THE ART AND ϚČÏΣƝČΣ OF SPEECH FEATURE ENGINEERING

Sriram Ganapathy, Samuel Thomas

Tutorial – T4, Sept. 14, Interspeech 2014

Introduction

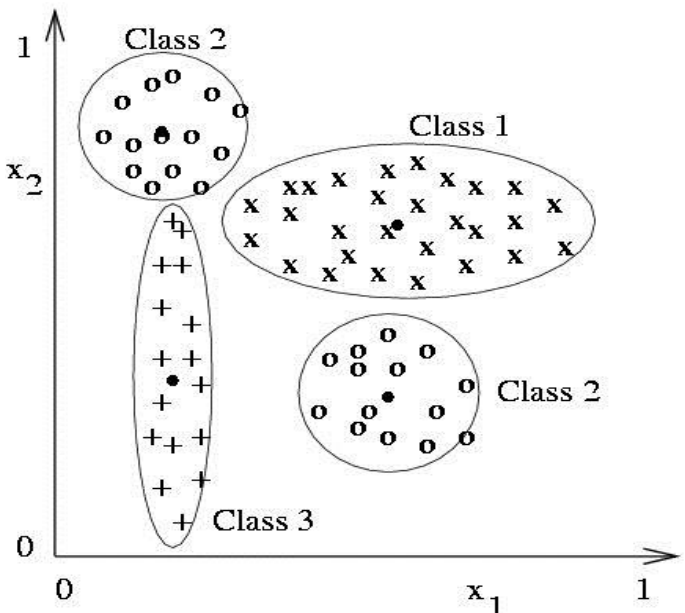
- Speech is a complex signal containing lots of information
  - Biometric – gender, language, speaker.
  - Content – word sequence, semantics, topic.
  - Higher level – emotion, environment

- Wide range of applications automatic speech systems
  - Coding and Enhancement
  - Speech and Speaker Recognition, Language identification, Emotion Recognition
  - Speech Synthesis and Voice Conversion.

- Accelerated interest in speech applications with the advances in mobile telecommunications.

Problem Formulation

- Traditional approaches to speech processing
  - Rule and heuristic based methodologies
  - Using small amounts of data
- Recently, most speech problems are addressed as statistical pattern recognition problem with big data.



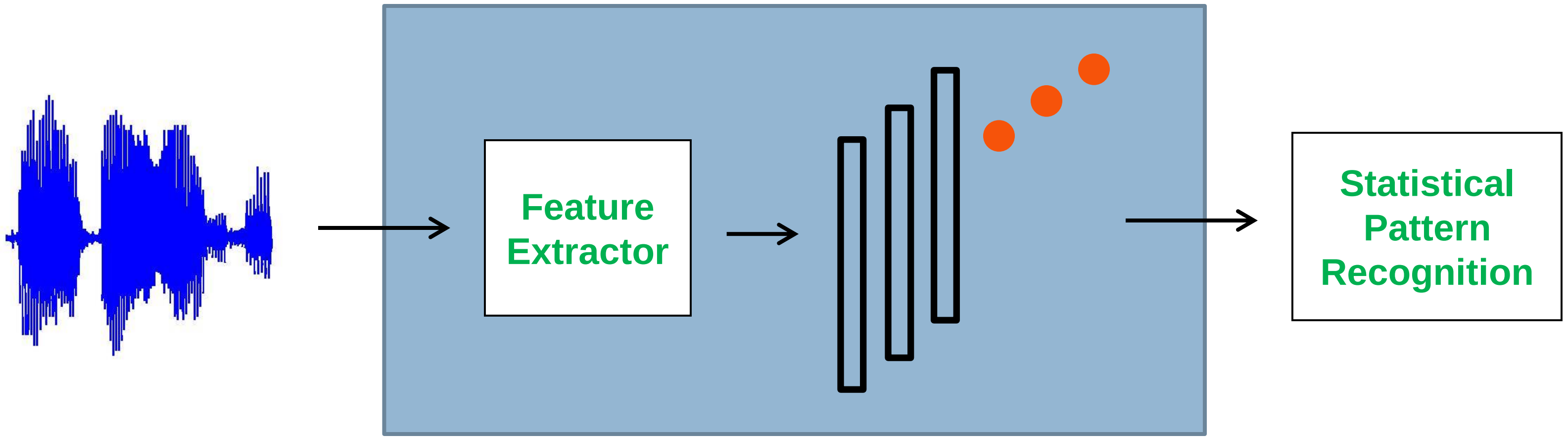
Problem Formulation

- Using the speech signal directly for pattern recognition
  - Speech is a time-varying non-stationary signal.
  - Information may lie in a small portion of signal.
  - May contain irrelevant information for the application.
  - Presence of noise and other distortions cause issues.
  - Size and dimensionality of the data.

- A need to transform the signal into lower dimensional descriptors called **features**.

Challenges

- Challenges involved in feature extraction
  - Preserving the relevant information for the application
  - Removing unwanted redundancies in the signal – separating the information pertinent to the task.
  - Resilience to noise and other degradations.



THE PAST …

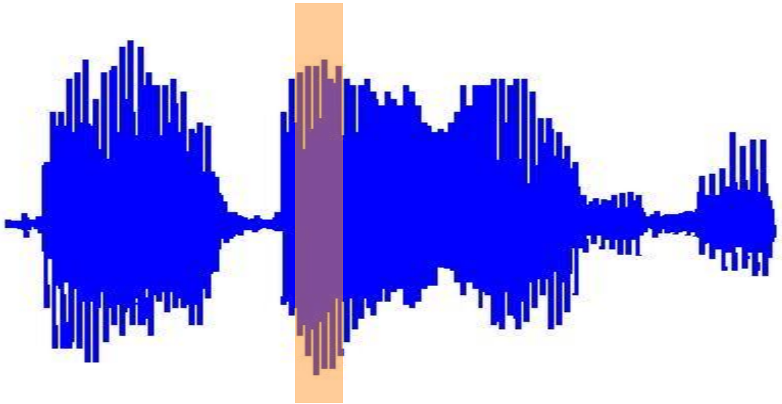
The farther back you can look, the farther forward you are likely to see.
-Winston Churchill

Outline

- Speech Coding Inspired Features
  - Mel Spectrogram and Linear Prediction
- Speech Synthesis Inspired Features
  - Pitch and Prosody
- Long Contextual Features
  - Delta Processing, RASTA Filtering and Modulation Features
- Normalization Techniques
  - Cepstral Mean Normalization and Spectral Subtraction

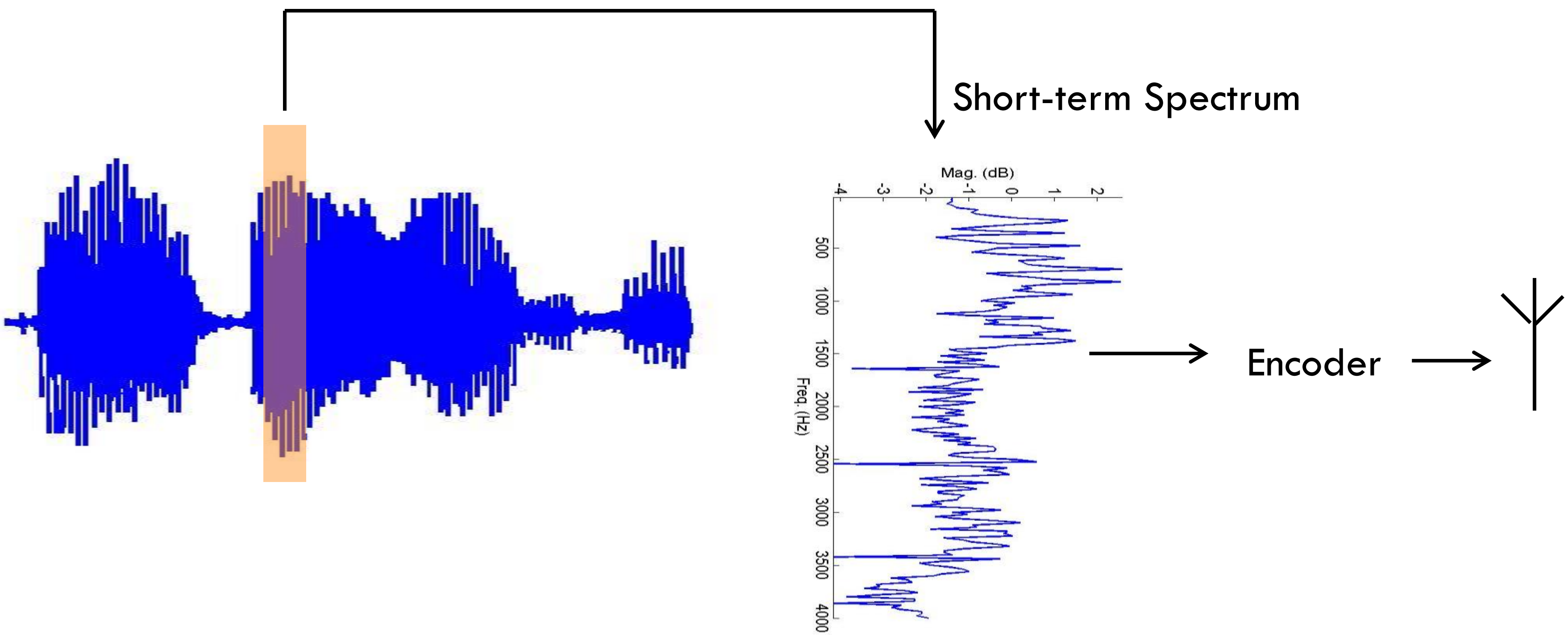
Speech Coding Inspired Features

- Coding – Transmitting the speech signal across a communication channel with small number of bits, having low latency.
  - Encoding the short-term spectrum.
  - Low latency processing



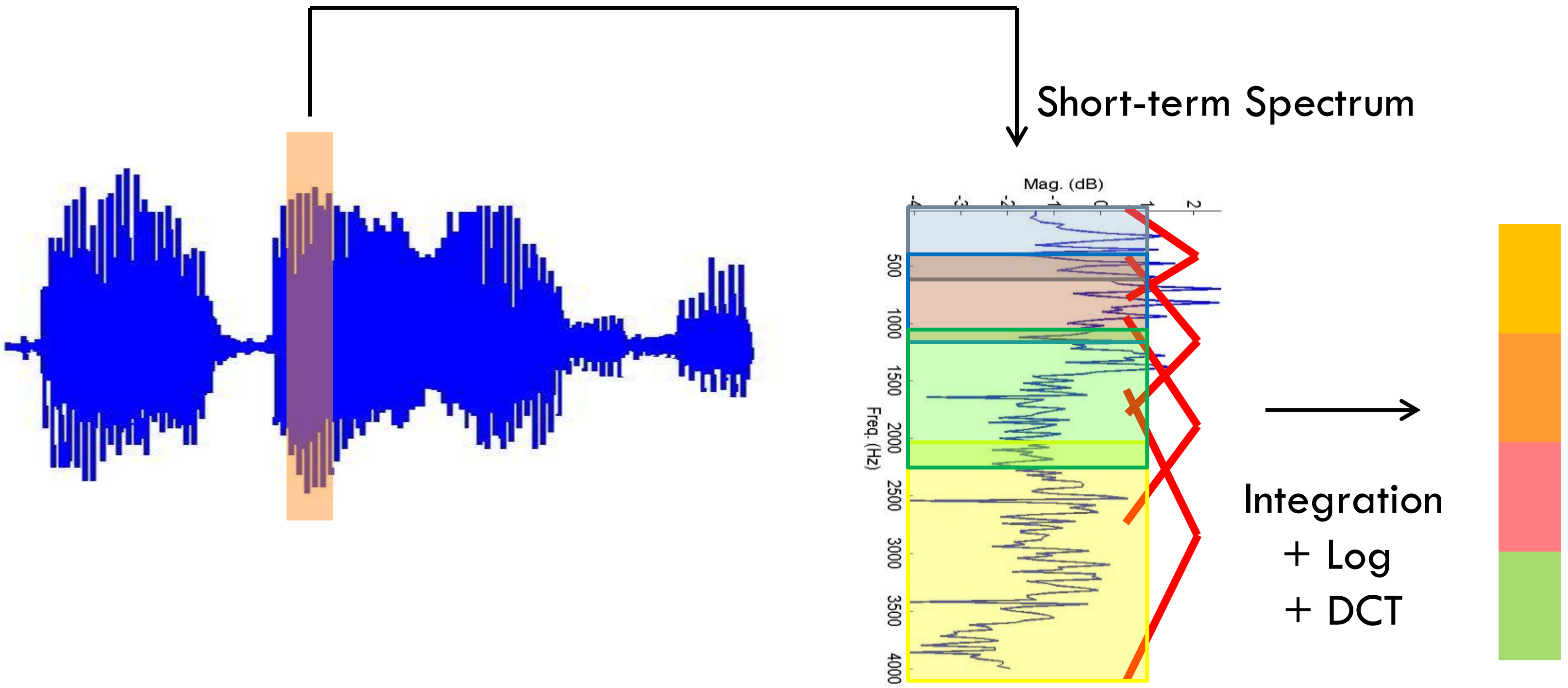
Speech Coding Inspired Features

- Coding – Transmitting the speech signal across a communication channel with small number of bits, having low latency.
  - Encoding the short-term spectrum.
  - Low latency processing



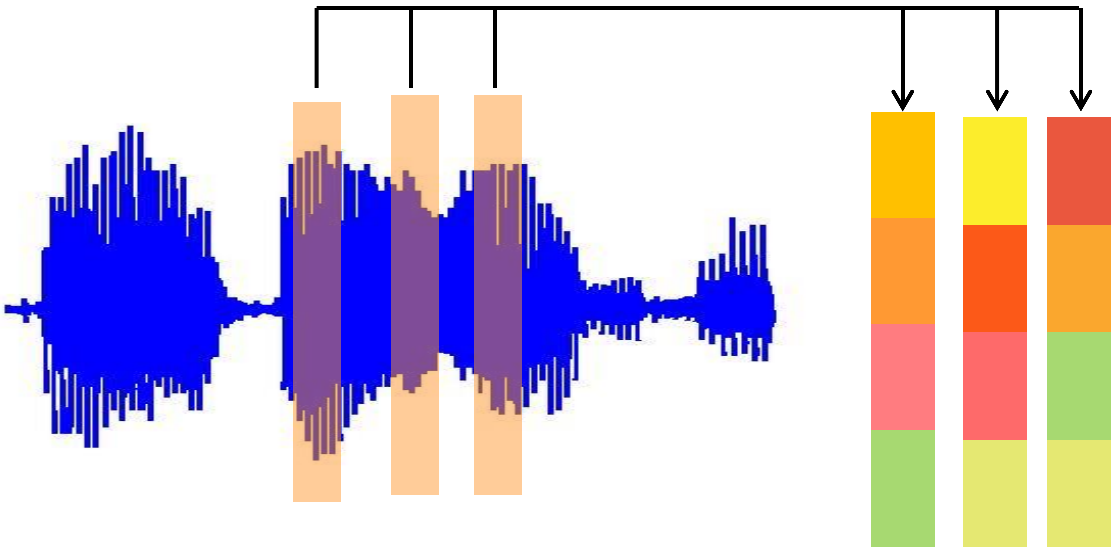
MFCC

- Short-term spectra integrated in mel frequency bands followed by log compression + DCT – mel frequency cepstral coefficients (MFCC) [Davis and Mermelstein, 1979].



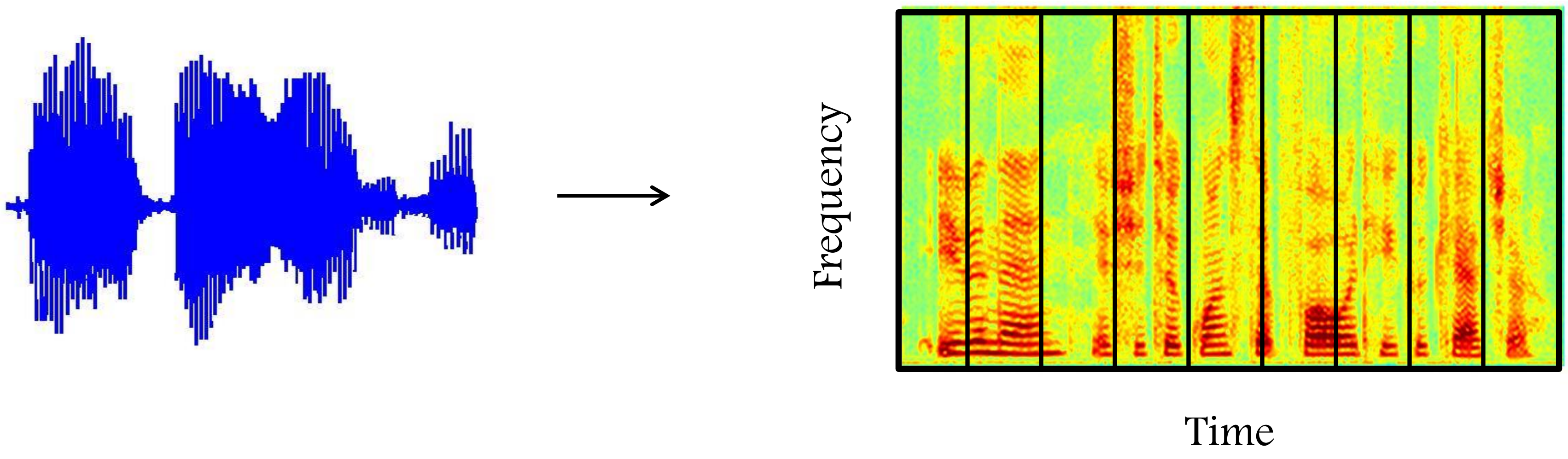
MFCC

- MFCC processing repeated for every short-term frame yielding a sequence of features.



Mel Spectrogram

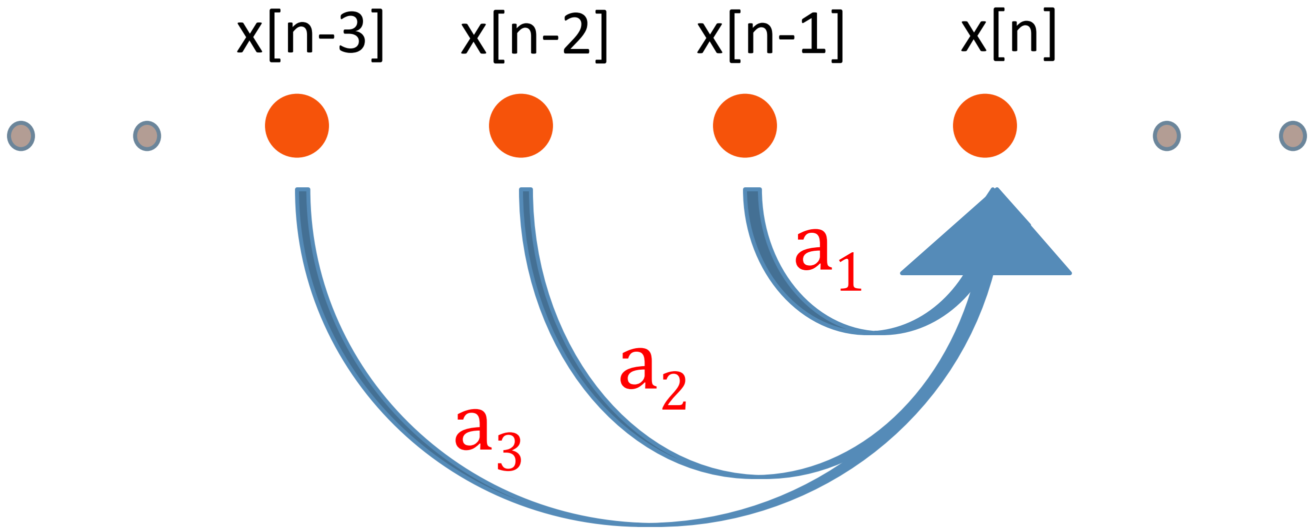
- Short-term spectra integrated in mel frequency bands followed by log compression – mel spectrogram [Davis and Mermelstein, 1979].
- Mel spectrogram constitutes an excellent tool for signal analysis and feature representation for speech.



Linear Prediction

- Current sample expressed as a linear combination of past samples [Atal, 1972],

$$x[n] \cong \sum_{k=1}^{p} a_k x[n-k]$$



Linear Prediction

- Prediction error defined as the difference between actual value and the estimate [Makhoul, 1975],

$$\mathbf{e}[n] = x[n] - \sum_{k=1}^{p} a_k x[n-k] = x[n] * d[n]$$

where the filter, $d = [1 \quad -a_1 \quad -a_2 \quad \ldots \quad -a_p]$

$$\mathcal{E}(\omega) = \sum_{n=0}^{N-1} \mathbf{e}[n] e^{-j\omega n} = X(\omega) D(\omega)$$

- Model parameters obtained by minimizing the L2-norm of the error.

$$E_p = \sum_{n=0}^{N-1} |\mathbf{e}[n]|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\mathcal{E}(\omega)|^2 \, d\omega$$

Linear Prediction

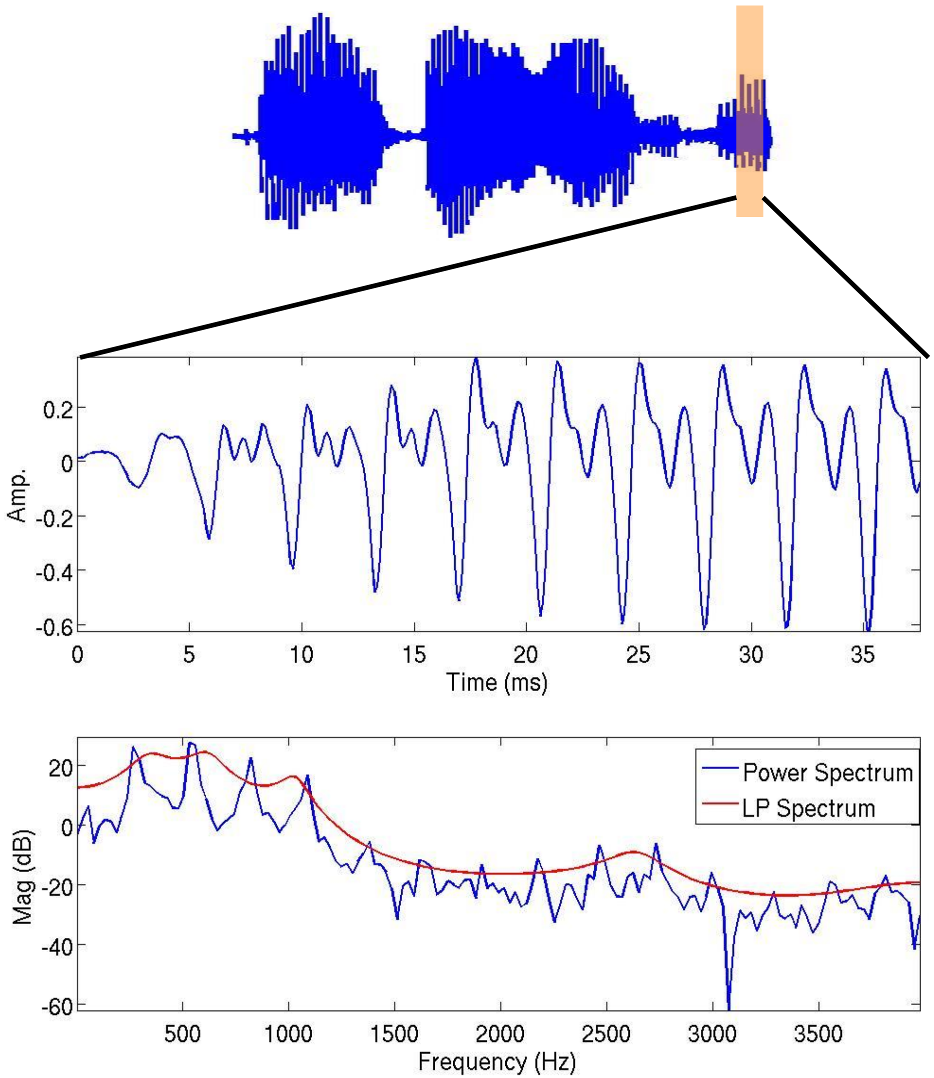
- Linear set of equations with signal autocorrelations obtained from the inverse of Fourier transform of the power spectrum $\{r_0, r_1, \ldots, r_p\}$.

- Solution of LP yields the filter coefficients, $\{a_1, \ldots, a_p\}$. Efficient coding scheme (code excited linear prediction - CELP) transmits LP coefficients with a few bits describing the residual signal.

- The inverse of the filter response multiplied by the signal variance gives an all-pole estimate of the power spectrum of the signal.

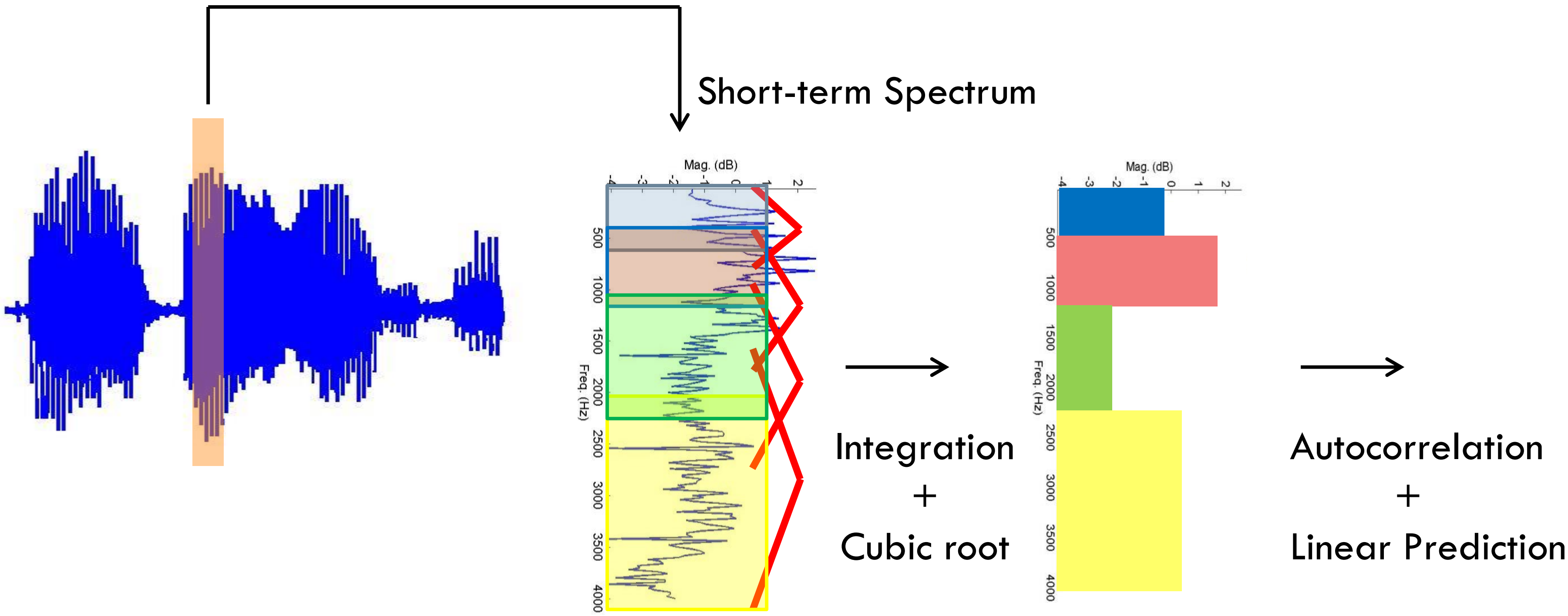
$$\widehat{P_x}[\omega] = \frac{\sigma^2}{|D(\omega)|^2} = \frac{\sigma^2}{|1 - \sum_{k=1}^{p} a_k e^{-jk\omega}|^2}$$

Linear Prediction

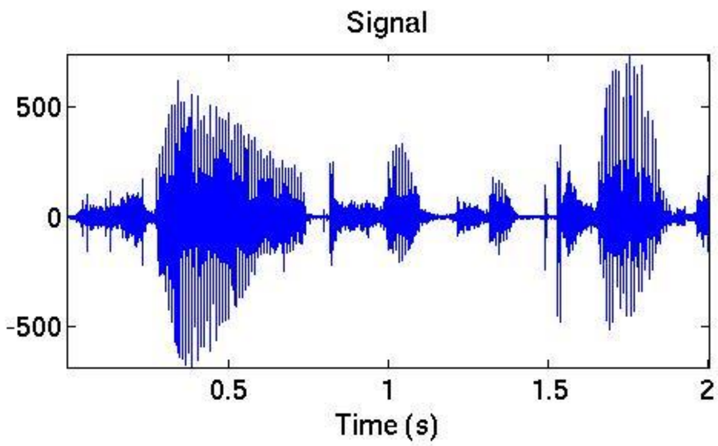


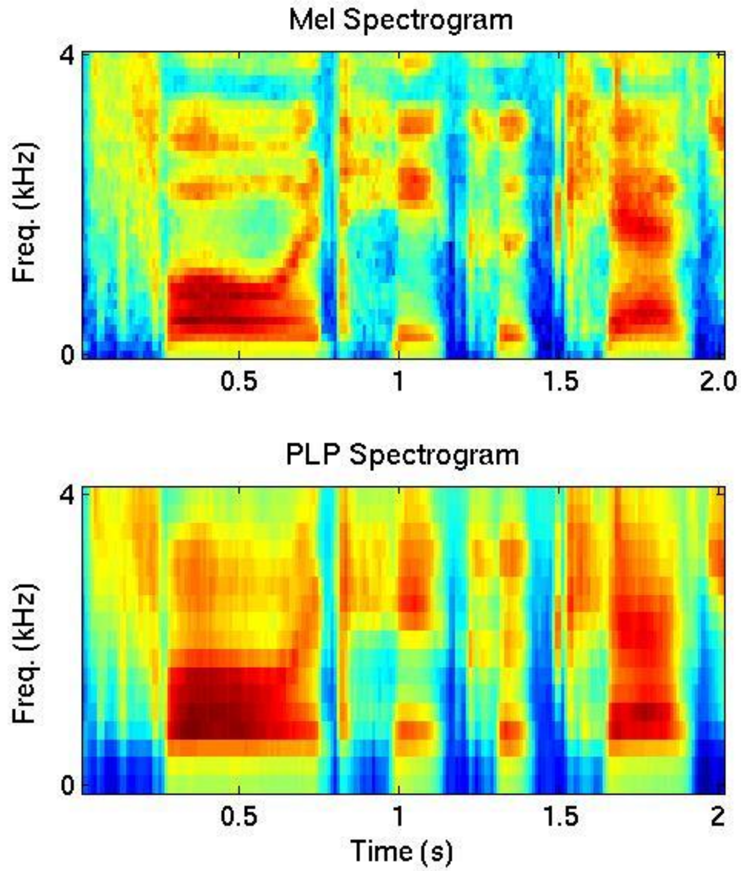
Perceptual Linear Prediction

- Critical band integration and compression to original power spectrum – convert to autocorrelation estimates – linear prediction [Hermansky, 1991].



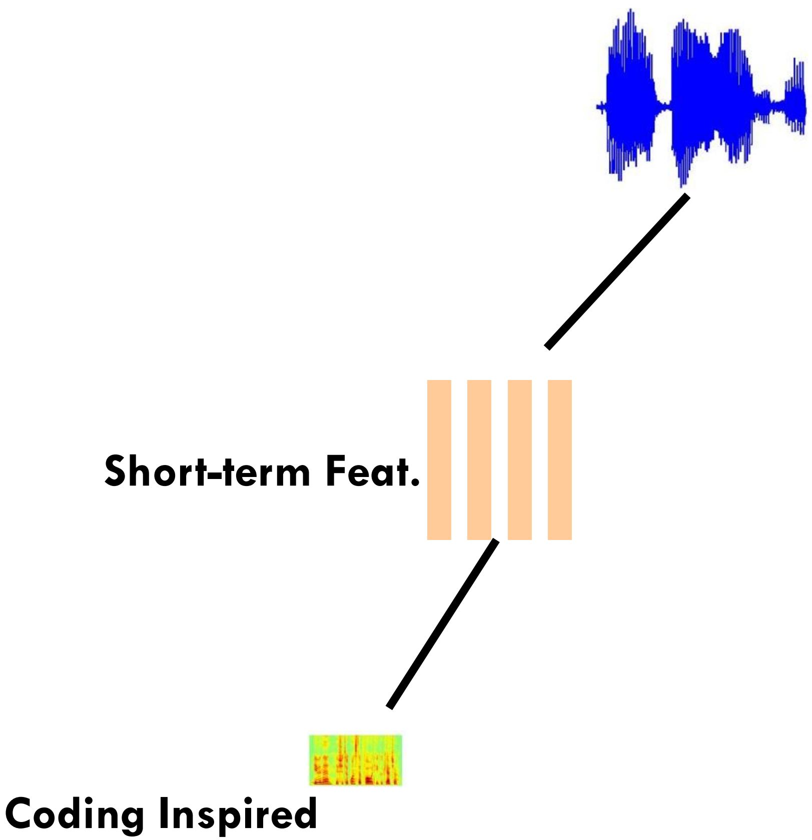
Perceptual Linear Prediction





- PLP provides smooth representation which is more robust.

Past – Discussion Summary

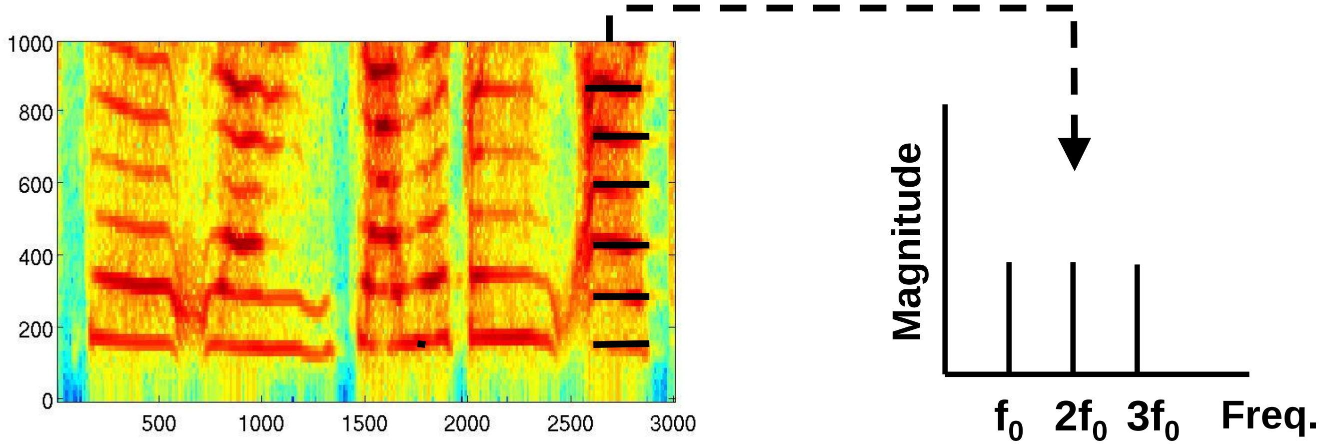


Outline

- Speech Coding Inspired Features
  - Mel Spectrogram and Linear Prediction
- Speech Synthesis Inspired Features
  - Pitch and Prosody
- Long Contextual Features
  - Delta Processing, RASTA Filtering and Modulation Features
- Normalization Techniques
  - Cepstral Mean Normalization and Spectral Subtraction

Pitch

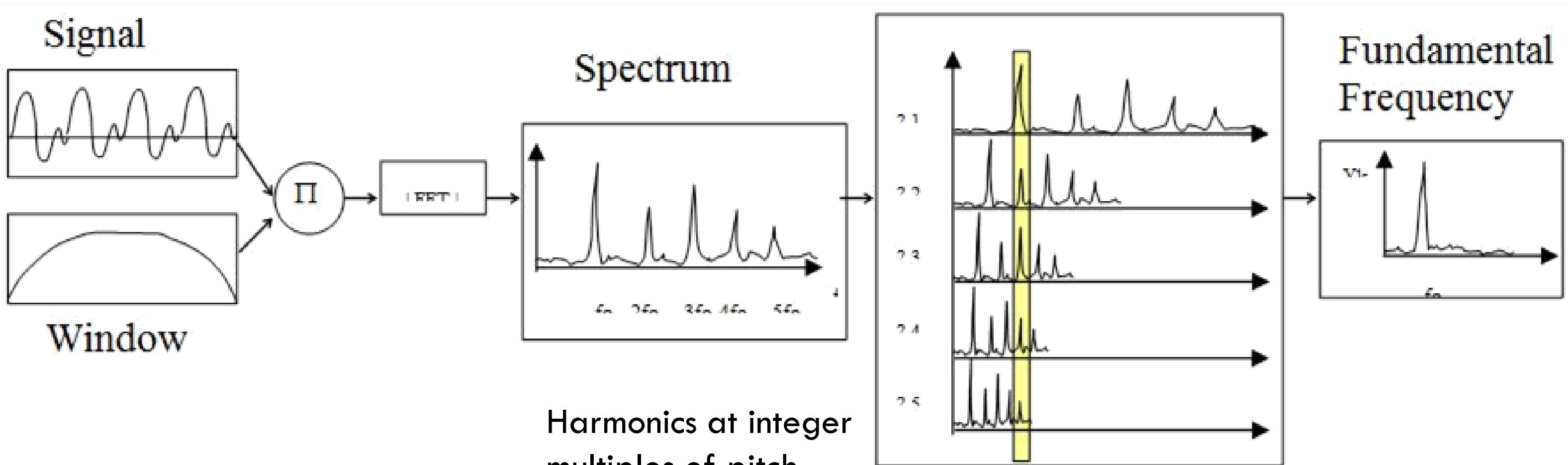
- Voiced speech exhibits harmonic properties
  - Pitch is a psycho-acoustic measure
  - The fundamental harmonic frequency is a way of quantifying pitch.
  - Varies based on speaker and content ~(50 – 400 Hz).
  - Useful in speech recognition, emotion recognition and speaker verification



Estimating Pitch – Frequency Domain

- Estimating the signal spectrum with different warping factors.
- Finding the peak in the product [Noll, 1969].

$$Y(\omega) = \prod_{r=1}^{R} |X(r\omega)|$$

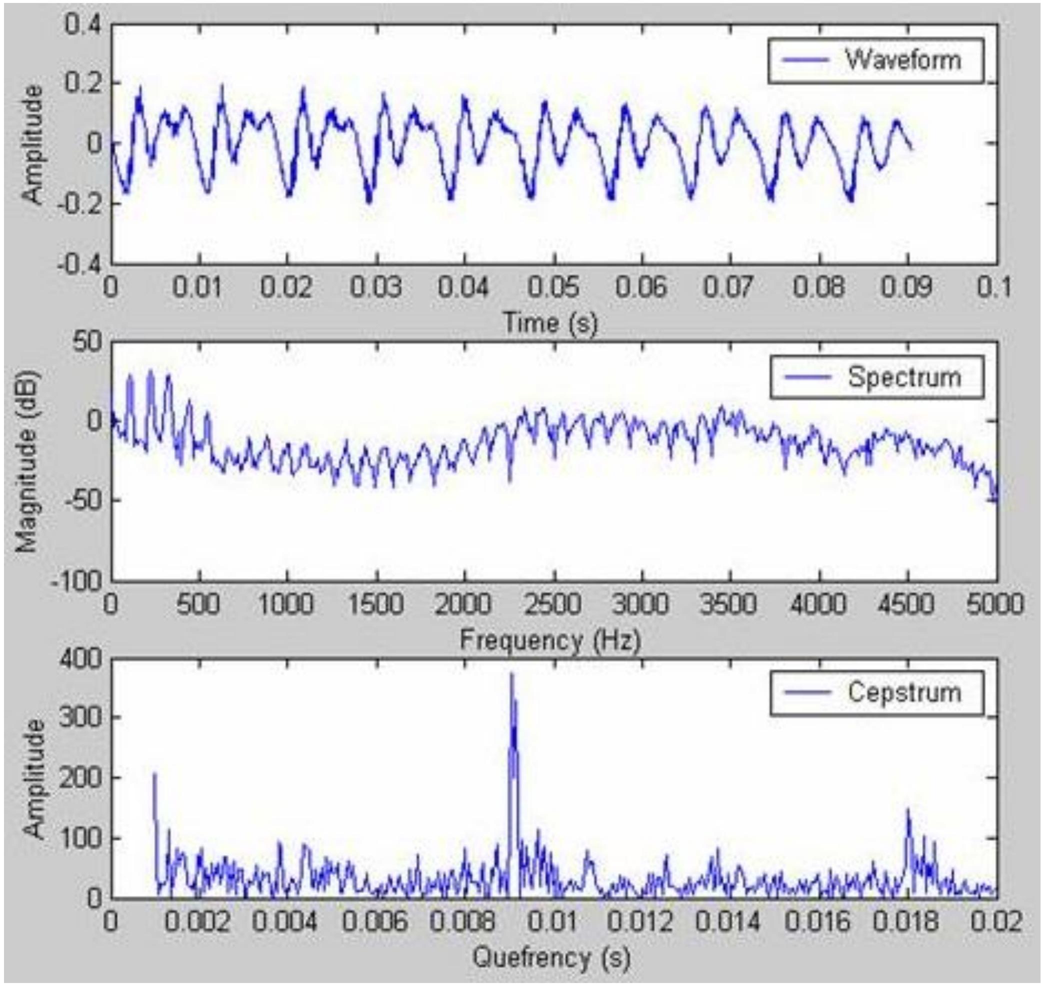


Harmonics at integer multiples of pitch

[Cuadra 2001]

Estimating Pitch – Cepstral Domain

- The log magnitude spectrum contains regularly spaced harmonic, thus can be viewed as a periodic signal and the period is pitch [Childers, 1977].
- Spectrum of log magnitude spectrum (cepstrum) will provide the pitch estimates



Courtesy – Yang et al.

Estimating Pitch – Time Domain

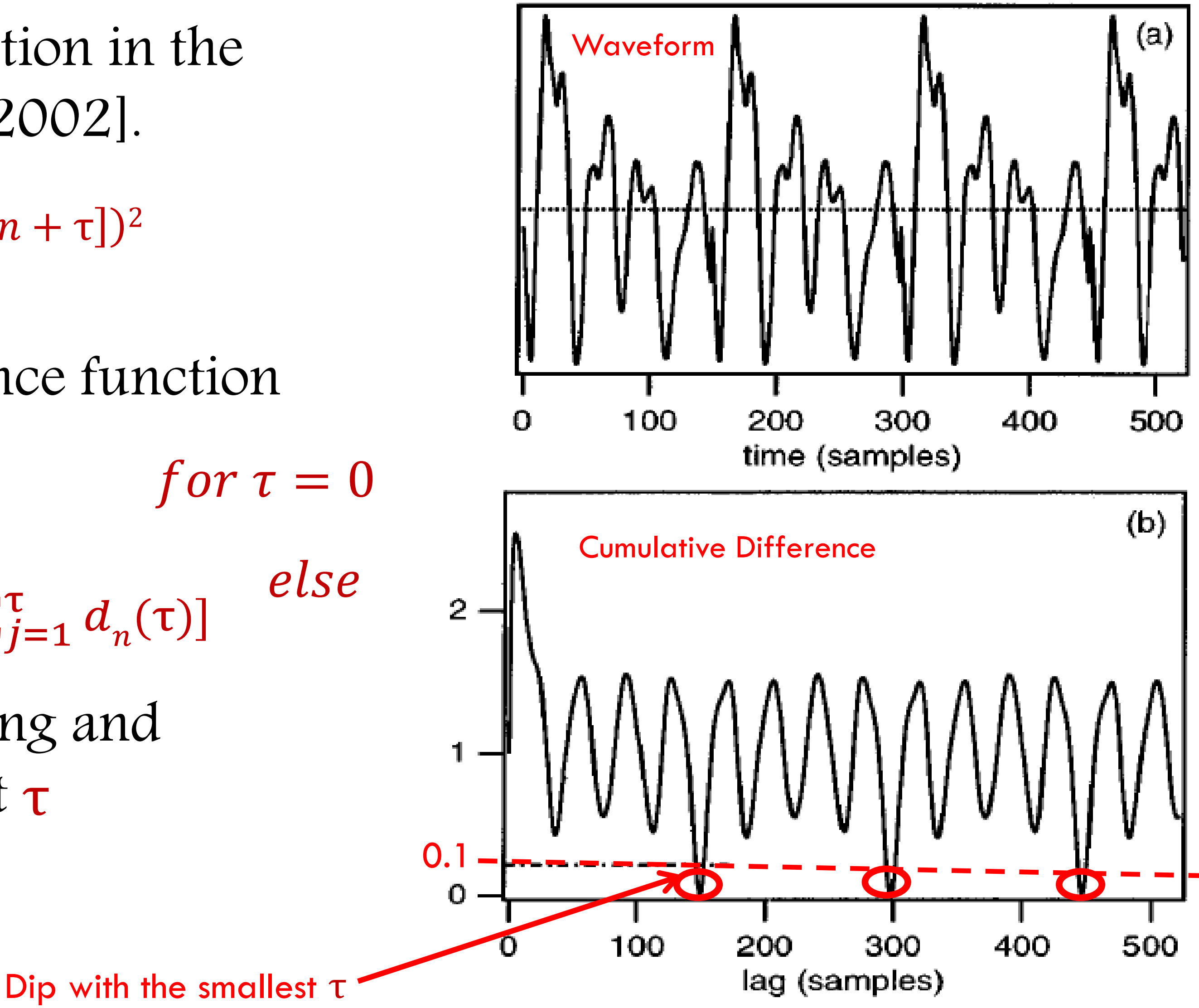
- The difference function in the time domain [YIN, 2002].

$$d_n(\tau) = \sum_{n=1}^{N} (x[n] - x[n+\tau])^2$$

- Cumulative difference function

$$D_n(\tau) = \begin{cases} 1 & for\ \tau = 0 \\ d_n(\tau) \Big/ \frac{1}{\tau}\left[\sum_{j=1}^{\tau} d_n(\tau)\right] & else \end{cases}$$

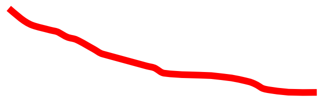
- Absolute thresholding and picking the smallest τ



Prosody

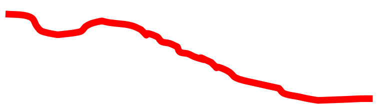
- Prosody is intonation, stress and rhythm of speech.

- Example with pitch contours
  - DECALARATIVE: "You are going home"



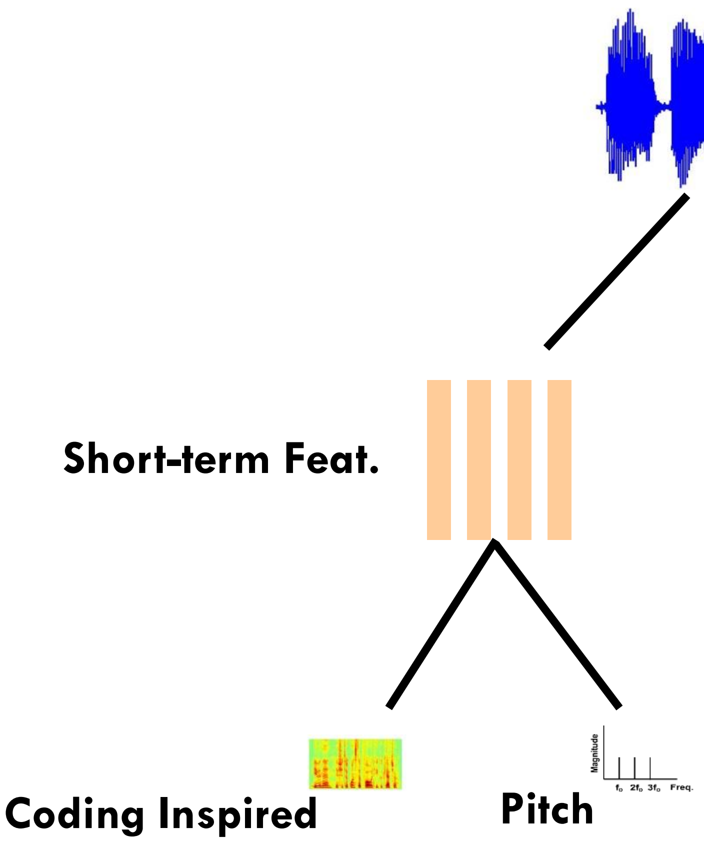
  - INTEROGATIVE: "You are going home?" (voice is raised at end of sentence)

  - IMPERATIVE: "You ARE going home!" (are is emphasized)



- Prosodic features
  - Pitch Contours, Pause durations [Shriberg 2000].

Past – Discussion Summary

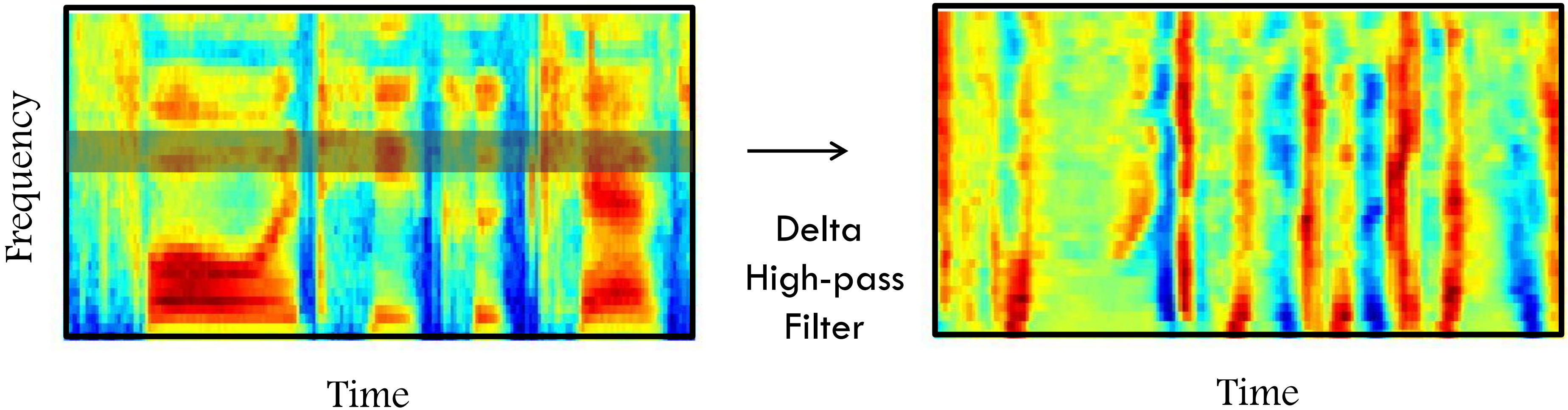


Outline

- Speech Coding Inspired Features
  - Mel Spectrogram and Linear Prediction
- Speech Synthesis Inspired Features
  - Pitch and Prosody
- Long Contextual Features
  - Delta Processing, RASTA Filtering and Modulation Features
- Normalization Techniques
  - Cepstral Mean Normalization and Spectral Subtraction

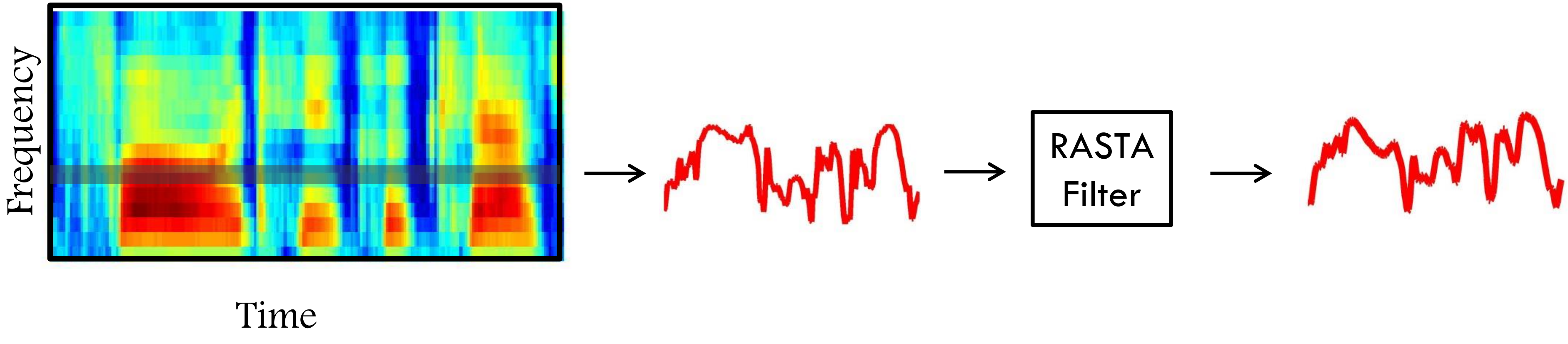
Delta Processing

- Filtering the trajectories using a high pass filter to derive deltas – implemented using simple difference operations [Furui, 1986].
- Enhancing the temporal changes in spectrogram.
- Widely used configuration for speech processing – spectrogram + deltas + double-deltas.



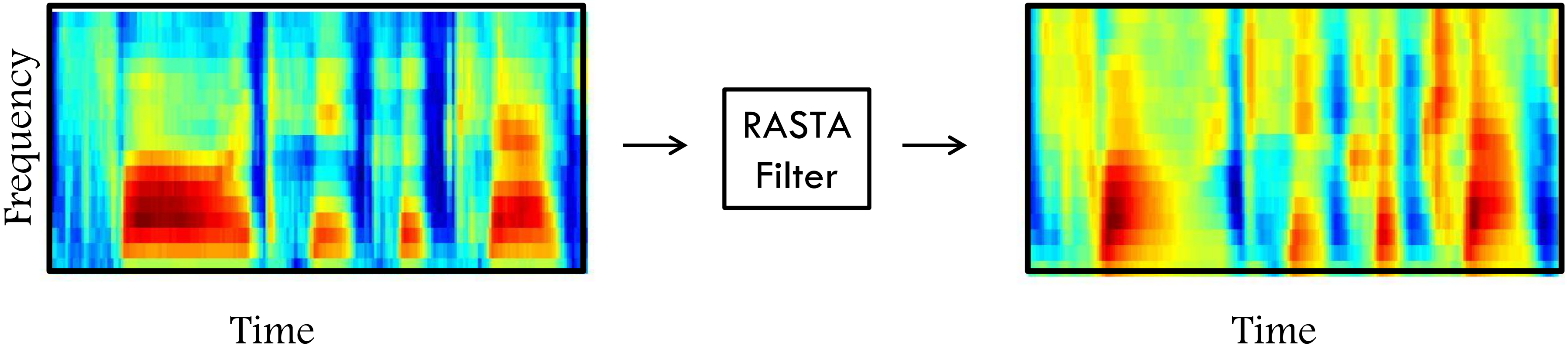
RASTA Filtering

- Human perception of speech modulations suggest a band-pass characteristic with a peak around 4-8Hz [Drullman, 1994].
- Relative Spectra (RASTA) [Hermansky,1994] - application of a band-pass infinite impulse response (IIR) filter on the temporal envelope of sub-band energy emulating human modulation processing.



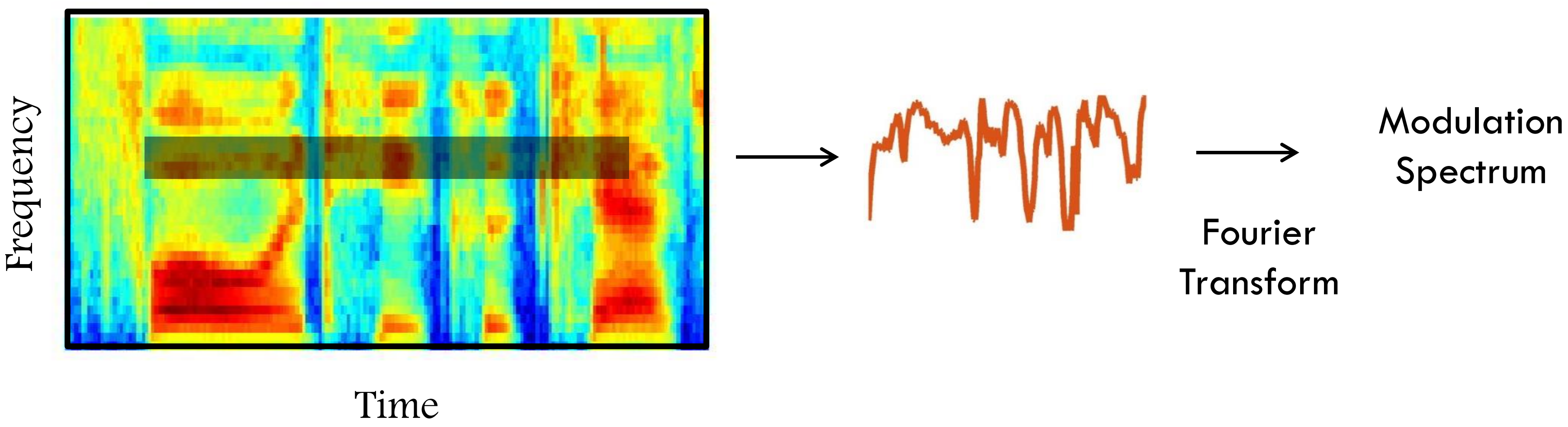
RASTA Filtering

- RASTA filtering – emphasizes slow changes and suppresses constant regions of the spectrogram as well as transients.
- Robustness to channel noise achieved through RASTA filtering.



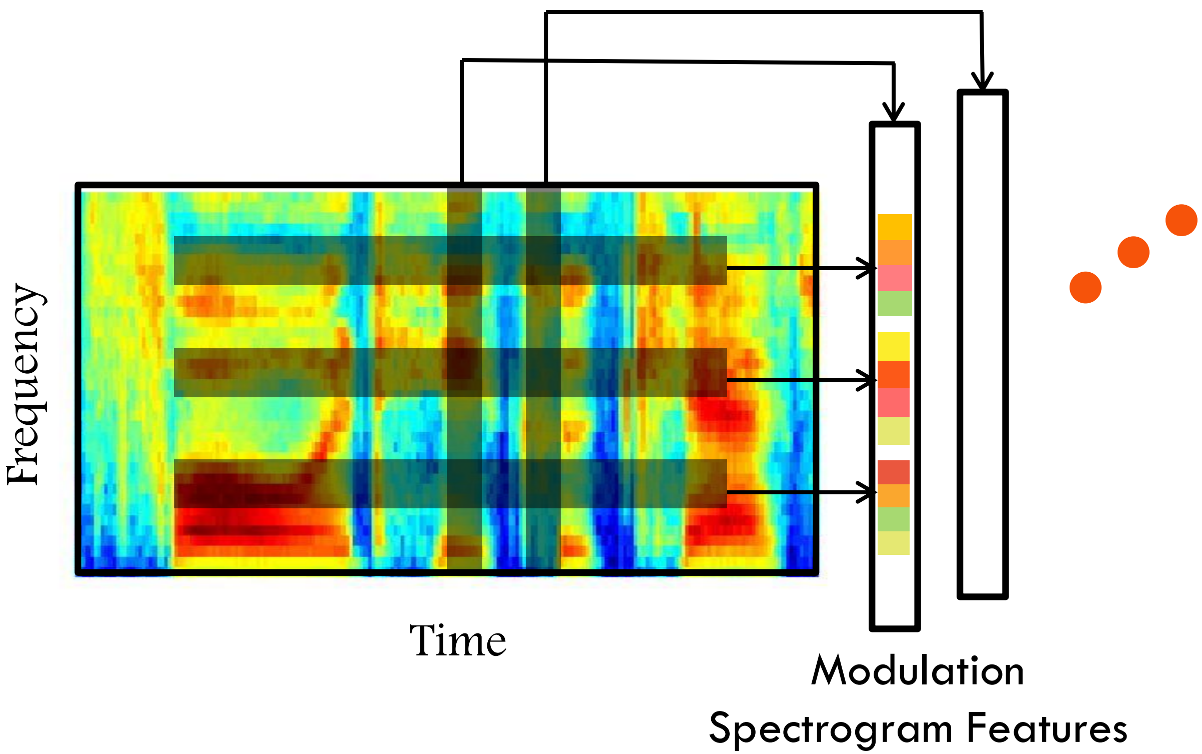
Modulation Spectrum of Speech

- Modeling the trajectories of individual sub-bands over a long duration [Kay, 1982].
- Typically used with a temporal context of 200-500ms around the current frame.



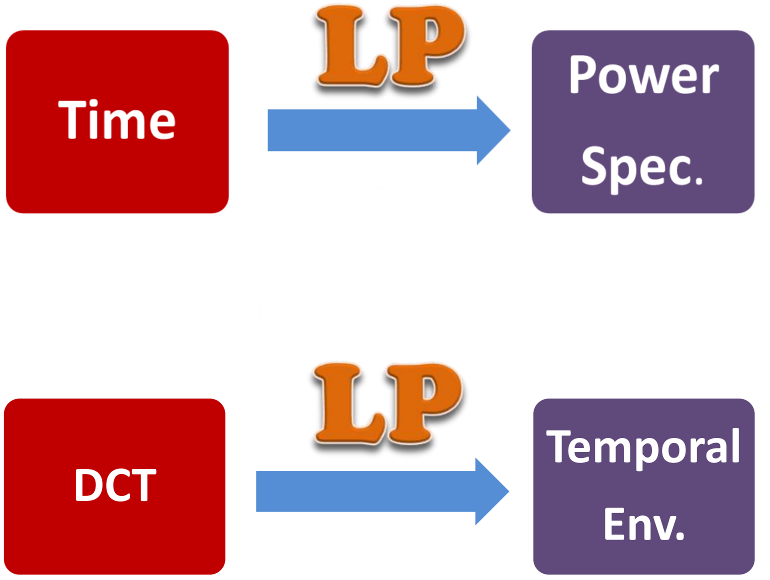
Modulation Spectrogram Features

- Stacking modulation spectral components from sub-bands – Modulation spectrogram of speech [Kingsbury et al., 1998].
- Useful representation in neural network acoustic models.



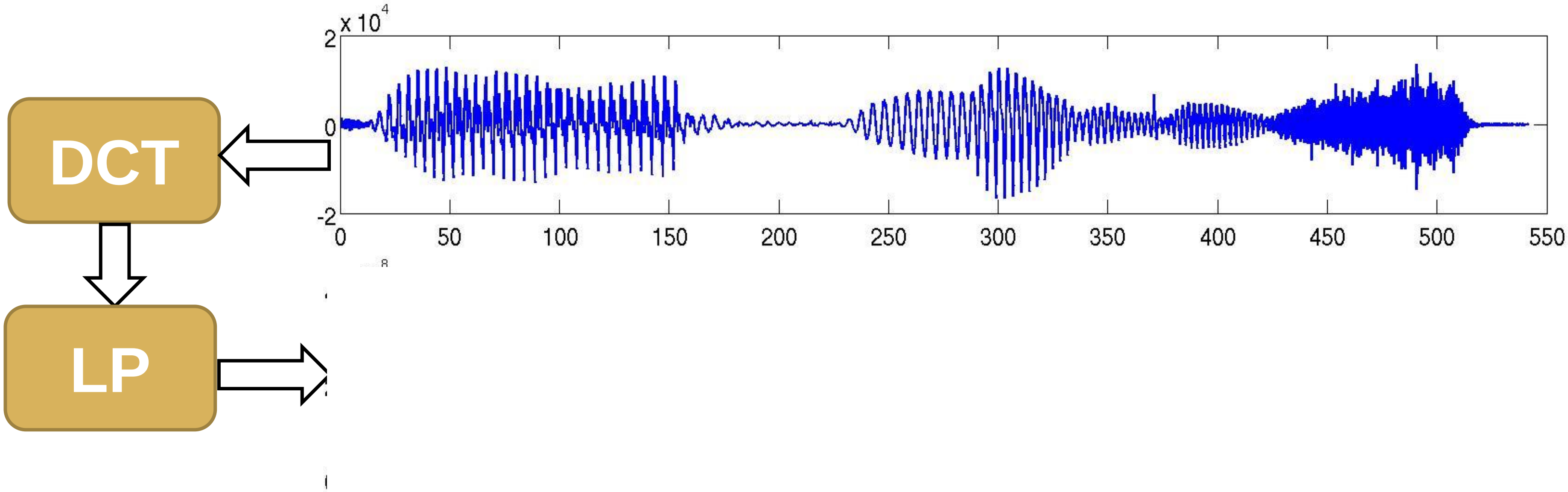
Frequency Domain Linear Prediction

- Predicting the trajectories of sub-band envelopes using linear prediction in the frequency domain [Athineos, 2003].
- Time-frequency duality.



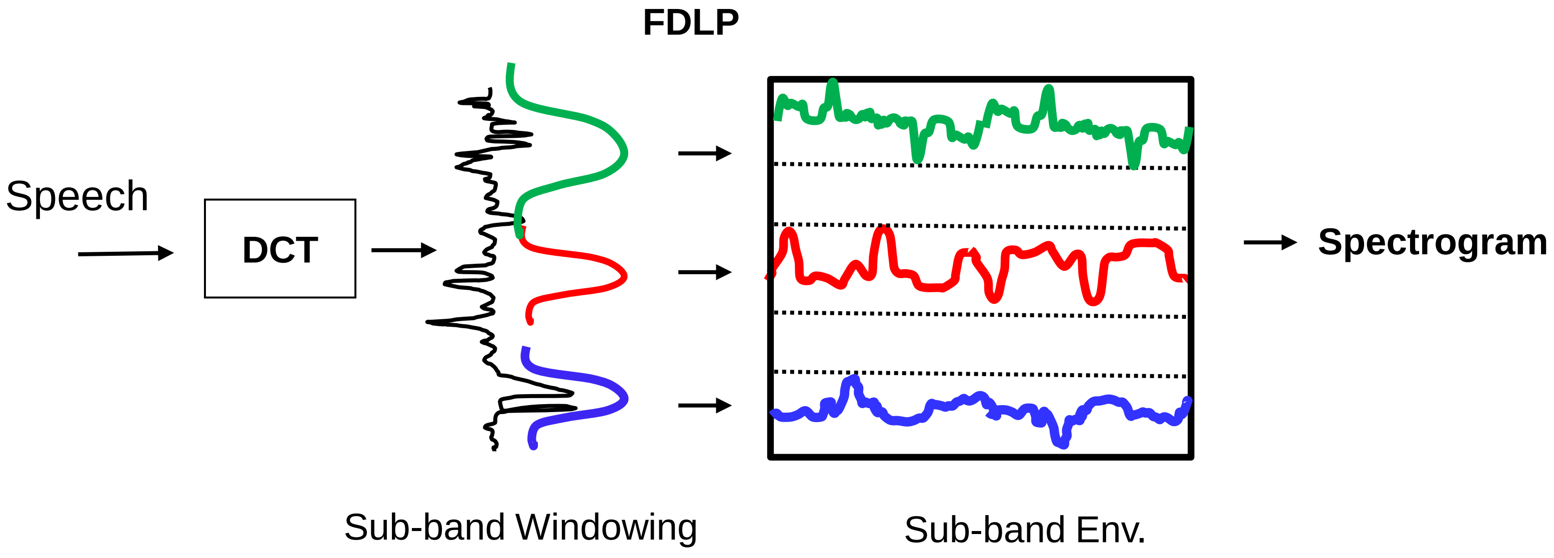
Frequency Domain Linear Prediction

Linear prediction on the cosine transform of the signal



Frequency Domain Linear Prediction

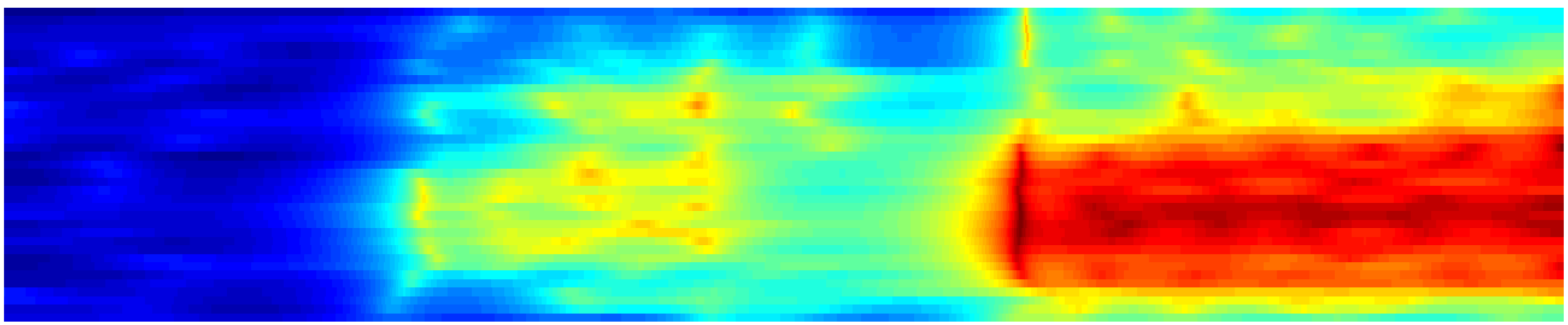
- Features from FDLP [Ganapathy, 2009].
- DCT of a long term signal (1000ms).
- Sub-band Windowing of DCT.
- Linear prediction on each sub-band DCT to derive envelopes.
- Stacking the envelopes to form the spectrogram.



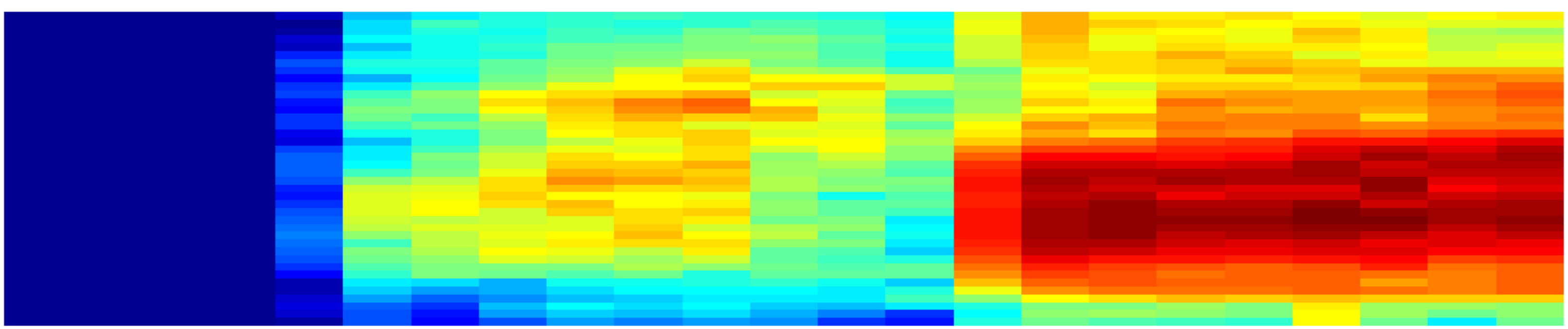
Frequency Domain Linear Prediction

- Higher temporal resolution is achieved with FDLP.
- Short-term and modulation features can be derived from the FDLP spectrogram.

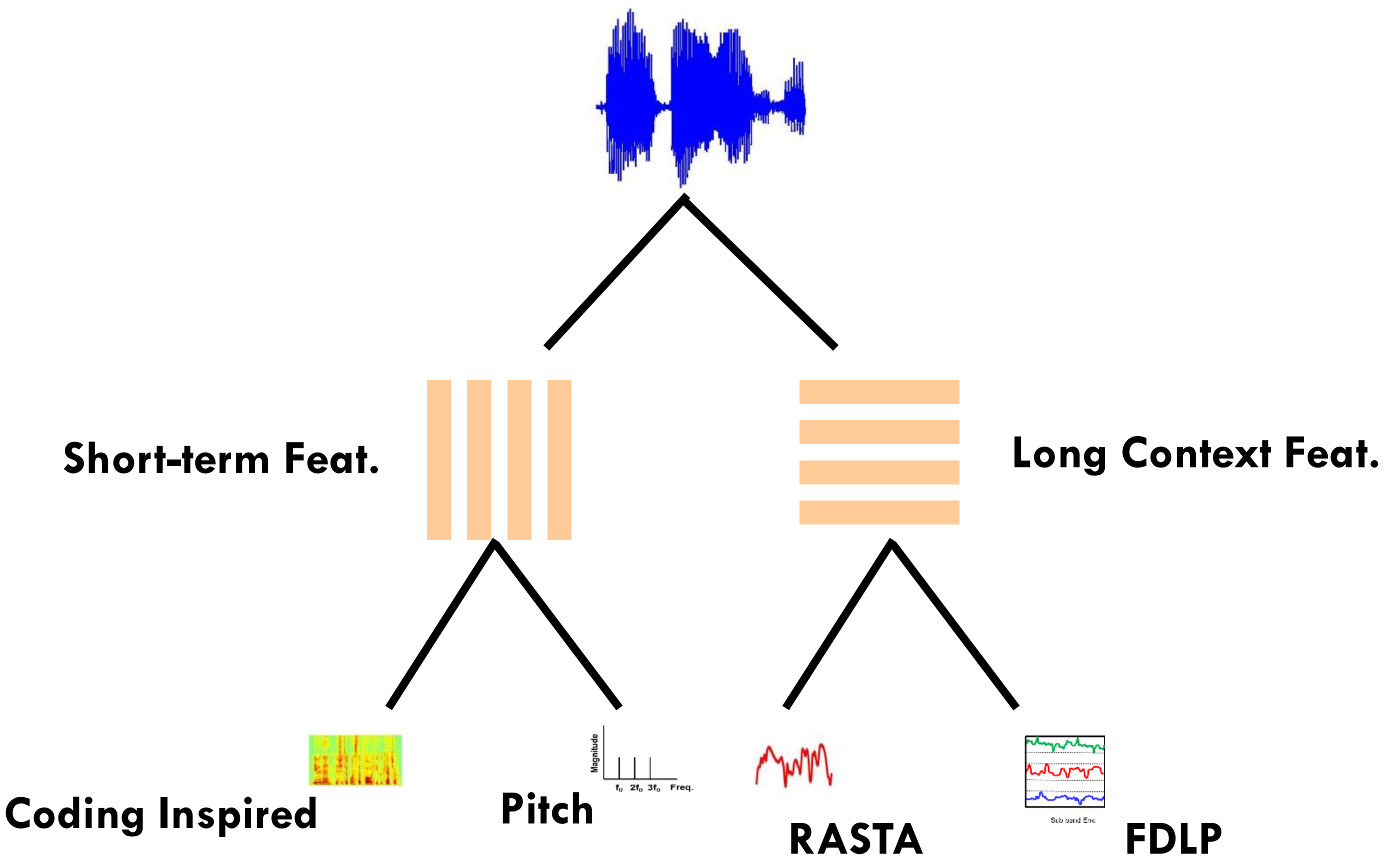
FDLP



Mel



Past – Discussion Summary



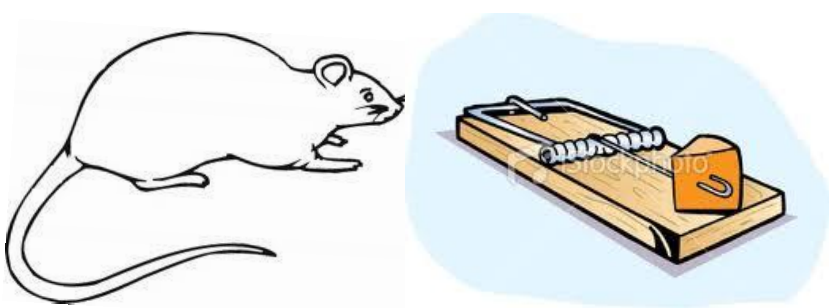
Outline

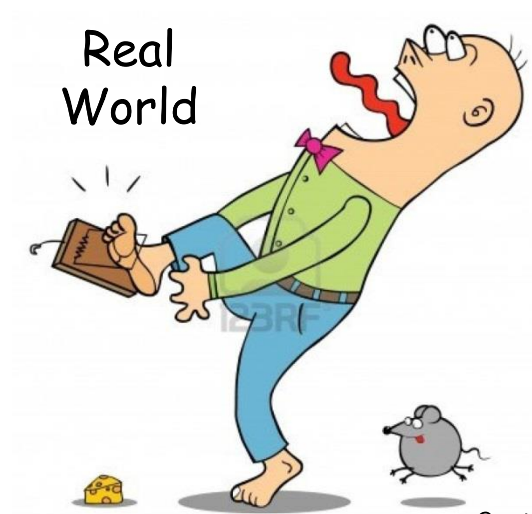
- Speech Coding Inspired Features
  - Mel Spectrogram and Linear Prediction
- Speech Synthesis Inspired Features
  - Pitch and Prosody
- Long Contextual Features
  - Delta Processing, RASTA Filtering and Modulation Features
- Normalization Techniques
  - Cepstral Mean Normalization and Spectral Subtraction

Robustness in Feature Processing

To expect the unexpected shows a thoroughly modern intellect.
Oscar Wilde

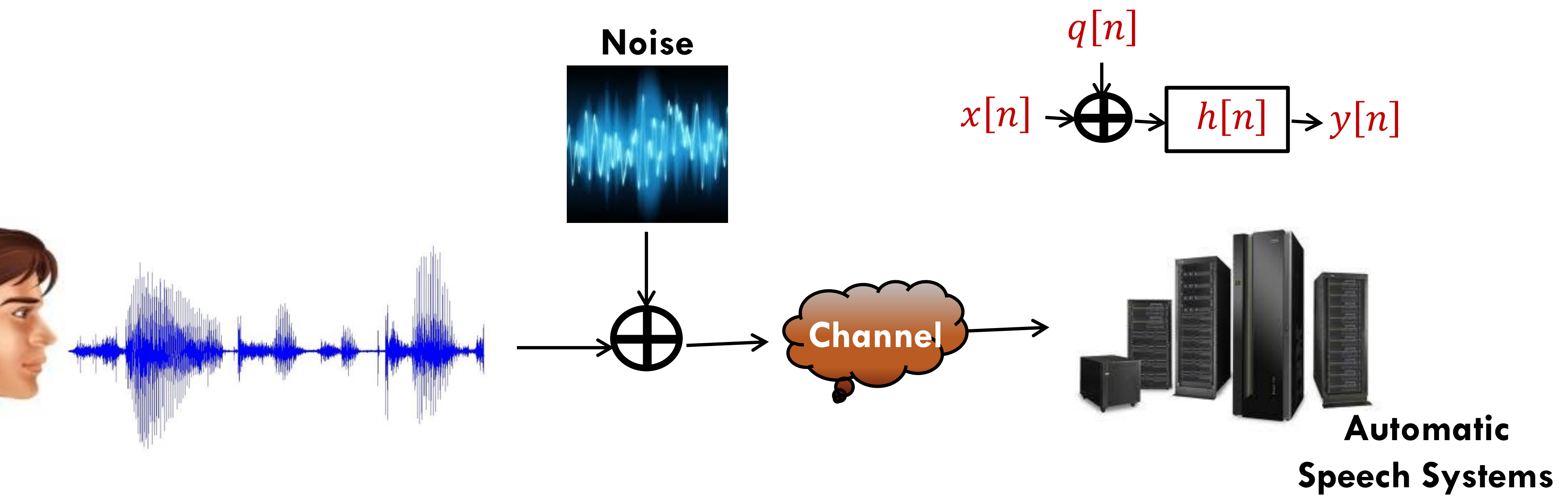
Expectation !





Courtesy – Google Images

Robustness in Feature Processing



Courtesy – Google Images

Cepstral Mean Normalization

- Linear channel has a convolutive effect on the speech signal.

$$y[n] = x[n] * h[n]$$

- Telephone channel/microphone channel have filter response $h[n]$ which has small time constant (less than 25ms). For frame index k,

$$y_k[n] = x_k[n] * h[n]$$

- Short-term cepstra (DCT of power spectrum) is modified by the linear channel,

$$Y_k(\omega) = X_k(\omega)H(\omega)$$

$$\log(|Y_k(\omega)|^2) = \log(|X_k(\omega)|^2) + \log(|H(\omega)|^2)$$

$$C_y[k] = C_x[k] + C_h$$

Cepstral Normalization

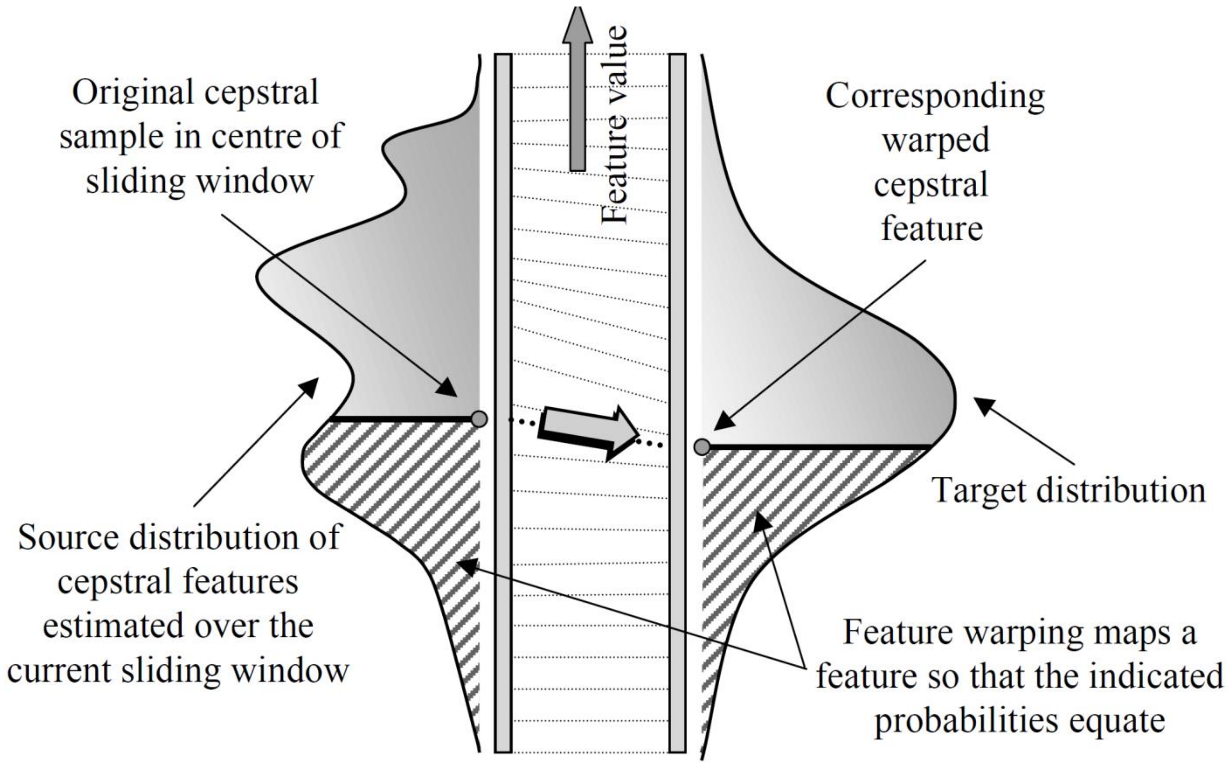
- The mean of the cepstra over all frames in the recording is the sum of the mean of signal cepstra and channel cepstra C_h.
- Hypothesis [Reynolds, 1994]–
  - mean of the signal cepstra is not useful component for speech processing.
  - removing the mean suppresses the effect of the linear channel.

- Cepstral variance normalization to increase the robustness to signal scale changes (due to different speakers, recording devices etc).

- Significant robustness achieved by cepstral normalization
  - Widely used in most of the speech signal processing applications.

Feature Warping

- Mapping the distribution of the cepstral features to standard Gaussian distribution [Pelecanos, 2001].



- Robustness against noise and linear channel – widely used in speaker and language recognition.

Reverberation

- Recording speech signal in a far-field environment – Received signal is the summation of the direct component and weighted-delayed components.
- Modeled as a long-term convolutive effect

$$y[n] = x[n] * r[n]$$

- Different from short-term convolutive effect like telephone channel
  - Telephone channel filters have time constants < 25ms
  - Reverberant room response functions has time time constant values [200-900] ms.

Long-term Log Spectral Subtraction

- Suppressing reverberation using long-term log spectral subtraction [Avendanos, 1996, Gelbert, 2001].
- Taking ~1000ms DFT of the signal (not true for short-term DFT).

$$Y_k(\omega) = X_k(\omega)R(\omega)$$

$$\log(|Y_k(\omega)|) = \log(|X_k(\omega)|) + \log(|R(\omega)|)$$

- Here k denotes the index of 1000ms segments.
- Computing the mean of $\log(|Y_k(\omega)|)$ and subtracting the mean suppresses reverberation.
  - The phase of $Y_k(\omega)$ is used in the reconstruction.

Additive Noise

- When speech signal is distorted with additive noise

$$y[n] = x[n] + q[n]$$

- Assuming stationary noise which is uncorrelated with the signal,

$$|Y_k(\omega)|^2 = |X_k(\omega)|^2 + |Q(\omega)|^2$$

- Effect of additive noise can be mitigated by subtracting the estimate of the noise from the signal [Boll, 1979].

- Using a voice activity detector, the noise power spectral estimate is obtained as mean of the power spectrum in the non-speech region.

$$|\hat{X}_k(\omega)|^2 = |Y_k(\omega)|^2 - |\hat{Q}(\omega)|^2$$

Spectral Subtraction

- Simple estimate of noise – average value of non-speech frames.
- Smoothed time-varying estimate (applied only on noisy frames)

$$|\hat{Q}_k(\omega)|^2 = \alpha\,|\hat{Q}_{k-1}(\omega)|^2 + (1-\alpha)|Y_k(\omega)|^2$$

- Reducing the musical noise by spectral flooring.

$$|X_k(\omega)|^2 = \begin{cases} |Y_k(\omega)|^2 - \alpha\,|\hat{Q}_k(\omega)|^2 & \text{if } |Y_k(\omega)|^2 > (\alpha+\beta)\,|\hat{Q}_k(\omega)|^2 \\ \beta\,|\hat{Q}_k(\omega)|^2 & \text{else} \end{cases}$$

- Wiener filtering –Estimating the gain function

$$G_k(\omega) = 1 - \frac{|\hat{Q}_k(\omega)|^2}{|Y_k(\omega)|^2}$$

Minimum Mean Square Error Estimation

- Assuming a Gaussian distribution for the spectral coefficients of clean speech and noise (dropping the frequency and frame index).

$$p_X(x) = \frac{1}{\pi {\sigma_X}^2} \exp(-\frac{|x|^2}{2{\sigma_X}^2}) \qquad p_Q(q) = \frac{1}{\pi {\sigma_Q}^2} \exp(-\frac{|q|^2}{2{\sigma_Q}^2})$$

- Minimum mean square error (MMSE) of the noise power spectrum [Eprhaim, 1984] given the noisy signal power spectrum Y.

$$|\hat{Q}|^2 = \min \mathbf{E}\,[|\hat{Q}|^2 - |Q|^2 \,|\, Y]$$

- This estimate is the posterior mean

$$|\hat{Q}|^2 = \mathbf{E}\,(|Q|^2 \,|\, Y)$$

MMSE Estimator

- Let ξ denote a-proiri signal-to-noise ratio

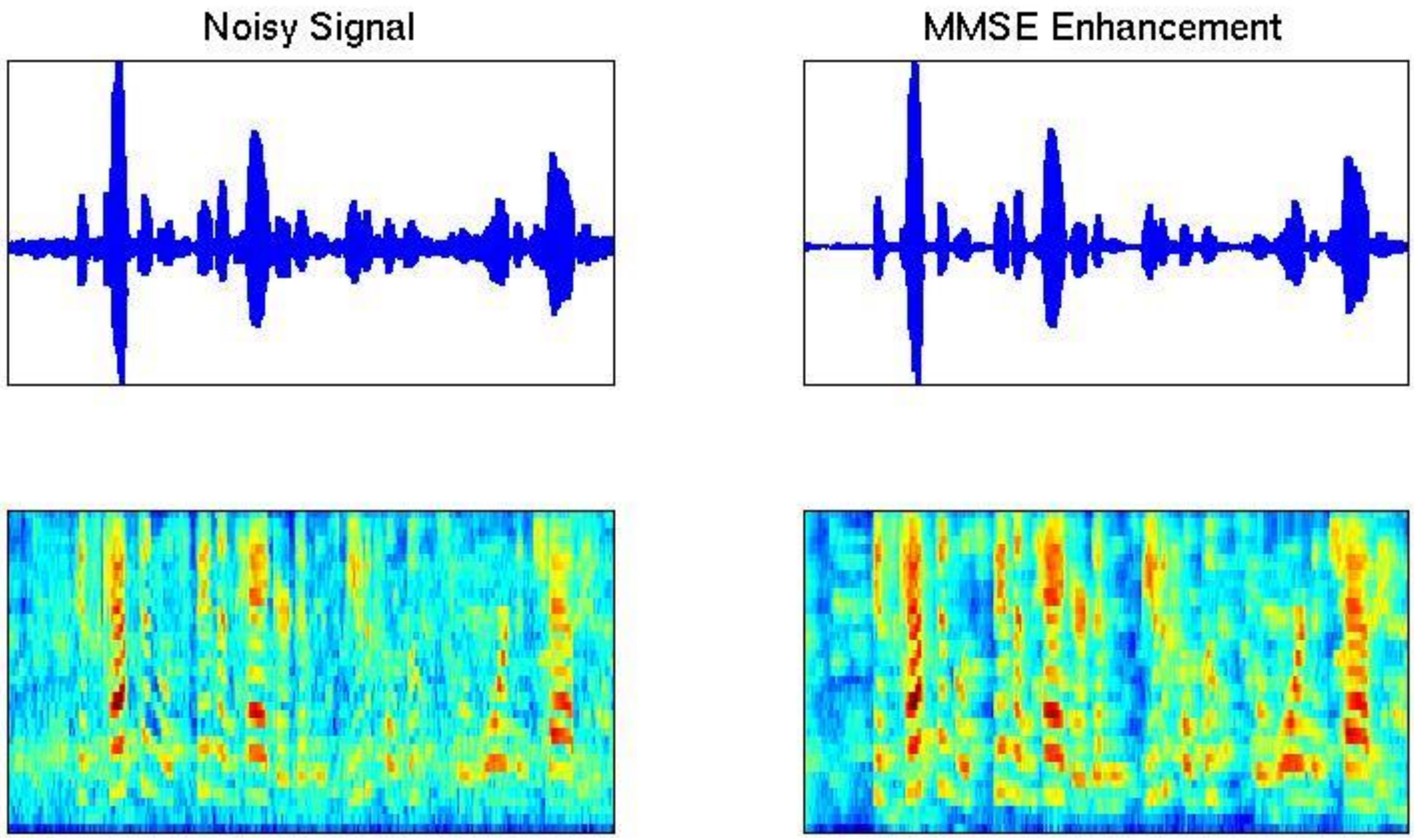
$$\xi = \frac{\sigma_X{}^2}{\sigma_N{}^2}$$

- Then, the posterior mean can be shown to be [Wolfe, 2001].

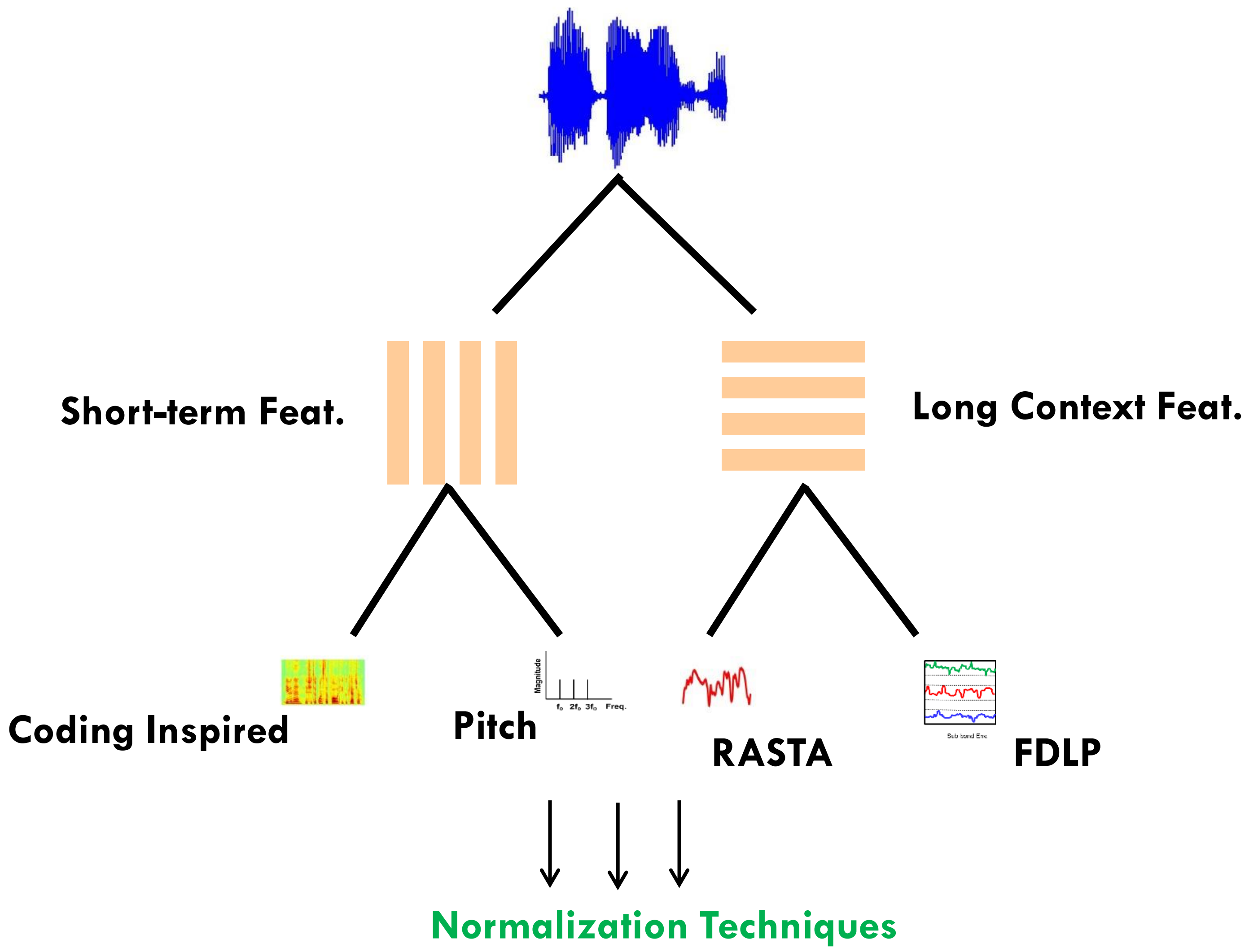
$$|\hat{Q}|^2 = \mathbf{E}\,[(|Q|^2\,|\,Y)] = \left(\frac{1}{1+\xi}\right)^2 |Y|^2 + \left(\frac{\xi}{1+\xi}\right)\sigma_N{}^2$$

- Smoothed approach to estimate the apriori SNR [Cappe, 1994]
  - With the estimate from the adjacent frames
  - Suppresses musical noise

MMSE Estimator



Past – Discussion Summary



THE PRESENT…

"Yesterday is gone. Tomorrow has not yet come. We have only today. Let us begin."
- Mother Teresa

Outline

- Normalizing Reverberation Artifacts
- Bio-inspired Spectro-temporal Filtering Approaches
- Unsupervised Data Driven Features – ivectors
- Supervised Data Driven Features

Normalizing Reverberation Artifacts

- When speech is corrupted with convolutive distortion like room reverberation

$$y[n] = x[n] * r[n]$$

- In the long-term DFT domain, this is a multiplication

$$Y[\omega] = X[\omega] \times R[\omega]$$

- In the m^{th} sub-band,

$$Y_m[\omega] = X_m[\omega] \times R_m[\omega]$$

- In narrow bands, $R_m[\omega]$ is slowly varying,

$$R_m[\omega] \cong X_m[\omega] \times R_m$$

Normalizing Reverberation Artifacts

- FDLP envelope of m^{th} band found by linear prediction on $R_m[\omega]$ outputs all-pole parameters $\{a_1, \ldots a_p\}$

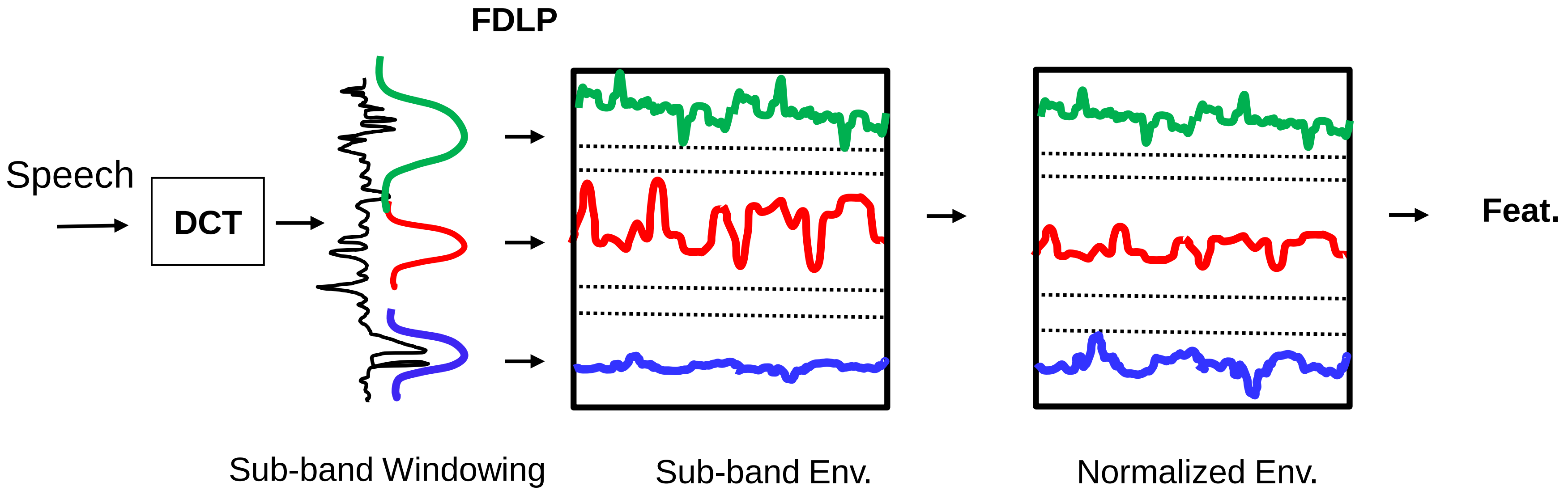
$$\widehat{E_m}[n] = \frac{G}{|1 - \sum_{k=1}^{p} a_k e^{\frac{-j2\pi kn}{N}}|^2}$$

- For reverberant speech, $X_m[\omega]$ is multiplied by R_m which modifies the gain G in the FDLP envelope.

- Normalization to convolutive distortions is achieved by reconstructing the FDLP envelope with $G = 1$.

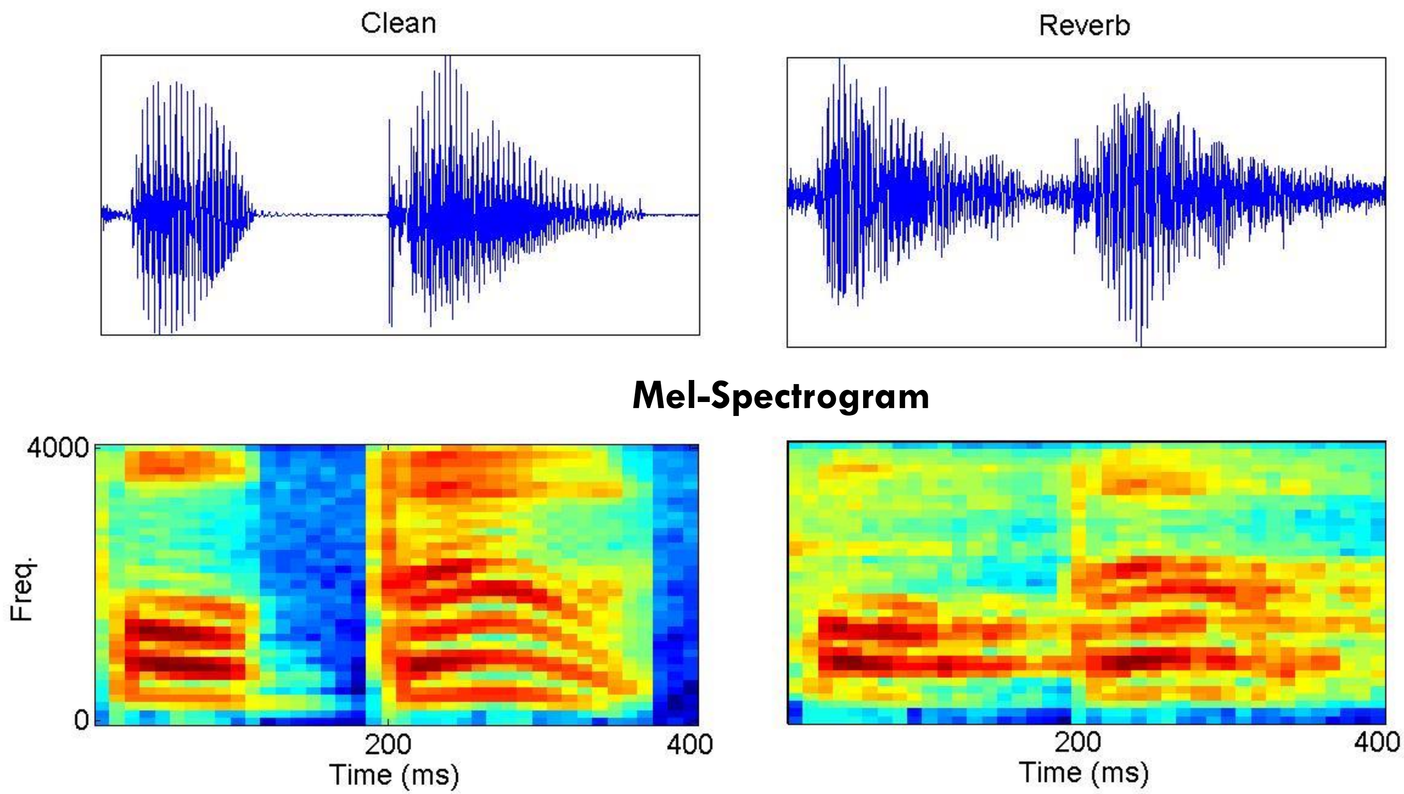
Gain Normalization in FDLP

- Sub-band decomposition into large number of sub-bands applied on a long-term DCT.
- Derive long-term sub-band envelopes with FDLP.
- Normalize the gain $G = 1$ on each sub-band to suppress reverberation artifacts.



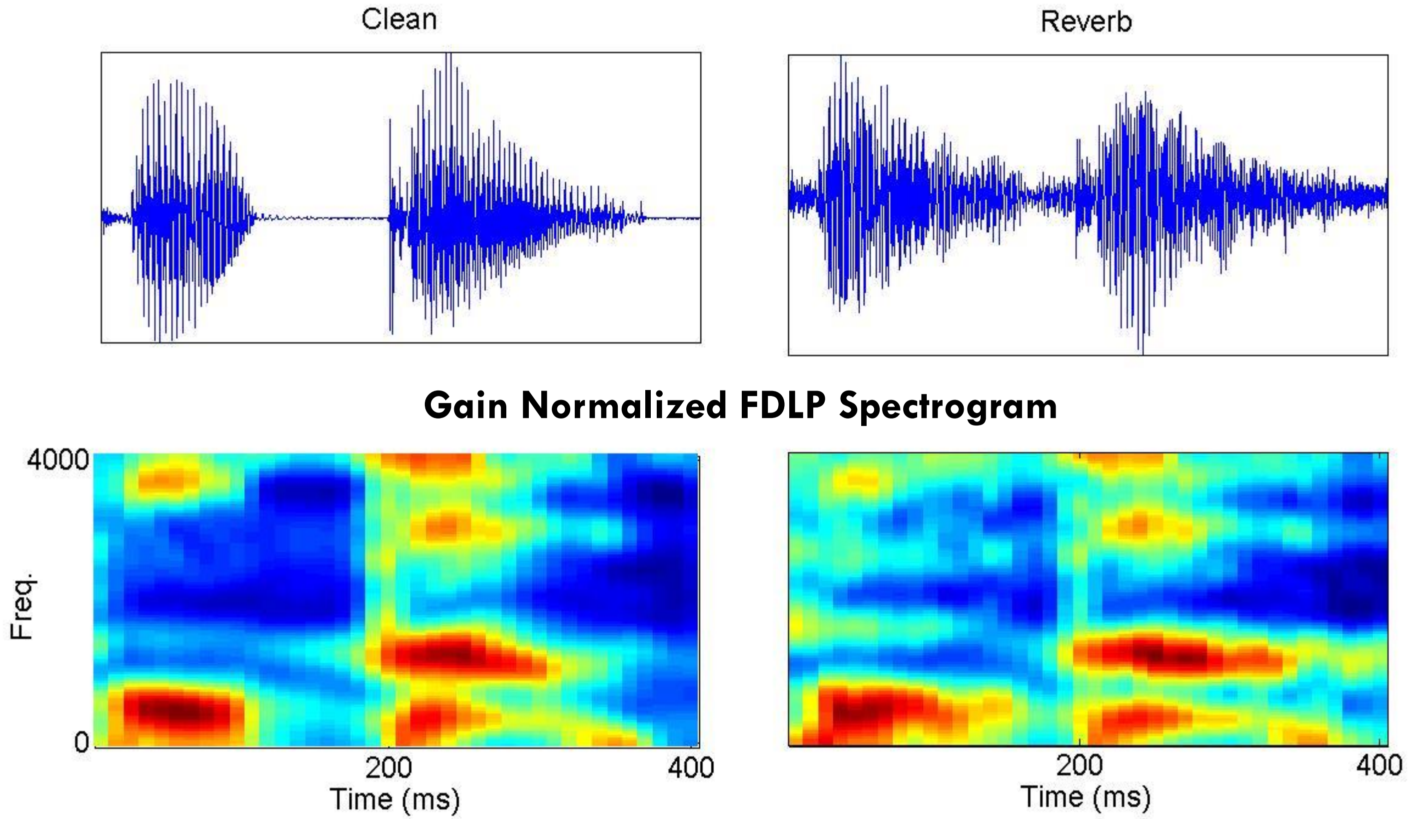
Normalizing Reverberation Artifacts

- Reverberation causes temporal smearing.
- Conventional mel spectrogram representation cannot provide invariant representation to these artifacts



Normalizing Reverberation Artifacts

- Removing the gain of FDLP model in long-term trajectories [Thomas, 2008] – suppresses reverberation artifacts.
  - Robust features extraction from FDLP spectrogram.

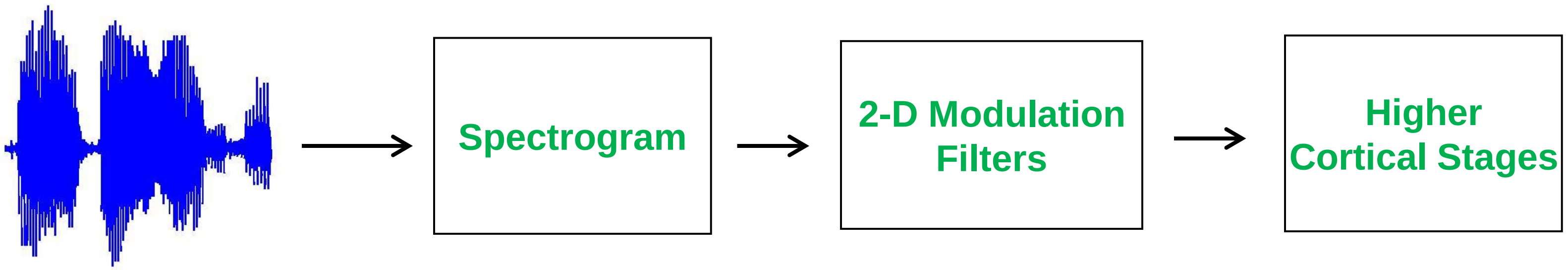


Outline

- Normalizing Reverberation Artifacts
- Bio-inspired Spectro-temporal Filtering Approaches
- Unsupervised Data Driven Features – ivectors
- Supervised Data Driven Features

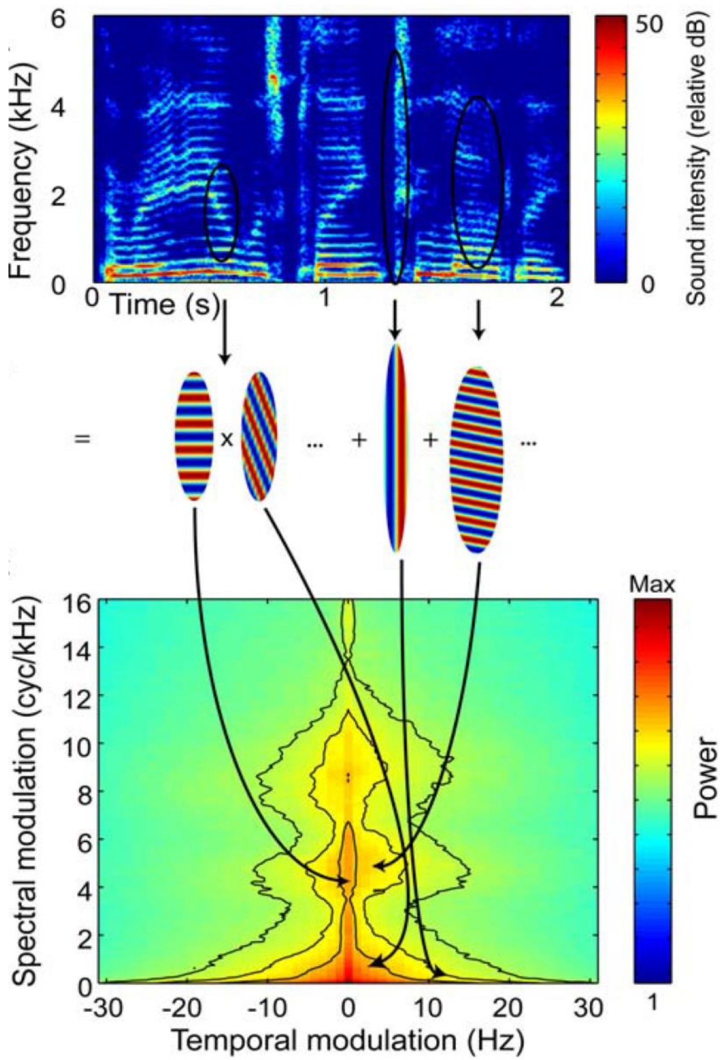
Bio-inspired Approaches

- Human audio perception is highly robust to noise and channel degradations.
- Several studies attributes the robustness to spectro-temporal filtering achieved in the cortical stages [Shamma, 2004].
  - 2-D modulation filters applied on the spectrogram with different high-pass/band-pass/low-pass characteristics.
  - Frequency along temporal axis – rate (Hz) and frequency along spectral axis – scale (cycles per kHz).



Bio-inspired Approaches

- Different speech sounds are characterized by different modulation properties. [Elliott, 2009].
- For example, vowels and stationary sounds are low-rate, while plosives and stops have high-rate.
- Most important speech information is band-pass in temporal modulations (1-16Hz) and low-pass in spectral modulations (0-3 cyc/kHz).



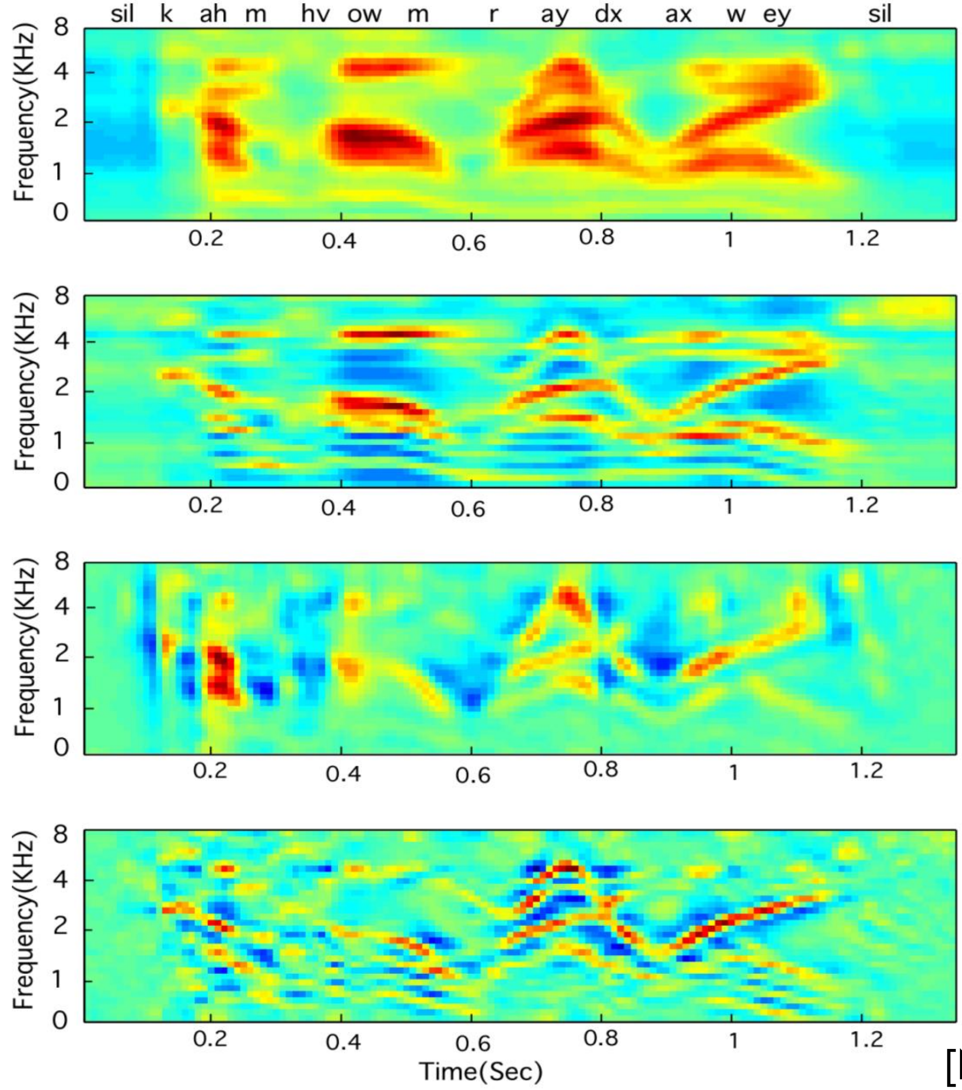
Emphasizing Spectro-temporal Modulations

Low-scale Low-rate

High-scale Low-rate

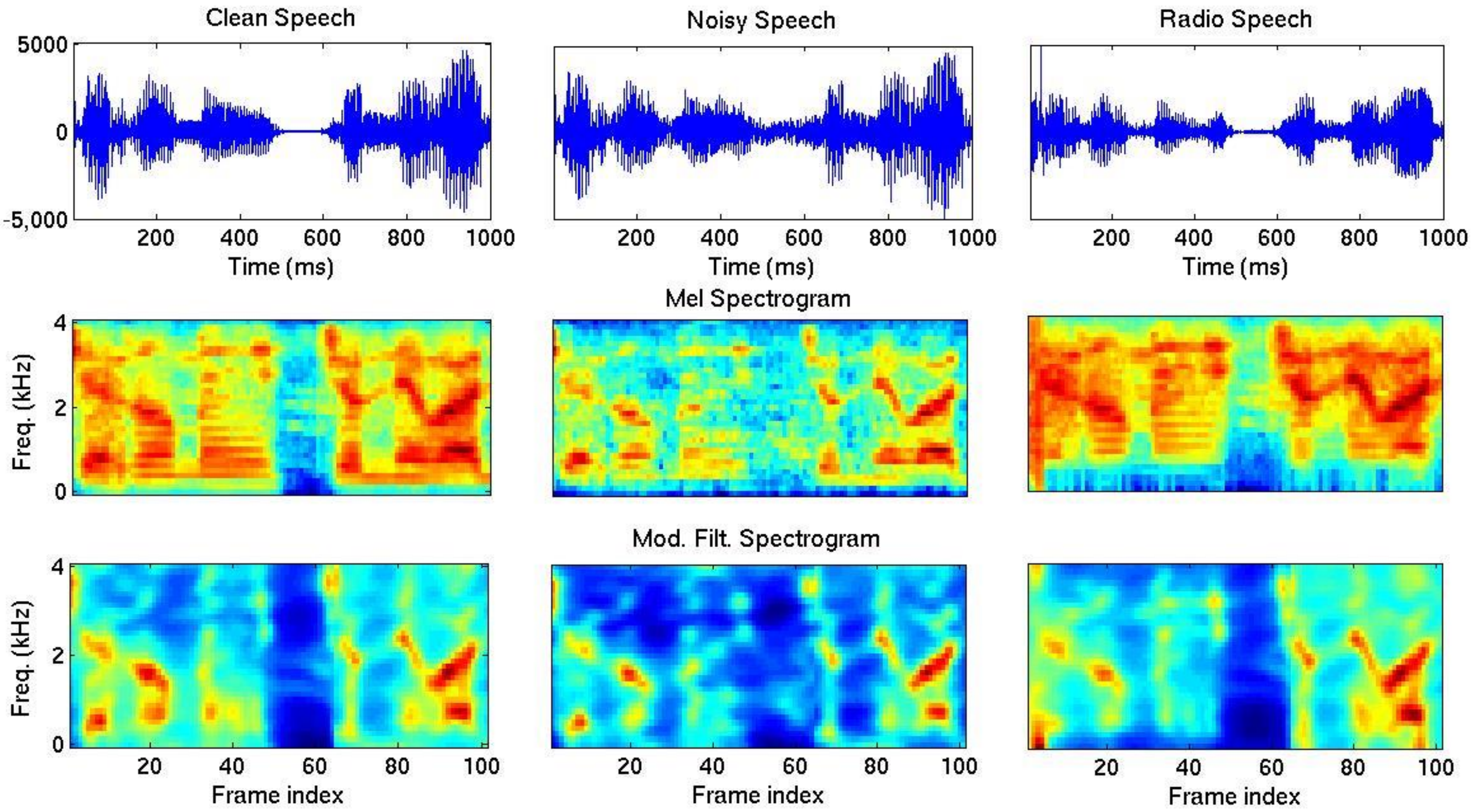
Low-scale High-rate

High-scale High-rate



[Nemala,2013]

Robustness to Noise



Outline

- Normalizing Reverberation Artifacts
- Bio-inspired Spectro-temporal Filtering Approaches
- Unsupervised Data Driven Features – ivectors
- Supervised Data Driven Features

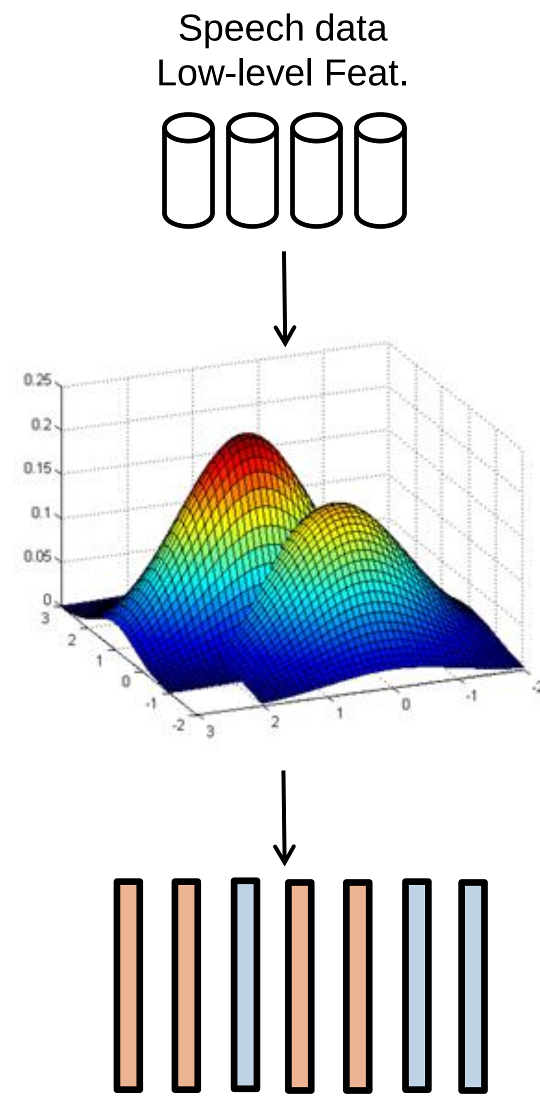
Data Driven Features

- Low-level features capture the acoustic signal information from the recording.
- For many applications, the statistical summary of the low-level features over the entire recording is useful.
  - Example, for speaker and language verification, these average statistic is a good representation and widely used.
  - Avoids dependency on the duration of the audio recording.

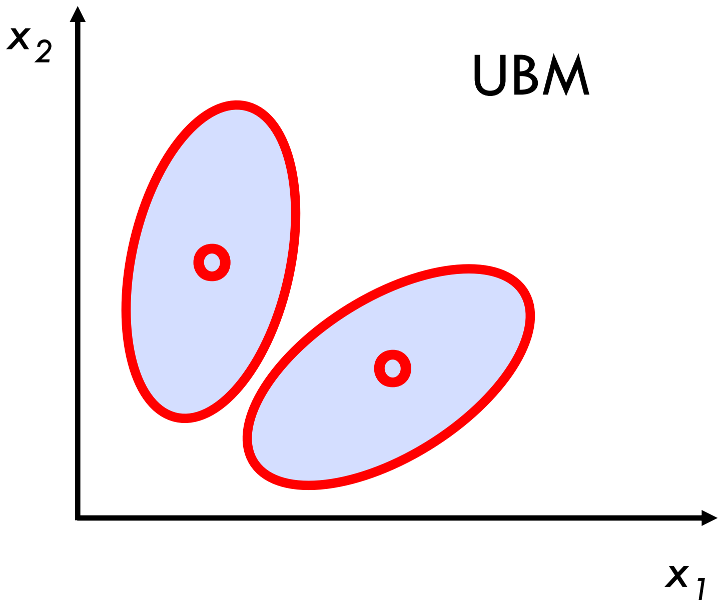
- This statistical summary can be derived from a universal background model (UBM).

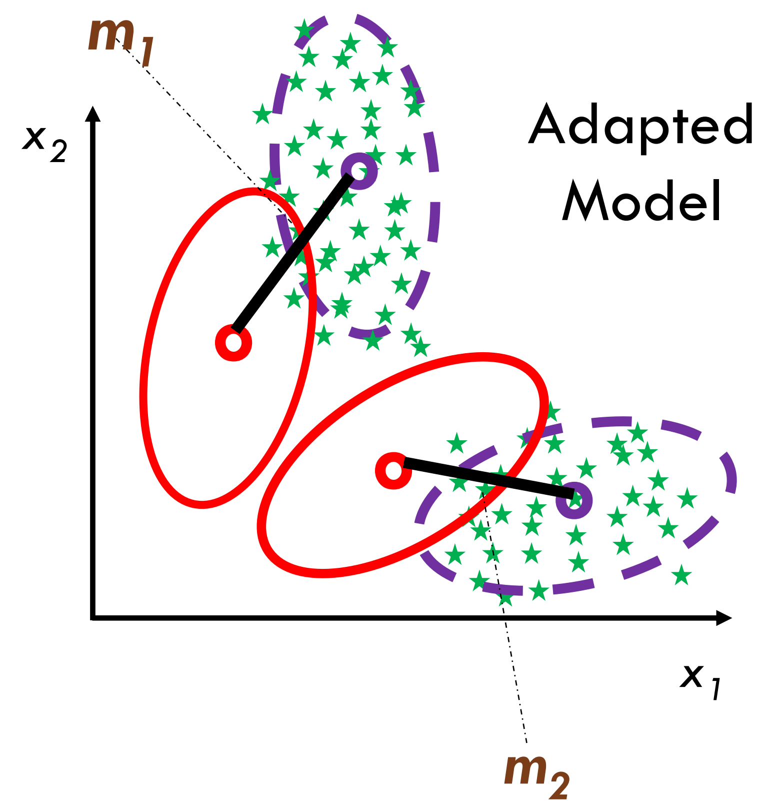
Overview of UBM Based Features

- Higher level features can be derived from lower level features by training an acoustic model. For example,
  - Derive low-level features like MFCC.
  - Training a Gaussian mixture model from a large number of speech recordings.
  - Aligning the low-level features with the GMM model.
  - Deriving model based features based on the alignment statistics.



Overview of i-vector Features





- The i-vector model is $\begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = Vy$ where y is the i-vector

i-vector Feature Extraction

- A popular GMM based feature is the i-vector [Kenny, 2005]
- The GMM-UBM with C mixtures is typically trained with a EM algorithm on large number of recordings from a corpus.
- Let $\lambda = \{\pi_c, \mu_c, \Sigma_c\}$ denote the parameters of the GMM-UBM

$$p_\lambda(x) = \sum_{c=1}^{C} \pi_c N(x ; \mu_c, \Sigma_c)$$

- Here, F is the dimension of μ_c and Σ_c is assumed diagonal $F \text{ x } F$
- Let supervector $\boldsymbol{M}_0$ be the concatenation of μ_c for $c = 1..C$ with dimension $CF \text{ x } 1$
- Let $\boldsymbol{\Sigma}$ be $CF \text{ x } CF$ block diagonal matrix with diagonal blocks $\Sigma_1 \ldots . \Sigma_C$

i-vector Feature Extraction

- Let $X(s)$ denote the low-level feature sequence for input recording with $X(s) = \{x_i^s, i = 1 \ldots H(s)\}$ where s denotes the recording index and i denotes the frame index, $H(s)$ denotes number of frames. Each $\boldsymbol{x}_i^s$ is a F dimensional feature vector.
- Let $\boldsymbol{M}(s)$ denote the CF x 1 supervector formed by the concatenation of means for the recording s.
- The i-vector model is

$$\boldsymbol{M}(s) = \boldsymbol{M}_{\mathbf{0}} + \boldsymbol{V}\boldsymbol{y}(s)$$

- $\boldsymbol{V}$ is of dimension CF x R known as total-variability matrix.
- The i-vector $\boldsymbol{y}(s)$ is of random vector of dimension R and assumed to be $N(\mathbf{0}, \boldsymbol{I})$

i-vector Model Estimation

- Outline of the iterative i-vector model estimation using EM algorithm (details of the proofs **Appendix-A**).
  - Step 1 – Finding the posterior distribution $p_{\boldsymbol{V},\lambda}(\boldsymbol{y}\,|X(s))$ of the i-vector given the recording $X(s)$ and the current estimates of $\boldsymbol{V}$.

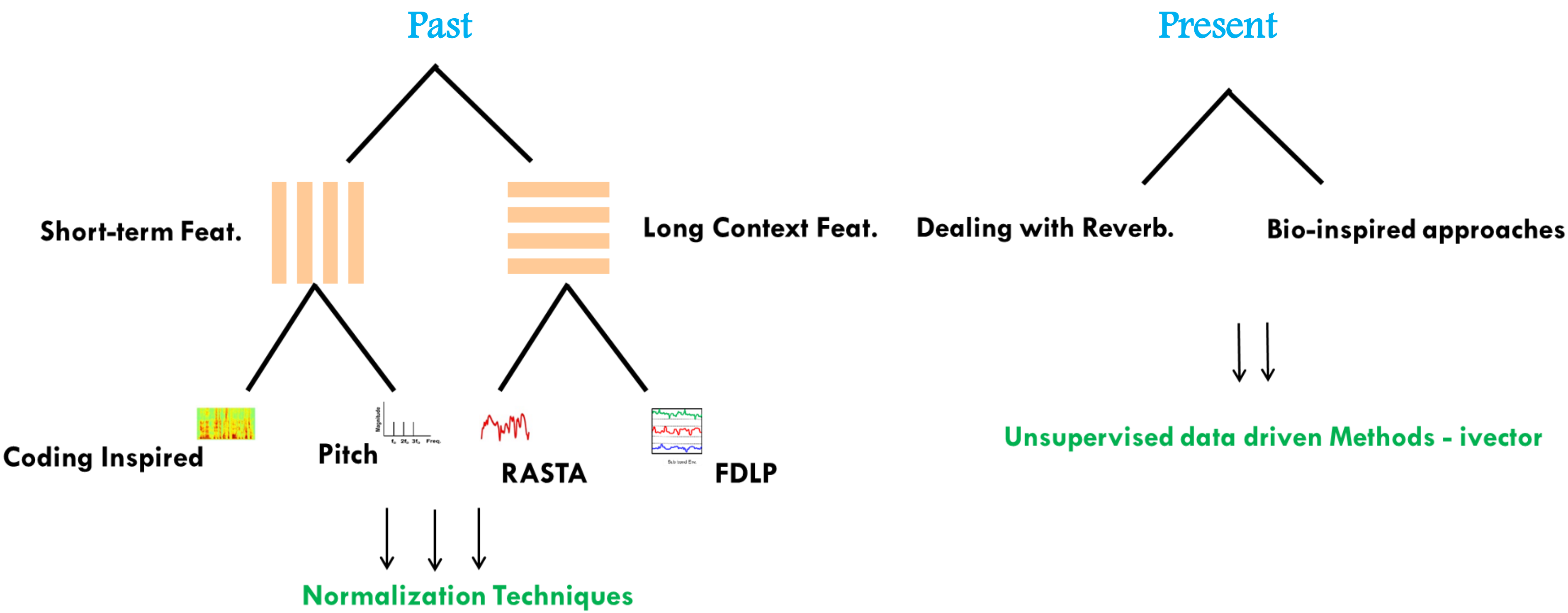
$$\boldsymbol{y}(s) = \underset{\boldsymbol{y}}{\operatorname{argmax}}\ p_{\boldsymbol{V},\lambda}(\boldsymbol{y}\,|X(s))$$

This posterior distribution is a Gaussian and the mode is the mean.

  - Step II – Update the estimate of $\boldsymbol{V}$ using the entire set of recordings and the $s = 1 \ldots .S$ and the estimates $\boldsymbol{y}(s)$

$$\boldsymbol{V} = \underset{\boldsymbol{V}}{\operatorname{argmax}} \prod_{s=1}^{S} p_{\boldsymbol{V},\lambda}(X(s)\,|\,\boldsymbol{y}\,(s))$$

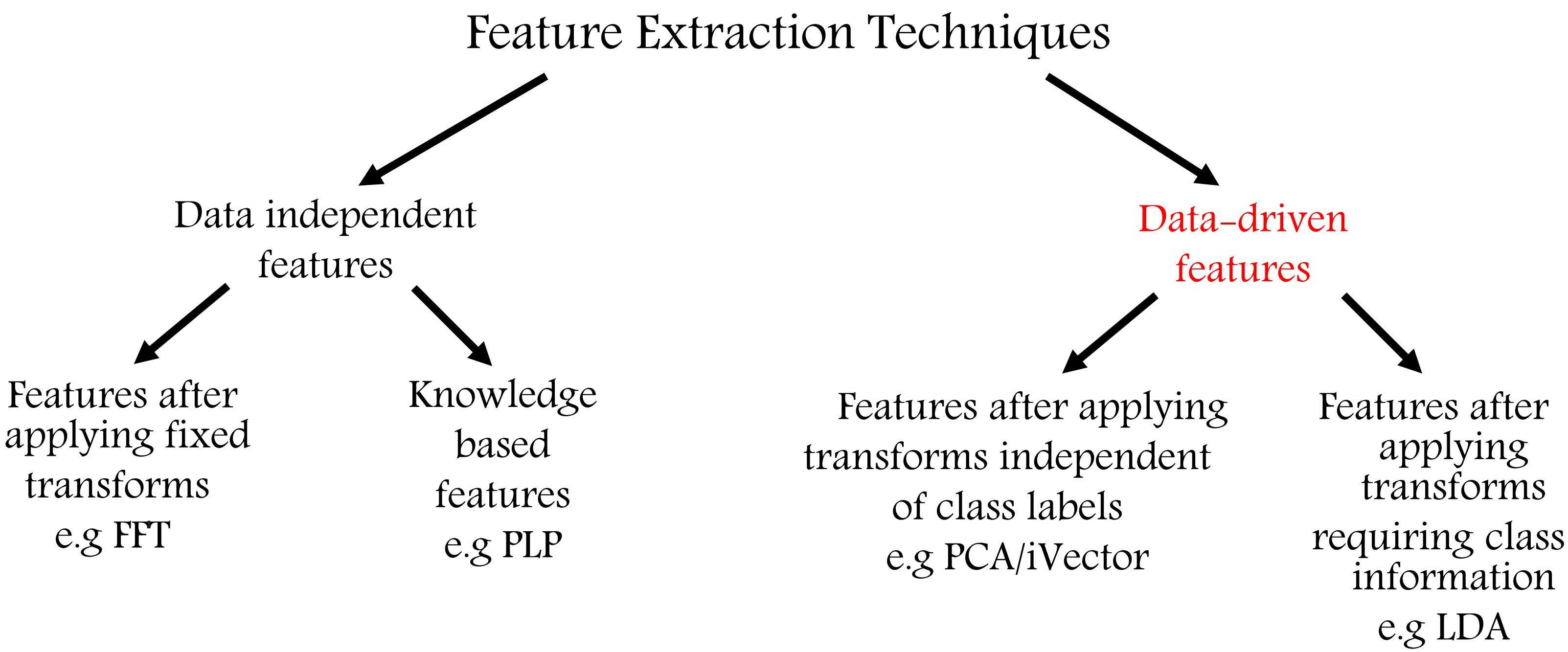
Discussion Summary



Outline

- Normalizing Reverberation Artifacts
- Bio-inspired Spectro-temporal Filtering Approaches
- Unsupervised Data Driven Features – ivectors
- Integrating Data with Feature Extraction

Broad classification of data-driven feature extraction techniques



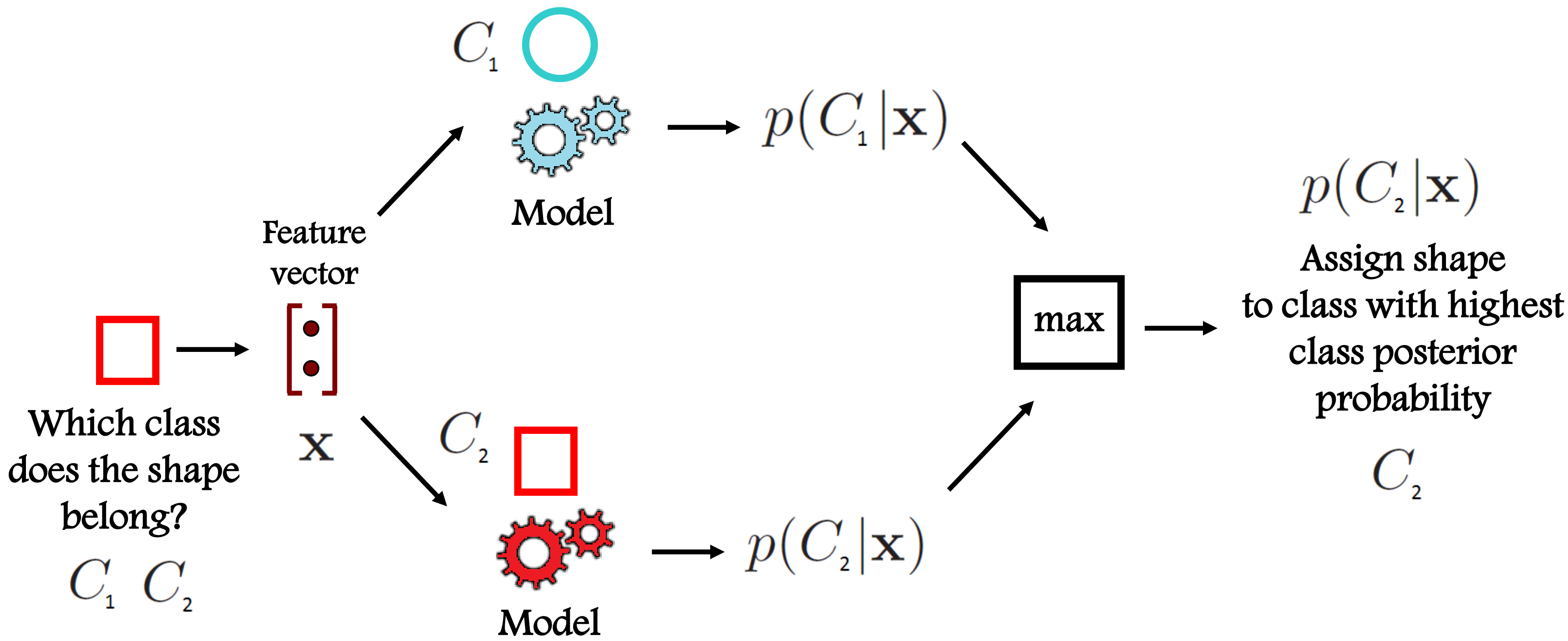
Integrating data with feature extraction

- Introduction
- Variants of data-driven features
  - PCA/LDA
  - Manifold Learning
  - Neural Networks
  - Application specific training criteria

Improving feature extraction with data

- Lower level acoustic features can be transformed to better represent the data and task at hand
- The transformation can be learnt from the data itself

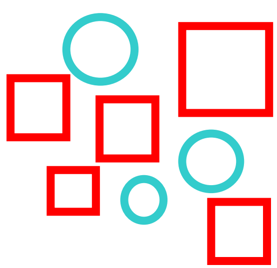
Probabilistic models for classification



Probabilistic models for classification

$$p(C_j|\mathbf{x}) = \frac{p(\mathbf{x}|C_j)p(C_j)}{p(\mathbf{x})}$$

Model directly estimates $p(C_j|\mathbf{x})$ — Discriminative Models



Data used to train models

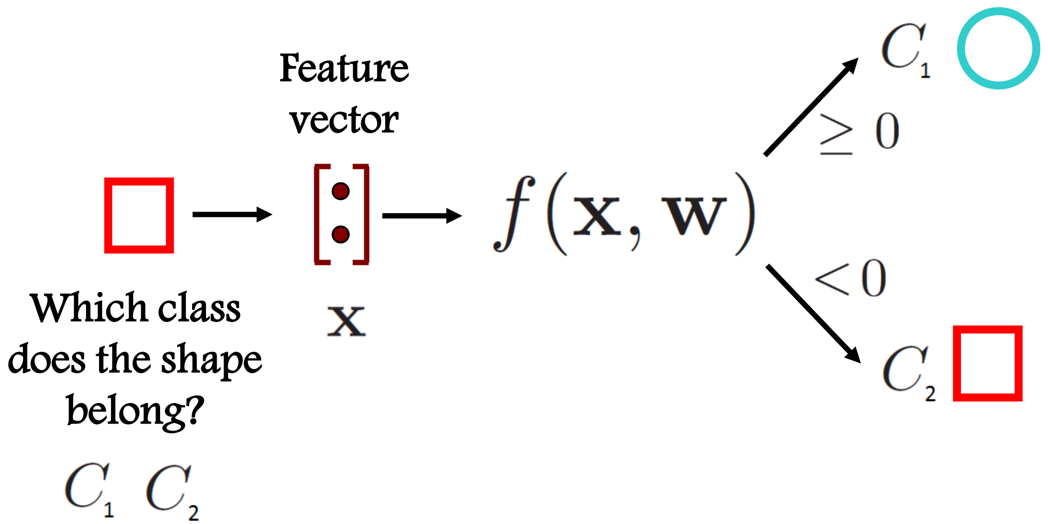
Indirect estimates via $p(\mathbf{x}|C_j)\,p(C_j)$ — Generative Models

Discriminant functions for classification

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + w_0$$

$\mathbf{w}$ – Weight vector

w_0 – bias



Discriminant functions for ~~classification~~ feature extraction

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + w_0$$

- The transform is being applied to the feature vector
- Transform is learnt from data – **information from the training data can be incorporated into feature extraction**
- Projection to one dimension might be disadvantageous – but useful to improve features

Discriminant functions for ~~classification~~ feature extraction

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + w_0$$

- The components of $\mathbf{w}$ can be adjusted to maximally separate classes
- Example – A projection such that there is maximal separation between class means and variance within each class is minimum
  - Fisher's linear discriminant

$$\mathcal{F}(\mathbf{w}) = trace(S_w^{-1} S_b)$$

Generalized linear discriminant functions

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

$$\downarrow$$

$$y(\mathbf{x}, \mathbf{w}) = f\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})\right)$$

$$y(\mathbf{x}, \mathbf{w}) = f\left(\sum_{j=1}^{M} w_j \phi_j(\mathbf{x})\right)$$

Linear combination

Basis functions

Predicted output

Activation Function

Input vector

Weight coefficients

Generalized linear discriminant functions

Perceptron model for classification

Generalized linear discriminant function

$$y(\mathbf{x}, \mathbf{w}) = f\left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})\right)$$

Activation Function

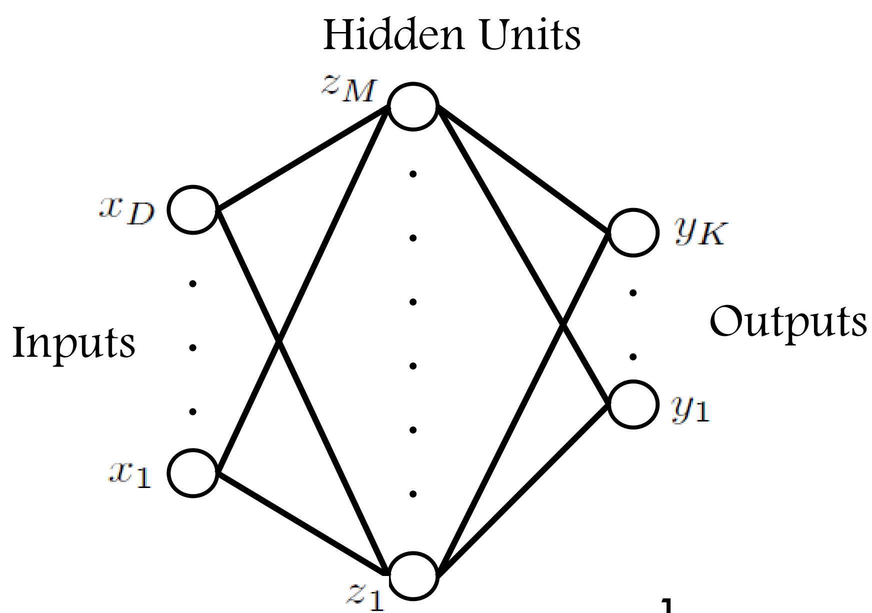
$$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0 \end{cases}$$

Perceptron training criteria

$$E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}_n t_n$$

Generalized linear discriminant functions

Neural Network Models for classification



$$y_k(\mathbf{x}, \mathbf{w}) = \underset{4}{\sigma} \left(\sum_{j=1}^{M} w_{kj}^{(2)} \underset{2}{h} \left(\overbrace{\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}}^{1} \right) + w_{k0}^{(2)} \right)$$

$$\underbrace{\sum_{j=1}^{M} w_{kj}^{(2)} h \left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)}}_{3}$$

Generalized linear discriminant functions

Neural Network Models for ~~classification~~ feature extraction

$$y(\mathbf{x}, \mathbf{w}) = f\left(\sum_{j=1}^{M} w_j \phi_j(\mathbf{x})\right)$$

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right)$$

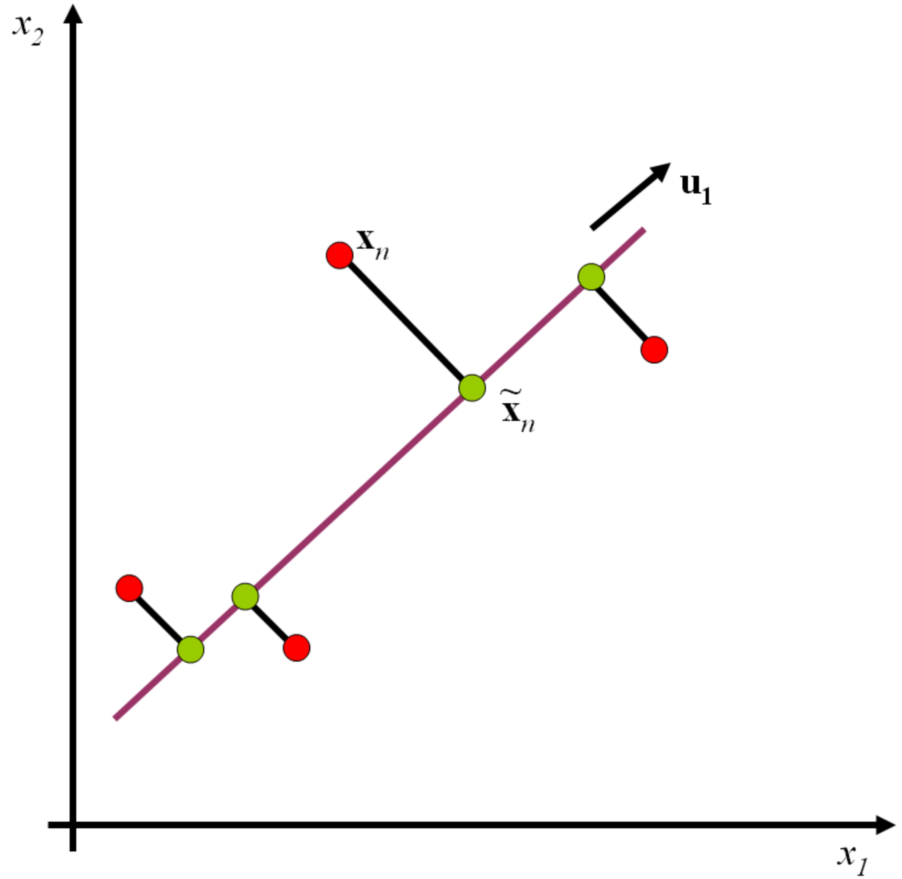
Data-driven feature transforms

Posterior probabilities of output classes

Data driven transformations without class information

- Learn class-independent distributions of the data with certain constraints
- Example – Find an orthogonal projection of data onto a lower dimensional linear space such that the variance of the projected data is maximized
  - Principal Component Analysis or Karhunen-Loeve transform

Data driven transformations without class information

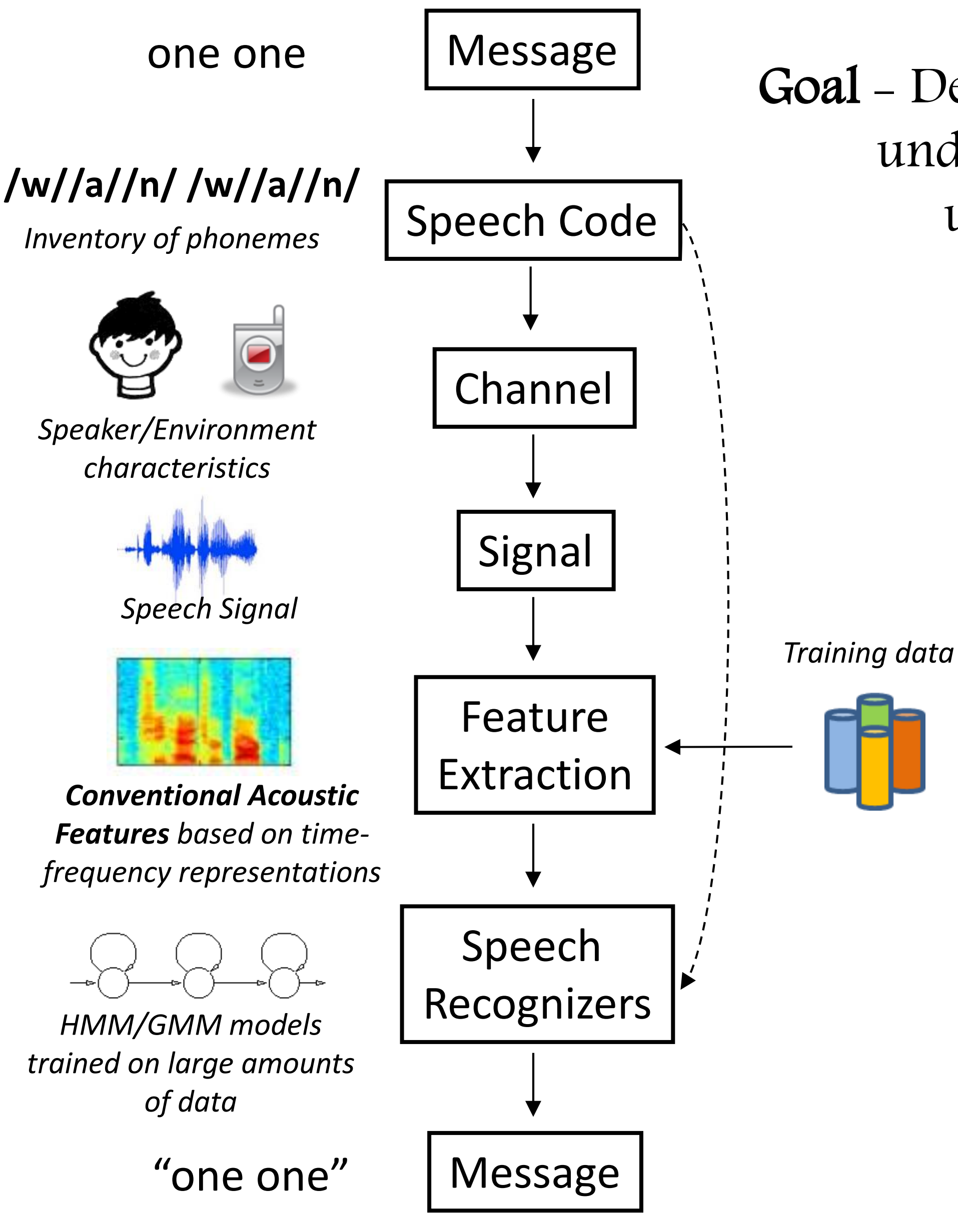


An alternate formulation of PCA is based on minimizing the sum-of-squares of the projection errors

Integrating data with feature extraction

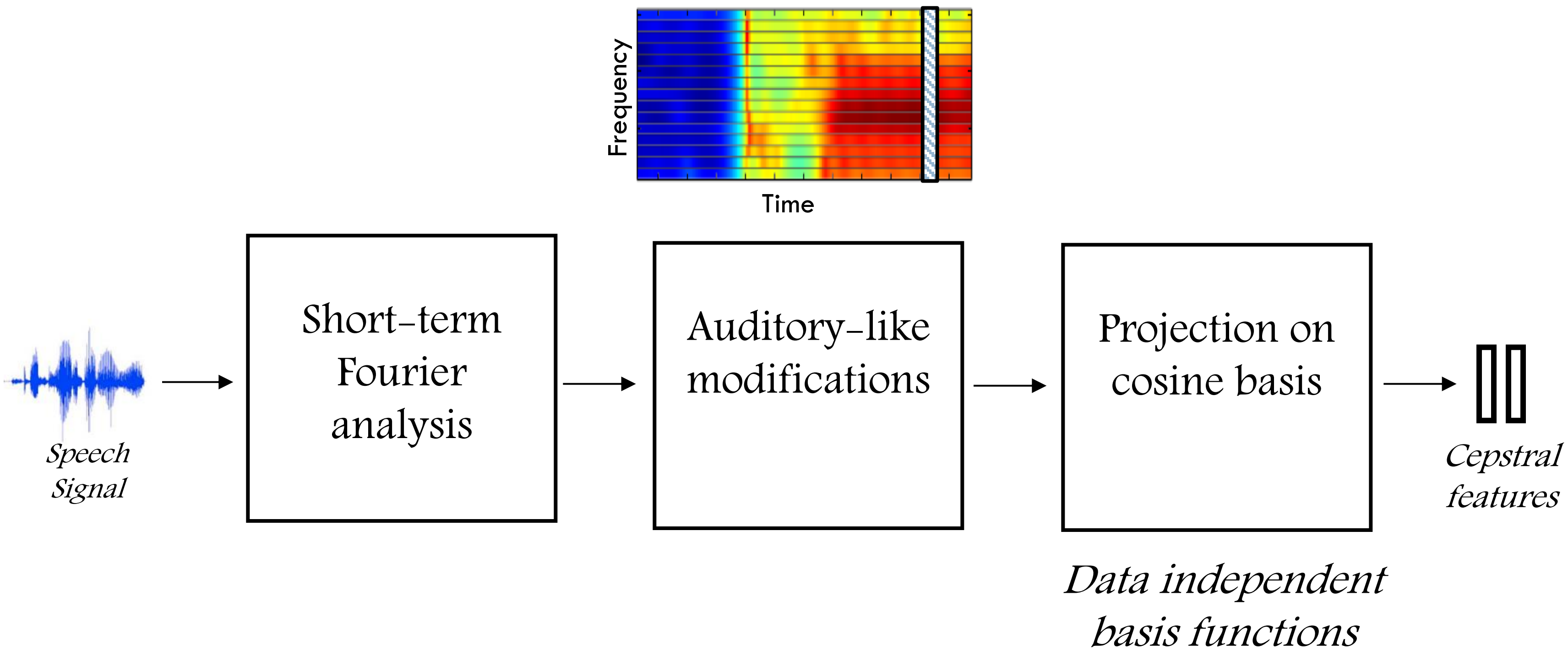
- Introduction
- Variants of data-driven features based on
  - PCA/LDA
  - Manifold Learning
  - Neural Networks
  - Application specific training criteria

Data-driven features for ASR

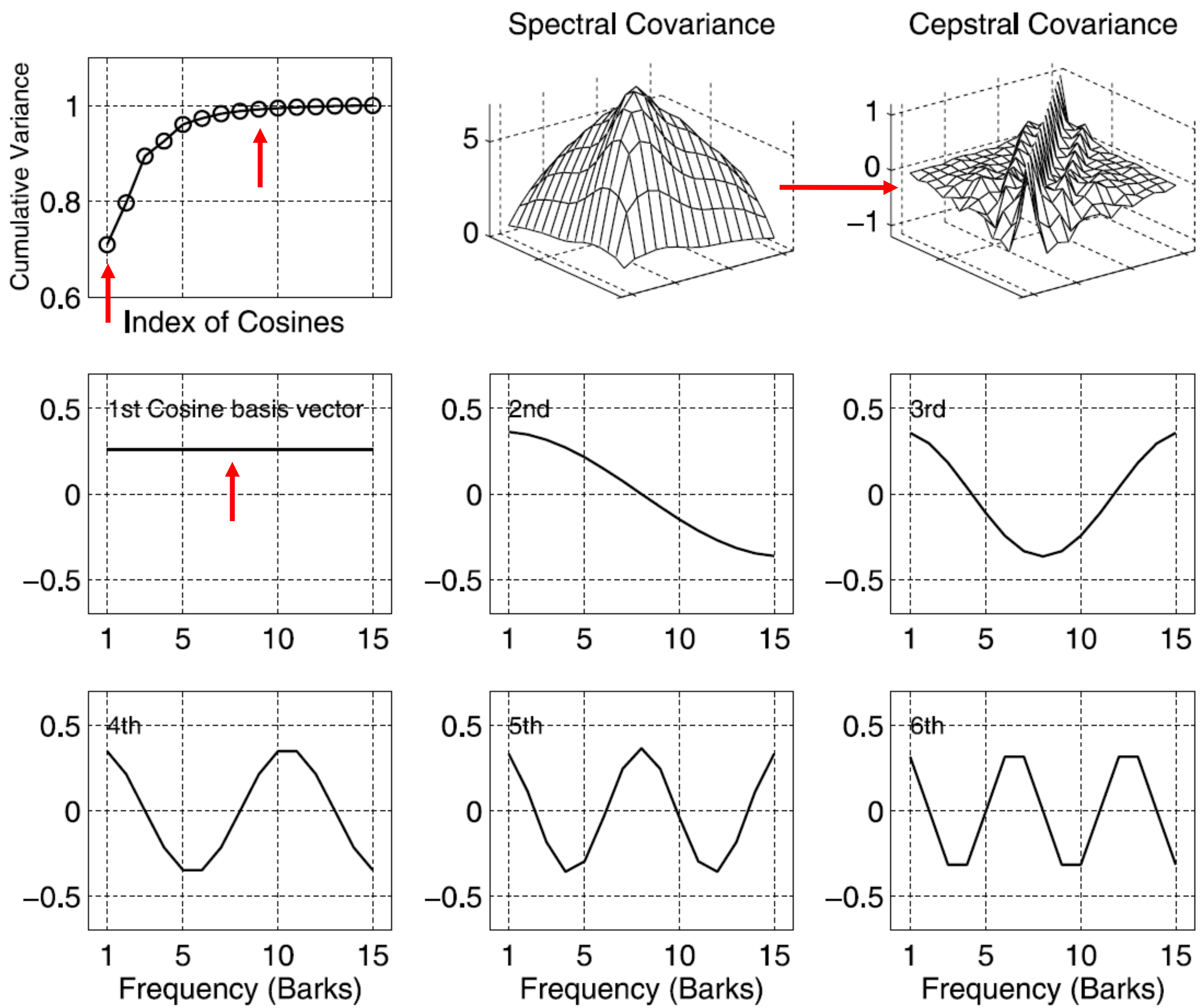


Goal – Design features more representative of the underlying message (speech code with un-desired speaker and channel characteristics removed)

Data-driven features for ASR



Data-driven features – projection on fixed basis



Variance captured by the first cosine basis vectors amounts to 70% of the total variability

The first 10 cosine basis capture almost the entire variance present in the data

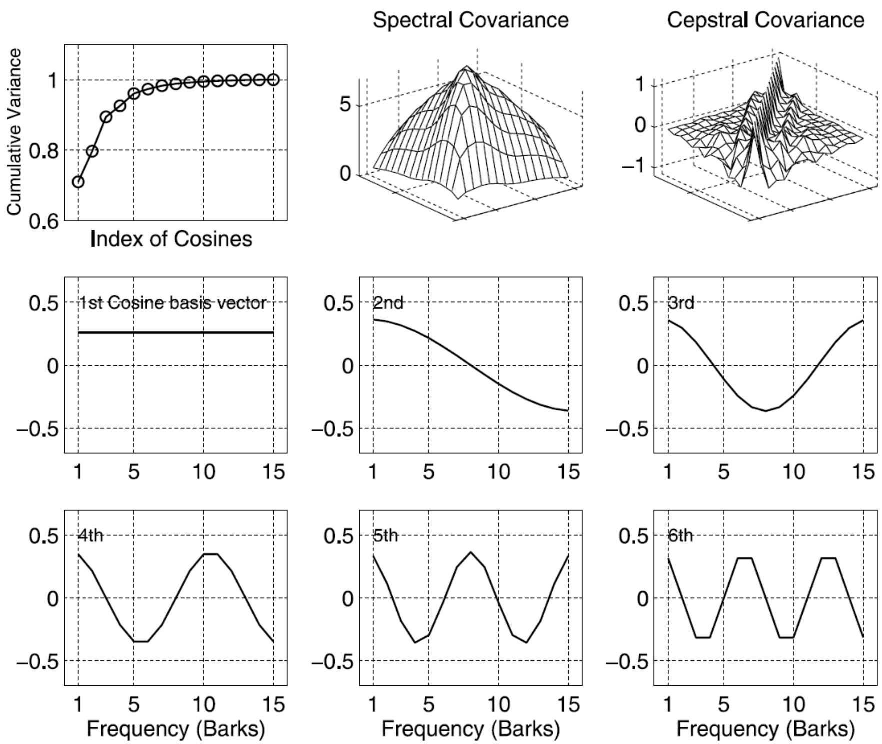
Spectral covariance matrix far from diagonal – projection on the first 8 vectors, makes it partially diagonalized

First cosine basis is flat across all bands – majority of the variance in the speech spectrum is caused by variation in the average energy

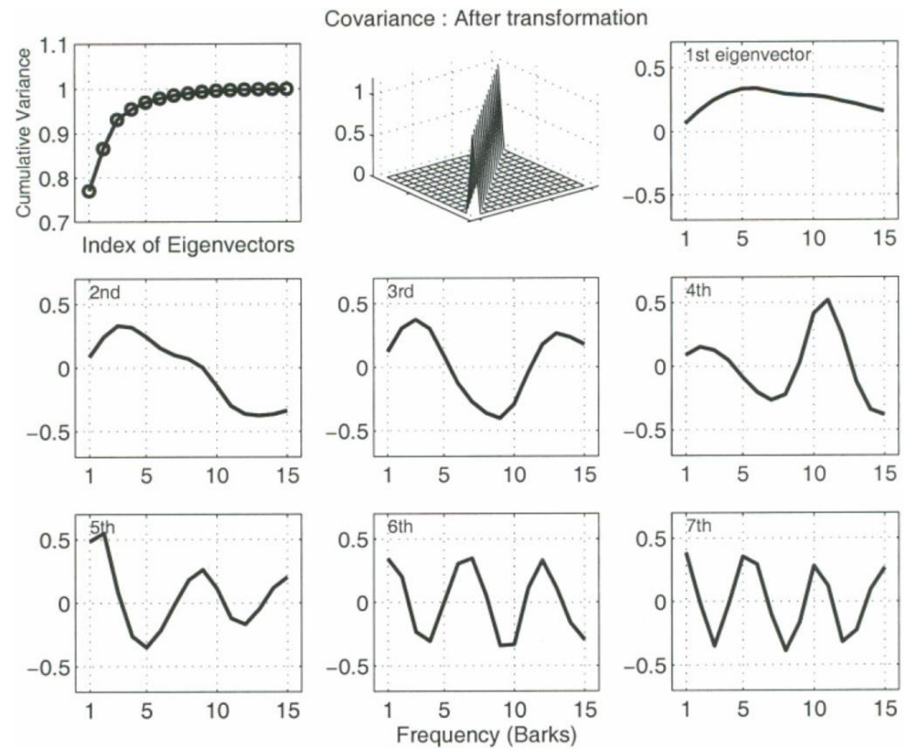
N. Malayath and H. Hermansky, "Data-driven spectral basis functions for automatic speech recognition", Speech Communications 2002.

Data-driven features - PCA

DCT basis functions



PCA basis functions

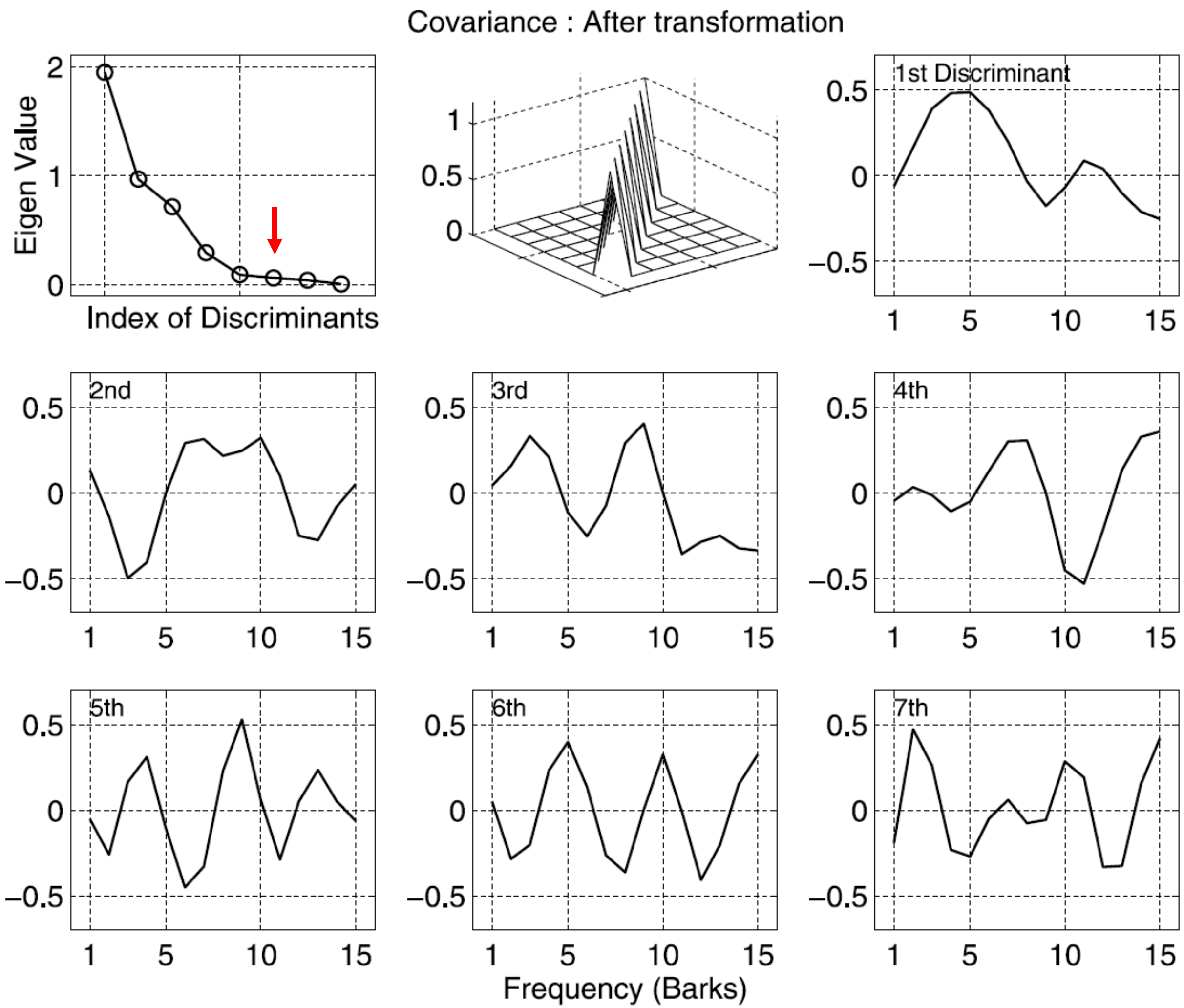


The PCA basis vectors are **reminiscent** of cosine functions – first vector measures spectral energy, higher basis functions similar to cosine-like functions with decreasing periods

DCT basis **very similar** to PCA basis – DCT is indeed a good choice for decorrelation and dimensionality reduction, which also results in minimum reconstruction error

N. Malayath, "Data-driven methods for extracting features from speech", PhD Thesis, OGI 2000.

Data-driven features - LDA



The first 7 Eigen vectors seem to have significant Eigen values

The first discriminant appears to evaluate spectral energy in the first formant region

The second and third discriminants seem to be focusing on spectral ripples in the central part of the critical-band spectrum.

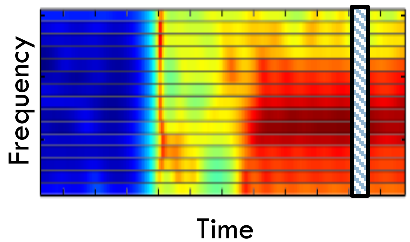
The fourth one analyzes the portion of the spectrum that lies above 5 barks.

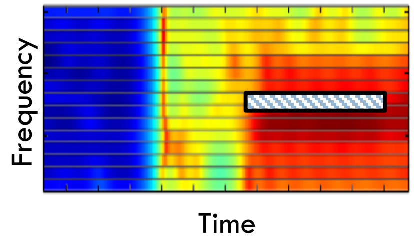
The fifth discriminant vector is sensitive to spectral ripples with a 5 bark period.

The fifth and sixth discriminants are very similar to sinusoidal functions.

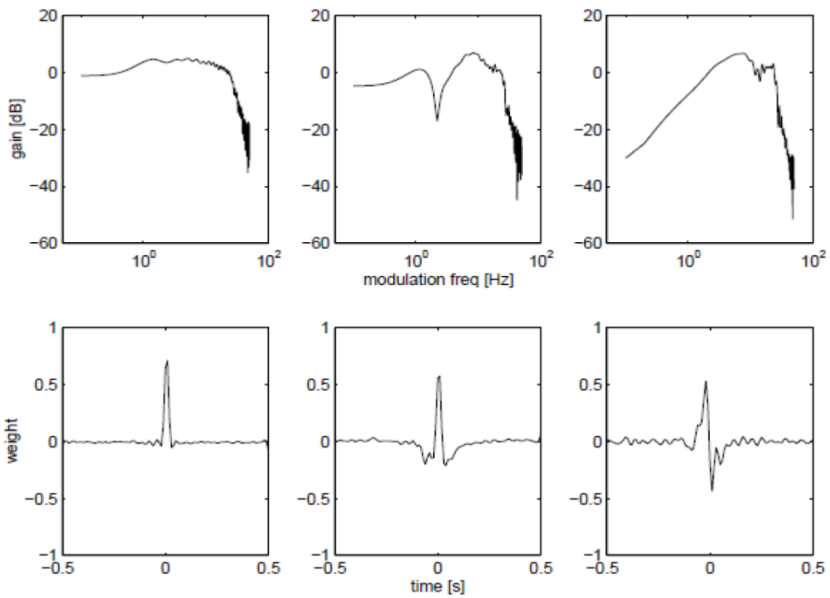
N. Malayath and H. Hermansky, "Data-driven spectral basis functions for automatic speech recognition", Speech Communications 2002.

Data-driven features - LDA

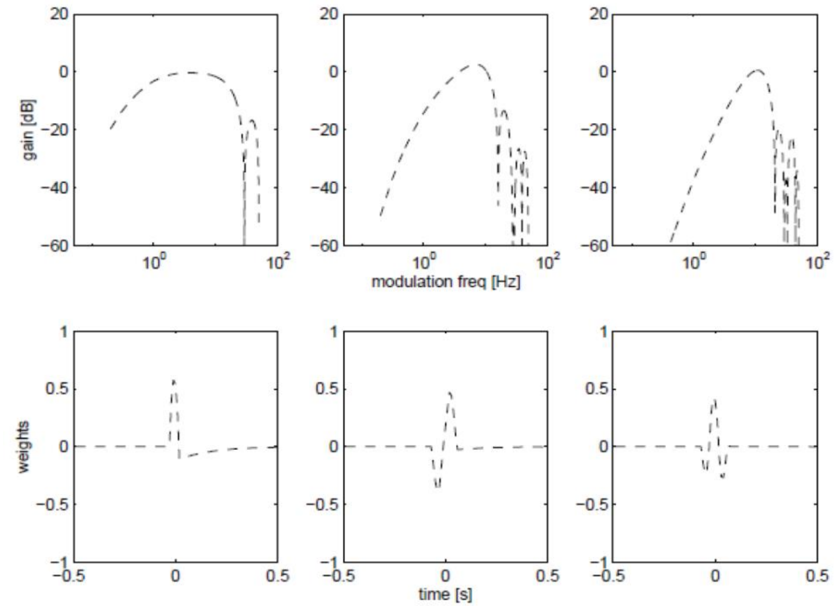




LDA basis functions



RASTA filter and the RASTA filter combined with the delta and double-delta filters



The first discriminant vector, explains about 80% of the variability in the data. The frequency response of the first discriminant vector agrees well with the frequency response of hand designed RASTA filter

The frequency characteristic of the second and third discriminant vectors are somewhat comparable to the second and third orthogonal polynomials approximating the time trajectory of the feature within a 9 frame time interval.

S. van Vuuren and H. Hermansky, "Data-driven design of RASTA-like filters", Eurospeech 1997.

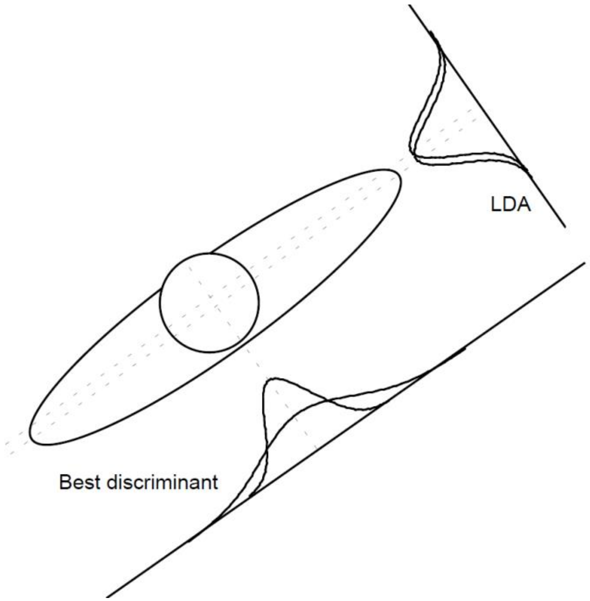
Extensions of LDA

- LDA is related to the maximum-likelihood estimation of a Gaussian model with two *a priori* assumptions
  - All class-discrimination information resides in a *p*-dimensional sub-space of the *n*-dimensional feature space
  - The within-class variances are equal for all the classes

N. A. Campbell, "Canonical Variate analysis - A General Model Formulation", 1984.

Extensions to LDA

- LDA is suited for classifier models where the class distributions have equal variance
- LDA is not the optimal transform when the class distributions are heteroscedastic



Two classes have almost the same mean, but the variances are different in one direction. It would be best for the classifier, if the data was projected along the direction where the variances are different and un-equal variance models can be used in the classifier design

N. Kumar and A. Andreou, "A Generalization Of Linear Discriminant Analysis In Maximum Likelihood Framework", 1996

Extensions to LDA - HLDA

Let θ be a full rank linear transformation that transforms x into y. First p columns of θ carry components of y that carry class-discrimination information. Partition the parameter space of the means and variances in y as -

Different across classes

Common for all classes

$$\mu_j = \begin{bmatrix} \mu_{j,1} \\ \vdots \\ \mu_{j,p} \\ \mu_{0,p+1} \\ \vdots \\ \mu_{0,n} \end{bmatrix} = \begin{bmatrix} \mu_j^p \\ \mu_0 \end{bmatrix}$$

$$\Sigma_j = \begin{bmatrix} \Sigma^p_{j(p\times p)} & 0 \\ 0 & \Sigma^{(n-p)}_{(n-p\times n-p)} \end{bmatrix}$$

The probability density of x_i under the preceding model is given as

$$P(x_i) = \frac{|\theta|}{\sqrt{(2\pi)^n |\Sigma_{g(i)}|}} \; exp\left(-\frac{(\theta^T x_i - \mu_{g(i)})^T \Sigma^{-1}_{g(i)} (\theta^T x_i - \mu_{g(i)})}{2} \right)$$

Techniques based on the generalized EM algorithm are then used to find the best transform.

N. Kumar and A. Andreou, "A Generalization Of Linear Discriminant Analysis In Maximum Likelihood Framework ", 1996

Extensions of LDA

- LDA has been used from very early on for speech recognition
  - To improve features
    - M. Hunt, "A statistical approach to metrics for word and syllable recognition," 1979.
    - P. Brown, "The acoustic-modeling problem in automatic speech recognition," 1987.
  - To improve the discrimination between HMM states
    - G. Doddington, "Phonetically sensitive discriminants for improved speech recognition," 1989.
  - As feature rotation and reduction technique in a maximum likelihood setting
    - E. Schukat-Talamazzini, J. Hornegger, and H. Niemann, "Optimal linear feature transformations for semi-continuous hidden Markov models," 1995.
- An alternate definition of HLDA (sometimes referred to as HDA) uses weighted contributions of classes to the LDA objective function
    - G Saon, "Maximum likelihood discriminant feature spaces" 2000
- When $p = n$ (no dimensionality reduction), HLDA transformation becomes a diagonalization transform – A popular such transform is the Maximum Likelihood Linear Transform (MLLT)
    - R. Gopinath. "Maximum likelihood modeling with Gaussian distributions for classification",1998.

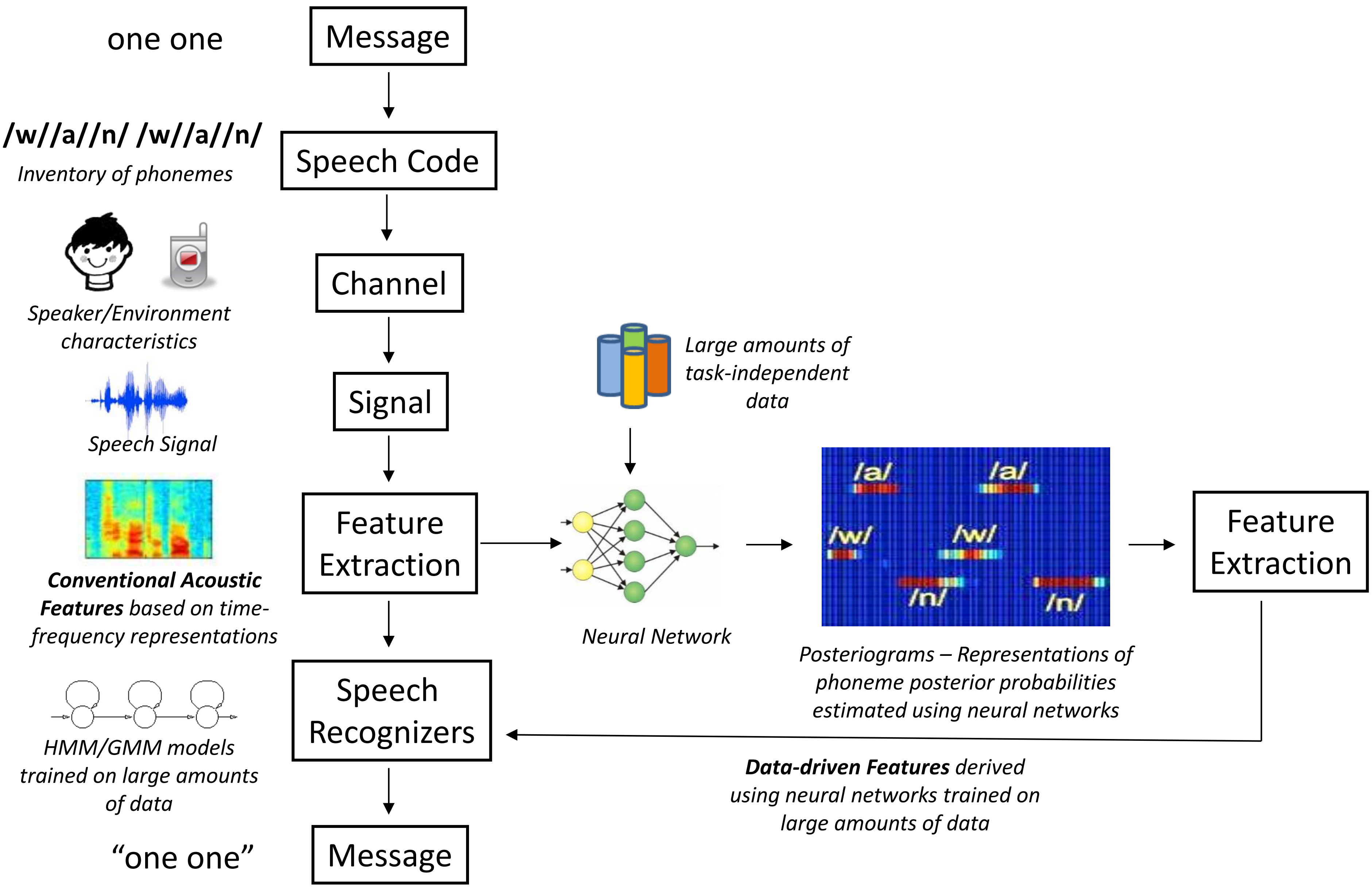
Manifold Learning

- While the PCA and LDA techniques described above are useful in describing transforms in the Euclidean space, manifold based techniques characterize data as being embedded in a manifold space

- Speech is produced by a set of articulators that have only few degrees of freedom – hence there should exist a lower dimensional manifold of the high dimensional space of all possible sounds

- Learning problems are usual solved as optimization problems or as generalized eigenvector problems.
  - A. Jansen and P. Niyogi, "Intrinsic Fourier analysis on the manifold of speech sounds," 2006.
  - V. Jain and L. Saul, "Exploratory analysis and visualization of speech and music by locally linear embedding," 2004.
  - A. Errity and J. McKenna, "An investigation of manifold learning for speech analysis," 2006.

Data-driven features for ASR



Neural Networks For Feature Extraction

- Introduction
- Types of Neural Networks
  - Deep Neural Networks
  - Deep Belief Networks
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Autoencoder Networks

Neural network based features – **Key differentiators** – *Training criteria*

- Neural networks are trained to **discriminate between output classes** using non-linear basis functions, with its cross-entropy training criteria.

- For acoustic modeling in speech recognition, MLP based systems **estimate posterior probabilities of output classes** [Appendix-B] like phonemes, conditioned on the input features.

- Training can also be **scaled efficiently to work on large amounts of training data**.

Neural network based features – **Key differentiators** – *Input assumptions*

- Neural networks can **model high dimensional input features** without any strong assumptions about the probability distribution of these features.

- Several different kinds of correlated feature streams can also be integrated together since there are also **no strong assumptions on statistical independence**

Neural network based features – **Key differentiators** – *Output representations*

For speech recognition, MLP based acoustic models –

- trained on large amounts of data from a diverse collection of speakers and environments, can achieve **invariance to these unwanted variabilities**.

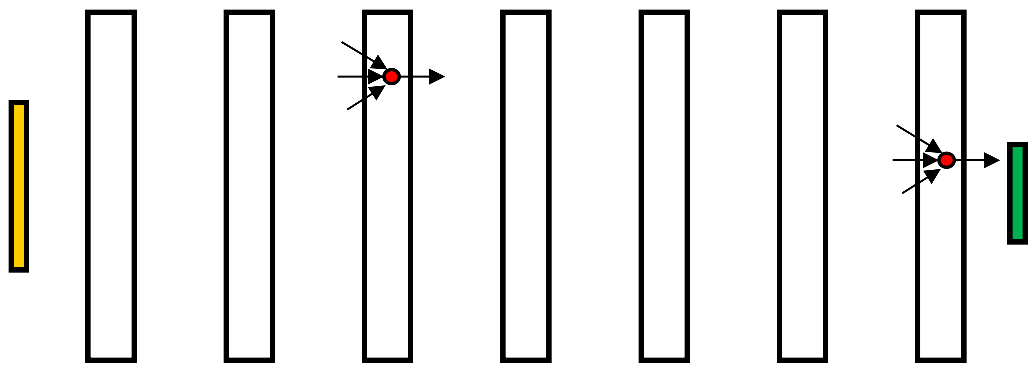
- outputs from several networks trained on different feature representations can be **combined in a multi-stream fashion to improve the final posterior estimations**.

Neural network based features – Variants - DNNs

- A **deep neural network (DNN) is a** feed-forward, artificial neural network that has more than one layer of hidden units between its inputs and its outputs

Intermediate layers $y_j = \text{logistic}(x_j) = \frac{1}{1+e^{-x_j}}, \qquad x_j = b_j + \sum_i y_i w_{ij}$

Output layer $p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)}$

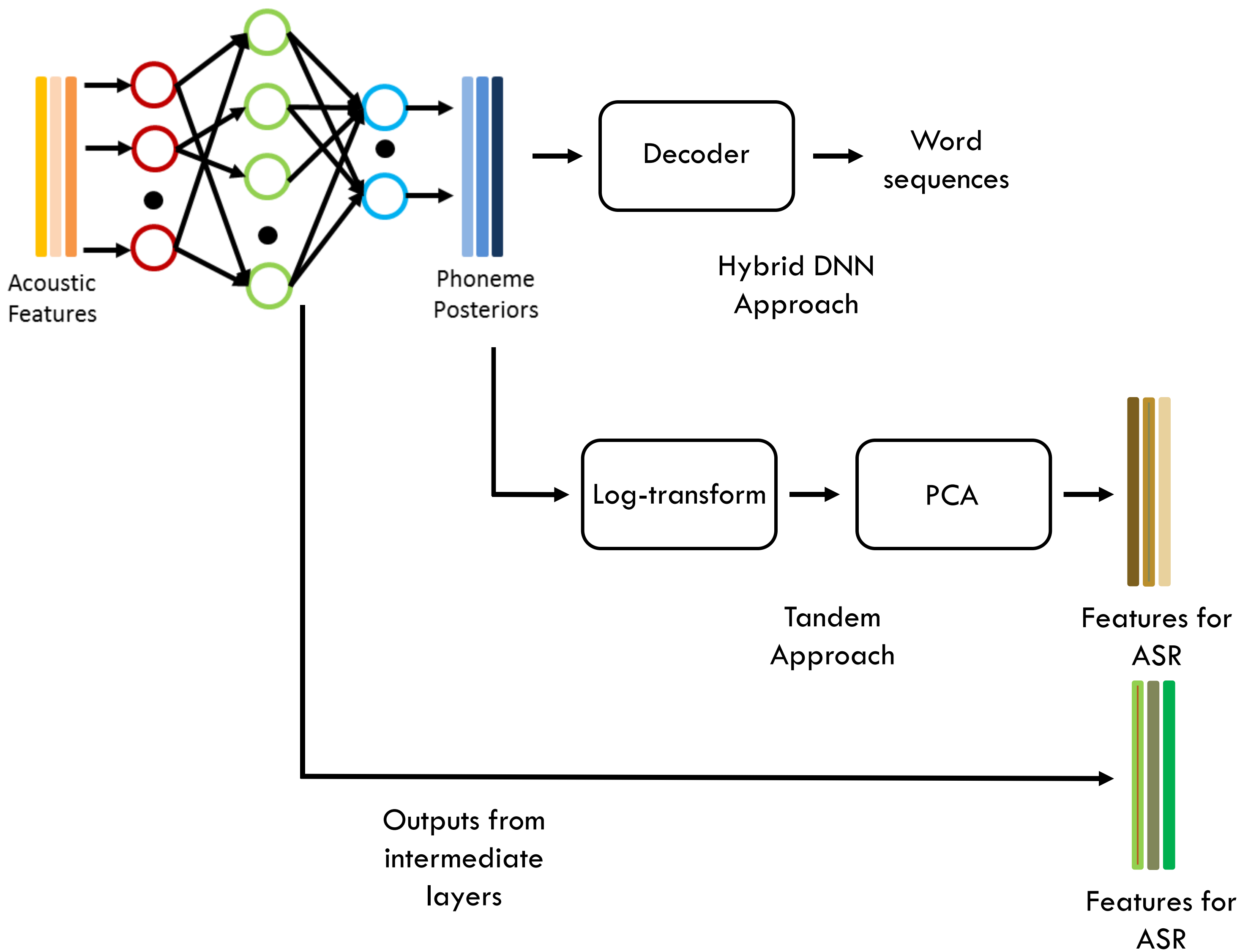


Training criteria cost function $C = -\sum_j d_j \log p_j$

Parameter updates $\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) - \epsilon \frac{\partial C}{\partial w_{ij}(t)}$

G. Hinton et.al , "Deep Neural Networks for Acoustic Modeling in Speech Recognition ", 2012

Neural network based features



Neural network based features

Model	400 hours – Broadcast News (dev04f)	300 hours – Conversational Telephony (Hub 5)
Baseline GMM/HMM	16.0	14.5
Hybrid DNN	15.1	12.2
Neural network features	13.1	11.5

T. Sainath et.al, "Deep convolutional neural networks for LVCSR ", 2013

Neural Networks For Feature Extraction

- Introduction
- Types of Neural Networks
  - Deep Neural Networks
  - Deep Belief Networks
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Autoencoder Networks

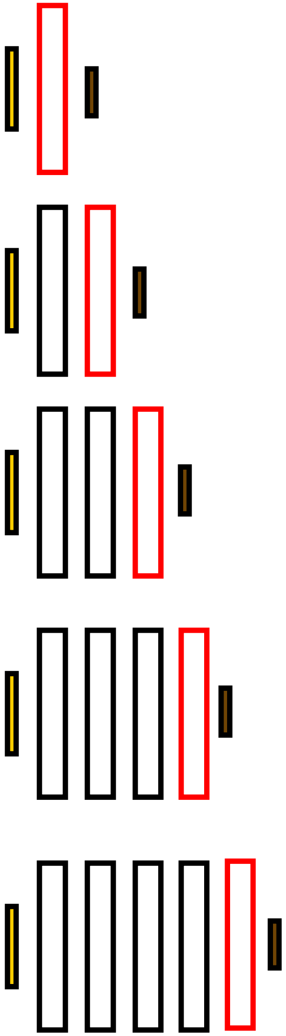
Neural network based features – Variants - DNNs

DNNs have large number of parameters

– hard to optimize

– right initialization

Discriminative pre-training is a layer-by-layer initialization technique using labeled training data.



Neural network based features – Variants - DBNs

- An alternate pre-training technique exists which **does not require labeled training data**
- The goal is design feature detectors that **model the structure of the data** rather than discriminate between classes
- The generative pre-training **finds a region of the weight space that allows the discriminative fine-tuning to make rapid progress**, and it also significantly reduces over-fitting

G. Hinton et.al , "Deep Neural Networks for Acoustic Modeling in Speech Recognition ", 2012

Neural network based features – Variants - DBNs

Probability of x using functions of the form $f(x;\Theta)$ where Θ are model parameters

$$p(x;\Theta) = \frac{1}{Z(\Theta)} f(x;\Theta)$$

$$Z(\Theta) = \int f(x;\Theta)\, dx$$ - Partition function

The model parameters are learnt by maximizing the probability of the training data

$$p(\mathbf{X};\Theta) = \prod_{k=1}^{K} \frac{1}{Z(\Theta)} f(x_k;\Theta)$$

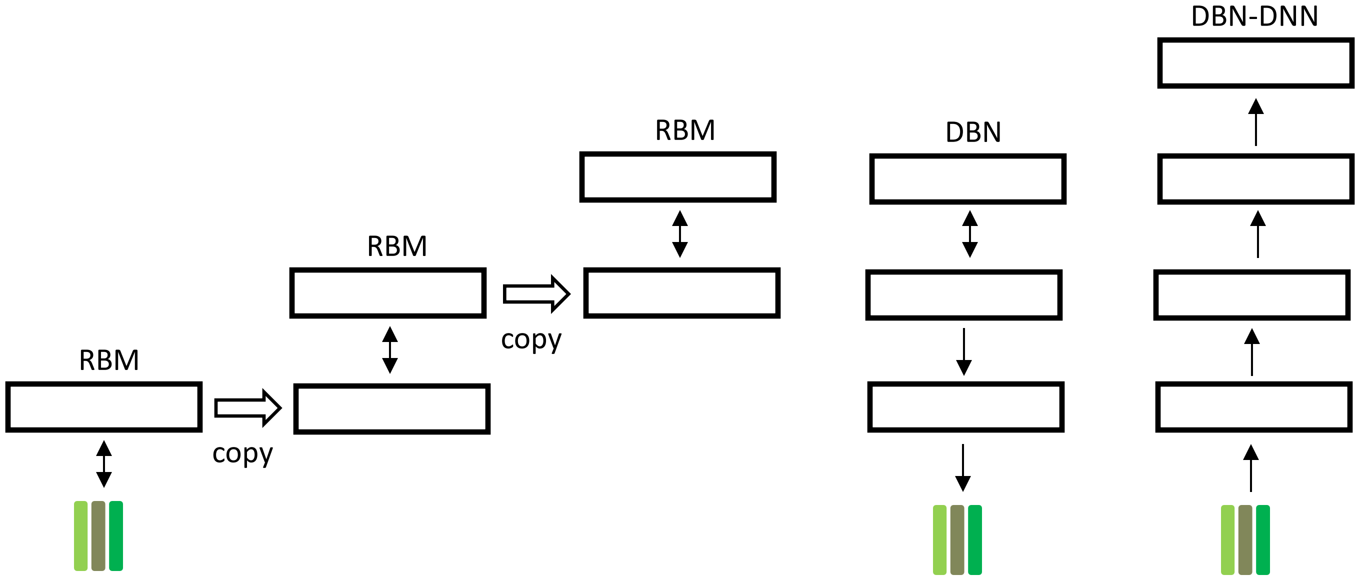
or minimizing the negative log of $p(\mathbf{X};\Theta)$, also called the energy $E(\mathbf{X};\Theta)$

$$E(\mathbf{X};\Theta) = \log Z(\Theta) - \frac{1}{K} \sum_{k=1}^{K} \log f(x_k;\Theta)$$

Gradient descent methods are used to find a local minimum of the energy function

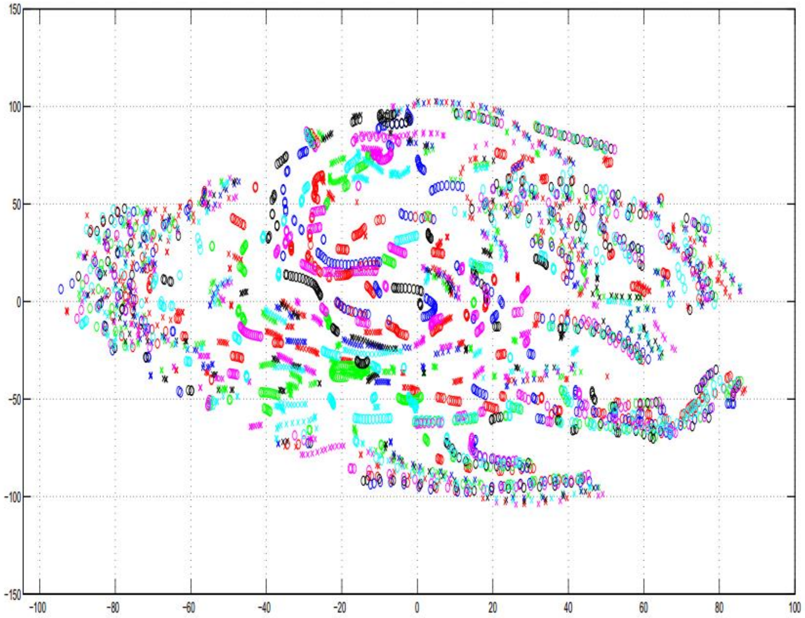
Neural network based features – Variants – DBNs

- **Restricted Bolzmann's machines** are a type of graphical models that have been shown to be useful in building such generative models
- A learning procedure called **contrastive divergence** is useful in training these models
- The RBMs in a stack can be combined in a surprising way to produce a single, multilayer generative model called a **deep belief net (DBN)**

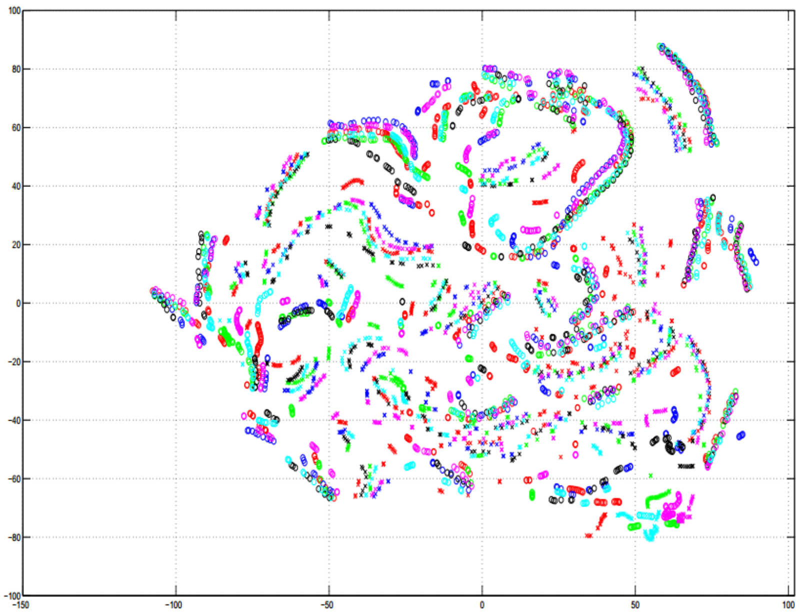


G. Hinton et.al , "Deep Neural Networks for Acoustic Modeling in Speech Recognition ", 2012

Neural network based features – Variants - DNNs



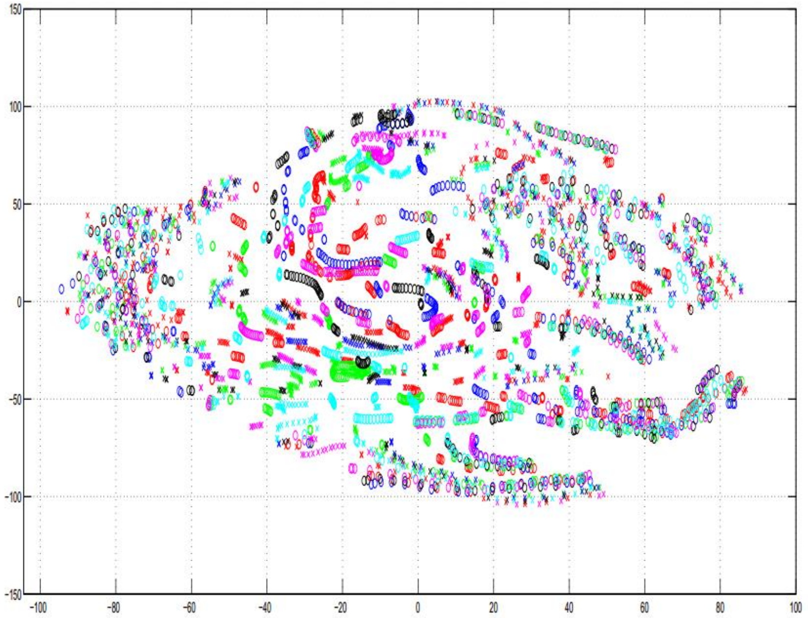
t-SNE 2-D map of fbank feature vectors



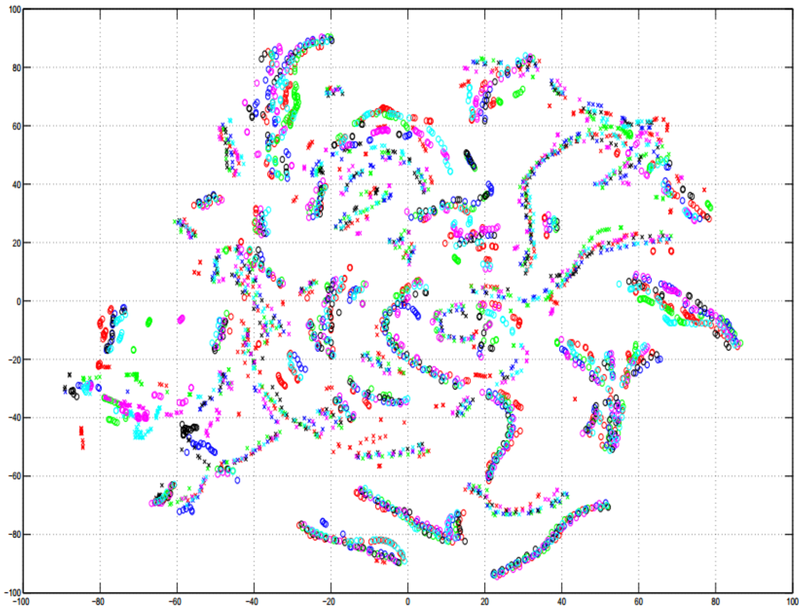
t-SNE 2-D map of the 1st layer of the fine-tuned hidden activity vectors using fbank inputs

A. Mohamed, G. Hinton, G. Penn, "Understanding how Deep Belief Networks perform acoustic modelling", 2012.

Neural network based features – Variants - DNNs



t-SNE 2-D map of fbank feature vectors

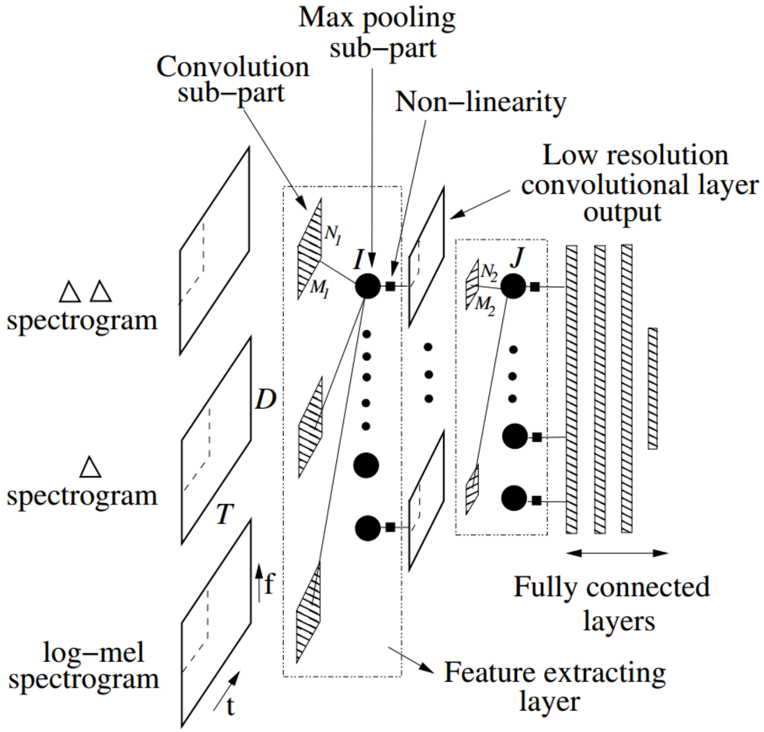


t-SNE 2-D map of the 8th layer of the fine-tuned hidden activity vectors using fbank inputs

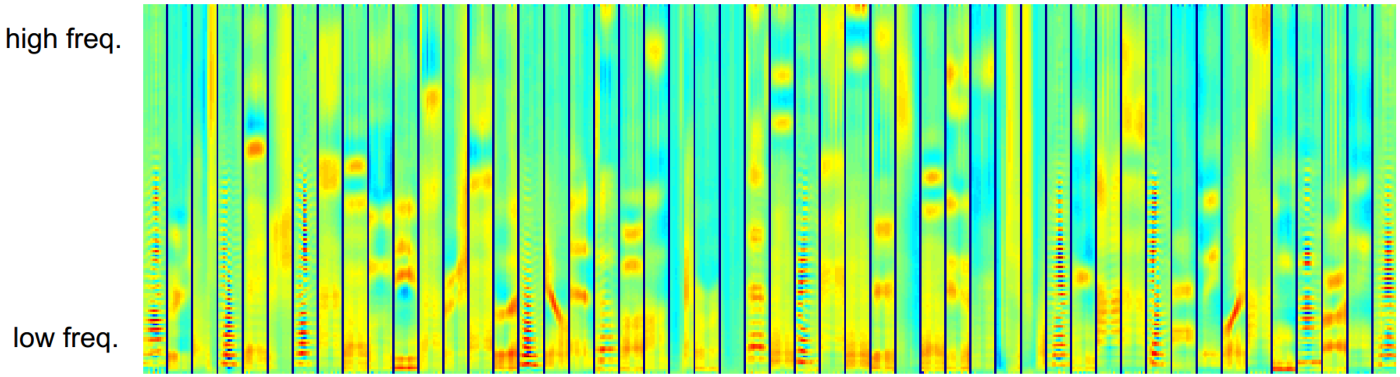
A. Mohamed, G. Hinton, G. Penn, "Understanding how Deep Belief Networks perform acoustic modelling", 2012.

Neural network based features – Variants - CNNs

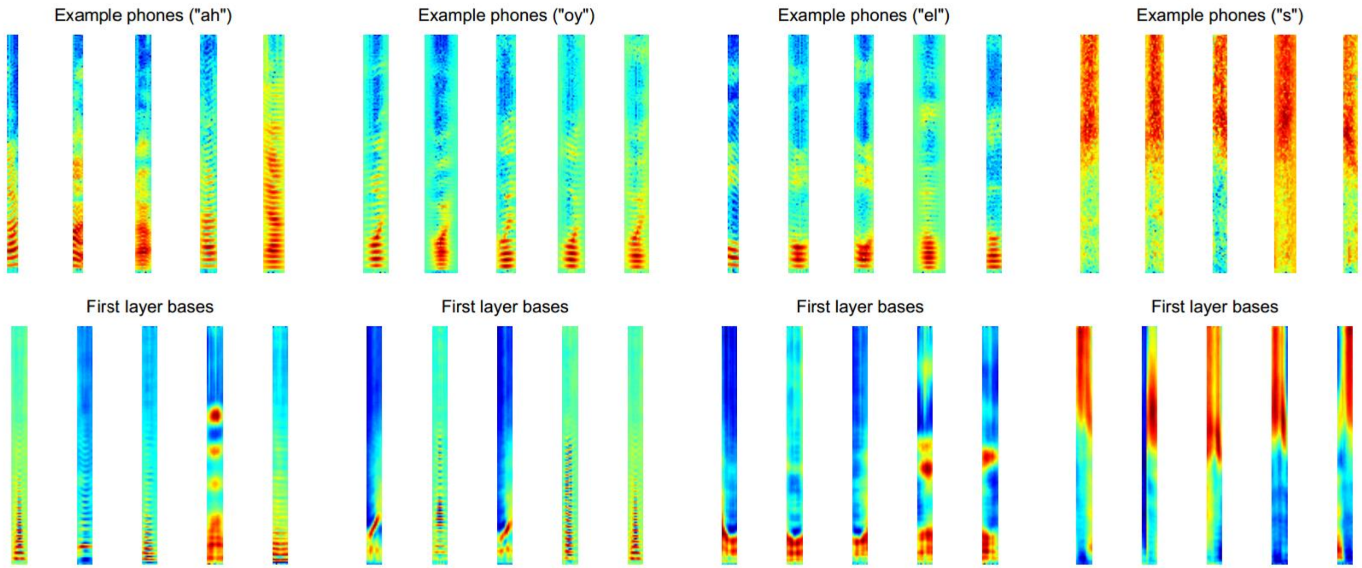
- Convolutional neural networks (CNN) are very similar to conventional deep neural networks – the difference between these models, being the additional CNN feature extracting layers
- These layers generate features for succeeding layers instead of pre-processed features that are usually input to the DNNs



Neural network based features – Variants - CNNs



Visualization of randomly selected first-layer CDBN bases trained on the TIMIT data

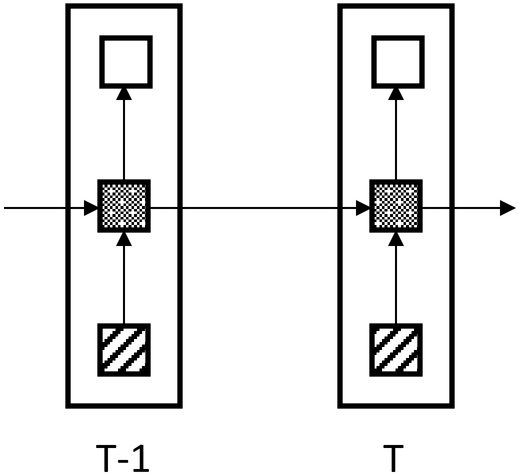


Visualization of the four different phonemes and their corresponding first-layer CDBN bases.

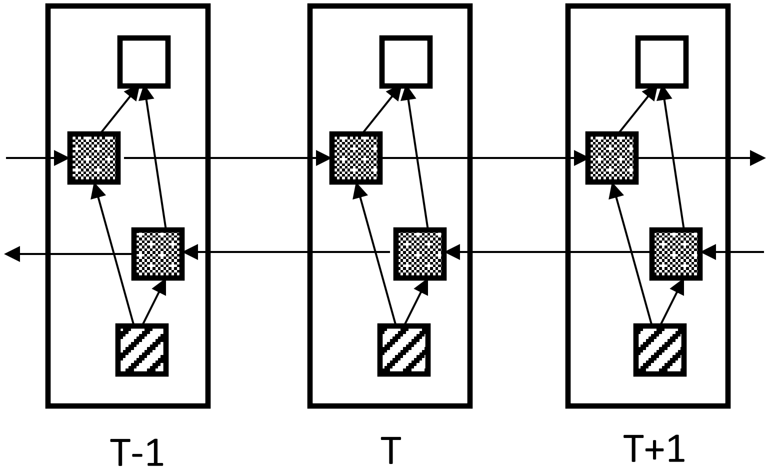
H. Lee et.al , "Unsupervised feature learning for audio classification using convolutional deep belief networks", 2009

Neural network based features – Variants - RNNs

- Integrating temporal information via neural networks – feed-back connections instead of only feed-forward connections



Unidirectional RNNs

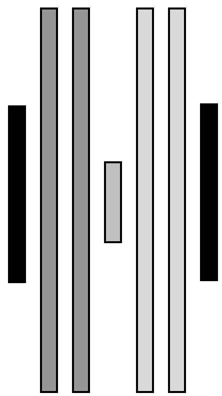


Bidirectional RNNs

M. Schuster and K. K. Paliwal, "Bidirectional Recurrent Neural Networks", 1997

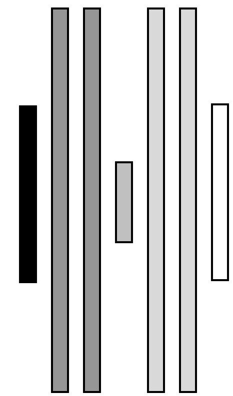
Neural network based features – Variants – Autoencoders

High-dimensional data can be converted to a lower dimension by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors.



Autoencoders for dimensionality reduction of data

Denoising autoencoders are variants of basic autoencoders to reconstruct a clean input from a noisy corrupted version



Autoencoders for denoising data

G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks", 1996

P. Vincent et. al, "Extracting and composing robust features with denoising autoencoders", 2008

Integrating data with feature extraction

- Introduction
- Variants of data-driven features based on
  - PCA/LDA
  - Manifold Learning
  - Neural Networks
  - Application specific training criteria

Feature transforms based on application specific training criteria

- Feature transforms based on speaker data
  - Vocal tract length normalization (VTLN)
  - Constrained maximum likelihood linear regression (cMLLR)
- Feature transforms based on an acoustic model training criteria
  - fMPE – feature based minimum phone error rate training
  - fMMI – feature based maximum mutual information training

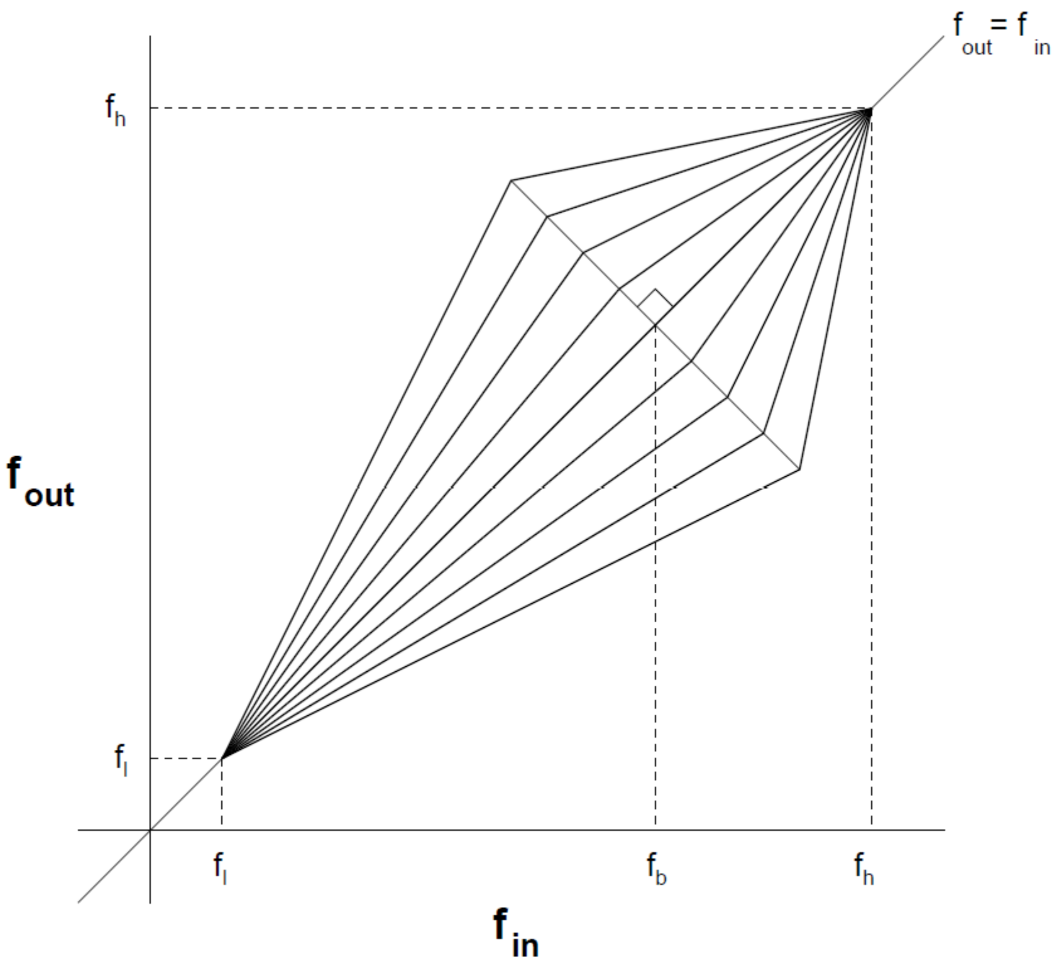
Feature transforms based on speaker data - VTLN

- A major contributing factor for speaker variability is the **length of the speaker's vocal tract**
- Scaling of the vocal tract length from L to kL corresponds to scaling of the frequency axis by $1/k$
- Vocal tract length normalization (VTLN) is a procedure of finding the scaling factor k for each speaker that best matches speech from the speaker to speech from a "canonical" speaker who has an average vocal tract length.

L. Lee and R. C. Rose, "A frequency warping approach to speaker normalization," 1998

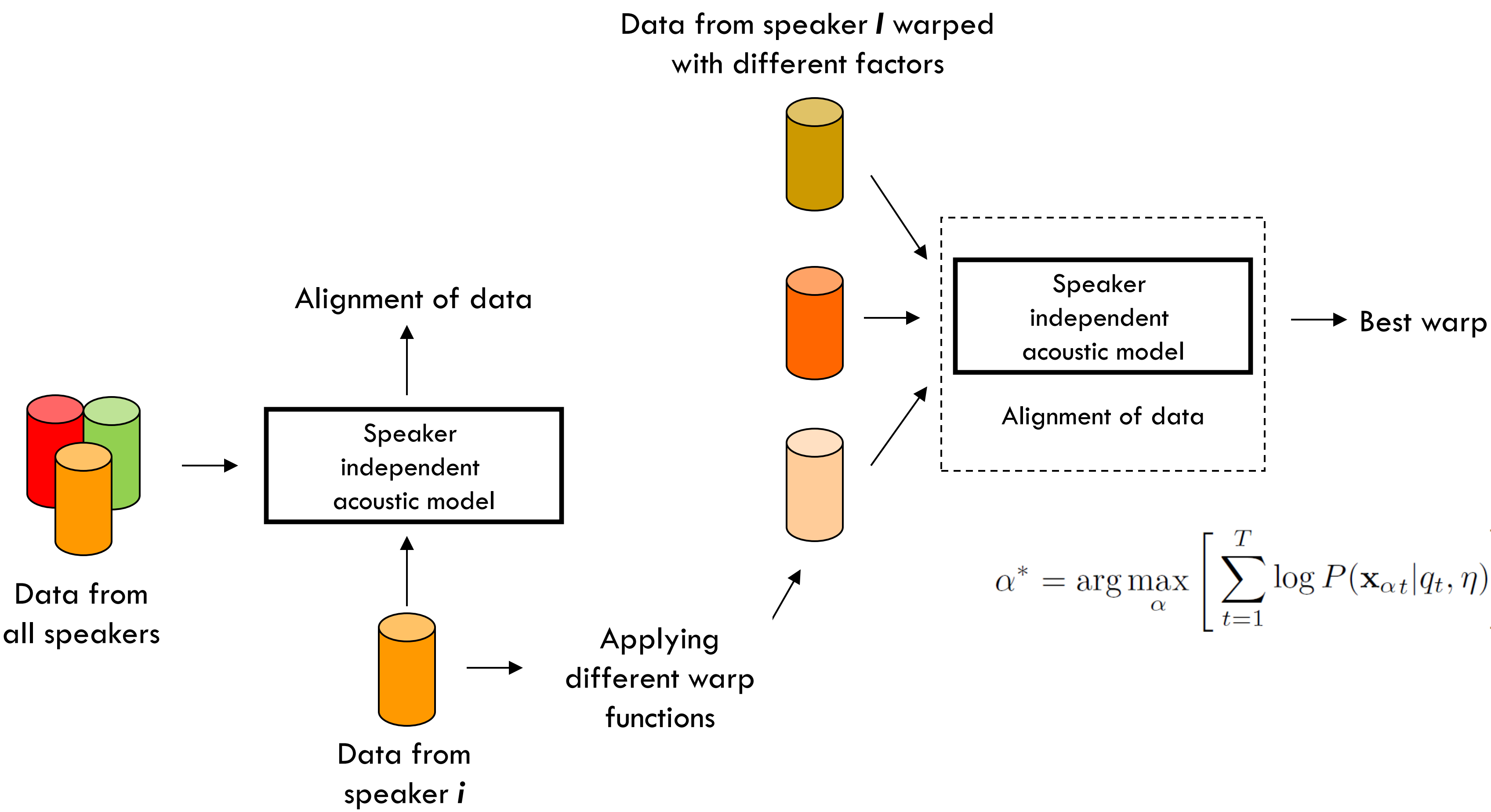
H. Soltau et. al, "Attila: The IBM Speech Recognition Toolkit", 2010

Feature transforms based on speaker data - VTLN



H. Soltau et. al, "Attila: The IBM Speech Recognition Toolkit", 2010

Feature transforms based on speaker data - VTLN



$$\alpha^* = \arg\max_{\alpha} \left[\sum_{t=1}^{T} \log P(\mathbf{x}_{\alpha t} | q_t, \eta) \right]$$

Feature transforms based on speaker data – constrained MLLR

- Adaptation technique used in ASR to reduce the mismatch between acoustic features from a speaker and trained models

Constrained MLLR transform

$$\hat{\mu} = \mathbf{A}_c\mu - \mathbf{b}_c$$
$$\hat{\Sigma} = \mathbf{A}_c\Sigma\mathbf{A}_\mathbf{c}^\mathbf{T}$$

- Equivalent to transforming the features

$$\hat{\mathbf{o}}_t = \mathbf{A}_c^{-1}\mathbf{o}_t + \mathbf{A}_c^{-1}\mathbf{b}_c$$

- Transformation parameters are estimated with EM to maximize the likelihood of the adaptation data

M. J. F. Gales. "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition", 1998

Feature transforms based on speech recognition - fMMI

- Given an observation sequence, **O**, and corresponding word sequence, **W**, there should be minimal uncertainty about the correct answer (i.e., minimize the conditional entropy of the word sequence given the observation):

$$H(W \mid O) = -\sum_{w,o} P(W = w, O = o) \log P(W = w \mid O = o)$$

- To accomplish this, the probability of the word sequence given the observation must increase – the recognizer should make good guesses

- The mutual information, I(W;O), between W and O:

$$I(W;O) = H(W) - H(W \mid O) \equiv$$
$$H(W|O) = H(W) - I(W;O)$$

- Two choices: minimize H(W) or maximize I(W;O)

K. Vertanen. "An Overview of Discriminative Training for Speech Recognition".

Feature transforms based on speech recognition - fMMI

- Maximizing the mutual information is equivalent to choosing the parameter set λ to maximize:

HMM corresponding to the transcription w

Probability of the word sequence w as determined by the language model

$$\mathcal{F}_{\mathrm{MMIE}}(\lambda) = \sum_{r=1}^{R} \log \frac{P_{\lambda}(O_r \mid M_{w_r}) P(w_r)}{\sum_{\hat{w}} P_{\lambda}(O_r \mid M_{\hat{w}}) P(\hat{w})}$$

Sums over each possible word sequences

- Maximization involves increasing the numerator term (maximum likelihood estimation – MLE) or decreasing the denominator term (maximum mutual information estimation – MMIE)
- The denominator term is accomplished by reducing the probabilities of incorrect, or competing, hypotheses.

L.R Bahl et.al, “Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition,”1986

Feature transforms based on speech recognition - fMMI

- fMMI is a form of discriminative training that optimizes the same objective function as MMI but does so by modifying the features

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{M}\mathbf{h}_t$$

High dimensional feature vector of Gaussian derived posteriors

Transformation matrix trained using the MMI criteria

- fMPE is a similar transformation except that the objective function is the Minimum Phone Error criteria

$$\mathcal{F}_{\mathrm{MPE}}(\lambda) = \sum_{r=1}^{R} \sum_{s} P_{\lambda}^{\kappa}(s|\mathcal{O}_r)\mathrm{A}(s, s_r)$$

Average of the transcription accuracies of all possible sentences **s**, weighted by the probability of **s** given the model

D. Povey et.al, “Boosted MMI for Model and Feature-space Discriminative Training ,”2008

D. Povey et.al, “fMPE: Discriminatively trained features for Speech Recognition ,”2008

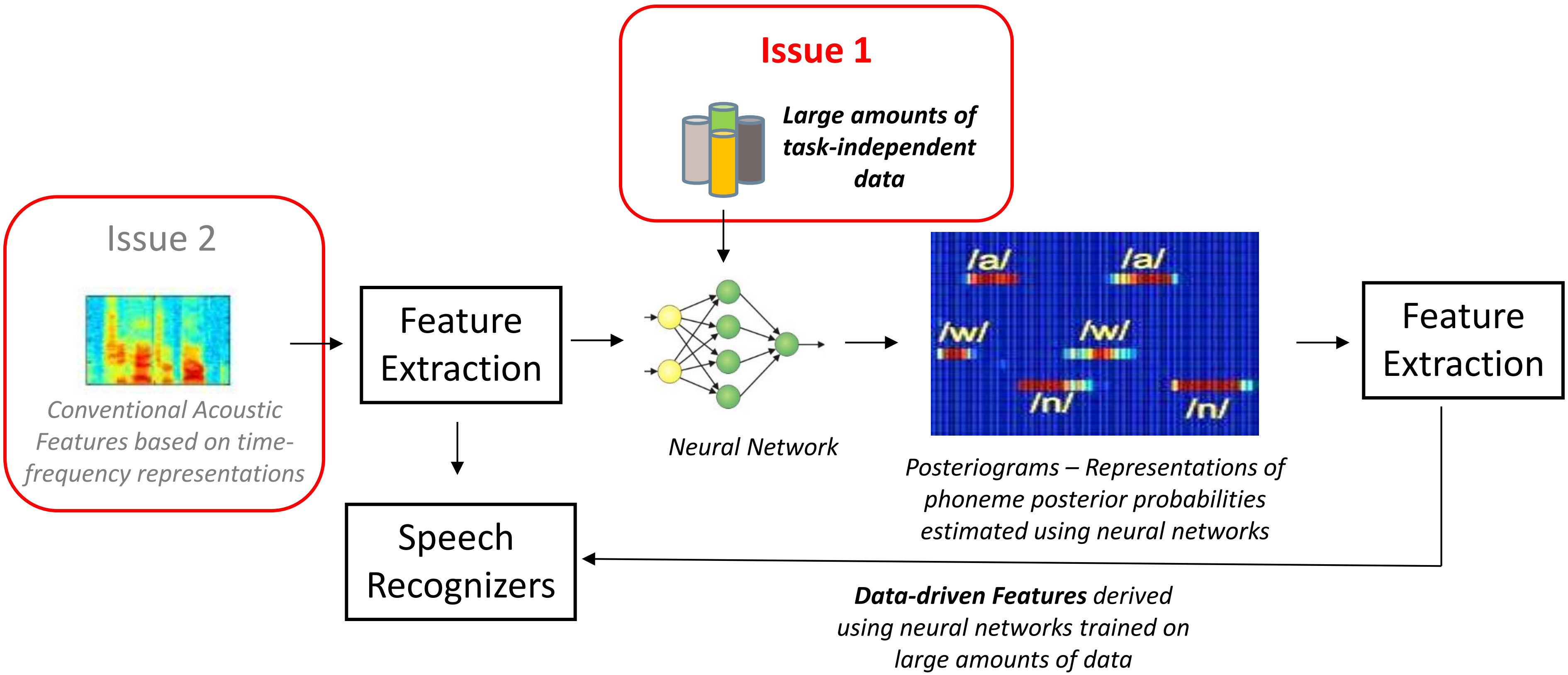
THE FUTURE …

The distinction between past, present, and future is only a stubbornly persistent illusion.
-Albert Einstein

Future Directions

- Speech technologies in newer languages with limited supervised data
  - Need for multi-lingual data driven approaches
  - Large amount of un-transcribed data is continuously being generated – semi-supervised approaches.
- Handling noisy and mis-matched acoustic data
  - Commercial and military applications of speech data with varying recording devices and environments.
  - Ubiquitous speech technologies

Future Directions

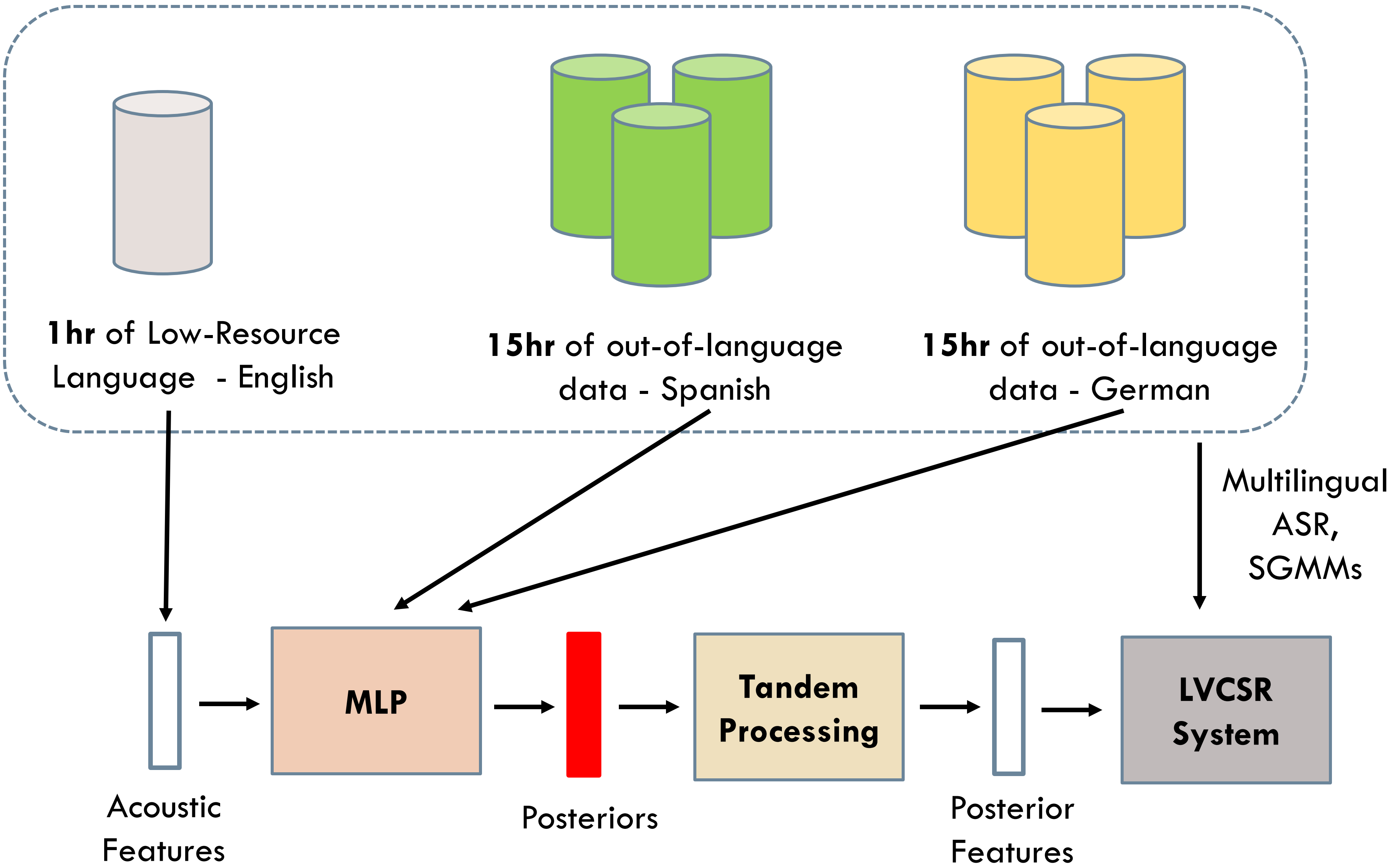


Low Resource ASR

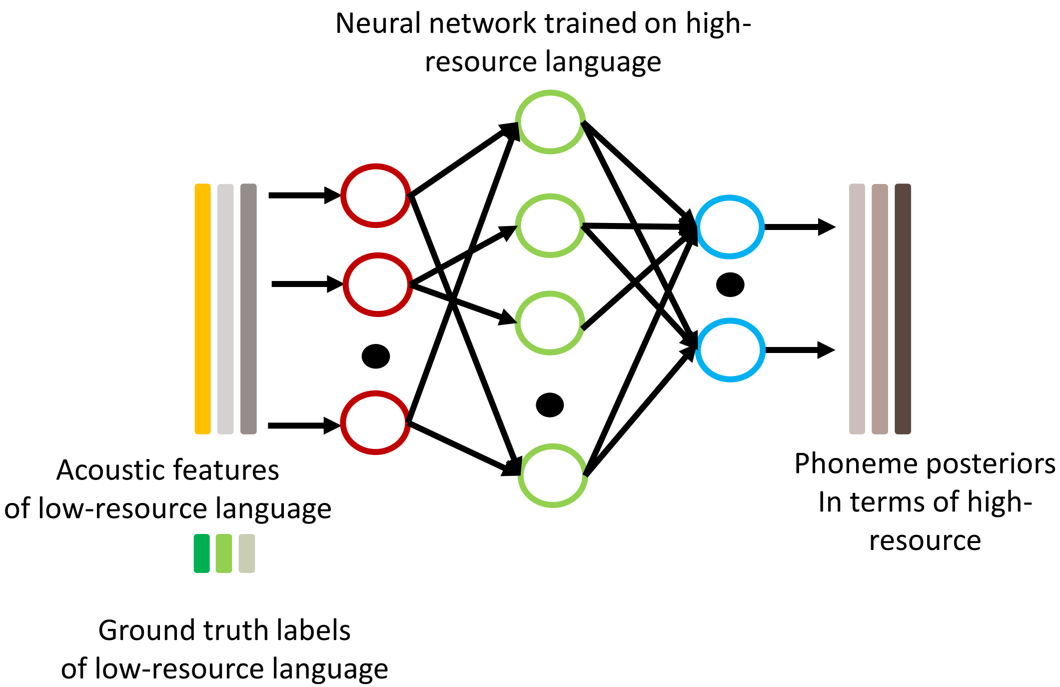
Word Recognition Error Rates (%) - Callhome English

Feature Configuration	Word Error Rate (%)
PLP features - 15 hours	53.5
PLP features - 1 hour	71.2
Data-driven features - 1 hour	70.0

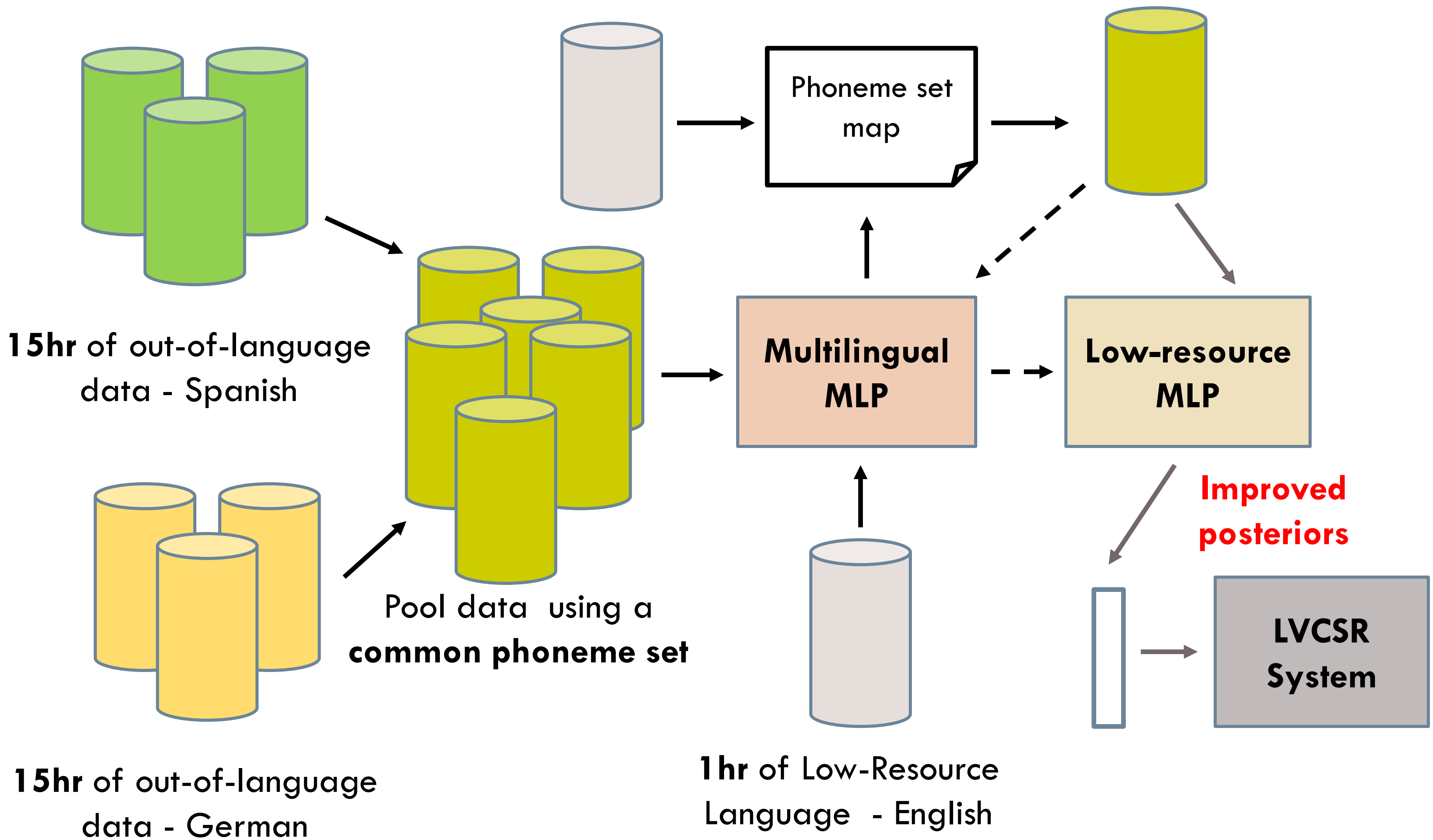
Low Resource ASR



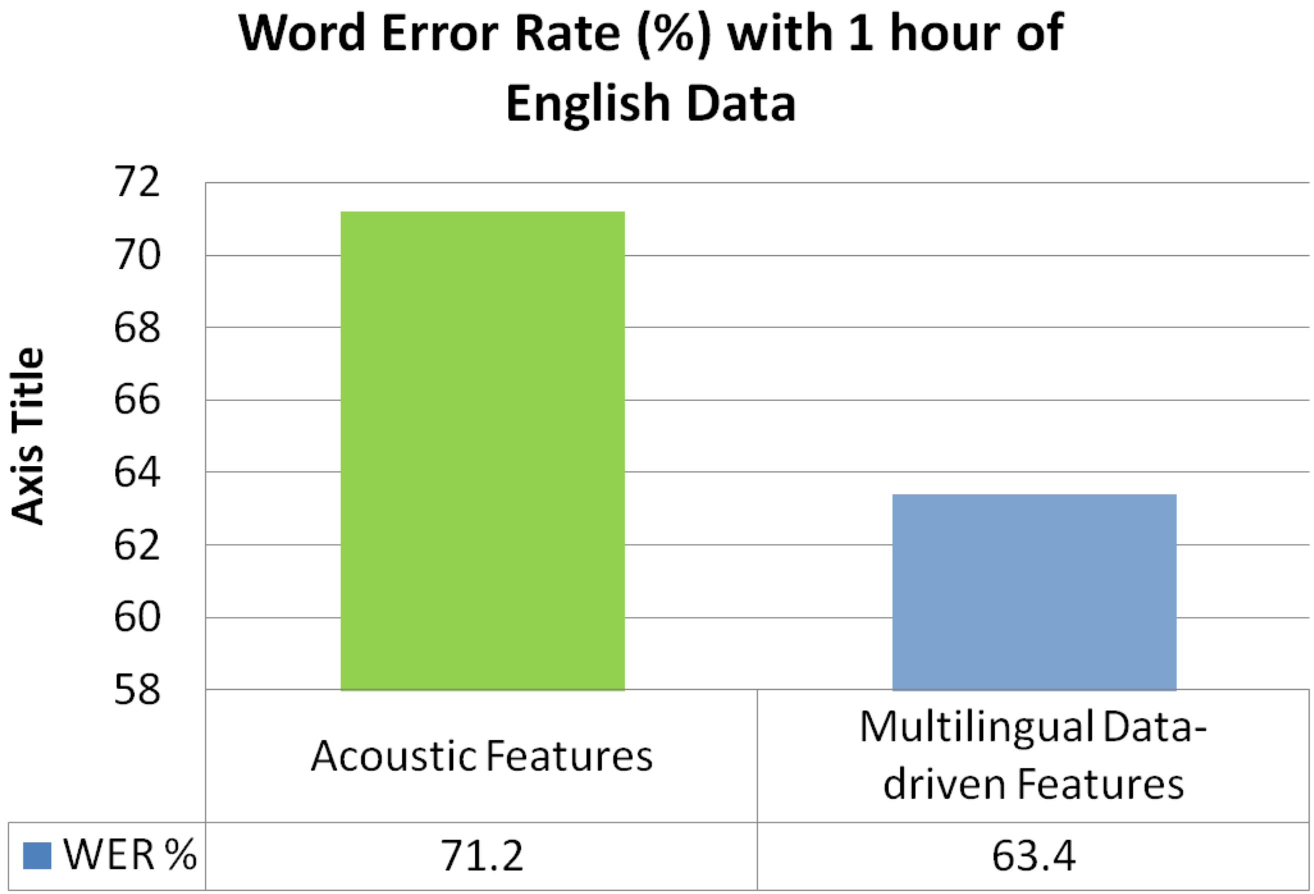
Low resource ASR – Solution I



Low resource ASR – Solution I

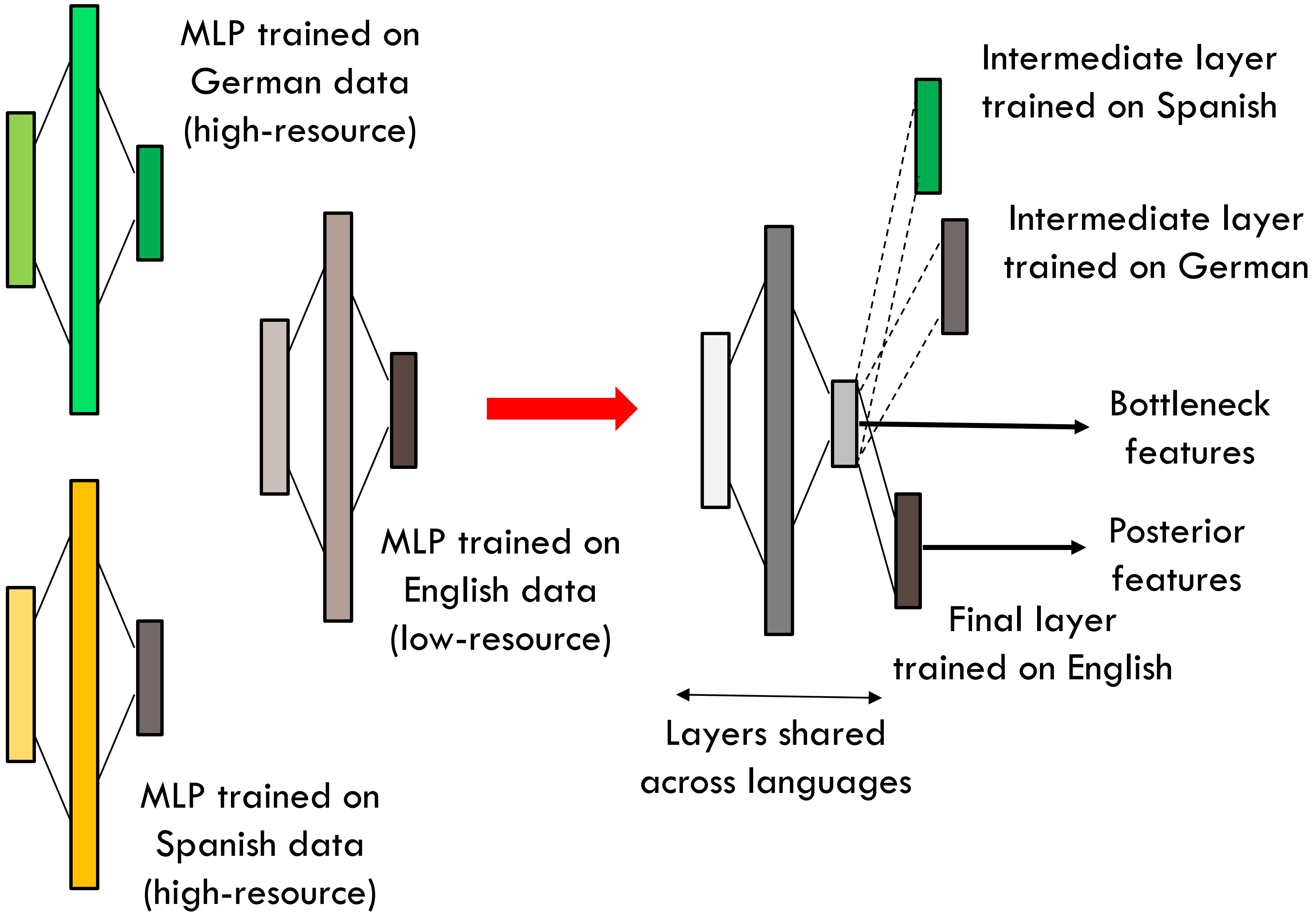


Low resource ASR – Solution I

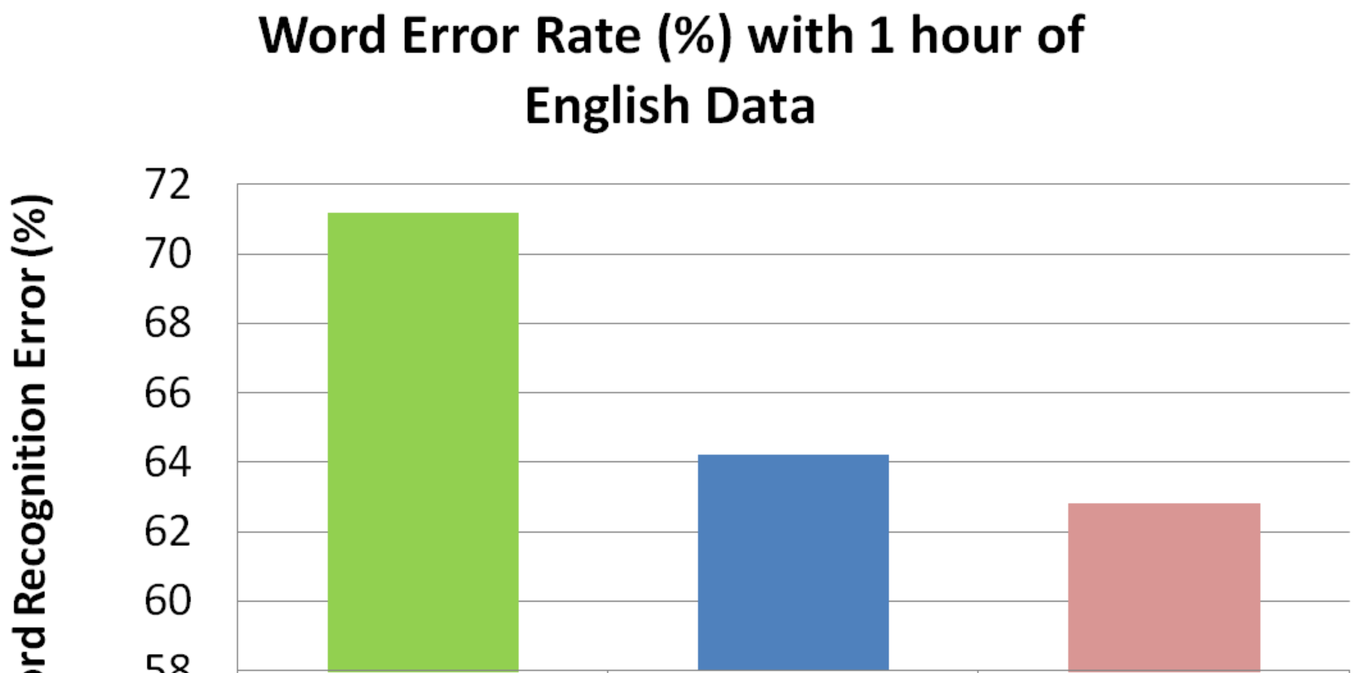


S. Thomas, S. Ganapathy and H. Hermansky, "Cross-lingual and Multi-stream Posterior Features for Low-resource LVCSR Systems", 2010.

Low resource ASR – Solution II



Low resource ASR – Solution II

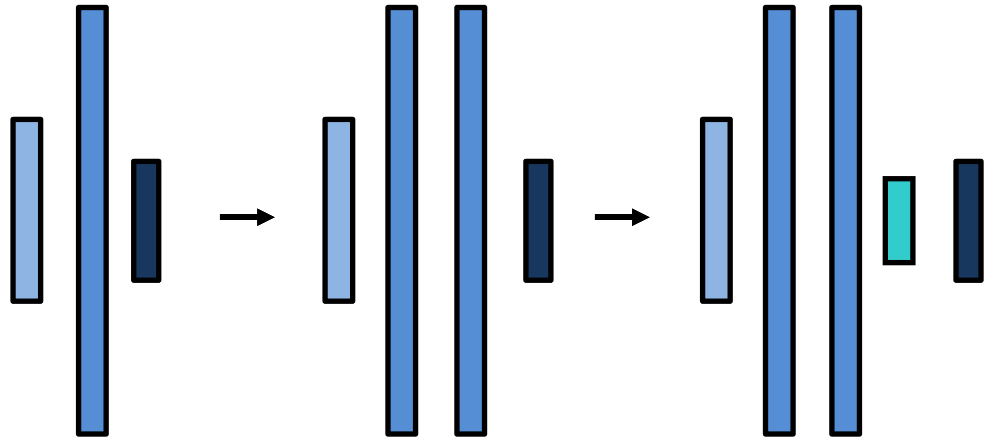


	PLP	Tandem Features	Bottleneck Features
WER (%)	71.2	64.2	62.8

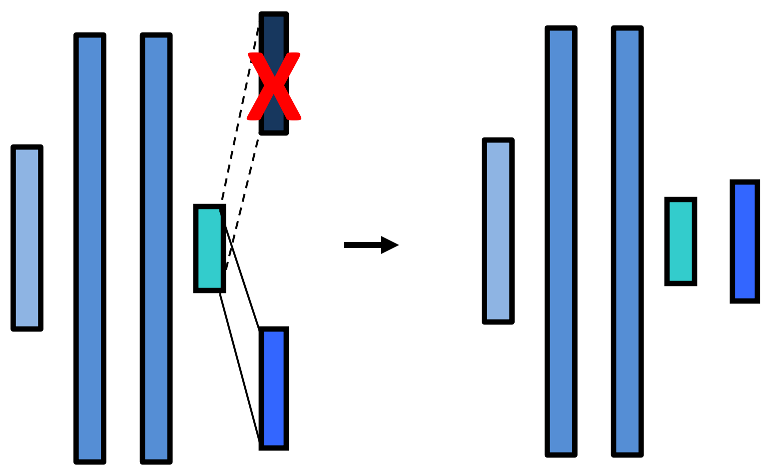
S. Thomas, S. Ganapathy and H. Hermansky, "Multilingual MLP features for Low-resource LVCSR systems", 2012.

Multilingual DNN based features

Multilingual Pre-
training
and fine-tuning



Adaptation to the
low-resource
language
and fine-tuning



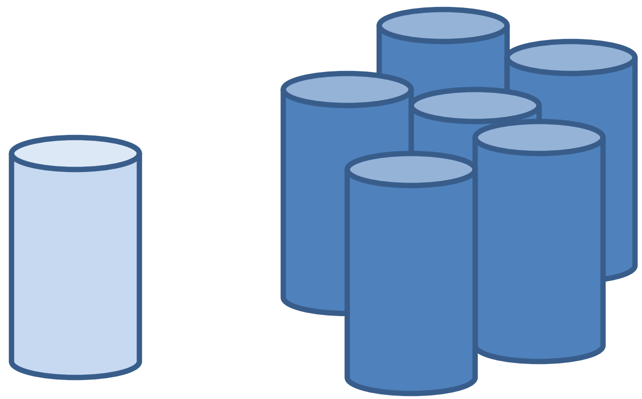
Multilingual DNN based features

System	Word Error Rate (%)
1 hour transcribed English	71.2
1 hour transcribed English + 31 hours German/Spanish – DNN features	59.0
15 hours transcribed English	53.5

Semi-supervised Training

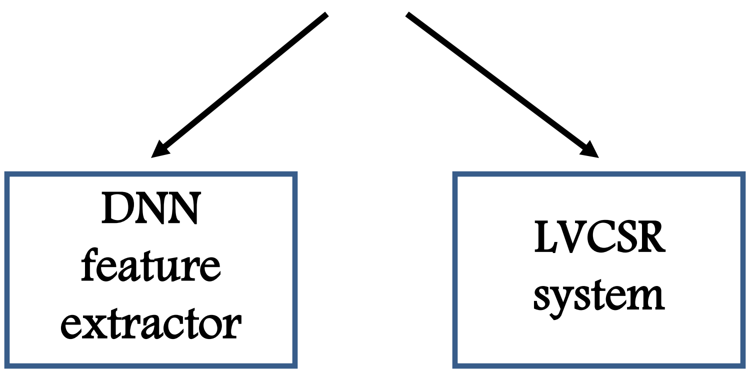
Low resource setting

Audio with transcriptions



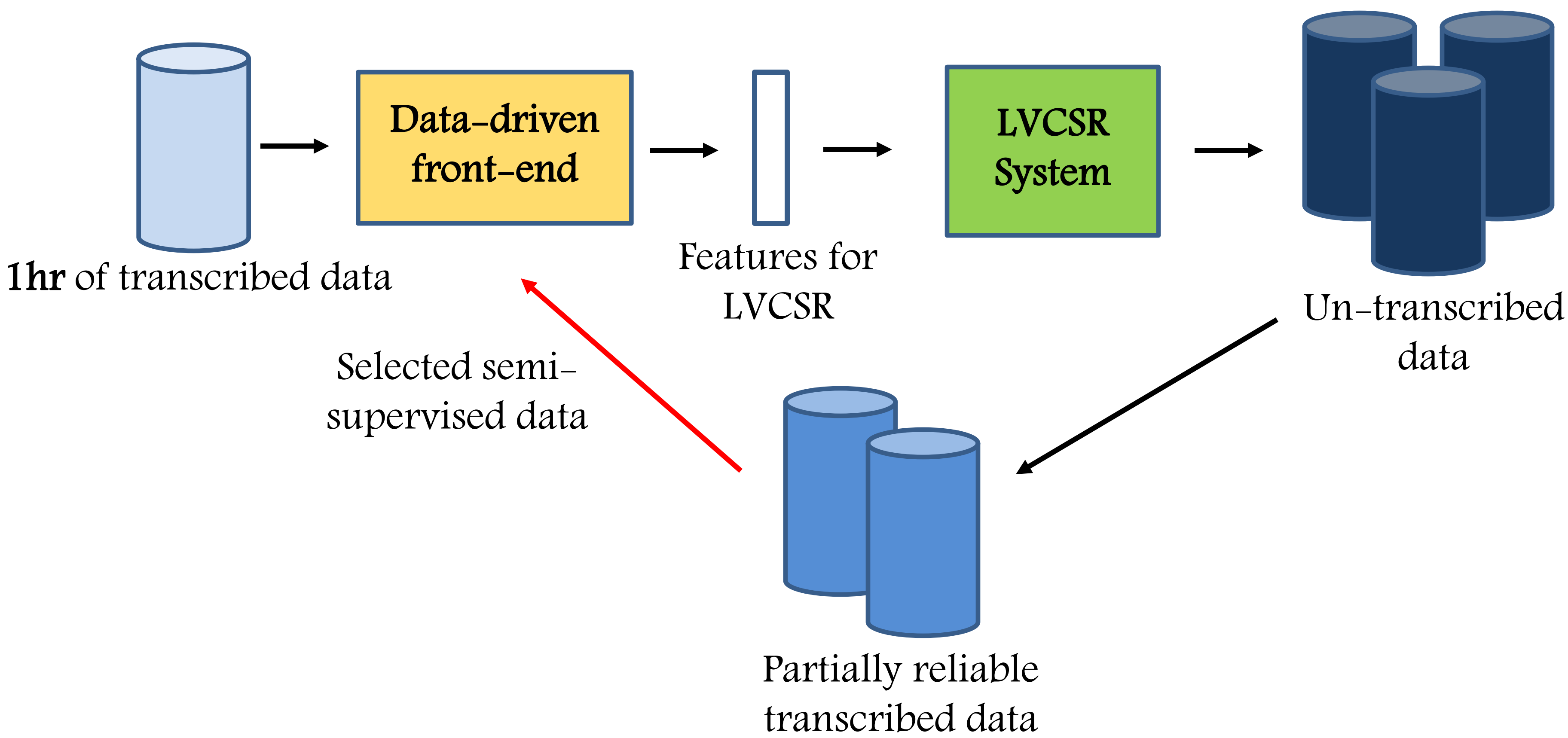
Audio without transcriptions

How do we make use of lots of audio (without transcriptions)?



Semi-supervised Training

Semi-supervised Training



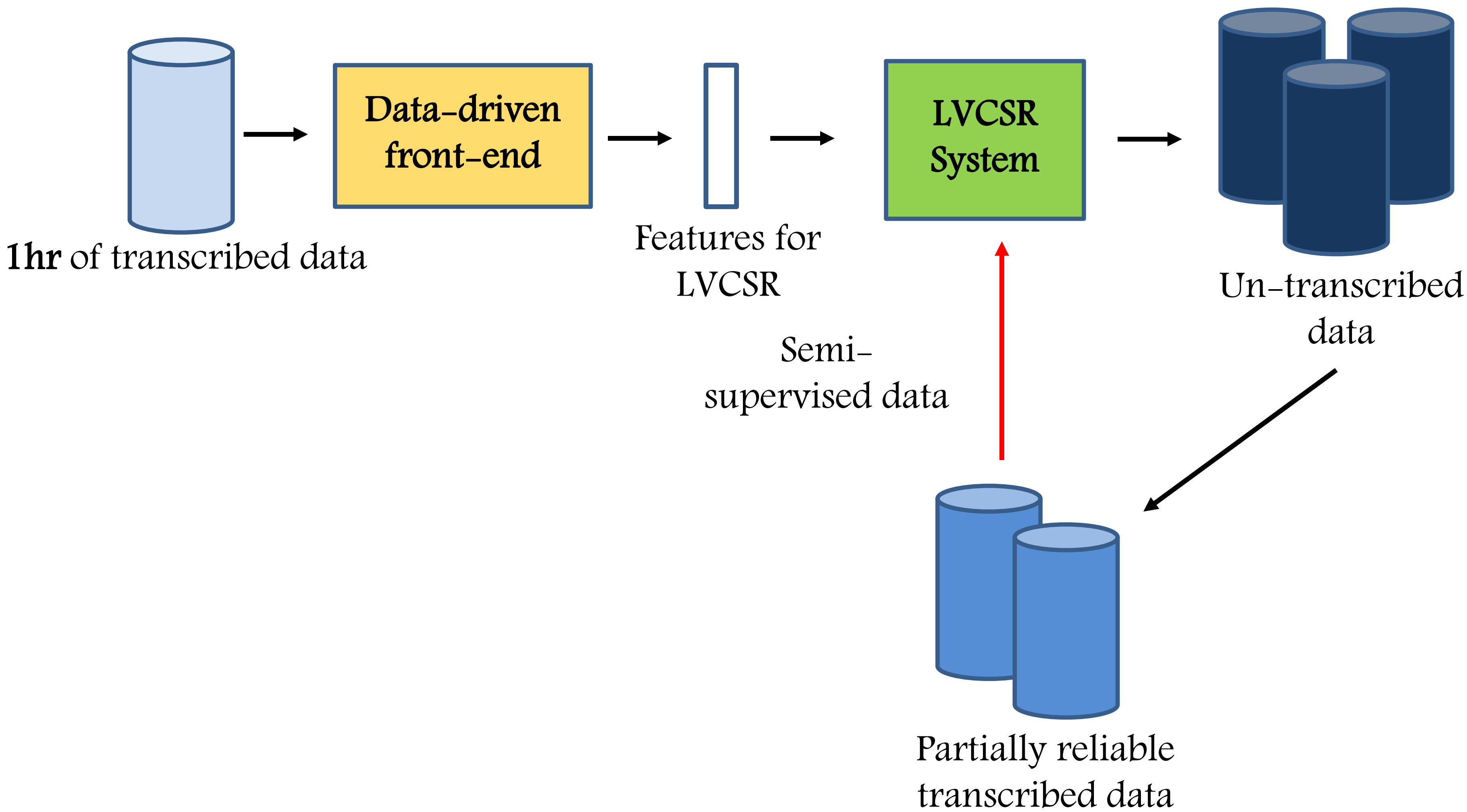
Semi-supervised Training

- ASR based word confidence scores
  - Sentences that have high lattice based word confidences
- MLP posteriogram based confidence scores
  - Sentences that have high phoneme occurrence counts
- Logistic regression is used to combine these scores

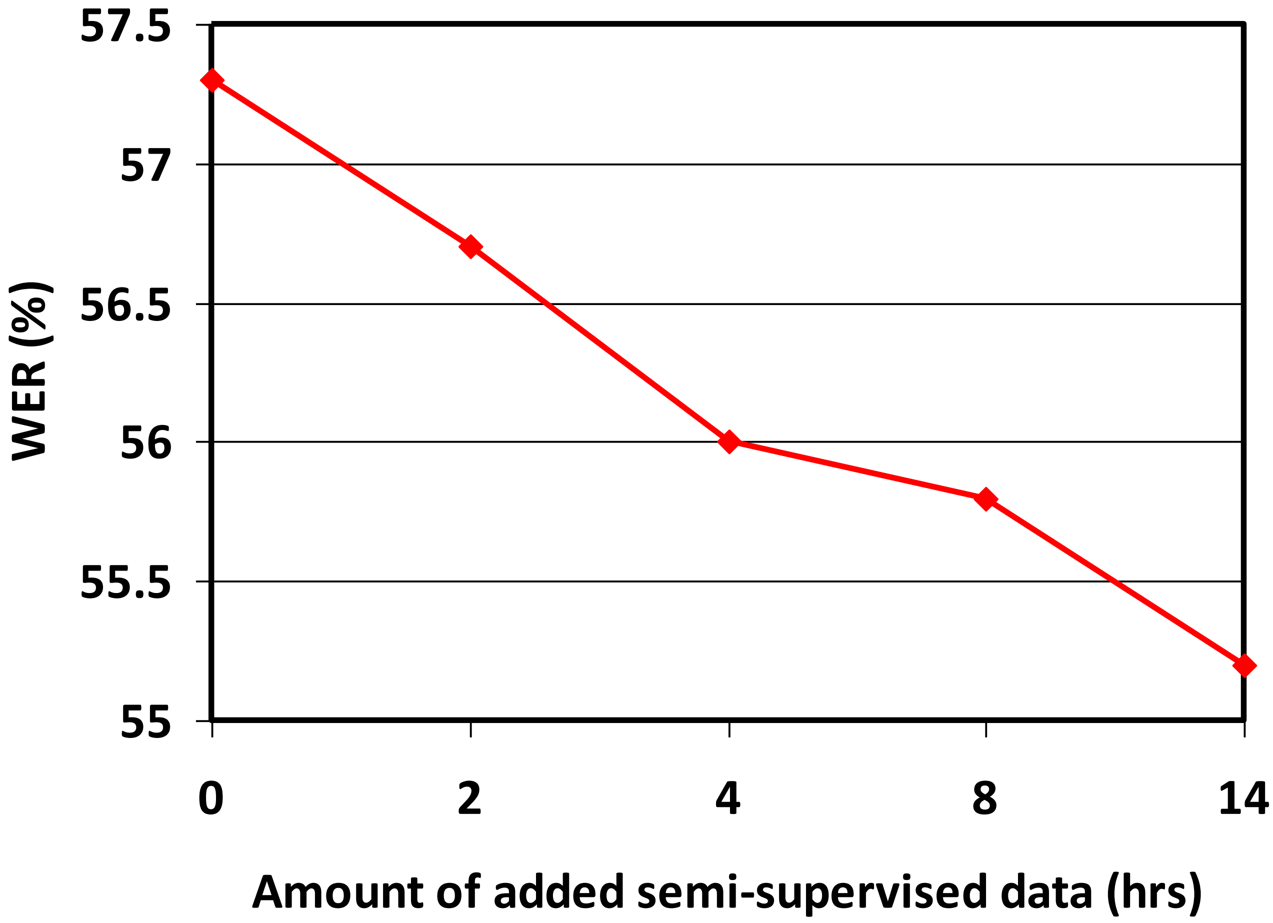
Semi-supervised Training

System	Word Error Rate (%)
1 hour transcribed English	71.2
1 hour transcribed English + 31 hours German/Spanish – DNN features	59.0
Semi-supervised data to DNN training	57.3
15 hours transcribed English	53.5

Semi-supervised Training



Semi-supervised Training

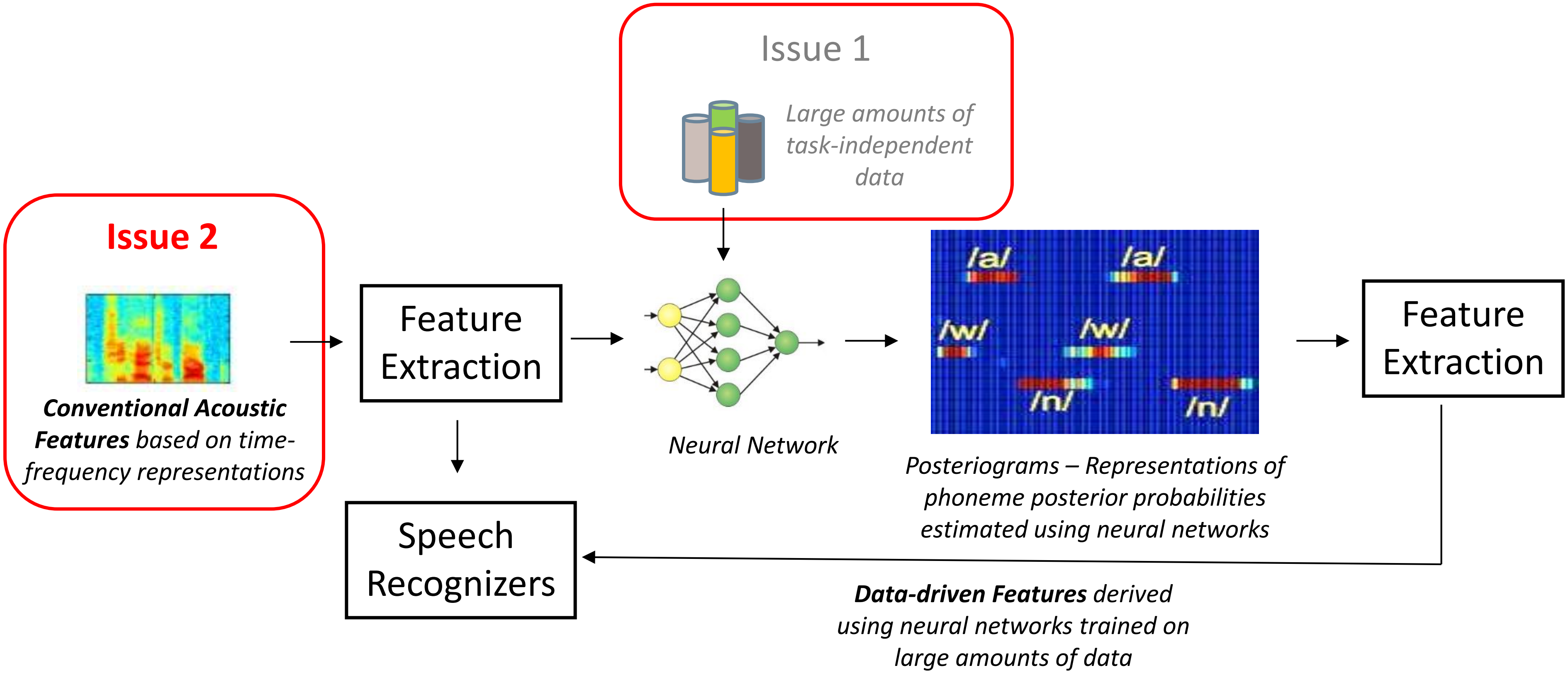


Low resource ASR

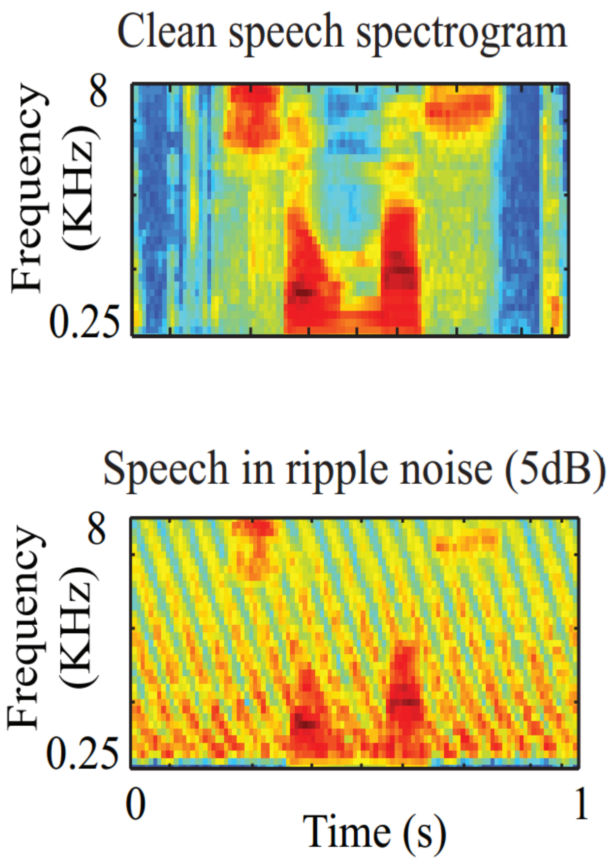
Feature Configuration	Word Error Rate (%)
PLP features - 15 hours	53.5
PLP features - 1 hour	71.2
Data-driven features - 1 hour	70.0
Multilingual MLP	62.8
Multilingual deep network	59.0
Self training with Multilingual deep network	57.3
Self training with ASR with deep network features	55.2

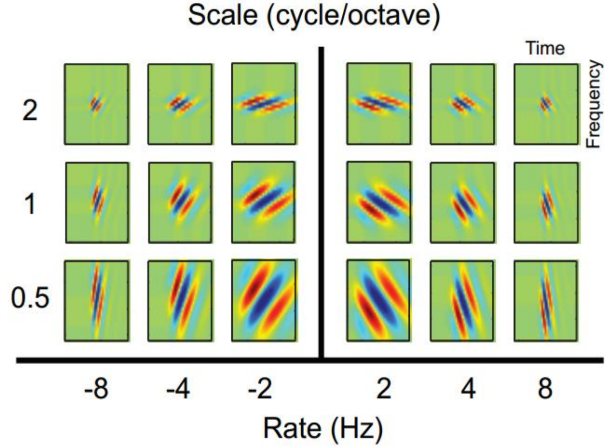
S. Thomas, et.al, "Deep Neural Network Features and Semi-supervised training for Low Resource Speech Recognition", 2013.

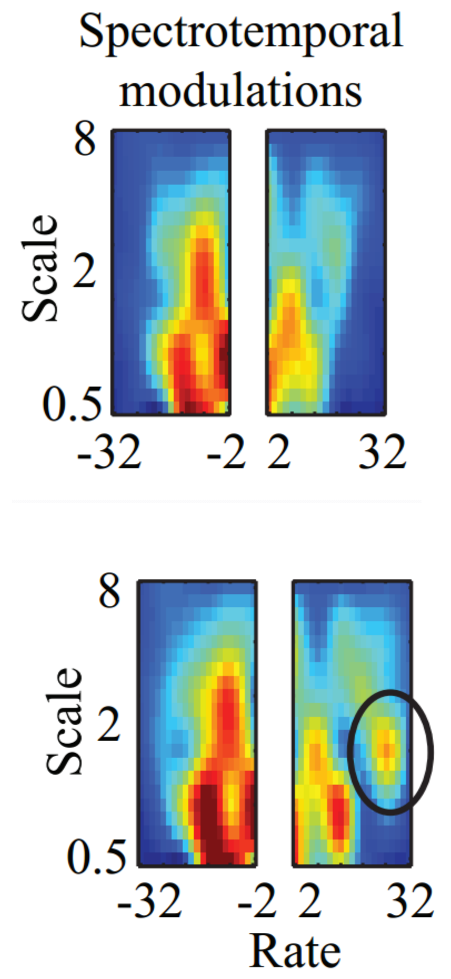
Future Direction



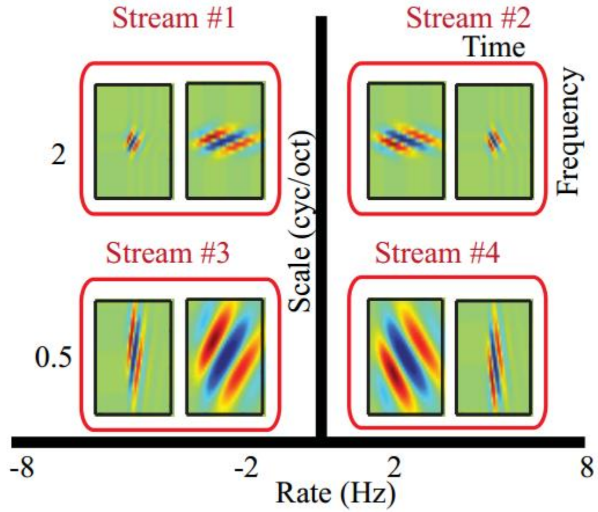
Dealing with noise – Multi-stream Idea

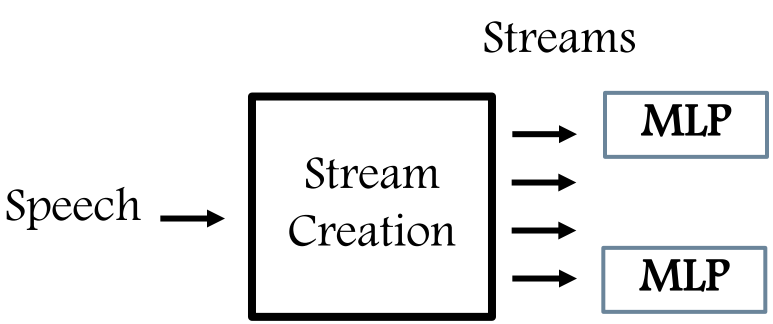




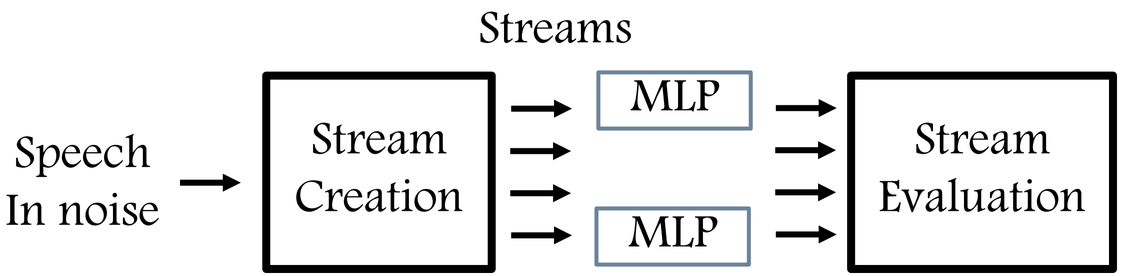


Dealing with noise – Stream creation

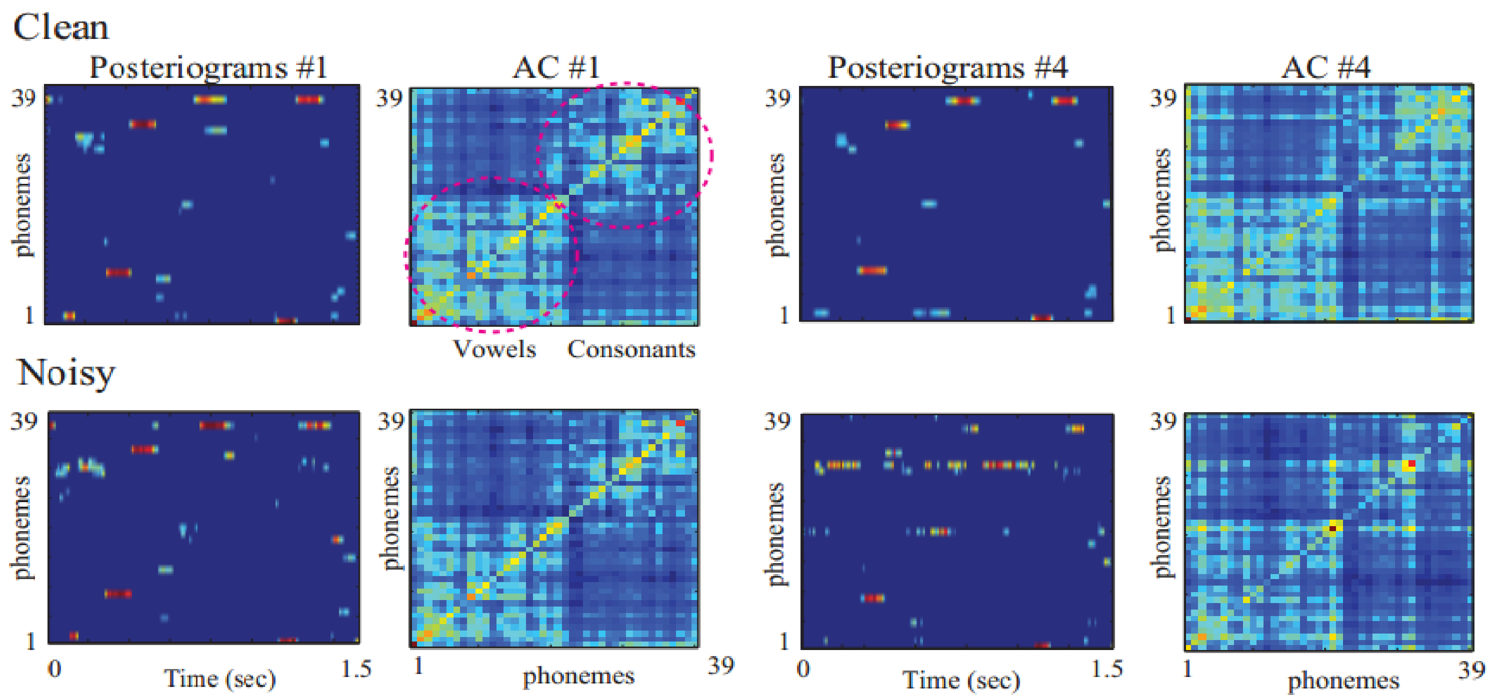




Dealing with noise – Stream Evaluation

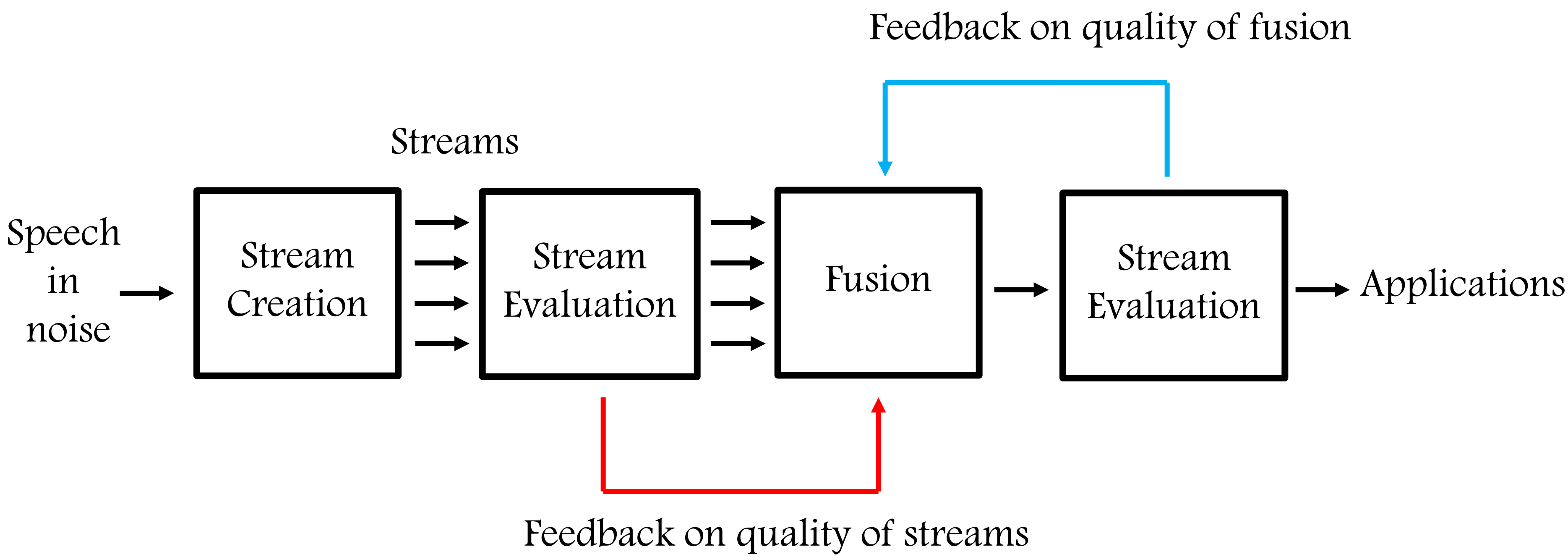


$$AC_j = \frac{1}{N}\sum_{n=1}^{N} P_j(n)\, P_j(n)^T$$



N. Mesgarani, S. Thomas and H. Hermansky, A Multistream Multiresolution Framework for Phoneme Recognition, 2010.

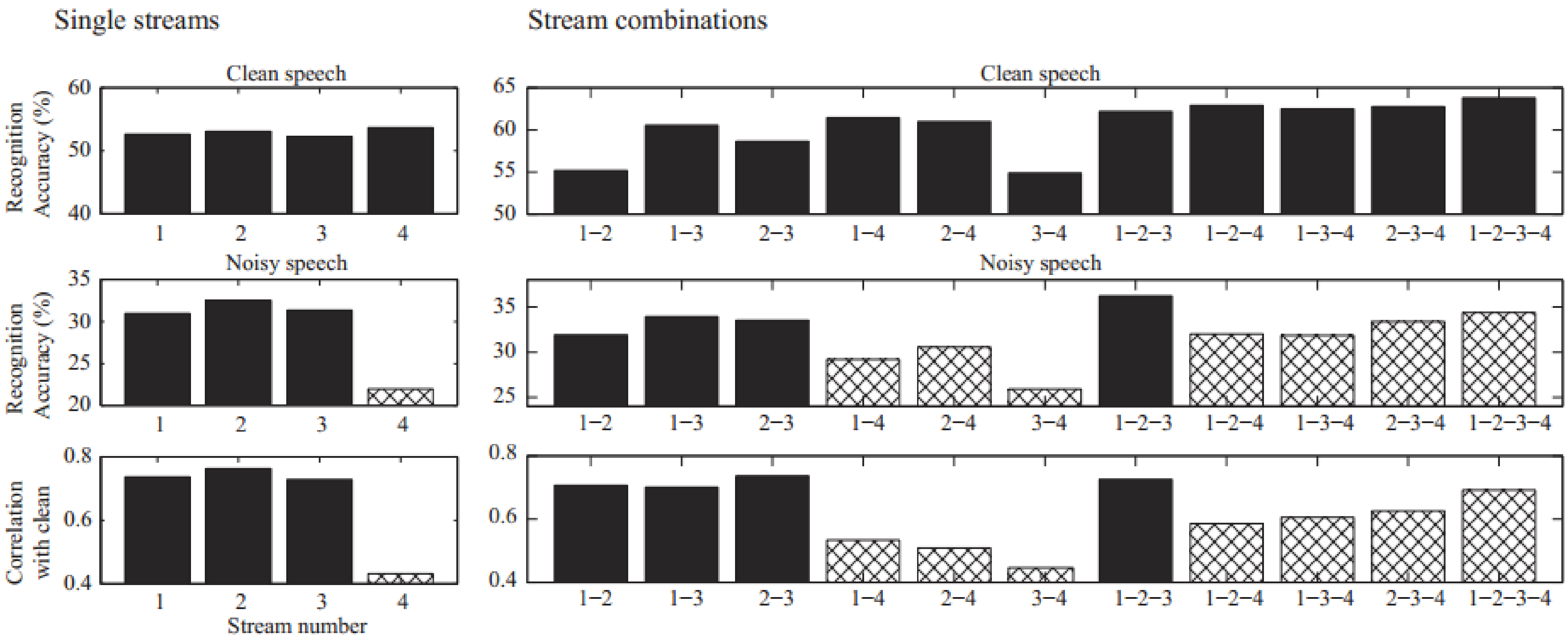
Dealing with noise – Stream Fusion



Issues with data-driven features – Dealing with noise – Stream Fusion

$$AC_j = \frac{1}{N}\sum_{n=1}^{N} P_j(n) P_j(n)^T$$

$$r = \frac{AC_{clean} AC_{noisy}}{\|AC_{clean}\| \|AC_{noisy}\|}$$



N. Mesgarani, S. Thomas and H. Hermansky, Toward Optimizing Stream Fusion in Multistream Recognition of Speech, 2011

Issues with data-driven features – Dealing with noise

Static Weights - Minimize the mean-squared –error between actual and estimated posteriors

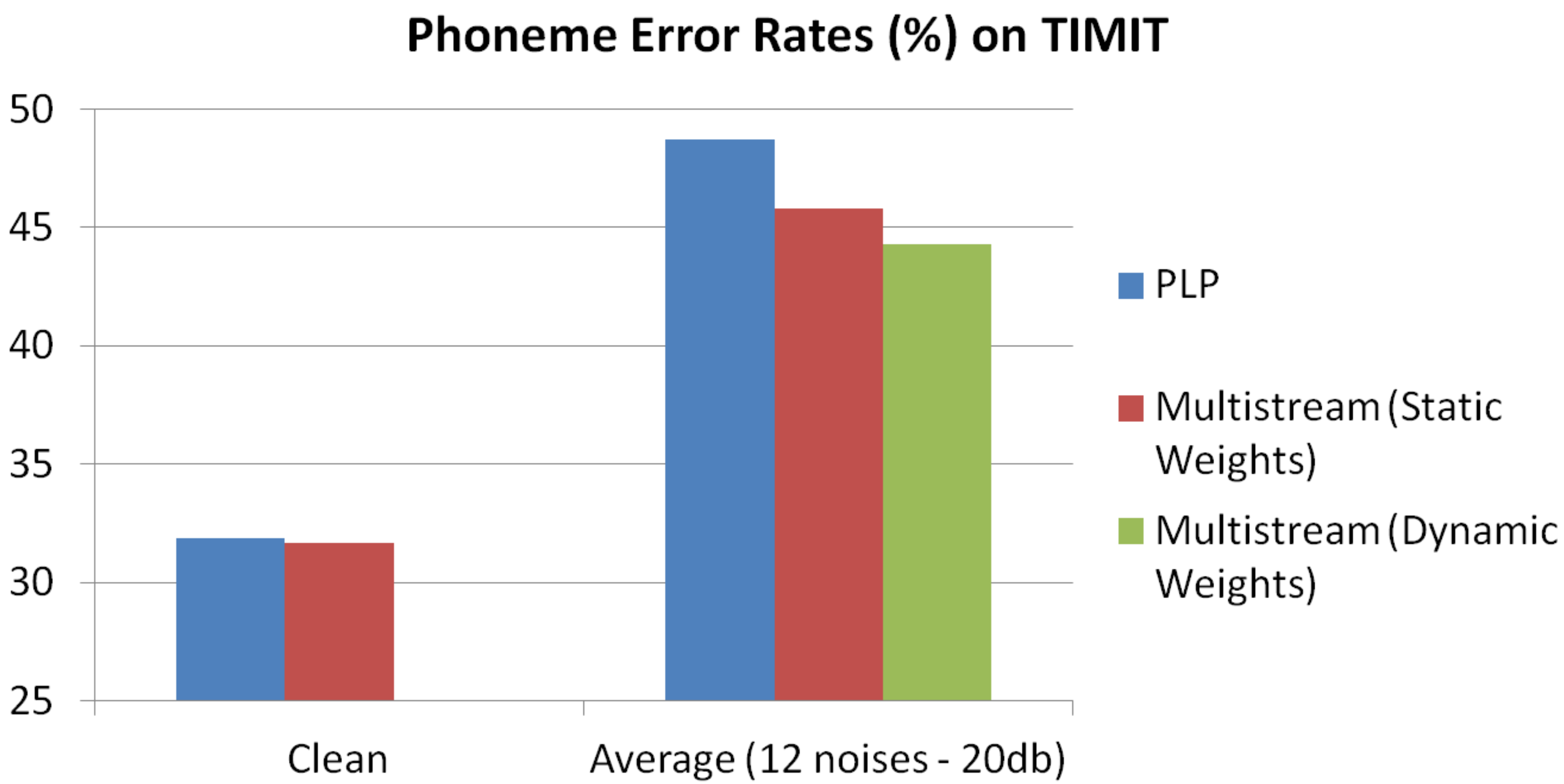
$$e = \sum_t \sum_k \left(p_k(t) - \hat{p}_k(t) \right)^2$$

$$W = \frac{\hat{P} P^t}{\hat{P} \hat{P}^t}$$

Cross-correlation of estimated posteriograms of each stream with the actual labels, normalized by the autocorrelation of stream posteriograms

N. Mesgarani, S. Thomas and H. Hermansky, Adaptive Stream Fusion in Multistream Recognition of Speech, 2011

Dealing with noise – Multi-stream Idea



N. Mesgarani, S. Thomas and H. Hermansky, Adaptive Stream Fusion in Multistream Recognition of Speech, 2011

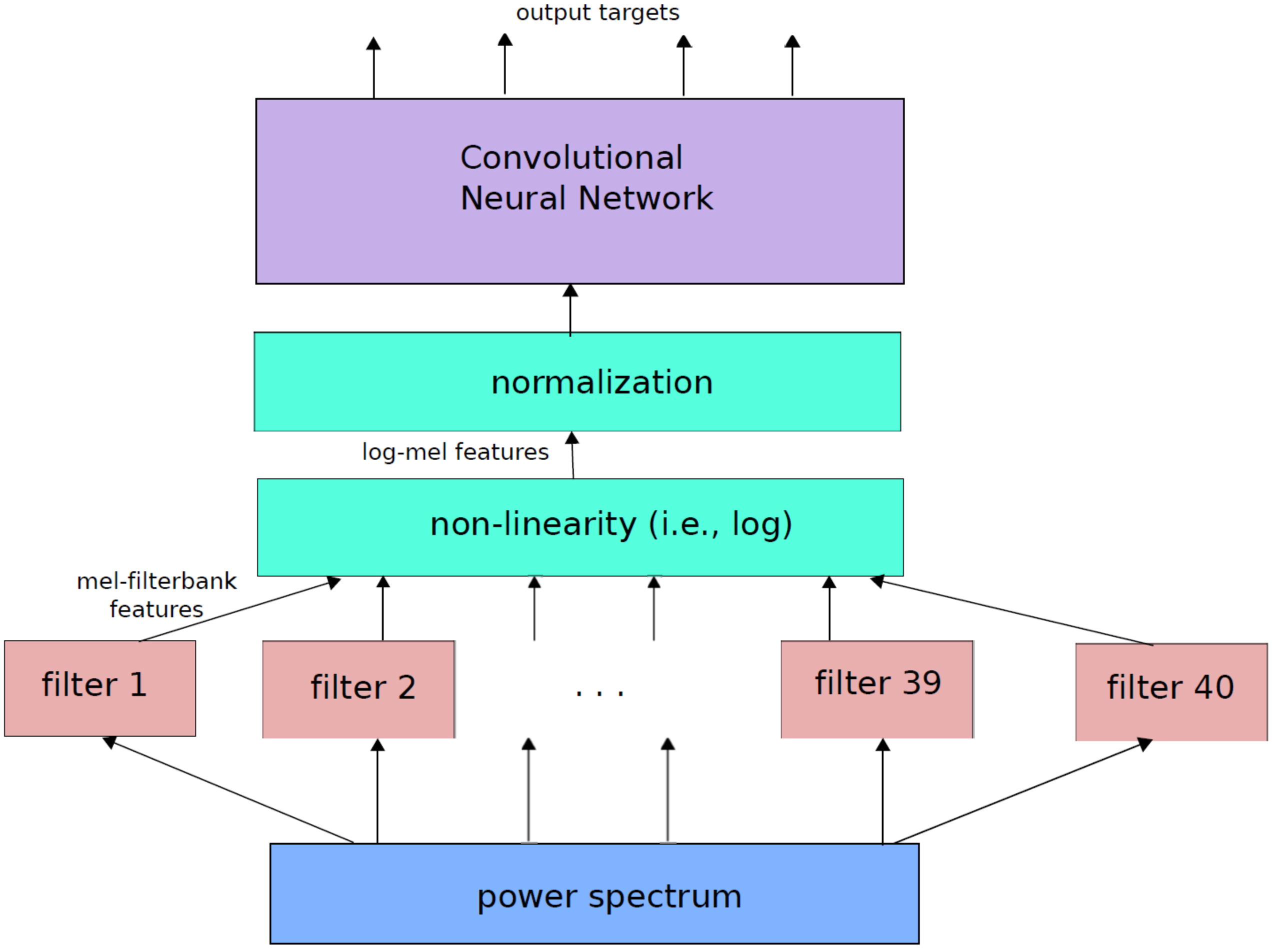
CONCLUSIONS

Summary

Challenges	Past	Present	Future
Preserving the relevant information for the application	MFCC/PLP	Multiple Data Representations	Adaptive Stream Combination
Removing unwanted redundancies in the signal – separating the information pertinent to the task.	Normalization Techniques	Data-driven Features	End-to-end Systems
Resilience to noise and other degradations	Spectral Subtraction	Multi-condition Training	Unsupervised Adaptation

Fuzzy Distinction Between Features and Models

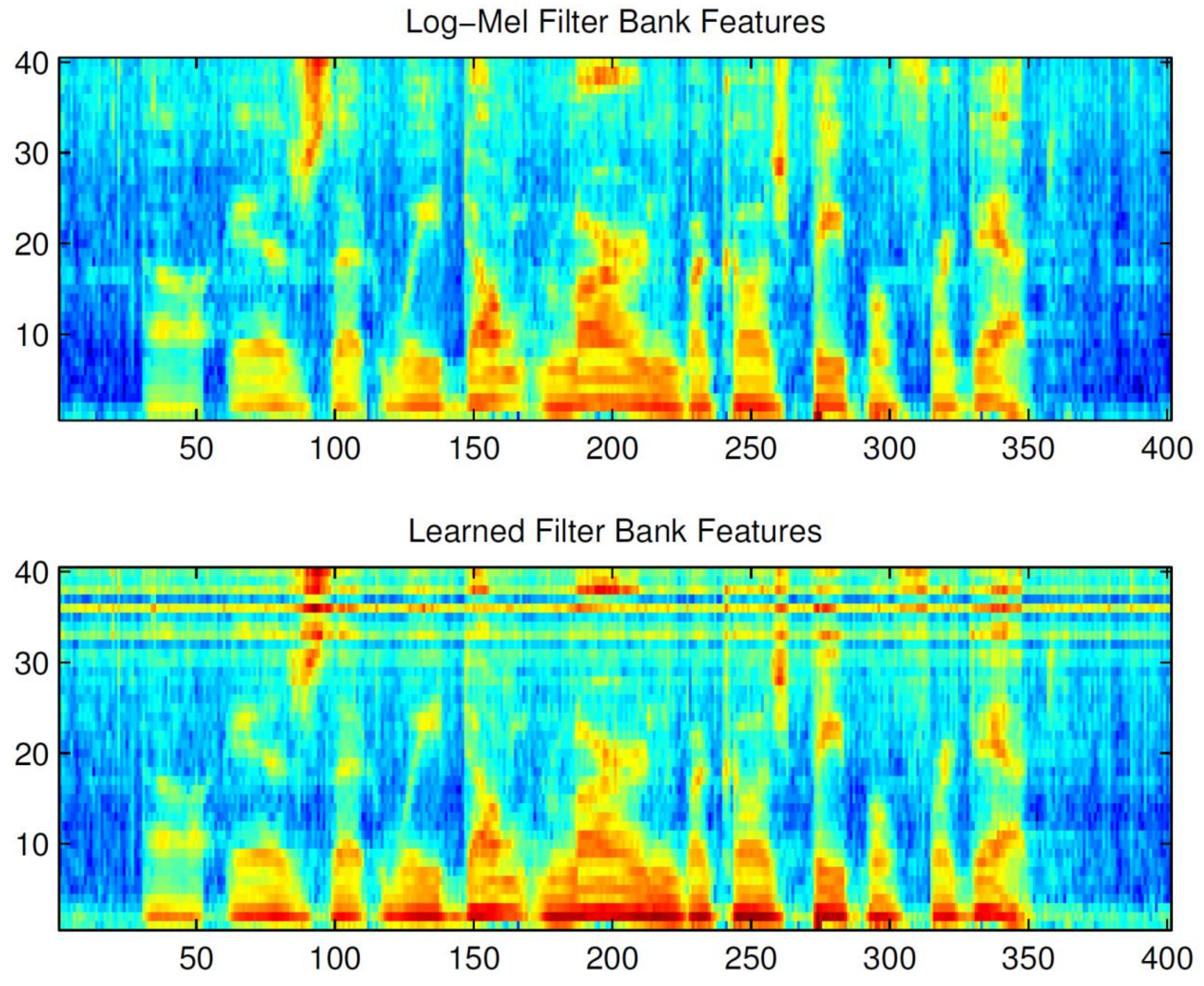
Learning Filter-banks directly from data



T. Sainath, B. Kingsbury, A. Mohamed, B. Ramabhadran, "Learning Filter-Banks Within A Neural Network Framework", ASRU 2013.

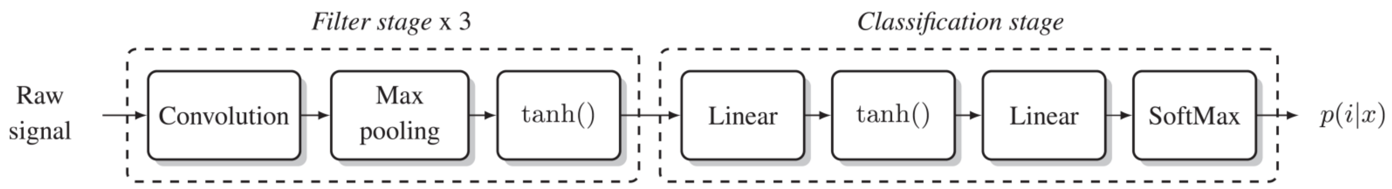
Fuzzy Distinction Between Features and Models

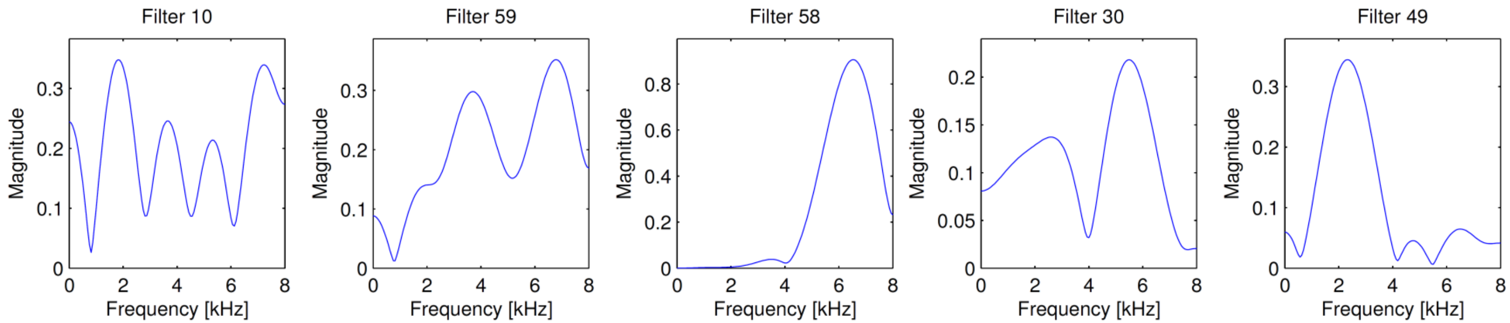
Learning Filter-banks directly from data



T. Sainath, B. Kingsbury, A. Mohamed, B. Ramabhadran, "Learning Filter-Banks Within A Neural Network Framework", ASRU 2013.

Fuzzy Distinction Between Features and Models





D. Palaz, R. Collobert, M. Doss, "Estimating Phoneme Class Conditional Probabilities from Raw Speech Signal using Convolutional Neural Networks", 2013.

Do We Need A Feature Extraction Step ?

- Pros
  - Extracting features and learning the model can be single step with the same target cost function
  - Not constrained by the assumptions in windowing and filtering prevalent in the current features
  - Purely data-driven

- Cons
  - Noise Robustness could be a huge challenge
  - Models may be bigger - prone to over-training
  - May require more data and computation

THANK YOU

Open Source Tools

[Dan Ellis, Feature Extraction Toolbox] - http://www.ee.columbia.edu/ln/rosa/matlab/

[Malcolm Slaney, Auditory Toolbox] - https://engineering.purdue.edu/~malcolm/interval/1998-010/

[Van Der Maaten, et al, Dimensionality Reduction Toolbox] - http://homepage.tudelft.nl/19j49/Matlab_Toolbox_for_Dimensionality_Reduction.html

[HTK ASR Toolkit] - http://htk.eng.cam.ac.uk/download.shtml

[ICSI Quicknet MLP Toolkit] - http://www1.icsi.berkeley.edu/Speech/qn.html

[Kaldi ASR Toolkit] - http://kaldi.sourceforge.net/about.html Povey, Daniel, et al. "The Kaldi speech recognition toolkit." *Proc. ASRU*. 2011.

[Ganapathy, FDLP Feature Extraction Toolkit] - http://old-site.clsp.jhu.edu/~sriram/software/soft.html

References

[Davis and Mermelstein, 1979] - Davis Steven, and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 28.4 (1980): 357-366.

[Makhoul, 1975] - Makhoul, John. "Linear prediction: A tutorial review." *Proceedings of the IEEE*63.4 (1975): 561-580.

[Hermansky, 1991] - Hermansky, Hynek. "Perceptual linear predictive (PLP) analysis of speech." *the Journal of the Acoustical Society of America* 87.4 (1990): 1738-1752.

[Noll, 1969] - Noll, A. M., "Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate," 1969.

[Childers, 1977] - Childers, D. G., Skinner, D.P., and Kemerait, R.C., "The cepstrum: A guide to processing," 1977.

[YIN, 2002] - Cheveigne, A. de, and Kawahara, H., "YIN, a fundamental frequency estimator for speech and music," 2002.

[Shriberg, 2000] - Shriberg, Elizabeth, et al. "Prosody-based automatic segmentation of speech into sentences and topics." *Speech communication* 32.1 (2000): 127-154.

[Furui, 1986] - Furui, Sadaoki. "Speaker-independent isolated word recognition using dynamic features of speech spectrum." *Acoustics, Speech and Signal Processing, IEEE Transactions on* 34.1 (1986): 52-59.

References

[Drullmann, 1994] - Drullman, Rob, Joost M. Festen, and Reinier Plomp. "Effect of temporal envelope smearing on speech reception." *The Journal of the Acoustical Society of America* 95.2 (1994): 1053-1064.

[Hermansky, 1994] - Hermansky, Hynek, and Nelson Morgan. "RASTA processing of speech."*Speech and Audio Processing, IEEE Transactions on* 2.4 (1994): 578-589.

[Kay, 1982] - Kay, R. H. "Hearing of modulation in sounds." *Physiol Rev* 62.3 (1982): 894-975.

[Kingsbury, 1998] - Kingsbury, Brian ED, Nelson Morgan, and Steven Greenberg. "Robust speech recognition using the modulation spectrogram." *Speech communication* 25.1 (1998): 117-132.

[Athineos, 2003] - Athineos, Marios, and Daniel PW Ellis. "Frequency-domain linear prediction for temporal features." *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*. IEEE, 2003.

[Ganapathy, 2009] - Ganapathy, Sriram, Samuel Thomas, and Hynek Hermansky. "Modulation frequency features for phoneme recognition in noisy speech." *The Journal of the Acoustical Society of America* 125.1 (2009): EL8-EL12.

[Reynolds, 1994] - Reynolds, Douglas A. "Experimental evaluation of features for robust speaker identification." *Speech and Audio Processing, IEEE Transactions on* 2.4 (1994): 639-643.

[Pelecanos, 2001] - Pelecanos, Jason, and Sridha Sridharan. "Feature warping for robust speaker verification." (2001): 213-218.

[Avendanos, 1996] - Avendano, Carlos, and Hynek Hermansky. "Study on the dereverberation of speech based on temporal envelope filtering." *ICSLP*. 1996.

References

[Gelbart, 2001] - Gelbart, David, and Nelson Morgan. "Evaluating long-term spectral subtraction for reverberant ASR." *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on.* IEEE, 2001.

[Boll, 1979] - Boll, Steven. "Suppression of acoustic noise in speech using spectral subtraction." Acoustics, Speech and Signal Processing, IEEE Transactions on 27.2 (1979): 113-120.

[Ephraim, 1984] - Ephraim, Yariv, and David Malah. "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator." Acoustics, Speech and Signal Processing, IEEE Transactions on 32.6 (1984): 1109-1121.

[Wolfe, 2001] - Wolfe, Patrick J., and Simon J. Godsill. "Efficient alternatives to the Ephraim and Malah suppression rule for audio signal enhancement." EURASIP Journal on Advances in Signal Processing 2003.10 (1900): 1043-1051.

[Cappe, 1994] - Cappé, Olivier. "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor." *IEEE Transactions on Speech and Audio Processing* 2.2 (1994): 345-349.

[Thomas, 2008] - Thomas, Samuel, Sriram Ganapathy, and Hynek Hermansky. "Recognition of reverberant speech using frequency domain linear prediction." *Signal Processing Letters, IEEE* 15 (2008): 681-684.

[Shamma, 2004] - Mesgarani, Nima, Shihab Shamma, and Malcolm Slaney. "Speech discrimination based on multiscale spectro-temporal modulations." *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on.* Vol. 1. IEEE, 2004.

[Kenny, 2005] - Kenny, Patrick, Gilles Boulianne, and Pierre Dumouchel. "Eigenvoice modeling with sparse training data." *Speech and Audio Processing, IEEE Transactions on* 13.3 (2005): 345-354.

APPENDIX - A

i-vector Estimation

- The proofs follow the details presented in [Kenny, 2005].
- Defining sufficient statistics from the recording. For the recording $X(s)$ and UBM $\lambda = \{\pi_c, \mu_c, \Sigma_c\}$, Let $p_\lambda(c \mid xi)$ denote the posterior probability of the mixture component given the feature vector x_i of dimension F for $c = 1 \ldots C$ and $i = 1 \ldots H(s)$
- The sufficient statistics are

$$N_c(s) = \sum_{i=1}^{H(s)} p_\lambda(c \mid x_i) \qquad S_{X,c}(s) = \sum_{i=1}^{H(s)} p_\lambda(c \mid x_i)\,(xi - \mu_c)$$

$$S_{XX,c}(s) = \sum_{i=1}^{H(s)} p_\lambda(c \mid x_i)\,(x_i - \mu_c)(x_i - \mu_c)^*$$

- Let $\boldsymbol{N}(s)$ denote the CF x CF block diagonal matrix with diagonal blocks $N_1(s)I, \ldots, N_c(s)I, \ldots N_C(s)I$ and I is F x F identity matrix.
- Let $\boldsymbol{S}_X(s)$ be the CF x 1 vector by concatenating $S_{X,1}(s), \ldots, S_{X,C}(s)$

i-vector Estimation

Theorem I The log-likelihood function is given by

$$log(p_V(X(s)\,|\,\boldsymbol{y}(s))) = G(s) + H_V(s, \boldsymbol{y}(s)) \qquad \textbf{(1)}$$

where

$$G(s) = \sum_{c=1}^{C}\left(N_c(s)\,log\frac{1}{(2\pi)^{F/2}|\Sigma_c|^{1/2}} - \frac{1}{2}tr(\Sigma_c{}^{-1}S_{XX,c}(s))\right)$$

$$H_V(s,\boldsymbol{y}) = \boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s) - \frac{1}{2}\boldsymbol{y}^*\boldsymbol{V}^*\boldsymbol{N}(s)\boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{y}$$

Proof

Let $\boldsymbol{O} = \boldsymbol{V}\boldsymbol{y}$ be vector of dimension CF with O_c denoting the c^{th} block of $\boldsymbol{O}$ and of dimension F. Also, let

$$S_{XX,c}(O_c) = \sum_{i=1}^{H(s)} p_{V,\lambda}(c\,|x_i)\,(x_i - \mu_c - O_c)(x_i - \mu_c - O_c)^* \qquad \textbf{(2)}$$

i-vector Estimation

Then, the likelihood function $p_V(X(s) \mid \boldsymbol{y}(s))$ is sequence of Gaussian models with $N_c(s)$ frames assigned to the c^{th} mixture component having a mean vector $\mu_c + O_c$,and diagonal covariance Σ_c. This gives (removing (s) for ease)

$$log(p_V(X \mid \boldsymbol{y})) = \sum_{c=1}^{C} \left(N_c \, log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} - \frac{1}{2} tr(\Sigma_c{}^{-1} S_{XX,c}(O_c)) \right) \qquad \textbf{(3)}$$

Expanding $S_{XX,c}(O_c)$ from Eq. (2) gives

$$S_{XX,c}(O_c) = S_{XX,c} - S_{X,c} O_c{}^* - O_c S_{X,c}{}^* + NcOcO_c{}^*$$

$$tr(\Sigma_c{}^{-1} S_{XX,c}(O_c)) = tr(\Sigma_c{}^{-1} S_{XX,c}) - 2 S_{X,c}{}^* \Sigma_c{}^{-1} O_c + O_c{}^* \Sigma_c{}^{-1} N_c O_c$$

$$\sum_{c=1}^{C} tr(\Sigma_c{}^{-1} S_{XX,c}(O_c)) = \sum_{c=1}^{C} tr(\Sigma_c{}^{-1} S_{XX,c}) - 2\boldsymbol{O}^* \boldsymbol{\Sigma}^{-1} \boldsymbol{S}_X + \boldsymbol{O}^* \boldsymbol{N} \boldsymbol{\Sigma}^{-1} \boldsymbol{O} \qquad \textbf{(4)}$$

i-vector Estimation

Substituting Eq. (4) in Eq. (3),

$$log(p_V(X \mid \boldsymbol{y})) = \sum_{c=1}^{C} \left(N_c \, log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} - \frac{1}{2} tr(\Sigma_c{}^{-1} S_{XX,c}) \right) + \boldsymbol{O}^* \boldsymbol{\Sigma}^{-1} \boldsymbol{S}_X + \boldsymbol{O}^* \boldsymbol{N} \boldsymbol{\Sigma}^{-1} \boldsymbol{O}$$

$$= G(s) + H_V(s, \boldsymbol{y}(s))$$

where the definition of $\boldsymbol{O} = \boldsymbol{V}\boldsymbol{y}$ was invoked in the last step. Thus, theorem-1 is proved for the likelihood function.

Theorem II The posterior distribution $p_\lambda(\boldsymbol{y} \mid X)$ is Gaussian with covariance matrix $\boldsymbol{l}^{-1}(s)$ and mean value $\boldsymbol{l}^{-1}(s)\, \boldsymbol{V}^* \, \boldsymbol{\Sigma}^{-1} \boldsymbol{S}_X(s)$ where $\boldsymbol{l}(s)$ is R x R

$$\boldsymbol{l}(s) = \boldsymbol{I} + \boldsymbol{V}^* \boldsymbol{\Sigma}^{-1} \boldsymbol{N}(s) \boldsymbol{V}$$

i-vector Estimation

Proof This is perhaps the most important component where the mean and covariance of the posterior distribution are found. In order to prove Theorem–II, it is enough to show that

$$p_{\boldsymbol{V},\lambda}(\boldsymbol{y} \mid X) \propto \exp\left(-\frac{1}{2}\big(\boldsymbol{y}-\boldsymbol{a}(s)\big)^{*}\boldsymbol{l}(s)\big(\boldsymbol{y}-\boldsymbol{a}(s)\big)\right)$$

where $\boldsymbol{a}(s) = \boldsymbol{l}^{-1}(s)\,\boldsymbol{V}^{*}\,\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_{X}(s)$. Ignoring the index s, using the Gaussian prior distribution of $\boldsymbol{y}$ and the results from Theorem–1 (Eq. (1)),

$$\begin{aligned}
p_{\boldsymbol{V},\lambda}(\boldsymbol{y} \mid X) &\propto p(X \mid \boldsymbol{y})\, N(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{I}) \\
&\propto \exp\left(\boldsymbol{y}^{*}\boldsymbol{V}^{*}\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_{X} - \frac{1}{2}\boldsymbol{y}^{*}\boldsymbol{V}^{*}\boldsymbol{N}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{y} - \frac{1}{2}\boldsymbol{y}^{*}\boldsymbol{y}\right) \\
&= \exp\left(\boldsymbol{y}^{*}\boldsymbol{V}^{*}\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_{X} - \frac{1}{2}\boldsymbol{y}^{*}\boldsymbol{l}\boldsymbol{y}\right) \\
&\propto \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{a})^{*}\boldsymbol{l}(\boldsymbol{y}-\boldsymbol{a})\right)
\end{aligned}$$

i-vector Estimation

Thus, Theorem–II is proved. Note that, since the posterior distribution of $\boldsymbol{y}$ is a Gaussian, the optimal estimate of i–vector $\underset{\boldsymbol{y}}{\operatorname{argmax}}\ p_{V,\lambda}(\boldsymbol{y}\,|X(s))$ is the mean given by $\boldsymbol{l}^{-1}(s)\ \boldsymbol{V}^{*}\ \boldsymbol{\Sigma}^{-1}\boldsymbol{S}_{X}(s)$

Theorem III Given initial estimate $\boldsymbol{V}_0$, the i–vector posterior distribution is given by Theorem–II. Using the conditional moments of the posterior, $\mathrm{E}[\boldsymbol{y}(s)]$ and $\mathrm{E}[\boldsymbol{y}(\boldsymbol{s})\boldsymbol{y}^{*}(s)]$, let the new estimate of $\boldsymbol{V}$ be the solution of

$$\sum_{s=1}^{S}\boldsymbol{N}(s)\boldsymbol{V}\ \mathrm{E}[\boldsymbol{y}(\boldsymbol{s})\boldsymbol{y}^{*}(s)] = \sum_{s=1}^{S}\boldsymbol{S}_{X}(s)\ \mathrm{E}[\boldsymbol{y}^{*}(s)]$$

Then, this new estimate $\boldsymbol{V}$ improves the data likelihood

$$\sum_{s=1}^{S} log\ (p_{V}(X(s)) \geq \sum_{s=1}^{S} log\ (p_{V_{\boldsymbol{0}}}(X(s))$$

i-vector Estimation

Proof This proof completes the re-estimation of the parameters in the EM algorithm. To prove this, we invoke the Jensen's inequality,

$$\sum_{s=1}^{S} \int \left(log \frac{p_{V}\big(X(s), \boldsymbol{y}(s)\big)}{p_{V_0}\big(X(s), \boldsymbol{y}(s)\big)} \right) p_{V_0}\big(\boldsymbol{y}(s) \mid X(s)\big) d\boldsymbol{y} \le \qquad (5)$$

$$\sum_{s=1}^{S} log \int \left(\frac{p_{V}\big(X(s), \boldsymbol{y}(s)\big)}{p_{V_0}\big(X(s), \boldsymbol{y}(s)\big)} \right) p_{V_0}\big(\boldsymbol{y}(s) \mid X(s)\big) d\boldsymbol{y}$$

The right hand side of the inequality simplifies to

$$\sum_{s=1}^{S} \log p_{V}\big(X(s), \boldsymbol{y}(s)\big) - \sum_{s=1}^{S} \log\, p_{V_0}\big(X(s), \boldsymbol{y}(s)\big)$$

Thus, Theorem-III can be proved (non-decreasing likelihood) if the left hand side of the inequality Eq. (5) is non-negative. Now,

$$p_{V}\big(X(s), \boldsymbol{y}(s)\big) = p_{V}\big(X(s) \mid \boldsymbol{y}(s)\big)\, N(\boldsymbol{y} \mid \boldsymbol{0}, \boldsymbol{I})$$

i-vector Estimation

The left hand side of inequality Eq. (5) can be written as $\mathcal{A}_V - \mathcal{A}_{V_0}$ where

$$\mathcal{A}_V = \sum_s \int p_V(X(s) \mid \boldsymbol{y}(s))\, p_{V_0}(\boldsymbol{y}(s) \mid X(s))\, d\boldsymbol{y}$$

To summarize, we have shown that $\sum_{s=1}^{S} log\ (p_V(X(s)) \geq \sum_{s=1}^{S} log\ (p_{V_0}(X(s))$ if $\mathcal{A}_V \geq \mathcal{A}_{V_0}$. This is the standard EM formulation with the auxiliary function $\mathcal{A}_V$ and the above condition can be met by maximizing $\mathcal{A}_V$ with respect to $\boldsymbol{V}$. Using Theorem–I

$$\mathcal{A}_V(X(s)) = \sum_{s=1}^{S} \int [G(s) + H_V(s, y(s))]\, p_{V_0}(\boldsymbol{y}(s) \mid X(s))\, d\boldsymbol{y}$$

$$= \sum_{s=1}^{S} [G(s) + E[H_V(s, y(s))]]$$

where $E[H_V(s, y(s))]$ is the conditional expectation given $X(s)$. The term with $G(s)$ is independent of $\boldsymbol{V}$. Thus maximizing $\mathcal{A}_V$ reduces to maximizing $\sum_{s=1}^{S} E[H_V(s, y(s))]$

i-vector Estimation

Using the definition of $H_V(s, y(s))$ from Theorem–I

$$\sum_{s=1}^{S} E\left[H_V(s, y(s))\right] = \sum_{s=1}^{S} E\left[\boldsymbol{y}^*(s)\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s) - \frac{1}{2}\boldsymbol{y}^*(s)\boldsymbol{V}^*\boldsymbol{N}(s)\boldsymbol{\Sigma}^{-1}\boldsymbol{V}\boldsymbol{y}(s)\right]$$

$$= \sum_{s=1}^{S} E[\boldsymbol{y}^*(s)]\,\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s) - \frac{1}{2} tr\left[\boldsymbol{V}^*\boldsymbol{N}(s)\boldsymbol{\Sigma}^{-1}\boldsymbol{V}E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]\right]$$

$$= \sum_{s=1}^{S} tr\left[\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{V}^*\boldsymbol{S}_X(s)E[\boldsymbol{y}^*(s)] - \frac{1}{2}\boldsymbol{N}(s)\boldsymbol{V}E[\boldsymbol{y}(s)\boldsymbol{y}^*(s)]\boldsymbol{V}^*\right)\right]$$

Taking derivative of above w.r.t. $\boldsymbol{V}$ and equating to $\boldsymbol{0}$

$$\sum_{s=1}^{S} \boldsymbol{N}(s)\boldsymbol{V}\ \mathrm{E}[\boldsymbol{y}(\boldsymbol{s})\boldsymbol{y}^*(s)] = \sum_{s=1}^{S} \boldsymbol{S}_X(s)\ \mathrm{E}[\boldsymbol{y}^*(s)\ ]$$

Thus, Theorem–III is proved and provides the re-estimation formula for $\boldsymbol{V}$

APPENDIX - B

Estimating Posteriors with MLPs

- Neural networks estimate posterior probabilities of classes when trained using squared loss function for classification problem [Lippmann, 1991].
- Let X denote input vector $\{x_i, i = 1 \ldots D\}$ which is to be assigned to one of the classes $\{C_i, i = 1 \ldots M\}$. By Bayes theorem, the class posterior is

$$p(C_i \mid x) = \frac{p(x \mid C_i)}{p(x)}$$

- Let $\{y_i(X), i = 1 \ldots M\}$ denote the output of the network and $\{d_i, i = 1 \ldots M\}$ denote the desired outputs. For the classification problem, if X belongs to C_j, then $d_i = 1$ for $i = j$ and 0 otherwise.

Estimating Posteriors with MLPs

- The squared loss function is defined as

$$\Delta = \boldsymbol{E}\left\{\sum_{i=1}^{M}(y_i(X) - di)^2\right\} = \int\left\{\sum_{i=1}^{M}(y_i(X) - di)^2\right\}p(X)dX$$

$$= \int\sum_{j=1}^{M}\left\{\sum_{i=1}^{M}(y_i(X) - di)^2\right\}p(X, Cj)dX$$

$$= \int\left\{\sum_{j=1}^{M}\sum_{i=1}^{M}(y_i(X) - di)^2 p(C_j \mid X)\right\}p(X)dX$$

$$= \boldsymbol{E}\left\{\sum_{j=1}^{M}\sum_{i=1}^{M}(y_i(X) - di)^2 p(C_j \mid X)\right\}$$

- Expanding the function inside the expectation

Estimating Posteriors with MLPs

- The squared loss function is defined as

$$\Delta = \boldsymbol{E}\left\{\sum_{j=1}^{M}\sum_{i=1}^{M}(y_i{}^2(X)p(C_j \mid X) - 2diyi(X)p(C_j \mid X) + d_i{}^2p(C_j \mid X))\right\}$$

- Now, $y_i(X)$ is only a function of X and $\sum_{j=1}^{M} p(C_j \mid X) = 1$

$$\Delta = \boldsymbol{E}\left\{\sum_{i=1}^{M}\left[y_i{}^2(X) - 2y_i(X)\sum_{j=1}^{M} d_i p(C_j \mid X) + \sum_{j=1}^{M} d_i{}^2 p(C_j \mid X)\right]\right\}$$

$$= \boldsymbol{E}\left\{\sum_{i=1}^{M}[y_i{}^2(X) - 2yi(X)\,\boldsymbol{E}\{d_i \mid X\} + \boldsymbol{E}\{d_i{}^2 \mid X\}]\right\}$$

- Adding and subtracting the term $\boldsymbol{E}^2\{d_i \mid X\}$ to make a perfect square

Estimating Posteriors with MLPs

$$\Delta = \boldsymbol{E}\left\{\sum_{i=1}^{M}[y_i^2(X) - 2yi(X)\,\boldsymbol{E}\{d_i \mid X\} + \boldsymbol{E}^2\{d_i \mid X\} - \boldsymbol{E}^2\{d_i \mid X\} + \boldsymbol{E}\{d_i^2 \mid X\}]\right\}$$

$$= \boldsymbol{E}\left\{\sum_{i=1}^{M}[y_i(X) - \boldsymbol{E}\{d_i \mid X\}]^2\right\} + \boldsymbol{E}\left\{\sum_{i=1}^{M} var\{d_i \mid X\}\right\}$$

- The second term $var\{d_i \mid X\}$ is independent of the weights.
- For classification case ($d_i = \delta_{ij} = 1$ for $i = j$ and 0 otherwise for X belonging to class C_j) ,

$$\boldsymbol{E}\{d_i \mid X\} = \sum_{j=1}^{M} d_i\, p(C_j \mid X) = \sum_{j=1}^{M} \delta_{ij}\, p(C_j \mid X) = p(C_i \mid X)$$

- Thus, minimizing squared loss function Δ estimates the posterior probabilities $p(C_i \mid X)$