A Text-Independent Speaker Recognition System

Catie Schwartz

Advisor: Dr. Ramani Duraswaimi

Introduction

- Humans have the innate ability to recognize familiar voices within seconds of hearing a person speak.

- How to we teach a machine to do the same?

History of Speaker Recognition

- **Speaker Recognition** – the computational task of validating a user's identity based on their voice
- Research began in 1960 - models based on the analysis of x-rays(Biometrics.gov)

- Over the past 50 years, robust and highly accurate systems have been developed
- Applications include: *forensics*, *automatic password reset capabilities* and *home healthcare verification*

Biometrics.gov - Home. Web. 02 Oct. 2011. <http://www.biometrics.gov/>.

Example Application: Santrax® Telephony for Home Healthcare

- Used to ensure the right caregiver is serving the right patient at the right time
- Helps prevent fraudulent billing

"Sandata Technologies, LLC - Videos." *Sandata Technologies, LLC.* Web. 01 Oct. 2011. <http://www.sandata.com/about/videos.aspx?vid=speakerVerificationVideo>.

Example Application:
Santrax® Telephony for Home Healthcare

- Caregivers repeat the same phrase from enrollment to verify their identity (**text-dependent** system)
- System needs to be robust against **channel variability** (landline or mobile phone) and **speaker related variability** (health, mood, aging)

"Missouri estimates that they will achieve $8 million in projected saving in total funds over a 12-month period following the implementation of state-wide electronic verification systems for personal care"

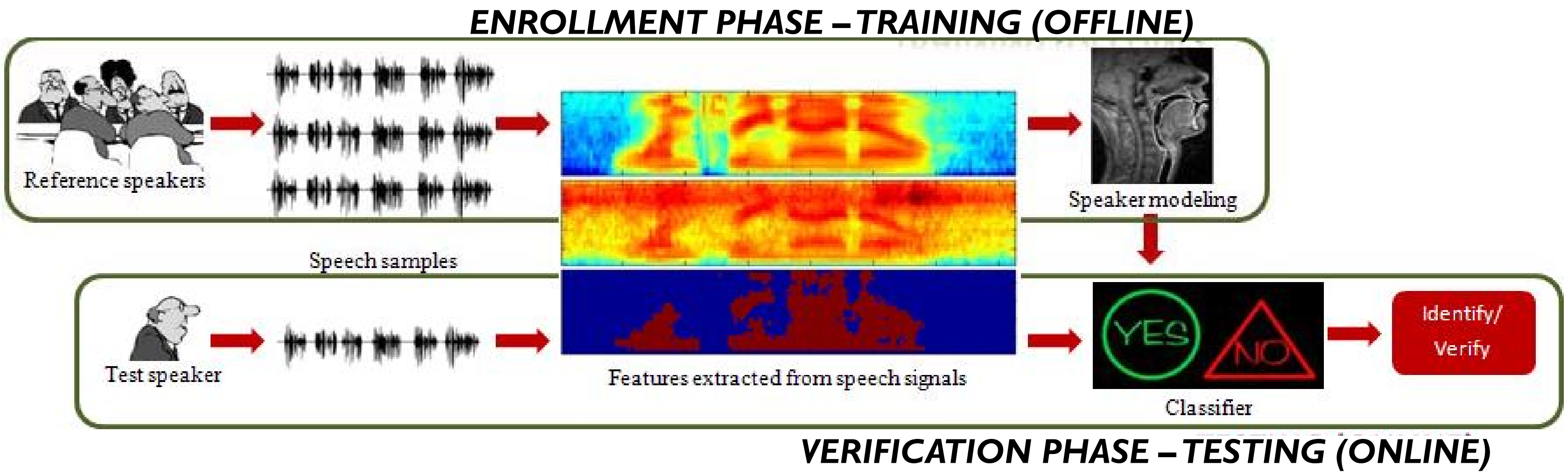
<http://www.marketwatch.com/story/enhanced-santrax-electronic-visit-verification-functionality-for-consumer-directed-services-programs-2011-09-15>.

Text-Dependent vs Text-Independent Speaker Recognition Systems

- **Text-Dependent** – Enrollment and verification phrases need to be identical
- **Text-Independent** – No requirement on the text used for the verification phase

- This project will focus on Text-Independent Speaker Verification

Speaker Recognition System



- All speaker recognition systems have an **enrollment phase** and a **verification phase**
- **Features** are extracted from speech samples, or utterances. Then **speaker models** are generated based on the features. **Classifiers** are used to determine if the test speaker is the same as the hypothesized model
- A **Universal Background Model (UBM)** or an average speaker model can be used in generating the speaker models
- Techniques can be used to minimize the affects of **channel variability** and **speaker related variability**

Project: *Implement a text-independent speaker recognition system*

- **Enrollment Phase:**
  - **Feature Extraction:**
    - Mel-frequency ceptral coefficients (MFCCs)
    - Energy based voice activity detector (VAD)
  - **Speaker Models:**
    - Gaussian Mixture Models (GMM) generated by adapting a UBM
    - Concatenate mean components of GMMs to create supervectors
    - Factor analysis (FA) techniques will be used to create i-vectors of low dimension to represent the speakers
    - Linear discriminant analysis (LDA) will be applied to i-vectors to compensate for intersession variability

- **Verification Phase:**
  - **Classifiers:**
    - Likelihood ratio test applied to GMM models
    - Cosine distance scoring applied to i-vectors and vectors from LDA

Feature Extraction

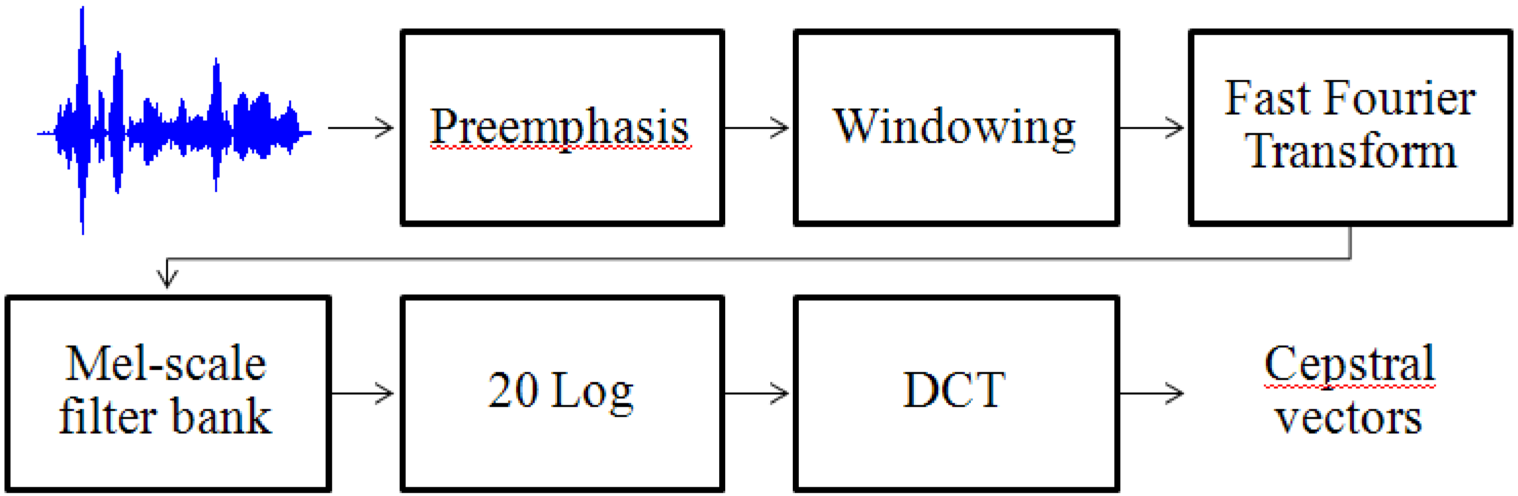
- Frames created using a 20 ms windowing processes with 10 ms overlap
- Low energy frames removed using a VAD algorithm
- MFCCs - relate to physiological aspects of speech
  - Mel-frequency scale – Humans differentiate sound best at low frequencies
  - Ceptral coefficients – Removes related timing information between different frequencies
  - Given an M-channel filterbank denoted by $Y(m), m = 1, \ldots, M$

  the MFCCs are founding using:

$$c_n = \sum_{m=1}^{M} [\log Y(m)] \cos\left[\frac{\pi n}{M}(m - \frac{1}{2})\right]$$

  were n is the index of the cepstral coefficient. The 19 lowest DCT coefficients will be used as the MFCCs.

Feature Extraction



MFCC extraction flow chart (courtesy of B. Srinivasan)

- Software written by D. Ellis will be used will be used to obtain MFCCs

@misc{Ellis05-rastamat, Author = {Daniel P. W. Ellis}, Year = {2005}, Title = {{PLP} and {RASTA} (and {MFCC}, and inversion) in {M}atlab}, Url = {http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/}, Note = {online web resource}}

π_k

Gaussian Mixture Models (GMM)

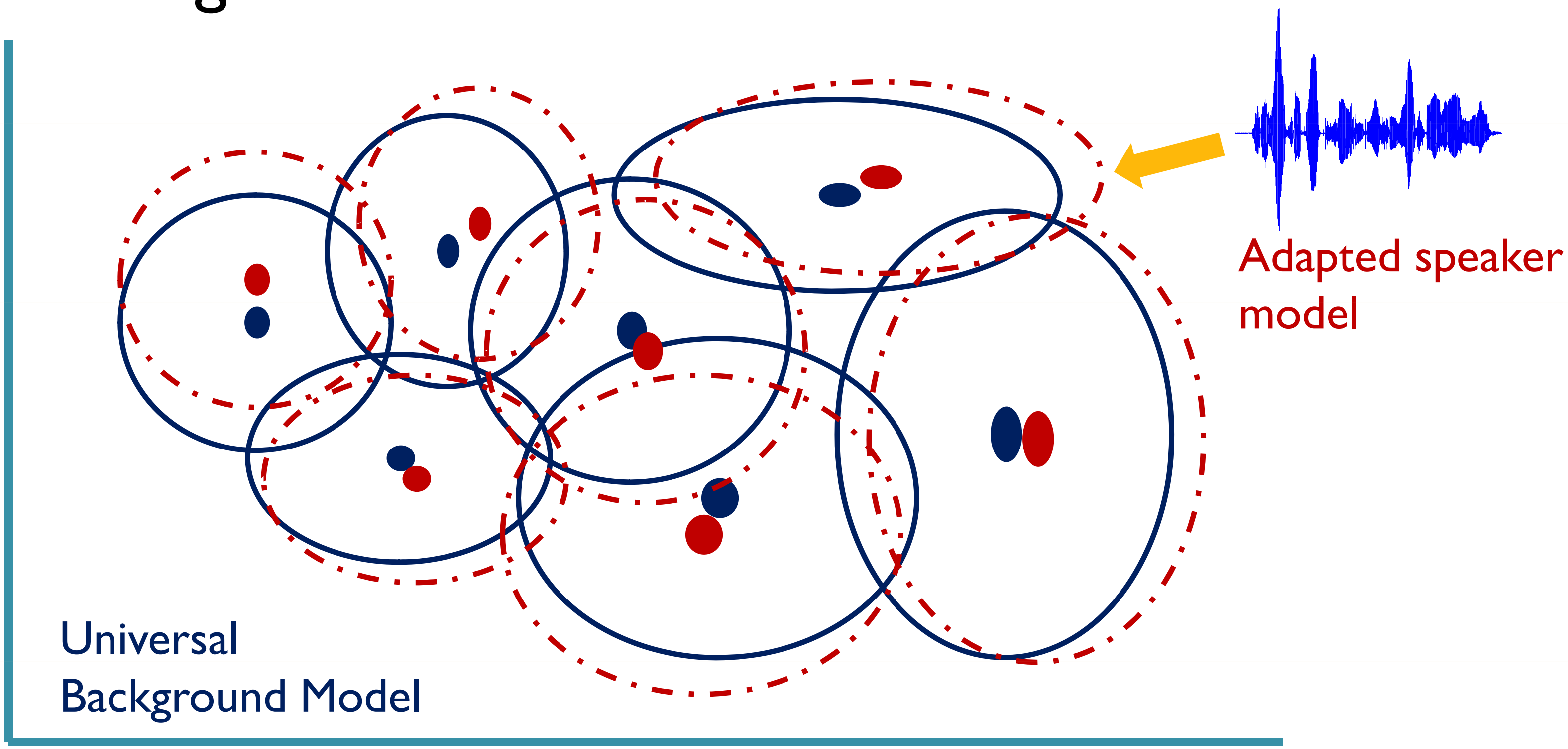
- Represent each speaker by a finite mixture of multivariate Gaussians

$$p(x|s_i) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)$$

where π_k, μ_k, Σ_k are learned using ML estimation techniques
- The UBM or average speaker model is trained using an expectation-maximization (EM) algorithm
- Speaker models learned using a maximum a posterior (MAP) algorithm

GMM - UBM adaptation (Reynolds et al., 2000)

- Adapt the UBM model to each speaker using the MAP algorithm



*D. Reynolds, T. Quatieri, and R. Dunn, "**Speaker verification using adapted Gaussian mixture models**," Digital Signal Processing, 10:19–41, 2000.*

Slide courtesy of B. Srinivasan

Expectation-maximization (EM)

- Iterative algorithm used to find GMM-UBM
- Expectation Step:
  - Conditional distribution of mixture component *c*

$$\gamma_t(c) = p(c|x_t, s) = \frac{\pi_c N(x|\mu_c, \Sigma_c)}{\sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)}$$

- Maximization Step:
  - Mixture Weights $\pi_c = \frac{1}{T}\sum_{t=1}^{T} \gamma_t(c)$
  - Means $\mu_c = \frac{\sum_{t=1}^{T} \gamma_t(c) x_t}{\sum_{t=1}^{T} \gamma_t(c)}$
  - Covariances $\sigma_c = \frac{\sum_{t=1}^{T} \gamma_t(c) x_t^2}{\sum_{t=1}^{T} \gamma_t(c)} - \mu_c^2$

Reynolds, Douglas A., and Richard C. Rose. "Robust Text-independent Speaker Identification Using Gaussian Mixture Speaker Models." *IEEE Transations on Speech and Audio Processing* IEEE 3.1 (1995): 72-83. Print.

Maximum a posteriori (MAP)

- Algorithm used to find parameters of the speaker models given UBM parameters π_c^{UBM}, μ_c^{UBM}
- First step is the same as EM – the values of π_c, μ_c are found using Bayesian statistics and ML estimations.
- Then, adapt old UBM parameters:

$$\hat{\pi}_c = [\alpha_c^w \pi_c + (1 - \alpha_c^w)\pi_c^{UBM}]\gamma$$

$$\hat{\mu}_c = \alpha_c^m \mu_c + (1 - \alpha_c^m)\mu_c^{UBM}$$

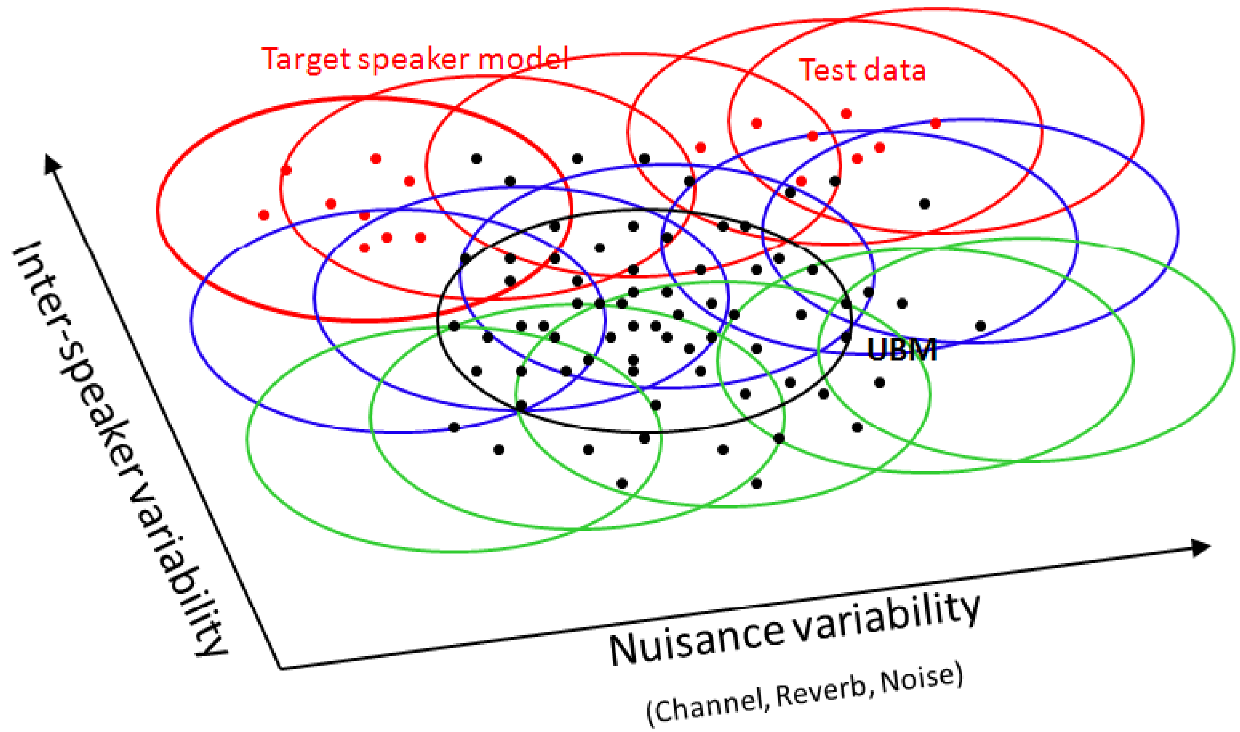
where

$$\alpha_c^\rho = \frac{\sum_{t=1}^{T} \gamma_t(c)}{\sum_{t=1}^{T} \gamma_t(c) + r^\rho}$$

for $\alpha_i^\rho, \rho \epsilon \{w, m\}$. Set $r^\rho = 16$

Reynolds, D. "Speaker Verification Using Adapted Gaussian Mixture Models." *Digital Signal Processing* 10.1-3 (2000): 19-41. Print.

How to account for variability?



- First, create supervectors from GMM model
- Then, find a space which inter-speaker variability is maximized and nuisance variability is minimized

Figure courtesy of B. Srinivasan

Adapted GMM ➔ Supervectors (Kinnunen et al., 2010)

- Supervector
  - High- and fixed-dimensional data
  - Has dimension $Kd \times 1$ where K is the number of Gaussian centers and d is the number of features

T. Kinnunen and H. Li, "**An overview of text-independent speaker recognition: From features to supervectors**," *Speech Communication*, 52:12–40, 2010.

Slide courtesy of B. Srinivasan

Factor Analysis

- **Factor Analysis** – a statistical method used to describe variability among observed variables in terms of potentially lower number of unobserved variables called factors
- Joint Factor Analysis (JFA) was the initial paradigm for speaker recognition:

$$\underset{\text{Speaker supervector}}{\mu_i} = \underset{\text{UBM}}{m} + V \overset{\text{Speaker factors}}{y_i} + U \underset{\text{Nuisance factors}}{x_i} + D \underset{\text{Residual factors}}{z_i}$$

JFA → Total variability (aka "i-vectors")

- Dehak et al. found that the subspaces U and V are not completely independent; therefore a combined "total variability" space that will be used in this project

Speaker factors

Speaker supervector

$$\mu_i = m + Vy_i + Ux_i + Dz_i$$

Residual factors

UBM

Nuisance factors

Speaker supervector

$$\mu_i = m + Tw_i$$

Intermediate / identity vectors (or) i-vectors

Total variability

UBM

Training total variability space

- The rank of T is set prior to training
- Concepts of Expectation-Maximization are used
- Training T similar to training V of the total variability matrix, except for training T all utterances from a given speaker are regarded as being produced by different speakers

Training total variability space

- Step 1: the Baum-Welsh statistics are calculated for a given speaker s and acoustic features $x_1, x_2, \ldots, x_T$ for each mixture component c:

0th order statistic $\longrightarrow$ $N_c(s) = \sum_{t=1}^{T} \gamma_t(c)$

1th order statistic $\longrightarrow$ $F_c(s) = \sum_{t=1}^{T} \gamma_t(c) x_t$

2th order statistic $\longrightarrow$ $S_c(s) = diag\left(\sum_{t=1}^{T} \gamma_t(c) x_t x_t^*\right)$

Kenny, Patrick, Pierre Ouellet, Najim Dehak, Vishwa Gupta, and Pierre Dumouchel. "A Study of Interspeaker Variability in Speaker Verification." *IEEE Transactions on Audio, Speech, and Language Processing* 16.5 (2008): 980-88. Print.

Training total variability space

- Step 2: Centralize 1st and 2nd order statistics

$$\tilde{F}_c(s) = F_c(s) - N_c(s)m_c$$

$$\tilde{S}_c(s) = S_c(s)(diag(F_c(s)m_c^* + m_c F_c(s)^* - N_c(s)m_c m_c^*)$$

Kenny, Patrick, Pierre Ouellet, Najim Dehak, Vishwa Gupta, and Pierre Dumouchel. "A Study of Interspeaker Variability in Speaker Verification." *IEEE Transactions on Audio, Speech, and Language Processing* 16.5 (2008): 980-88. Print.

Training total variability space

- Step 3: Define matrices and vectors based on Baum-Welsh statistics

$$NN(s) = \begin{bmatrix} N_1(s)I & & \\ & \ddots & \\ & & N_C(s)I \end{bmatrix}$$

$$SS(s) = \begin{bmatrix} \tilde{\tilde{S}}_1(s) & & \\ & \ddots & \\ & & \tilde{\tilde{S}}_c(s) \end{bmatrix}$$

$$F(s) = \begin{bmatrix} \tilde{\tilde{F}}_1(s) \\ \vdots \\ \tilde{\tilde{F}}_c(s) \end{bmatrix}$$

Lei, Howard. "Joint Factor Analysis (JFA) and i-vector Tutorial." *ICSI. Web. 02 Oct. 2011.*
http://www.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf

Training total variability space

- Step 4: Initial estimate of the speaker factors y

$$l_T(s) = I + T^{*}\Sigma^{-1}NN(s)T$$

- Posterior distribution of y(s) given data from speaker s is normally distributed with mean $l_T^{-1}(s)T^{*}\Sigma^{-1}\tilde{F}(s)$ and covariance matrix $l_T^{-1}(s)$

Lei, Howard. "Joint Factor Analysis (JFA) and i-vector Tutorial." *ICSI.Web. 02 Oct. 2011.*
http://www.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf

Training total variability space

- Step 5: Accumulate additional statistics across all the speakers

$$N_c = \sum_s N_c(s) \quad (c = 1, \ldots, C)$$

$$A_c = \sum_s N_c(s) l_T^{-1}(s) \quad (c = 1, \ldots, C)$$

$$\mathbb{C} = \sum_s FF(s) (l_T^{-1}(s) T^* \Sigma^{-1} FF(s))^* \quad (c = 1, \ldots, C)$$

Lei, Howard. "Joint Factor Analysis (JFA) and i-vector Tutorial." *ICSI.Web. 02 Oct. 2011.* http://www.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf

Training total variability space

- Step 6: Compute new estimate of T

$$T = \begin{bmatrix} T_1 \\ \vdots \\ T_C \end{bmatrix} = \begin{bmatrix} A_1^{-1}\mathbb{C}_1 \\ \vdots \\ A_1^{-1}\mathbb{C}_c \end{bmatrix}$$

where

$$\mathbb{C} = \begin{bmatrix} \mathbb{C}_1 \\ \vdots \\ \mathbb{C}_c \end{bmatrix}$$

Lei, Howard. "Joint Factor Analysis (JFA) and i-vector Tutorial." *ICSI.Web. 02 Oct. 2011.*
http://www.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf

Training total variability space

- Iterate through steps 4-6 approximately 20 times substituting new estimates of T into Step 4.
- Goal is for T*w*(s) to be as similar to FF(s) as possible
- Once the trained total variability space is obtained, can use knowledge that the expected value of an acoustic feature w(s) is $l_T^{-1}(s)T^*\Sigma^{-1}\tilde{F}(s)$

Lei, Howard. "Joint Factor Analysis (JFA) and i-vector Tutorial." *ICSI.Web. 02 Oct. 2011.*
http://www.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf

Linear Discriminant Analysis

- i-vectors from Factor analysis used in Linear discriminant analysis

$$\mu_i = m + Tw_i$$

Factor Analysis

$$w_i = Aw_i$$

Linear Discriminant Analysis

- Both methods used to reduce dimensionality

Linear Discriminant Analysis

$$\mathcal{w}_i = \boldsymbol{A}w_i$$

- Matrix A is chosen such that within-speaker (or speaker-dependent) variability is minimized and inter-speaker variability is maximized within the space
- Matrix A found by eigenvalue problem

$$J(A) = Tr\{s_W^{-1} s_B\}$$

Classifier: Log-likelihood test

- Once all GMM speaker models are test a sample speech to a hypothesized speaker

$$\Lambda(X) = \log p(X|s_{hyp}) - \log p(X|s_{UBM})$$

where $\Lambda(X) \geq \theta$ leads to verification of the hypothesized speaker and $\Lambda(X) < \theta$ leads to rejection.

Reynolds, D. "Speaker Verification Using Adapted Gaussian Mixture Models." *Digital Signal Processing* 10.1-3 (2000): 19-41. Print.

Classifier: Discrete cosine score

- The DCS can be applied to both the i-vectors and the intersession-compensated i-vectors using LDA

$$score(\omega_1, \omega_2) = \frac{\omega_1^t \omega_2}{\|\omega_1\| \|\omega_2\|} = \cos(\theta_{\omega_1, \omega_2})$$

where $score(\omega_1, \omega_2) \geq \varphi$ leads to verification of the hypothesized speaker and $score(\omega_1, \omega_2) < \varphi$ leads to rejection.

Implementation

- Completely implemented in Matlab using modern Dell desktop computer
- Software package that extracts MFCCs will be used
- Code will not be able to process large amounts of data which is typically necessary for a robust speaker recognition system. Numerical complexities and memory issues expected in this case

@misc{Ellis05-rastamat, Author = {Daniel P. W. Ellis}, Year = {2005}, Title = {{PLP} and {RASTA} (and {MFCC}, and inversion) in {M}atlab}, Url = {http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/}, Note = {online web resource}}

Databases

- The National Institute of Standards and Technology (NIST) has coordinated Speaker Recognition Evaluations (SRE) approximately every two years since 1996.
- Will use NIST 2008 SRE and NIST 2010 SRE Each contain speech data sampled at 8kHz in several different conditions including data from interviews, microphones, telephone conversations.
- The NIST 2010 SRE database contains around 12000 models.
- Files in *.wav or *.sph format

Validation and Test

Validation Metrics:

1. Equal error rates (ERR)
2. Detection error trade-off (DET) curves
3. MinDCF will be used to determine how well system is calibrated for a certain application

Validation will take place after GMM models created (using likelihood ratio test), after i-vector extraction (using DCS) and after LDA (using DCS). Results should improve at each step.

A variety of different tests using SRE database will be completed after first level validation completed on all code

Project Schedule (Fall 2011)

Phase I:	***~(5 weeks)***
Aug. 29 – Sept. 28 ~(4 weeks)	➢ Read a variety of Text-Independent Speaker Identification papers to obtain an understanding of the proposed project
Sept. 28 – Oct. 4 ~(1 week)	➢ Write proposal and prepare for class presentation
Phase II:	***~(4 weeks)***
Oct. 5 – Oct. 21 ~(2 weeks)	➢ Be able to extract MFCCs from speech data and apply simple VAD algorithm ➢ Understand SRE databases
Oct. 22 – Nov. 4 ~(2 weeks)	➢ Develop EM algorithm to trained UBM ➢ Add MAP algorithm to create speaker models ➢ Add likelihood ratio test as a classifier ➢ Validate results using likelihood ratio test as classifier with EER and DET curves, bug fix when necessary
Phase III:	***~(5 weeks)***
Nov. 5 – Dec. 2 ~(3 weeks + Thanksgiving Break)	➢ Create supervectors from GMMs ➢ Write code to train total variability space ➢ Add ability to extract i-vectors from the total variability space ➢ Add cosine distance scoring (CDS) as a classifier ➢ Validate results using the CDS classifier with EER and DET curves, bug fix when necessary
Dec. 3 – Dec. 9 ~(1 week) *overlap*	➢ Prepare Project Progress Report
Dec. 3 – Dec. 19 ~(2 week) *overlap*	➢ Implement LDA on the i-vectors ➢ Validate results using the CDS classifier with EER and DET curves, bug fix when necessary

Project Schedule (Spring 2012)

Phase IV:	***~(4 weeks)***
Jan. 25 – Feb. 24 ~(4 weeks)	➢ Obtain familiarity with vetted a speaker recognition system ➢ Test algorithms of Phase II and Phase III on several different conditions and compare against results of vetted system ➢ Bug fix when necessary
Phase V	***~(7 weeks)***
Feb. 25 – Mar. 2 ~(1 week) *overlap*	➢ Make Decision to either: (1) parallelize/optimize inefficient code, (2) Add more features, or (3) test in various conditions ➢ Read appropriate background material to make decision
Feb. 25 – Mar. 2 ~(1 week) *overlap*	➢ Work on Project Status Presentation
Mar. 3 – Apr. 20 ~(6 weeks + Spring Break)	➢ Update Schedule to reflect decision made in Phase IV ➢ Finish (1) or (2) in a 6 week time period including time for validation and test
Phase VI:	***~(3 weeks)***
Apr. 21 – May 10 ~(3 weeks)	➢ Create final report and prepare for final presentation

Milestones

Fall 2011

October 4
- Have a good general understanding on the full project and have proposal completed. Present proposal in class by this date.
  Marks completion of Phase I

November 4
- Validation of system based on supervectors generated by the EM and MAP algorithms
  Marks completion of Phase II

December 19
- Validation of system based on extracted i-vectors
- Validation of system based on nuisance-compensated i-vectors from LDA
- Mid-Year Project Progress Report completed. Present in class by this date.
  Marks completion of Phase III

Spring 2012

Feb. 25
- Testing algorithms from Phase II and Phase III will be completed and compared against results of vetted system. Will be familiar with vetted Speaker Recognition System by this time.
  Marks completion of Phase IV

March 18
- Decision made on next step in project. Schedule updated and present status update in class by this date.

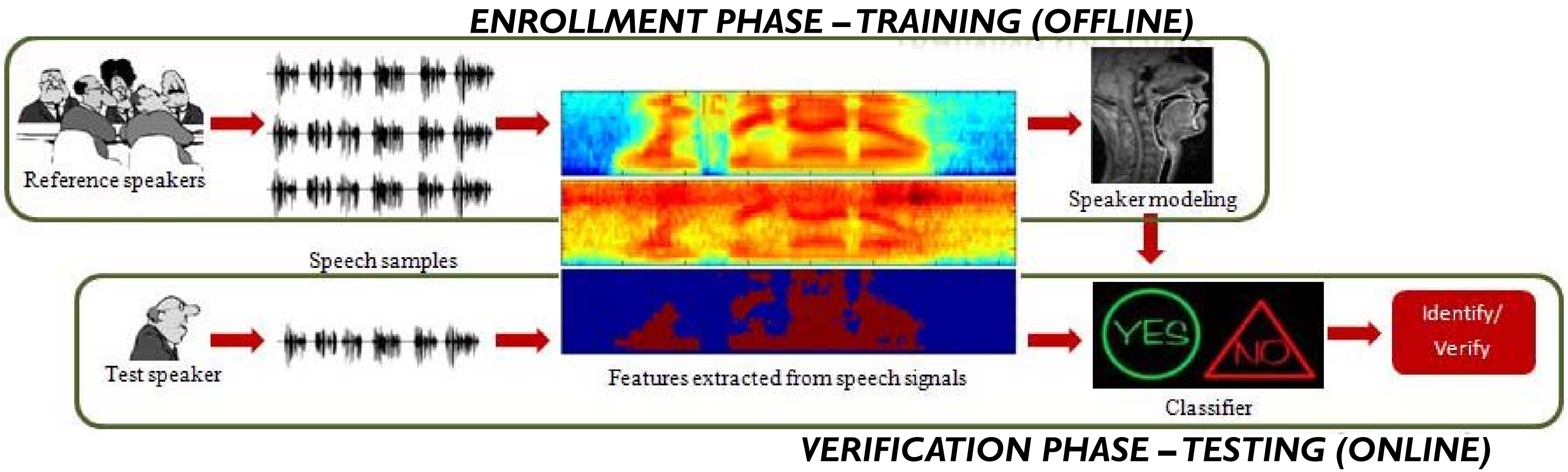
April 20
- Completion of all tasks for project.
  Marks completion of Phase V

May 10
- Final Report completed. Present in class by this date.
  Marks completion of Phase VI

Deliverables

- A fully validated and complete Matlab implementation of a speaker recognition system will be delivered with at least two classification algorithms.
- Both a mid-year progress report and a final report will be delivered which will include validation and test results.

Questions?


ENROLLMENT PHASE – TRAINING (OFFLINE)
Reference speakers
Speech samples
Speaker modeling
Test speaker
Features extracted from speech signals
YES
NO
Classifier
Identify/
Verify
VERIFICATION PHASE – TESTING (ONLINE)

Bibliography

- [1]*Biometrics.gov - Home.* Web. 02 Oct. 2011. <http://www.biometrics.gov/>.
- [2] Kinnunen, Tomi, and Haizhou Li. "An Overview of Text-independent Speaker Recognition: From Features to Supervectors." *Speech Communication* 52.1 (2010): 12-40. Print.
- [3] Ellis, Daniel. "An introduction to signal processing for speech." *The Handbook of Phonetic Science,* ed. Hardcastle and Laver, 2nd ed., 2009.
- [4] Reynolds, D. "Speaker Verification Using Adapted Gaussian Mixture Models." *Digital Signal Processing* 10.1-3 (2000): 19-41. Print.
- [5] Reynolds, Douglas A., and Richard C. Rose. "Robust Text-independent Speaker Identification Using Gaussian Mixture Speaker Models." *IEEE Transations on Speech and Audio Processing* IEEE 3.1 (1995): 72-83. Print.
- [6] "Factor Analysis." *Wikipedia, the Free Encyclopedia.* Web. 03 Oct. 2011. <http://en.wikipedia.org/wiki/Factor_analysis>.
- [7] Dehak, Najim, and Dehak, Reda. "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification." *Interspeech 2009 Brighton.* 1559-1562.
- [8] Kenny, Patrick, Pierre Ouellet, Najim Dehak, Vishwa Gupta, and Pierre Dumouchel. "A Study of Interspeaker Variability in Speaker Verification." *IEEE Transactions on Audio, Speech, and Language Processing* 16.5 (2008): 980-88. Print.
- [9] Lei, Howard. "Joint Factor Analysis (JFA) and i-vector Tutorial." *ICSI. Web. 02 Oct. 2011.* http://www.icsi.berkeley.edu/Speech/presentations/AFRL_ICSI_visit2_JFA_tutorial_icsitalk.pdf
- [10] Kenny, P., G. Boulianne, and P. Dumouchel. "Eigenvoice Modeling with Sparse Training Data." *IEEE Transactions on Speech and Audio Processing* 13.3 (2005): 345-54. Print.
- [11] Bishop, Christopher M. "4.1.6 Fisher's Discriminant for Multiple Classes." *Pattern Recognition and Machine Learning*. New York: Springer, 2006. Print.
- [12] Ellis, Daniel P.W. *PLP and RASTA (and MFCC, and Inversion) in Matlab. PLP and RASTA (and MFCC, and Inversion) in Matlab.* Vers. Ellis05-rastamat. 2005. Web. 1 Oct. 2011. <http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/>.

VAD energy based algorithm [Kinnunen]

```
E = 20*log10(std(Frames')+eps); % Energies
Max1 = max(E);
I = (E>max1-30) & (E>-55);  % Indicator
```

Detection threshold is 30dB below maximum and -55dB in absolute energy

GMMs

- A GMM is the composition of a finite mixture of multivariate Gaussian components and is characterized by its probability density function (PDF):

$$p(x|s_i) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)$$

- where represents the speaker of interest, *K* is the number of Gaussian components, is the prior probability of the Gaussian component and

$$N(x|\mu_k, \Sigma_k) = (2\pi)^{-\frac{d}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-\frac{1}{2}} (x-\mu_k)\}$$

- is the d-variate Gaussian density function with mean and covariance . Note that the prior probabilities are constrained as