Robustness Techniques for Speech Recognition

ShihHsiang 2006
Reference

• X. Huang et al. *Spoken Language Processing* (2001). Chapter 10
• C. P. Chen, K. Kirchhoff and J. Bilmes, "LOW-RESOURCE NOISE-ROBUST FEATURE POST-PROCESSING ON AURORA 2.0 “, UWEE Technical Report, ICSLP, 2002
• W. Zhu and D. Shaughnessy, "LOG-ENERGY DYNAMIC RANGE NORMALIZATON FOR ROBUST SPEECH RECOGNITION", ICASSP 2005
• L. Deng, A. Acero, M. Plumpe and X. Huang. Large-Vocabulary Speech Recognition under Adverse Acoustic Environments, ICSLP 2000
• Ni-chun Wang, Jeih-weih Hung and Lin-shan Lee, “Data-driven Temporal Filters Based on Multi-Eigenvectors for Robust Features in Speech Recognition, ICASSP 2003
Outline

• Introduction
• Review Noise Type
• Robustness Approaches
• Conclusion
Introduction

• The Diagram for Speech Recognition

  ![Speech Recognition Diagram]

  - **Acoustic Processing**
    - Feature Extraction
    - Likelihood computation
    - Acoustic model

  - **Linguistic Processing**
    - Linguistic Network Decoding
    - Language model
    - Lexicon

  **Recognition results**

• 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.)

- The main reason is that performance of existing recognition system which assume a noise-free tranquil environment.

\[ y(n) = \left\{ \begin{array}{c}
  s(n) \\
  + d_1(n) \left( h_{MIKE}(n) + d_2(n) \right) h_{CHAN}(n) + d_3(n)
\end{array} \right. \]

Hansen, 1996
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.

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)
- Inherently robust acoustic features (both model- and feature-based)
  - Discriminative acoustic features
Introduction (cont.)

• Speech recognition System can be divided into two parts
  – Front-End processing (Feature extraction)
    • Suppress the noise
    • Get more robust parameters
  – Back-End processing (HMM decoding)
    • Compensate for noise
    • Adapt the parameters

• In order to get more noise robust features, there are numerous efforts (based on MFCCs or PLP)
  – Add pre-processing
    • Noise reduction, speech enhancement
  – Incorporate algorithms in an MFCC calculation framework
    • Frequency masking, SNR-Normalization
  – Add feature post-processing techniques
    • CMS, MVN, HEQ … etc
The robustness techniques can be applied.
The Noise Types

\[ x[m] = s[m] * h[m] + n[m] \quad \text{Time-Domain} \]

\[ X(\omega) = S(\omega)H(\omega) + N(\omega) \quad \text{Frequency-Domain} \]
Additive Noises / Convolutional 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 power spectral density does not change over time
  - Non-stationary noises (most difficult)
    - Machine gun, door slams, keyboard clicks, radio/TV, and other speakers’ voices
    - the statistical properties change over time
- 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.
Noise Characteristics

• White Noise
  – The power spectrum is flat \( S_{\omega} = q \), a condition equivalent to different samples being uncorrelated, \( R_{mn}[m] = q \delta[m] \)
  – 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 noise 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
The Effectives of Active Noise

Figure 4.1: Plots of “corrupted-speech” distribution (solid), and maximum likelihood Gaussian distribution (dashed)
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 testing noisy conditions
  – *The model part is modified while the feature part remains unchanged*

• Inherently 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*
Robustness Approaches

• Linear approach
  – Spectral Subtraction (SS)
  – Wiener Filtering (WF)
  – Cepstral Mean subtraction (CMS)
  – Time filtering of log-FBE’s (RASTA)
  – Auto-Regression Moving Average (ARMA)
  – Log-energy Dynamic Range Normalization (ERN)

• Nonlinear approach
  – Histogram Equalization (HEQ)
  – Stereo-based Piecewise Linear Compensation (SPLICE)

• Retraining on Corrupted Speech

• Model Adaptation

• Inherently Robust Parameters for Speech
  – Multi Layer Peceptrons (MLP)
  – Discriminative Feature (PCA,LDA, HLDA)
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)

- 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 known to be just noise: \[ \hat{P}_n(\omega) = \frac{1}{M} \sum_{i=0}^{M-1} P_{n,i}(\omega) \]

Boll, 1979
Spectral Subtraction (cont.)

- 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

**FIGURE 9.1** The spectral subtraction technique, $\gamma=1$ gives magnitude subtraction, $\gamma=2$ gives power subtraction.
• Modification: Nonlinear Spectral Subtraction (NSS)

\[
\hat{P}_s(\omega) = \begin{cases} 
    P_X(\omega) - P_N(\omega), & \text{if } P_X(\omega) \geq P_N(\omega) \\
    P_N(\omega), & \text{otherwise}
\end{cases}
\]

\[
\bar{P}_X(\omega) \text{and } \bar{P}_N(\omega) \text{: smoothed noisy and noise spectrum}
\]

\[
\hat{P}_s(\omega) = \begin{cases} 
    P_X(\omega) - \phi(\omega), & \text{if } P_X(\omega) > \phi(\omega) + \beta \cdot P_N(\omega) \\
    \beta \cdot P_N(\omega), & \text{otherwise}
\end{cases}
\]

\[
\bar{P}_X(\omega) \text{and } \bar{P}_N(\omega) \text{: smoothed noisy and noise spectrum}
\]

\[
\phi(\omega) \text{: a non-linear function according to SNR}
\]

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

\[
\hat{P}_S(\omega) = P_X(\omega) - P_N(\omega)
\]

\[
= P_X(\omega) \left[ 1 - \frac{P_N(\omega)}{P_X(\omega)} \right] = P_X(\omega) \left[ \frac{P_S(\omega)}{P_S(\omega) + P_N(\omega)} \right] \quad \text{(supposed that } P_X(\omega) \approx P_S(\omega) + P_N(\omega))
\]

\[
= P_X(\omega) \left[ 1 + \frac{1}{R(\omega)} \right]^{-1} \quad (\ R(\omega) = \frac{P_S(\omega)}{P_N(\omega)}: \text{instantaneous SNR})
\]

\[
\therefore \text{The time varying suppression filter is given approximately by:}
\]

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

\text{Spectrum Domain}
Wiener Filtering (WF)

- From the Statistical Point of View
  - The process $x[m]$ is the sum of the random process $s[m]$ and the additive noise process $n[m]$
    \[ x[m] = s[m] + n[m] \]
  - Find a linear estimate $\hat{s}[m]$ in terms of the process $x[m]$:
    - Or to find a linear filter $h[m]$ such that the sequence $\hat{s}[m] = x[m] * h[m]$ minimizes the expected value of $(\hat{s}[m] - s[m])^2$

\[
\hat{s}[m] = x[m] * h[m] = \sum_{l=-\infty}^{\infty} h[l] x[m - l]
\]

\[
\begin{array}{ccc}
\text{Noisy Speech} & \text{A linear filter} & \text{Clean Speech} \\
\text{x[m]} & h[n] & \hat{s}[m] \\
\end{array}
\]
Wiener Filtering (cont.)

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

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

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

\[ \Rightarrow \forall k 2\left(\sum_{l=-\infty}^{\infty} h[l]x[m-l]\right)x[m-k] - 2s[m]x[m-k] = 0 \]

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

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

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

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

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

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

Take summation for \( k \)

Take Fourier transform

\( R_s[n] \) and \( R_x[n] \): are respectively the autocorrelation sequences of \( s[n] \) and \( x[n] \)

\( s[m] \) and \( n[m] \) are statistically independent!
Wiener Filtering (cont.)

• Minimize the expectation of the squared error (MMSE estimate)
  \[ S_{ss}(\omega) = H(\omega)S_{xx}(\omega) \]

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

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

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

• Approach 1: Ephraim(1992)
  – the use of an HMM where, if we know the current frame falls under, we can use its mean spectrum as
  – 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):
    \[ \hat{P}_S(t, \omega) = P_X(t, \omega)H(t-1, \omega), \quad (t: \text{frame index}, H(\cdot): \text{Wiener filter}) \]
    \[ \therefore H(t, \omega) = \frac{\hat{P}_S(t, \omega)}{\hat{P}_S(t, \omega) + P_N(\omega)} \]
    \[ \tilde{P}_S(t, \omega) = P_X(t, \omega)H(t, \omega) \]
  - The initial estimate of the speech spectrum can be derived from spectral subtraction.

- **Approach - III:**
  - Slow down the rapid frame-to-frame movement of the object speech power spectrum estimate by apply temporal smoothing:
    \[ \hat{P}_S(t, \omega) = \alpha \cdot \hat{P}_S(t-1, \omega) + (1 - \alpha) \cdot \hat{P}_S(t, \omega) \]
    Then use \( \hat{P}_S(t, \omega) \) to replace \( \tilde{P}_S(t, \omega) \) in
    \[ H(t, \omega) = \frac{\hat{P}_S(t, \omega)}{\hat{P}_S(t, \omega) + P_N(\omega)} \Rightarrow H(t, \omega) = \frac{\hat{P}_S(t, \omega)}{\tilde{P}_S(t, \omega) + P_N(\omega)} \]
Wiener Filtering (cont.)

Figure 13.3 Enancement 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.
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

\[ x[n] = s[n] * h[n] \]  

\[ X(\omega) = S(\omega)H(\omega) \]

\[ X^l = \log |SH|^2 = \log |S|^2 + \log |H|^2 = S^l + H^l \]  

\[ CX^l = C(S^l + H^l) = CS^l + CH^l \]

\[ \overline{CS^l} = \frac{1}{T} \sum_{t=0}^{T-1} CS^l_t \] \quad and \quad \[ \overline{CX^l} = \frac{1}{T} \sum_{t=0}^{T-1} (CS^l_t + CH^l_t) = \overline{CS^l} + CH^l \]

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

Training: \[ CX(1)^l = C(S^l + H(1)^l) = CS^l + CH(1)^l \]  

Testing: \[ CX(2)^l = C(S^l + H(2)^l) = CS^l + CH(2)^l \]

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

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

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.)

- 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)

**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 bi-gram language model.

*Cepstral delta and delta-delta features are computed prior to the CMN operation so that they are unaffected.*
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 coefficients are identical and equal to $\frac{1}{T}$, 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.

\[
\overline{CX}_i^t = \alpha \cdot CX_i^t + (1 - \alpha) \cdot \overline{CX}_{i-1}^t, \quad (\overline{CX}_i^t : \text{cepstral mean})
\]
Relative Spectral Temporal Filter (RASTA)

- **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 tract 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
Relative Spectral Temporal Filter (cont.)

- The IIR transfer function

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

- An other version

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

\[ \tilde{c}[t] = 0.98 \cdot \tilde{c}[t-1] + 0.2 \cdot c[t] + 0.1 \cdot c[t-1] - 0.1 \cdot c[t-3] + 0.2 \cdot c[t-4] \]

RASTA has a peak at about 4Hz (modulation frequency)
Auto-Regression Moving Average (ARMA)

wave $\rightarrow$ FE $\rightarrow$ C $\rightarrow$ MS+VN $\rightarrow$ $\tilde{C}$ $\rightarrow$ ARMA $\rightarrow$ $\tilde{C}$ $\rightarrow$ HMM

- The first step is standard mean subtraction (MS)
  \[ C'_{td} = C_{td} - \mu_d, \quad \mu_d = \frac{1}{T} \sum_{t=1}^{T} C_{td} \]
- The second step is variance normalization (VN)
  \[ \overline{C}_{td} = \frac{C'_{td}}{\sigma_d} = \frac{C_{td} - \mu_d}{\sigma_d}, \quad \sigma_d = \sqrt{\frac{1}{T} (C_{td} - \mu_d)^2} \]
- The third step is auto-regression moving average (ARMA)
  \[
  \tilde{C}_{td} = \begin{cases}
  \sum_{i=1}^{M} \overline{C}_{(t-i)d} + \sum_{j=0}^{M} \overline{C}_{(t+j)d} & \text{if } M < t \leq T - M \\
  2M + 1 & \text{otherwise}
  \end{cases}
  \]

  $C = \begin{bmatrix}
  C_{11} & C_{21} & \cdots & C_{T1} \\
  C_{12} & C_{22} & \cdots & C_{T2} \\
  \vdots & \vdots & \ddots & \vdots \\
  C_{1D} & C_{2D} & \cdots & C_{TD}
  \end{bmatrix}$

  feature vector (cepstral coefficient)

  Chen, 2002
Auto-Regression Moving Average (cont.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Train Condition</th>
<th>Word Acc.</th>
<th>WER Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline[1]</td>
<td>86.39</td>
<td>60.06</td>
<td>73.23</td>
</tr>
<tr>
<td>MS+VN ($M = 0$)</td>
<td>91.40</td>
<td>78.25</td>
<td>84.83</td>
</tr>
<tr>
<td>MS+VN+ARMA</td>
<td>92.55</td>
<td>84.97</td>
<td>88.76</td>
</tr>
</tbody>
</table>

Before feature processing.

After feature processing.
Log-energy Dynamic Range Normalization (ERN)

- Comparing with cepstral coefficients, the log-energy feature has quite different characteristics
  - Comparing with clean speech, the log-energy feature sequence of noisy speech are
    - Elevated minimum value
    - Valleys are buried by additive noise energy, while peaks are not affected as much

`Figure 1: Comparison of log energy feature sequences between clean and noisy speech.`

Zhu, 2005
Log-energy Dynamic Range Normalization (cont.)

- **Log-energy dynamic range** of the sequence

\[
D.R.(dB) = 10 \times \frac{\text{Max}(\log(\text{Energy}_i)_{i=1..n})}{\text{Min}(\log(\text{Energy}_i)_{i=1..n})}
\]

- **Algorithm**
  - Find \( \text{Max} = \text{Max}(\log(\text{Energy}_i)_{i=1..n}) \)
  - \( \text{Min} = \text{Min}(\log(\text{Energy}_i)_{i=1..n}) \)
  - Calculate target \( T_{\text{Min}} = \alpha \times \text{Max}(\log(\text{Energy}_i)_{i=1..n}) \)
  - If \( \text{Min}(\log(\text{Energy}_i)_{i=1..n}) < T_{\text{Min}} \)
    - For \( i=1..n \)
      - Liner
        \[
        \log(\text{Energy}_i) = \log(\text{Energy}_i) + \frac{T_{\text{Min}} - \text{Min}}{\text{Max} - \text{Min}} \times (\text{Max} - \log(\text{Energy}_i))
        \]
      - Non-Liner
        \[
        \log(\text{Energy}_i) = \log(\text{Energy}_i) + \frac{T_{\text{Min}} - \text{Min}}{\log(\text{Max}) - \log(\text{Min})} \times (\log(\text{Max}) - \log(\log(\text{Energy}_i)))
        \]
Log-energy Dynamic Range Normalization (cont.)

*Figure 2:* Schematic representation of scaling effect of log-energy dynamic range normalization algorithm.
Histogram Equalization (HEQ)

- Histogram normalization relies on two basic assumptions
  - The global statistics of the speech signal are independent of what is actually spoken
  - The feature space dimensions are oriented such that the variations that are tackled by histogram normalization are uncorrelated in each dimension

Figure 7.1: Schematic distribution of training and test data in a two-dimensional example feature space. The marginal distributions are plotted along both axes.
Histogram Equalization (cont.)
Histogram Equalization (cont.)

• The basic normalization algorithm is as follows
  – Compute a normalized histogram \( \tilde{p}(x) \) on the full training corpus
  – Compute the cumulative training data histogram \( \tilde{P}(x) \) which becomes the reference histogram
    \[
    \tilde{P}(x) = \int_{-\infty}^{x} dx' \tilde{p}(x')
    \]
  – Compute a normalized histogram \( p_r(x) \) from all data \( X_r \)
  – Compute the cumulative condition-dependent histogram \( \tilde{P}_r(x) \)
    \[
    \tilde{P}_r(x) = \int_{-\infty}^{x} dx' \tilde{p}_r(x')
    \]
  – Replace each value \( x \) by \( \tilde{x} \) that corresponds to the same point in the cumulative reference histogram
    \[
    x = \tilde{P}^{-1}(P_r(x))
    \]
Stereo-based **Piecewise Linear Compensation (SPLICE)**

- One major limitation of all SS techniques is its assumption of noise stationary
  - SS techniques are unable to exploit correlations among the frequency components, and they do not have knowledge what clean speech look like
- In SPLICE, $y$ is modeled by a mixture of Gaussians, and the a posterori probability of clean speech vector $x$ give the noisy speech and give the mixture component ($k$) is modeled using an **additive correction vector** $r_k$

$$p(x \mid y, k) = N(x; y + r_k, \Gamma_k)$$

Deng, 2000
Stereo-based Piecewise Linear Compensation (cont.)

- The essence of SPLICE algorithm is the application of MAP principle to deriving the optimal estimate for the noise-reduced speech. This gives:

$$
\hat{x} = \arg \max_x p(x \mid y) = \arg \max_x \frac{p(x, y)}{p(y)} \\
= \arg \max_x p(x, y) \\
= \arg \max_x \sum_{k=0}^{K-1} c_k p(x, y \mid k) \\
\approx \arg \max_x \arg \max_k c_k p(x, y \mid k) \\
= \arg \max_x \arg \max_k c_k p(y \mid k) p(x \mid y, k) \\
= \arg \max_x \left\{ \arg \max_k c_k N(y; \mu_k, \Sigma_k) \right\} N(x; y + r_k, \Gamma_k)
$$

Step 1  Step 2
Stereo-based Piecewise Linear Compensation (cont.)

- **Step 1:** finding the optimal mixture component

\[
\hat{k} = \arg \max_k c_k N(y; \mu_k, \Sigma_k)
\]

- **Step 2:** optimizes the second Gaussian pdf

\[
\hat{x} = y + r_k
\]

\[
\sum_{i=0}^{T-1} P(k | y_t)(x_t - y_t)
\]

\[
r_k = \frac{\sum_{i=0}^{T-1} P(k | y_t)}{\sum_{i=0}^{T-1} P(k | y_t)}
\]

the correction vectors are trained using the stereo recordings for both the clean and noisy speech data based on the maximum likelihood principle.

Assumption: the conditional mean of the a posteriori probability \(p(x|y)\) is a shifted version of the noisy data \(y\) is used for implementation simplicity only \(\Rightarrow\) a rotation on \(y\) is needed.
Retraining on Corrupted Speech

- **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 us 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](image-url) 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.

![Graph showing word error rate (%) vs. SNR (dB)](image)

**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
Spectral Entropy based Feature for Robust ASR

- For voiced sounds, spectra have clear formants
  - Entropies of such spectra will be low
  - On the other hand spectra of unvoiced sounds are flatter (entropies should be higher)
  - Entropy of a spectrum can be used as an estimate for voicing/un-voicing decision \( \Rightarrow \) VAD

- To compute entropy of a spectrum we converted the spectrum into a PMF like function by normalizing it

\[
x_i = \frac{X_i}{\sum_{i=1}^{N} X_i} \quad \text{for } i = 1 \text{ to } N
\]

- Entropy for each frame was computed by

\[
H = - \sum_{i=1}^{N} x_i \log_2 x_i
\]

Misra, 2004
Spectral Entropy based Feature for Robust ASR (cont.)

• Realizing that full-band entropy can capture only the gross peakiness of the spectrum but not the location of the formants
  – While multi-resolution entropy feature can capture the location of the formants

• To extract multi-band entropy features, we divided the full-band spectrum into J non-overlapping sub-bands of equal size
  – Entropy was computed for each sub-band and we obtained one entropy value for each sub-band
  – These sub-band entropy values indicate the presence or absence of formants in that sub-band
Spectral Entropy based Feature for Robust ASR (cont.)

- **Experiment Result**
  - Appending the time derivatives of the entropy feature to the entropy feature
  - Feature combination

<table>
<thead>
<tr>
<th>Feature</th>
<th>Clean</th>
<th>SNR12</th>
<th>SNR6</th>
<th>SNR0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLP</td>
<td>4.3%*</td>
<td>10.3%</td>
<td>20.1%</td>
<td>41.9%*</td>
</tr>
<tr>
<td>24-Mel</td>
<td>7.1%</td>
<td>12.1%</td>
<td>19.9%</td>
<td>37.7%</td>
</tr>
<tr>
<td>PLP + 24-Mel</td>
<td>4.2%*</td>
<td>9.7%</td>
<td>18.5%</td>
<td>41.1%*</td>
</tr>
</tbody>
</table>

**Table 3.** WERs for PLP feature, 24 Mel-band entropy feature and its time derivatives (24-Mel), and the two features appended (PLP + 24-Mel), in TANDEM system under different noise conditions. * indicates that the difference in performance is not significant.

<table>
<thead>
<tr>
<th>Feature</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-band Entropy</td>
<td>91.6%</td>
</tr>
<tr>
<td>2-bands Entropy</td>
<td>74.4%</td>
</tr>
<tr>
<td>3-bands Entropy</td>
<td>59.5%</td>
</tr>
<tr>
<td>4-bands Entropy</td>
<td>42.7%</td>
</tr>
<tr>
<td>8-bands Entropy</td>
<td>24.3%</td>
</tr>
<tr>
<td>16-bands Entropy</td>
<td>18.6%</td>
</tr>
<tr>
<td>24-bands Entropy</td>
<td>16.2%</td>
</tr>
<tr>
<td>32-bands Entropy</td>
<td>15.1%</td>
</tr>
<tr>
<td>24 Mel-bands Entropy</td>
<td>15.7%</td>
</tr>
</tbody>
</table>

**Table 1.** Word-Error-Rates (WERs) for clean speech for multi-band spectral entropy features in hybrid system for different number of sub-bands. Only Mel-bands are overlapping. Rest of the sub-bands are non-overlapping.
TempoRAI Patterns (TRAPs)

• Substitute a conventional spectral feature vector in phonetic classification by a 1 sec long temporal vector of critical band logarithmic spectral energies (Bark critical band)

• The TRAPS system consists of two stages of MLPs
  – In the first stage
critical band MLPs learn phone probabilities posterior on the input
  – In the second stage
A “merger” MLP merges the output of each of these individual critical band MLPs resulting in overall phone posteriors probabilities
Temporal Patterns (cont.)

- Input to each TRAP is a 1 sec long temporal vector
- Output of each TRAP is a vector of estimates of phoneme-specific likelihoods
- Output from the merging MLP is a vector of estimates of phoneme-specific posterior probabilities

![Diagram showing a critical-band spectrum and subsequent processing through TRAPs, MLPs, and PCA for HMM features.]
Principal Component Analysis

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.
PCA Applied in Inherently Robust Features

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

Original feature stream $\mathbf{x}_t$

The columns of $\mathbf{A}$ are the “first k” eigenvectors of $\Sigma_x$

Transformed feature stream $\mathbf{z}_t = \mathbf{A}^T \mathbf{x}_t$
• 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

\[
\begin{align*}
\mathbf{z}_k(n) &= [y_k(n) y_k(n+1) y_k(n+2) \ldots \ldots y_k(n+L-1)]^T \\
\mu_{z_k} &= \frac{1}{N-L+1} \sum_{n=1}^{N-L+1} \mathbf{z}_k(n) \\
\Sigma_{z_k} &= \frac{1}{N-L+1} \sum_{n=1}^{N-L+1} (\mathbf{z}_k(n) - \mu_{z_k}) (\mathbf{z}_k(n) - \mu_{z_k})^T \\
\end{align*}
\]

The element in the new feature vector

\[
\hat{x}(n,k) = e_k^T \mathbf{z}_k(n)
\]
PCA Applied in Inherently Robust Features (cont.)

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

\[ h_k \] is one of the eigenvectors of the covariance for \( x_k \)

---

Figure 1: The process of finding PCA-optimized filter bank coefficients
Most of the current approaches still have room for improvement