SAMSUNG

# Voice recognition

## Before and after Deep Learning

Mikhail Kudinov

Samsung R&D Center Russia

April 1, 2019

# Table of Contents

Introduction
Feature extraction
"Classic" approaches
Deep learning approaches

# We are here!

**Introduction**

Feature extraction

"Classic" approaches

Deep learning approaches

# Speaker recognition

## Problems

- Speaker verification
- Speaker identification
- Speaker diarization

## Environment

- Text-dependent
- Text-independent

# We are here!

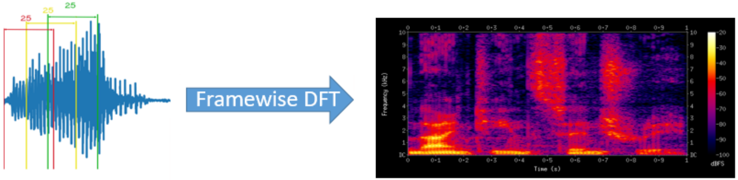

Introduction

**Feature extraction**

"Classic" approaches

Deep learning approaches

# Feature extraction: Short-time Fourier Transform

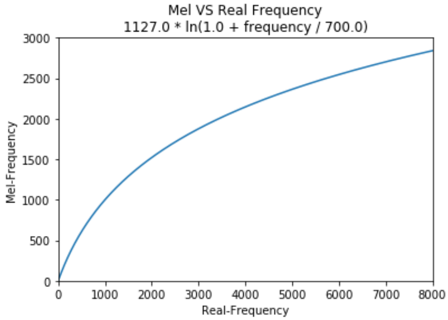



TL;DR: we use a 25ms sliding window with 10ms stride and apply FFT in each frame [1]

- Sound wave is preemphasized with a linear filter: $y[n] = x[n] - 0.95x[n-1]$ to boost higher frequencies
- Each frame is processed by a windowing function to prevent spectral distortion
- Each spectrum is post-filtered giving rise to different features extraction algorithms: LPC, MFSC, MFCC etc.

# Perceptially motivated features

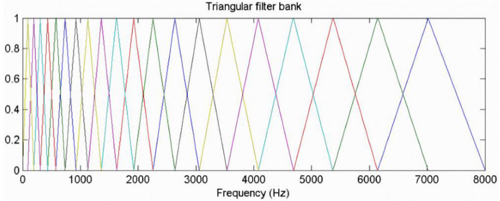
- Frequency resolution is not uniform for the human ear
- At low frequencies we hear small changes in frequency while at high-frequency range the audible difference is much higher
- Actually our frequency resolution mapping is logarithmic




Mel-frequency scale

# Perceptially motivated features: MFSC and MFCC

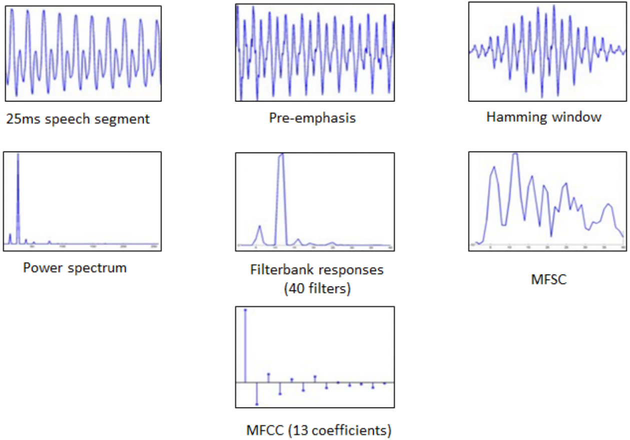
$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}$$




Mel filterbank formula (left) and graph (right)

- Filterbank with central frequencies and bandwidths set according to the mel-scale
- Mel-frequency spectral coefficients: logarithm of filter responses
- Mel-frequency cepstral coefficients: apply decorrelating transform (DCT or inv-DCT) to MFSC

# MFSC and MFCC: step-by-step




MFCC computation (left to right) [2]

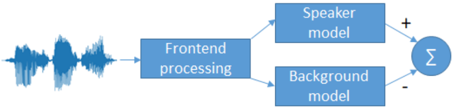
# We are here!

Introduction
Feature extraction

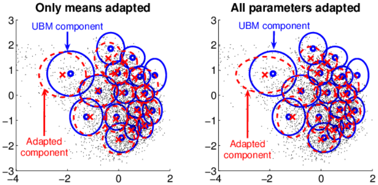
"Classic" approaches
Deep learning approaches

# Universal Background Model

- Criterion:
  $\Lambda(X) = \log p(X|\theta_s) - \log p(X|\theta_{bck}) \geq C$
- Train separate model $\theta_{bck}$ called Universal Background Model
- All frames are assumed to be independent
- $p(X_t|\theta)$ is a GMM with diagonal covariance matrices $\Sigma_i$
- $\theta_{bck}$ is trained with EM algorithm on the whole collection of speech data
- Speaker model $\theta_s$ is an MAP update of $\theta_{bck}$



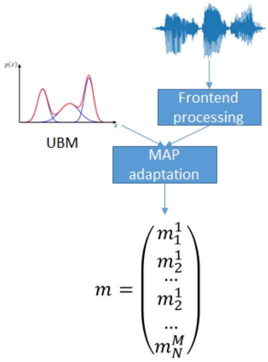

Likelihood ratio based system layout




MAP update of UBM [3]

# SVM on GMM supervectors

- Supervector $m$: do MAP-updates of means of the UBM and concatenate them
- Use lower-bound on KL-divergence as a distance measure:
  $d(m^a, m^b) = \frac{1}{2}\sum_i \pi_i (m_i^a - m_i^b)\Sigma_i^{-1}(m_i^a - m_i^b)$
- Use SVM on supervectors with the kernel defined above
- Nuisance attribute projection




Supervector computation

# Extensions of supervector model

- SVM with simple cosine kernel on preprocessed supervectors
- Within-class Covariance Normalization $k(m_1, m_2) = m_1 W^{-1} m_2$, where $W$ is a covariance matrix: $W = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{n_s}\sum_{i=1}^{n_s}(m_i^s - \overline{m_s})(m_i^s - \overline{m_s})^t$
- SVM on LDA-projected supervectors
- Nuisance Attribute Projection $\arg\min_P \sum_{i,j} W_{i,j}\|Pm_1 - Pm_2\|_2^2$, where $W_{i,j} = 0$ iff $w1$ and $w2$ are from e.g. different microphones (nuisance channel direction)
- Joint Factor Analysis
- Probabilistic LDA (PLDA)

# i-vectors

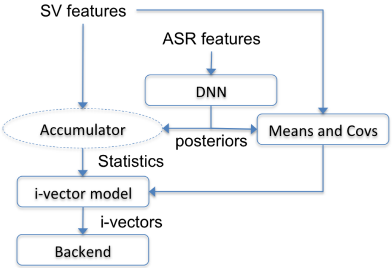
- Front-end Factor Analysis for a supervector $M$: $m = \mu + Tw$, where $\mu$ is the speaker- and channel-independent vector (UBM-vector)
- $T$ is a rectangular matrix of low rank
- $w \sim \mathcal{N}(0, \mathbb{I})$ is called *i-vector*
- i-vector estimation for known matrix $T$ for utterance $u$:
  - Use UBM $\Omega$ to get Viterbi or Baum-Welch estimations for GMM component $c$: $N_c = \sum_t P(c|y_t, \Omega)$; $F_c = \sum_t P(c|y_t, \Omega)(y_t - \mu_c)$
  - $w = (I + T^T \Sigma^{-1} N T)^{-1} T^T \Sigma^{-1} F$, where $N$ and $F$ are concatenations of $N_c$ and $F_c$ for all GMM components $\{c_i\}$
- $T$ is estimated using ML algorithm

# We are here!

Introduction
Feature extraction
"Classic" approaches
**Deep learning approaches**

# DNNs for better i-vector extraction

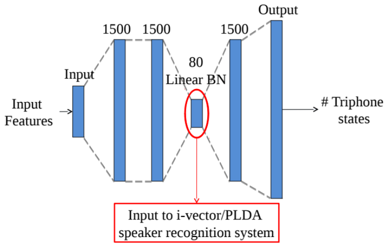
- Use DNN to obtain class-posterior for statistics collection *N* and *F* for i-vector calculation
- Classes are context-aware *senones*
- Requires ASR system trained separately
- Similarity score is computed on PLDA-projections




DNN/i-vector framework

# GMM-UBM on bottleneck features

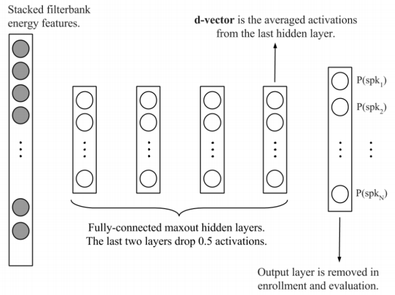
- Use DNN predicting senone classes to train a bottleneck mapping
- Use first layers of the DNN as a bottleneck features extractor
- Train traditional GMM-UBM/i-vector system on the bottleneck features




DNN for bottleneck features extraction [4]

# d-vectors

- Train a DNN to predict speaker label by a short speech segment
- Use last hidden layer as a feature vector
- At the enrollment stage average of last hidden layers for each speech frame is used as a speaker vector
- At the identification stage the decision is made according to a particular distance metric




DNN for d-vector extraction [5]

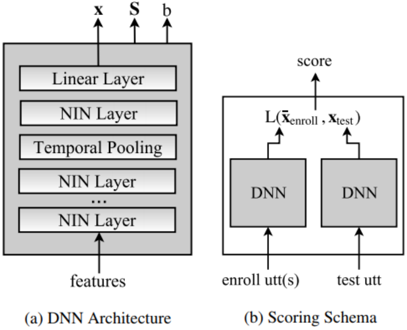
# Text-independent end-to-end speaker verification/x-vectors

- Add global average pooling to handle problem of different lengths
- Use objective similar to metric learning

$$P(x, y) = \frac{1}{1 + e^{-L(x,y)}}$$

$$L(x, y) = x^T y - xSx - ySy + b$$

$$E = - \sum_{(x,y) \in Same} \log P(x, y) - K \sum_{(x,y) \notin Same} \log(1 - P(x, y))$$




DNN scoring [6]

# THANK YOU

# References

## External resources


- [1] https://en.wikipedia.org/wiki/Spectrogram/media/File:Spectrogram-19thC.png
- [2] Bhiksha Raj and Rita Singh. Feature Computation: Representing the Speech Signal, http://www.cs.cmu.edu/afs/cs/user/bhiksha/WWW/courses/yahoo2009/01-02.featurecomputation.pdf
- [3] Jakub Galka. Voice Biometrics - how to recognize a speaker https://www.slideshare.net/TomaszZietek/voicepin-biometrics
- [4] Alicia Lozano-Diez, Anna Silnova et al. Analysis and Optimization of Bottleneck Features for Speaker Recognition, Odyssey 2016
- [5] Ehsan Variani, Xin Lei et al. DEEP NEURAL NETWORKS FOR SMALL FOOTPRINT TEXT-DEPENDENT SPEAKER VERIFICATION, ICASSP 2014
- [6] David Snyder, Daniel Garcia-Romero et al. X-VECTORS: ROBUST DNN EMBEDDINGS FOR SPEAKER RECOGNITION, ICASSP 2018