Rheinisch Westfälische Technische Hochschule Aachen
Lehrstuhl für Informatik VI
Prof. Dr.-Ing. Hermann Ney

Speech Recognition and Language Processing im SS 2003

Histogram Normalization

Haihua Luo

Supervisor: Florian Hilger

July 30, 2003

References

[1] S. Dharanipragada, M. Padmanabhan. A Nonlinear Unsupervised Adaptation Technique for Speech Recognition. In *Proc. Int. Conf. on Spoken Language Processing*, pages 556-559, Bejing, China, October 2000.

[2] F. Hilger, S. Molau, H. Ney. Quantile Based Histogram Equalization For Online Applications. In *Proc. Int. Conf. on Spoken Language Processing*, Vol. 1, pages 237-240, Denver, CO, USA, September 2002.

[3] S. Molau, M. Pitz, H. Ney. Histogram Based Normalization in the Acoustic Feature Space. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Vol. 1, pages 225-228, Madonna di Campiglio, Trento, Italy, December 2001.

[4] A. de la Torre, J.C. Segura, C. Benitez, A.M. Peinado and A.J. Rubio. Non-linear Transformations of the Feature Space for Robust Speech Recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 1, pages 401-404, Orlando, Florida, USA, May 2002.

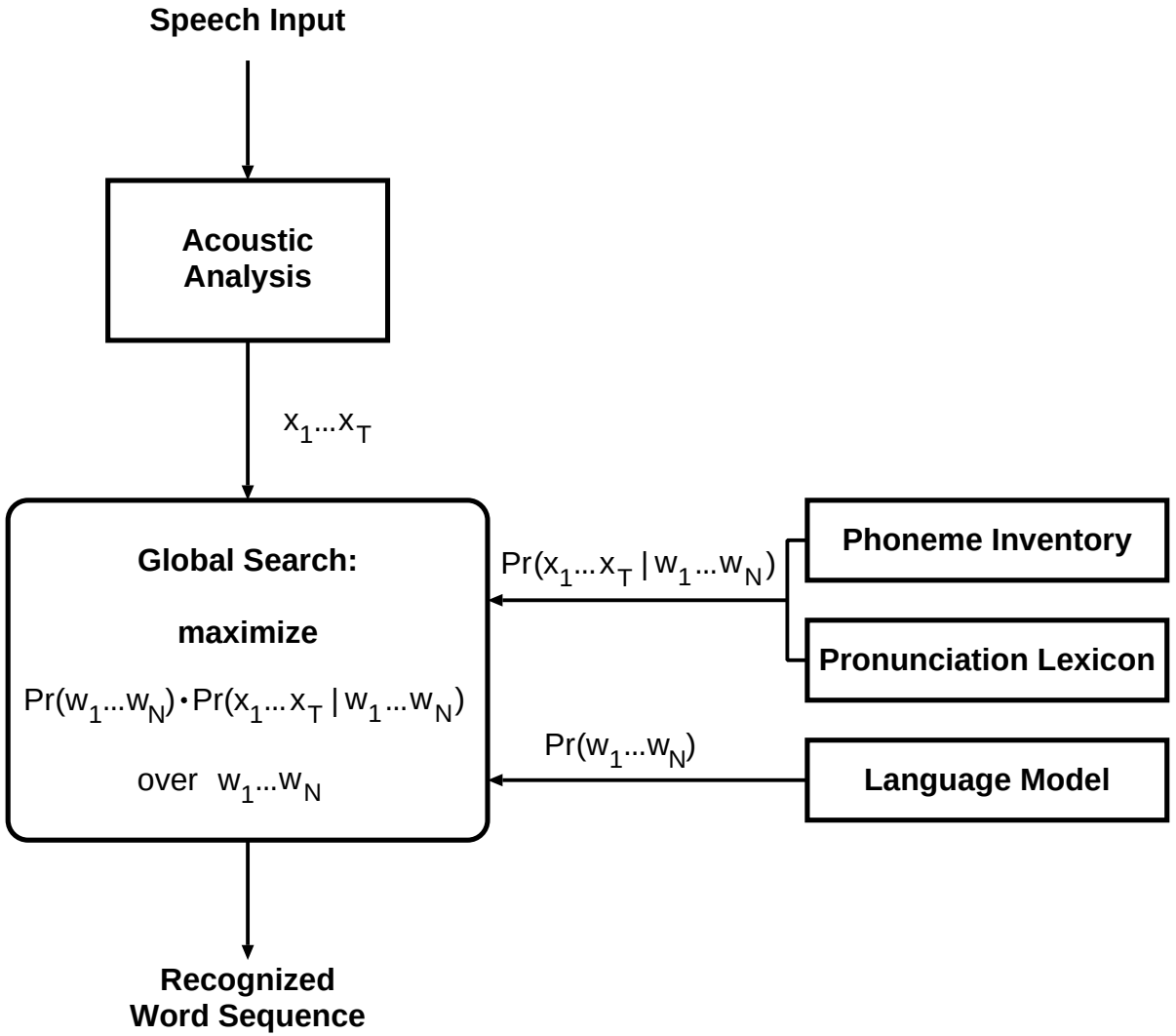
Overview

Overview:

- Introduction and Motivation
- Histogram Normalization
- Quantile Equalization
- Experiments and Recognition Results
- Conclusion

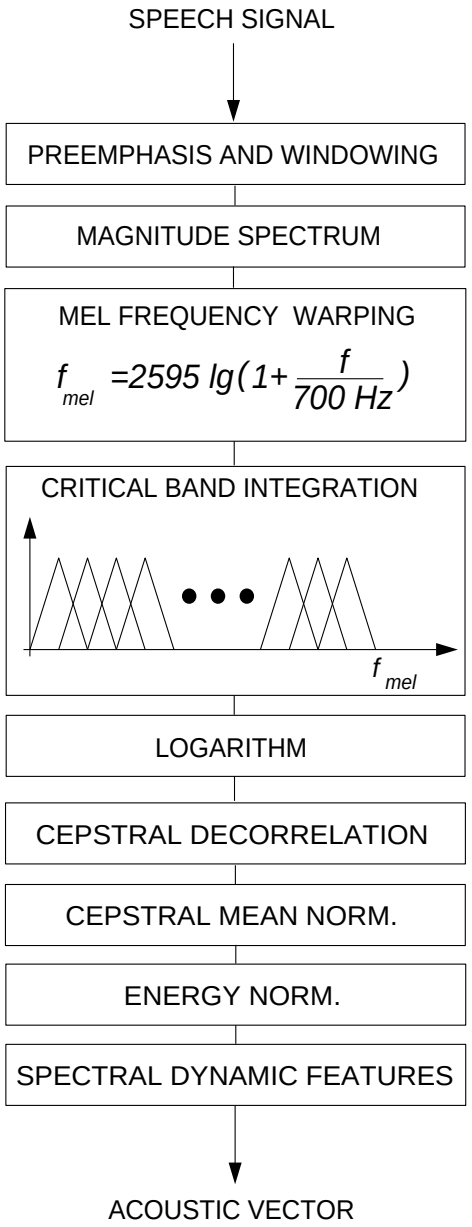
Speech Recognition Procedure:

- Acoustic Model
  - $x_1 \cdots x_T$: acoustic vectors
  - $P_r(x_1 \cdots x_T \mid w_1 \cdots w_N)$: probability of the acoustic vectors given a hypothesized word sequence $w_1 \cdots w_N$
- Language Model:
  - $P_r(w_1 \cdots w_N)$: probability of a word sequence
- Global Search (Bayes decision rule)



Introduction and Motivation

- **Windowing**: every 10ms-segment a 25ms wide Hamming window
- **Magnitude Spectrum**: Discrete Fourier Transformation or Fast Fourier Transformation
- **Mel Frequency Warping**: convert frequency to Mel frequency
- **Critical Band Integration**: sum magnitude of Fourier components over each critical band
- **Logarithm**
- **Cepstral Decorrelation**: Discrete Cosine Transformation
- **Cepstral Mean/Variance Normalization**: remove channel characteristics



Performance Evaluation:

1. Word Error Rate (WER):

- Portion of error words during the whole recognition:

$$WER(\%) = \frac{Sub + Del + Ins}{Total} \times 100\%$$

  - Sub, Del, Ins: number of word substitution, deletion, insertion errors
  - $Total$: total number of recognition words

- Widely used to evaluate the performance of speech recognition

2. Robustness:

- Preserve high recognition performance regardless of variations

Problems:

- Large variations of acoustic environment:
  background noise, distortion of transmission channels, speaking styles and accents of speakers
  $\Rightarrow$ Result in mismatch between recognition and training data in log-likelihood calculation:

$$-\log\ p(x \mid \mu, \sigma)$$

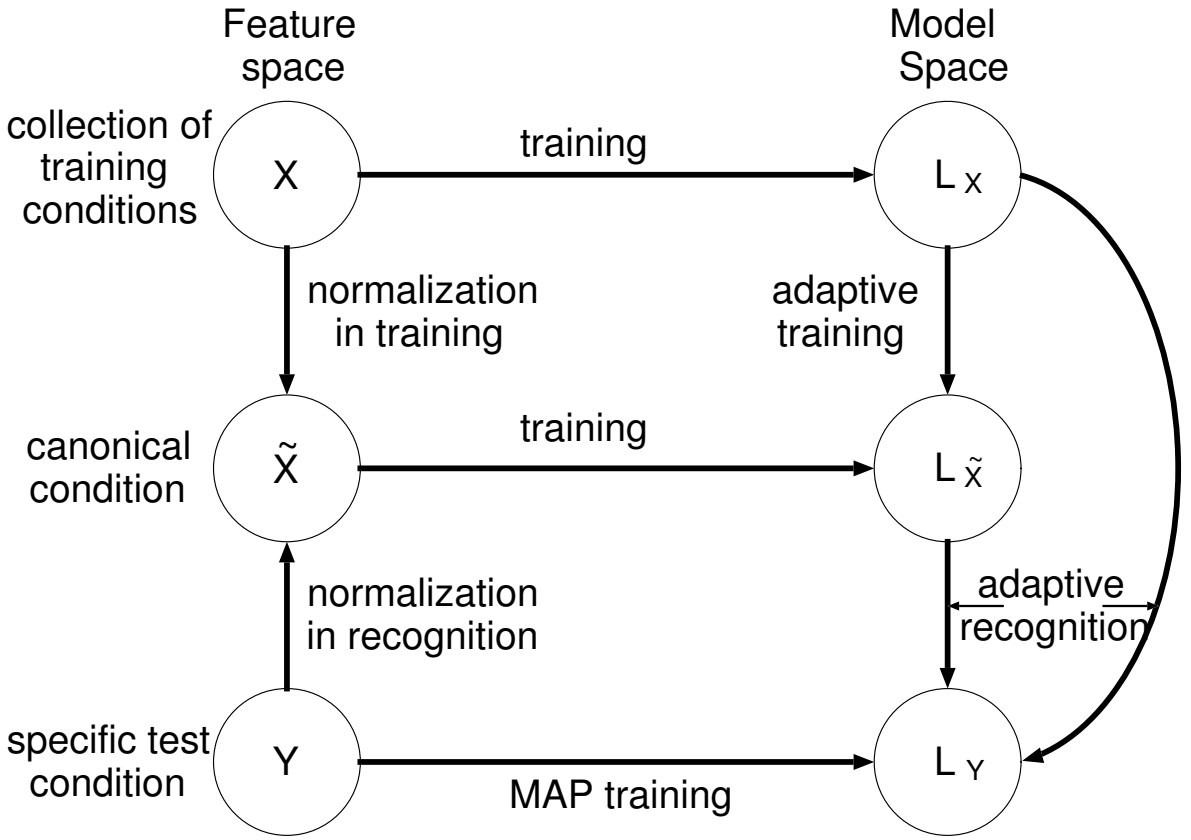
  - p: probability function
  - x: feature vector computed from speech signal to be recognized
  - μ, σ: references estimated on training data

Different approaches to reduce the mismatch:

- Normalize x to compensate variabilities during feature extraction
- Adapt references μ, σ to the noisy environment

Framework of Normalization and Adaptation approaches:

- Mismatch exists if the acoustic features Y and model L_X do not belong to the same level
- Normalization: transform the acoustic vectors Y to canonical form $\tilde{X}$
- Adaptation: adapt the acoustic model L_X to noisy environment L_Y



Examples for normalization methods:

- Cepstral Mean Normalization (CMN):
  subtract the mean of each cepstrum coefficient distorted by noise
- Cepstral Variance Normalization (CVN):
  normalize the variance of each cepstrum coefficient
- Histogram Normalization (HN):
  match the overall distribution of the training and test data in each feature space dimension
- Quantile Equalization (QE):
  apply Histogram Normalization for real-time recognition using quantiles

Adaptation methods:

- Maximum Likelihood Linear Regression (MLLR)
  (plann to be covered in another talk of the Seminar)

Histogram Normalization

History:

- Originally utilized in Digital Image Processing to adjust the brightness and constrastness.
- In 2000 Dharanipragada et al (IBM) applied HN into Automatic Speech Recognition.
- RWTH Informatik VI developed HN in consideration of silence fraction and quantile based online application.

Main idea:

- Transform the distribution of corrupted test utterance to that of the clean training data
- Use histogram to approximate the distribution
- Normalize test data distribution to training data distribution by the CDF matching

Computational Procedure:

1. Calculate the distribution of training data:

- Divide the collection of training data ranged from the minimum x_{min} to the maximum x_{max} uniformly into M non-overlapping bins:

$$x_{min} = b_1 < b_2 < ... < b_{M+1} = x_{max}$$

- Each bin $B_i = [b_i, b_{i+1})$
- Distribution histogram of training data:

$$p_X(x_t) = p_X(x_t \in B_i) = \frac{n_i}{N}$$

  - x_t: any value inside the collection of training data
  - n_i: number of values fall into B_i
  - N: total number of training data
  - $p_X(\ .\ )$: probability density function

Histogram Normalization

2. Compute the cumulative histogram of training data (target histogram):

- Approximate Cumulative Density Function (CDF) by cumulative histogram $P_X(x_t)$:

$$P_X(x_t) = P_X(x \leq x_t) = \sum_{j=1}^{i} \frac{n_j}{N}$$

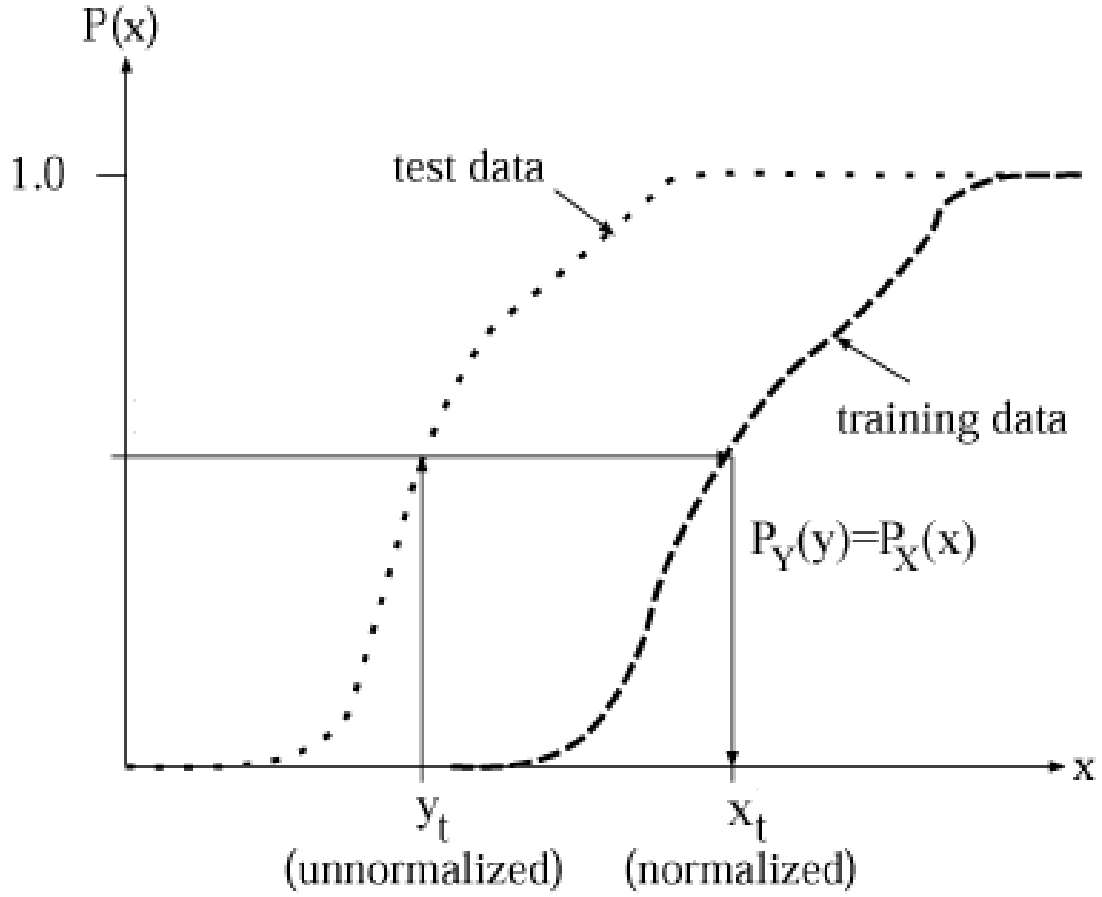
3. Calculate the cumulative histogram of recognition utterance $P_Y(y_t)$ similar as step 1. and 2.

4. Find out the transformation

- Record all pairs $(x_t, P_X(x_t))$ and $(y_t, P_Y(y_t))$ in two tables
- Obtain transformation $f : y_t \rightarrow x_t$ by combining two tables with condition $P_Y(y_t) = P_X(x_t)$

5. Normalize any noisy recognition feature y to a clean one:

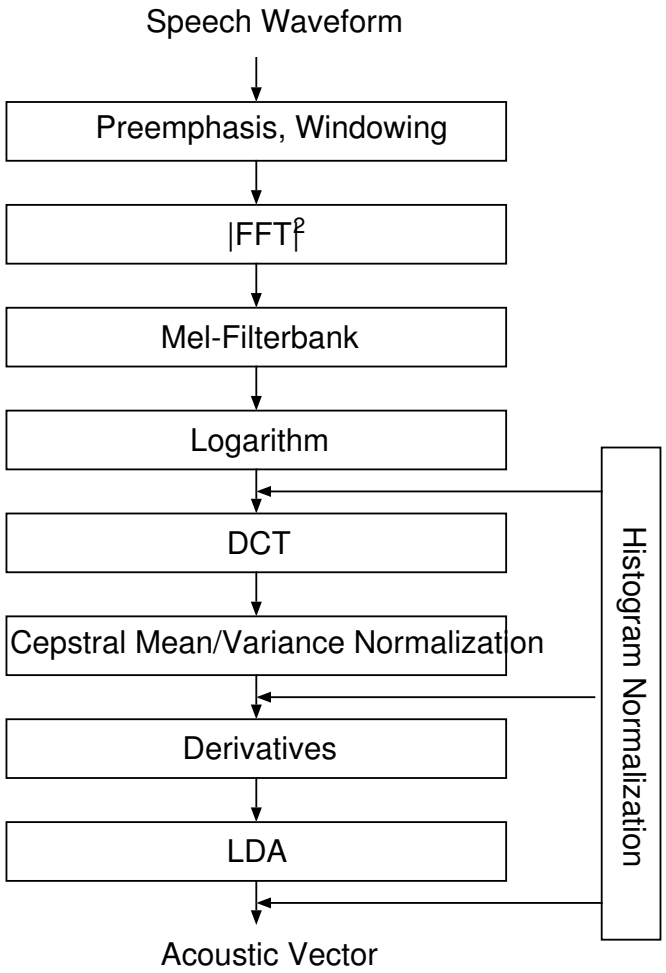
- Search the closest value to y in the set $\{y_t\}$
- Transform y to the corresponding clean value x_t using $f : y_t \rightarrow x_t$



Histogram Normalization

Applying HN at different stages of feature extraction is possible:

- Filterbank and Logarithm
- Cepstral Mean Normalization (CMN)
- Linear Discriminant Analysis (LDA)
- The best experimental result:
  Apply HN after Filterbank and Logarithm



Histogram Normalization

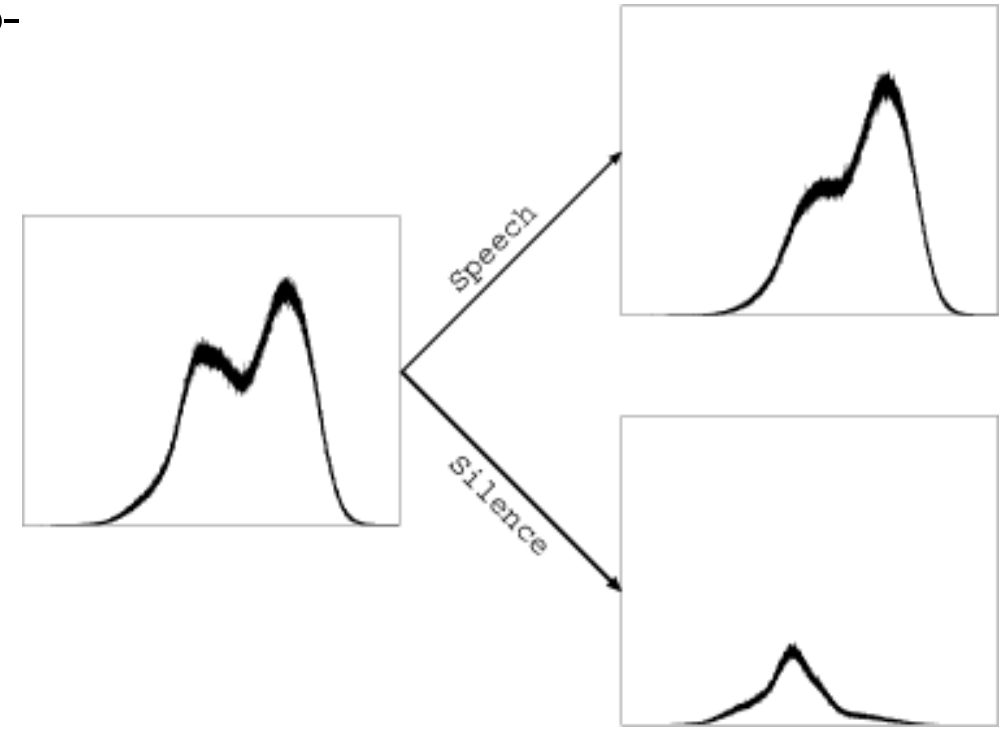
Silence Fraction Treatment:

- Compensate variations of silence fractions in the recordings
- Estimate two independent target histograms for silence and speech
- Compute an adapted target histogram:

$$P_\alpha = \alpha \cdot P_{sil} + (1 - \alpha) \cdot P_{speech}$$

  - P_α: adapted target histogram
  - α: silence fraction
  - P_{sil}: silence histogram
  - P_{speech}: speech histogram

- Use P_α as target histogram for the basic algorithm



Histogram Normalization

Limits of basic Histogram Normalization algorithm:

- Large amount of recognition data (usually several minutes) needed
- Computation Delay

$\Rightarrow$ Unsuitable for real-time (online) application

Solution:

- Approximate the CDF using a small number quantiles instead of the full histograms
- Utilize a parametric transformation function

Quantile Equalization

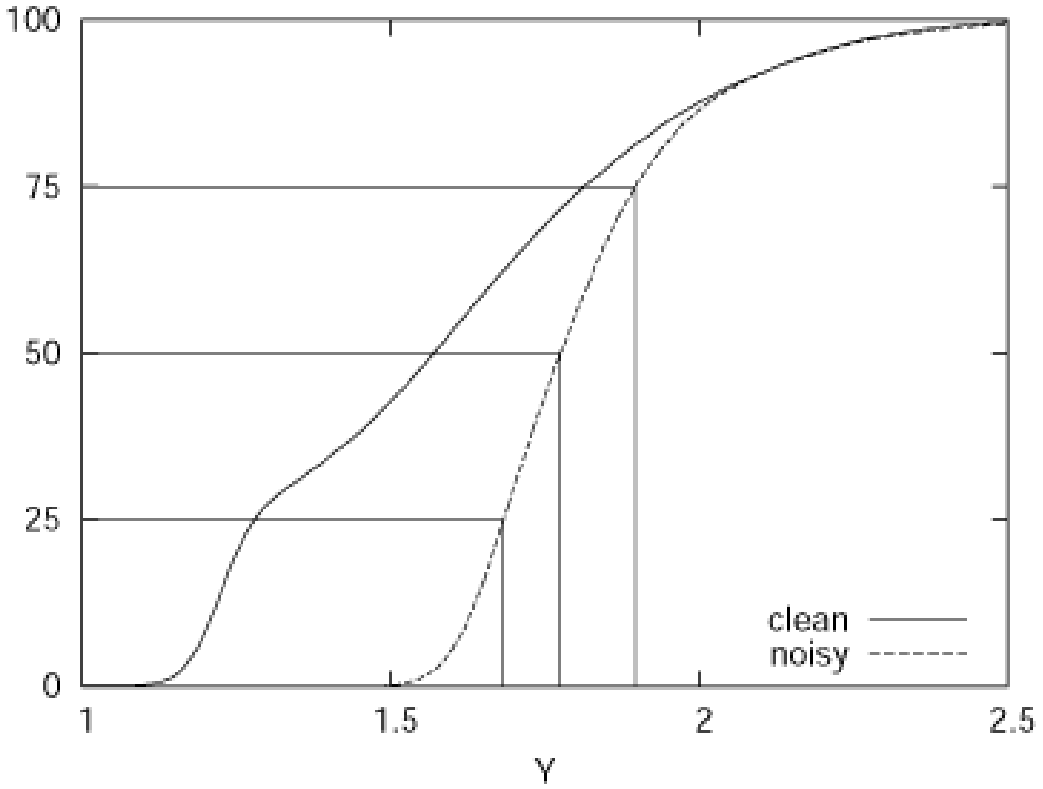
Main idea:

- Work also on CDF matching as Histogram Normalization
- Use a small number quantiles to estimate the CDF
- Utilize a parametric transformation function
- Apply the transformation function to match the training and recognition data

Quantile Equalization

Main computational procedure:

1. Find out a small number (typically four) of quantiles on training and test data:



Quantile Equalization

2. Power transformation function:

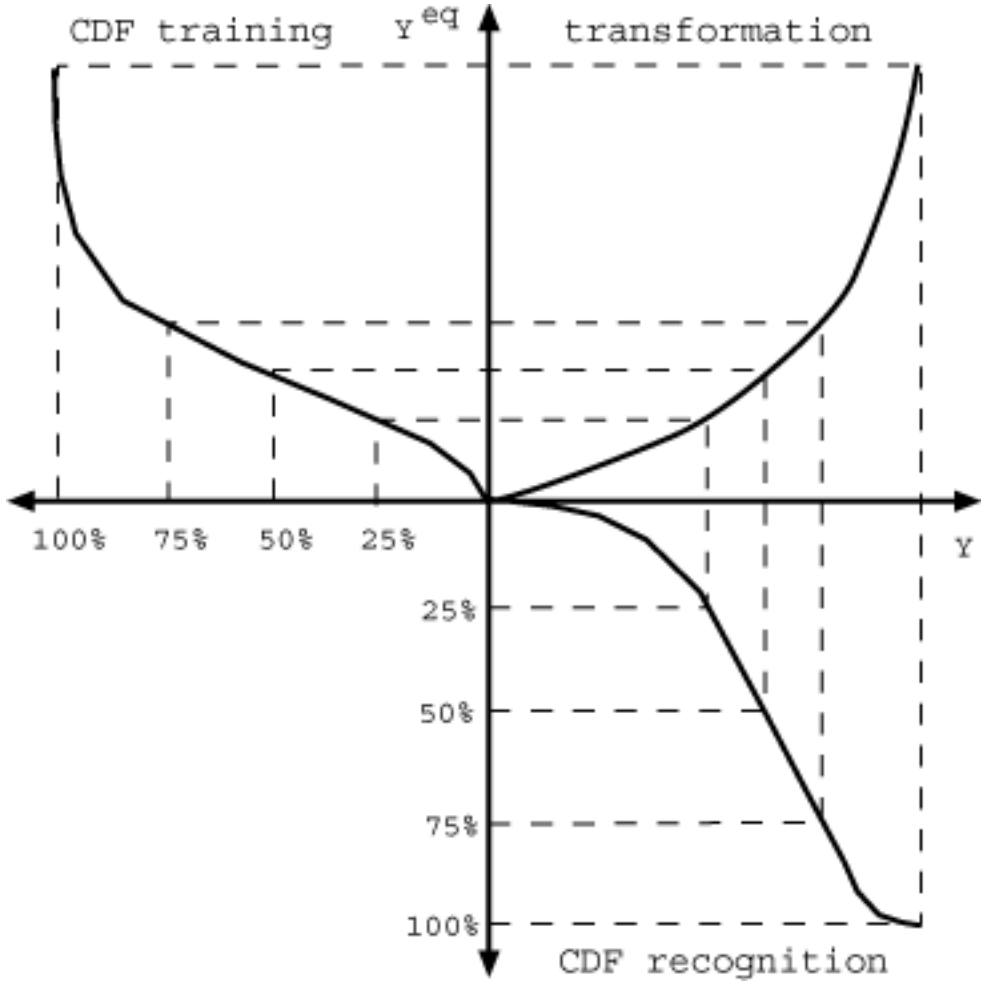
$$T_k(Y_k[t]) = Q_{k,N_Q}\left(\alpha_k\left(\frac{Y_k[t]}{Q_{k,N_Q}}\right)^{\gamma_k} + (1-\alpha_k)\frac{Y_k[t]}{Q_{k,N_Q}}\right)$$

- T_k^{eq}: equalized value of $Y_k[t]$
- T_k: transformation function
- $Y_k[t]$: output of the kth Mel scaled filter at time frame t
- N_Q: number of quantiles (typically 4)
- $Q_{k,i}$: the ith recognition quantile for filter channel k
- Q_i^{train}: the ith quantile on the training data
- Parameters γ_k, α_k:

$$\{\gamma_k, \alpha_k\} = \underset{\{\gamma_k,\alpha_k\}}{\text{argmin}}\left(\sum_{i=1}^{N_Q-1}(T_k(Q_{k,i},\gamma_k,\alpha_k) - Q_i^{train})^2\right)$$

Quantile Equalization

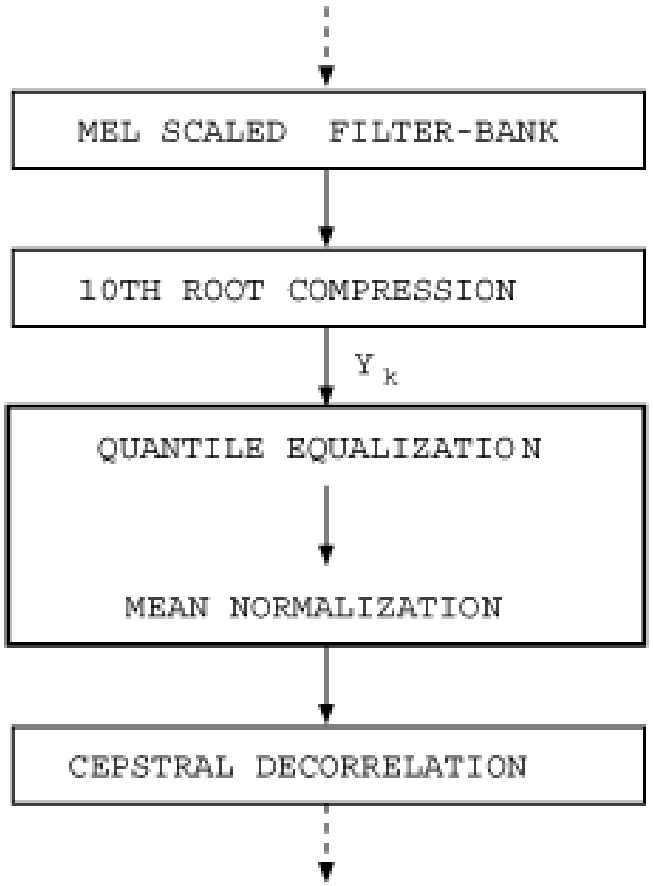
3. Utilize the transformation function to make the training and recognition data match:



Quantile Equalization

Implementation:

- Replace the usual Logarithm by the 10th root compression
- Apply QE after Mel scaled filtering
- Combine with Filterbank Mean Normalization (FMN)



Experiments and Recognition Results

Experiments:

1. Aurora

An experimental framework for the performance evaluation of feature extraction methods under noisy conditions

- Obtain comparable recognition results
- Predefined front-end and recognizer
- Defined databases:
  - Aurora-2: TI digit strings with noise added
  - Aurora-3: SpeechDat-Car database, recorded in cars

Experiments and Recognition Results

Recognizer setup:

- HTK speech recognition toolkit (fixed setup)
- MFCC feature extraction (Aurora WI007)
  23 filters;
  13 cepstral coeff. + 13 first deriv. + 13 second deriv.
  = 39 dimensional feature vector
- Word models of fixed length (16 states) for the digits
- Gender independent models
- Gaussians mixtures

Experiments and Recognition Results

Aurora 2 Database definition:

- Basis TI Digit strings
- Downsampled to 8kHz
- Filtered with G.712 and MIRS characteristics, which simulate a realistic frequency characteristic of a mobile phone
- Two training sets
  - Clean data (8440 utterances)
  - Multi condition data (8440 utterances)
    20 subsets (4 noises $\times$ 5 SNRs)
- Three test sets:
  - A: four noises: suburban train, babble, car, exhibition hall (28028 utterances)
    Same noises as used for multi condition training
  - B: four noises: restaurant, street, airport, train station (28028 utterances)
    Different noises as used for multi condition training
  - C: filtered with MIRS characteristic two noises: suburban train, street (14014 utterances)
  - Overall:
    $0.4 \times \text{setA} + 0.4 \times \text{setB} + 0.2 \times \text{setC}$

Experiments and Recognition Results

Recognition results for MFCC baseline and HN (by University of Granada, Spain):

Aurora-2 noisy TI digit strings					
		Set A	Set B	SetC	Overall
WER [%] for MFCC baseline	Multi	12.0	12.8	15.4	13.0
	Clean	41.3	46.6	34.0	41.9
	Average	26.6	29.7	24.7	**27.5**
WER [%] using HN	Multi	10.0	10.2	10.8	10.3
	Clean	19.7	18.6	19.6	19.2
	Average	14.9	14.4	15.2	**14.8**
Rel. Improv. [%]	Multi	17.8	24.2	33.3	23.5
	Clean	49.0	58.1	42.0	51.2
	Average	33.4	41.2	37.7	**37.4**

Experiments and Recognition Results

Recognition results for MFCC baseline und Quantile Equalization (by RWTH i6):

Aurora-2 noisy TI digit strings					
		Set A	Set B	SetC	Overall
WER [%] for MFCC baseline	Multi	12.0	12.8	15.4	13.0
	Clean	41.3	46.6	34.0	41.9
	Average	26.6	29.7	24.7	**27.5**
WER [%] using QE	Multi	10.2	10.8	10.8	10.5
	Clean	23.5	21.9	22.4	22.6
	Average	16.9	16.3	16.6	**16.6**
Rel. Improv. [%]	Multi	10.6	14.8	23.6	14.9
	Clean	43.4	59.9	42.0	49.7
	Average	27.0	37.4	32.8	**32.3**

2. Isolated Word Car Navigation Database:

Database description:

- a German isolated word database with a 2k-word closed vocabulary
- Record training data in a quiet office environment (SNR 21dB)
- The office test set: under the same conditions as training
- Two further test sets in driving cars: city (9dB) and highway traffic (6dB)
- Bandwidth: microphone

Statistics of the training and test corpora:

Isolated Word Car Navigation Database				
	Training	Test		
	Office	Office	City	High.
Duration [h]	18.8	1.7	1.7	1.8
Sil. Fraction [%]	60	69	73	75
Turn Duration [s]	785	425	450	468
# Speakers	86	14	14	14
# Words	61,742	2,069	2,100	2,100
Zerogram PP.	-	2,100	2,100	2,100

Experiments and Recognition Results

Recognizer setup:

- RWTH large vocabulary speech recognition system
- MFCC feature extraction (RWTH implementation)
  20 filters;
  16 cepstral coeff. + 16 first deriv. + 1 second deriv.
  = 33 dimensional feature vector
- Linear Discriminant Analysis (LDA)
- Triphone based modelling
- Classification and regression tree with 700 tied states
- 20k Gaussian densities
- 2,100 word vocabulary

Experiments and Recognition Results

Experiments on comparison with different normalization techniques (RWTH i6):

- Cepstral mean/variance normalization (CMN/CVN)
- Filterbank mean normalization (FMN)
- Quantile equalization (QE) in training and test (QETT)
- Histogram normalization (HN) with silence fraction treatment (HNSIL)
- Logarithm (log)
- 10th root compression (root)

Isolated Word Car Navigation Database				
#	Normalization	WER [%]		
		Office	City	High.
0	log, no normalization	2.8	68.0	99.0
1	log, CMN	2.9	31.6	74.2
2	log, CMN, CVN	4.2	20.8	39.7
3	root, FMN	2.8	19.9	40.1
4	root, FMN, QE	3.2	11.7	20.1
5	root, FMN, QETT	**3.3**	**10.1**	**16.7**
6	log, CMN, HN	2.8	10.2	16.6
7	log, CMN, HNSIL	**2.6**	**8.2**	**14.3**

Conclusion

Goal of Histogram Normalization:

- Reduce the mismatch caused by environment variations

Advantages of Histogram Normalization:

- Effective noise compensation:
  - No prior knowledge about the noise is needed
  - No assumptions about the noise (additive or convolutional noise) have to be made
- Low computational complexity
  - Approximate the data distributions of the recognition utterance and training data by histograms
  - Normalize noisy recognition data by simple table look-up

Conclusion

Further Improvements:

- Training data normalization
- Silence fraction treatment
- Quantile Equalization for online application
- Combination with other compensation techniques