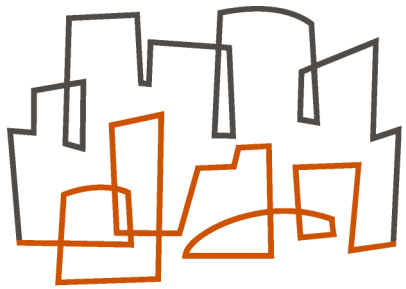


Data-Driven Factor Analysis for Characterizing Spoken Audio

Najim Dehak and Stephen Shum

Spoken Language Systems Group
MIT Computer Science and Artificial Intelligence Laboratory

Goals

- **Aim**
  - To provide an overview of the theory and operation of a low-dimensional representation of speech and its application to automatic speaker recognition, language identification, and audio diarization.
  - To demonstrate the plausibility of this approach to any sequential data classification problem

- **Participants should gain an introduction to and understanding of:**
  - *Subspace Representation of Speech Signals or Sequential Data*
  - *Algorithms for Joint-Factor Analysis and Total-Variability Modeling to handle the variabilities in the data.*
  - *Application of subspace representations to automatic speaker and language recognition systems*

Roadmap

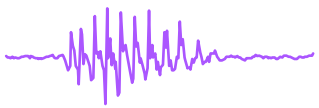
- **Introduction**
  - Terminology, tasks, and framework

- **Low-Dimensional Representation**
  - Speaker Verification
  - Sequence of features: GMM
  - Super-vectors: JFA
  - Low-dimensional vectors: i-vectors
  - Processing i-vectors: compensation and scoring
  - Data visualization

- **Other Applications**
  - Speaker diarization
  - Language recognition

Extracting Information from Speech

Goal: Automatically extract information transmitted in speech signal



Speech Signal

Speech Recognition → Words
"How are you?"

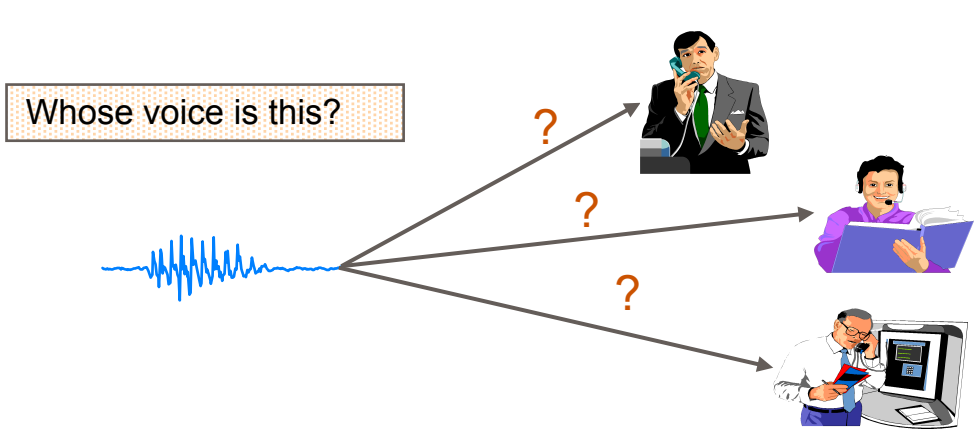
Language Recognition → Language Name
English

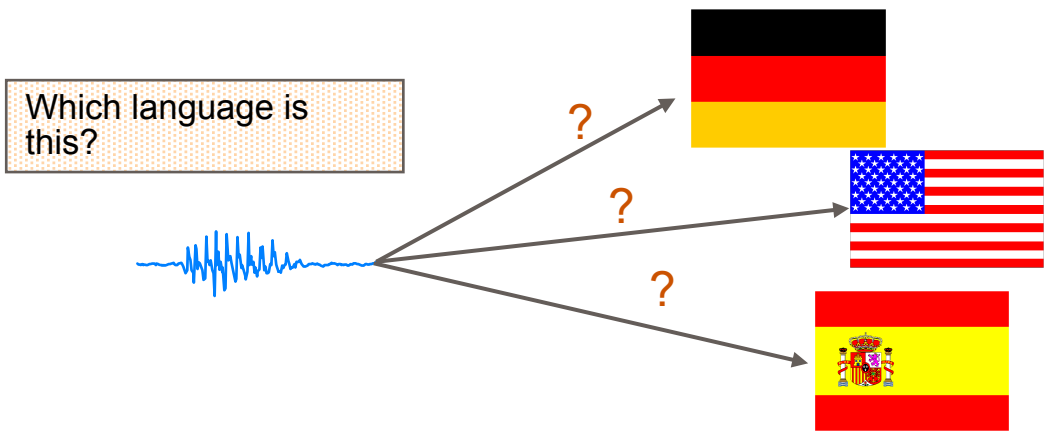
Speaker Recognition → Speaker Name
James Wilson

Emotion Recognition → Emotion category
Happy, angry …

Identification

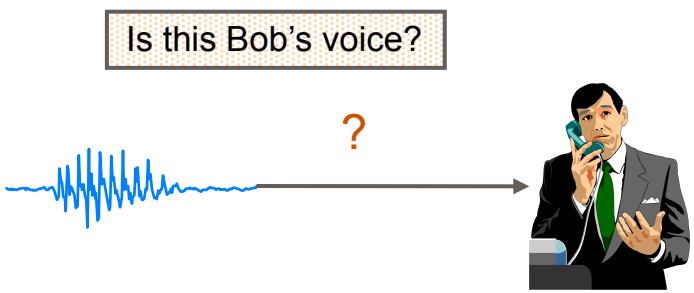
- Determine whether unknown speaker (language) matches one of a set of known speakers (languages)
- One-to-many mapping
- Often assumed that unknown voice must come from a set of known speakers – referred to as **closed-set** identification

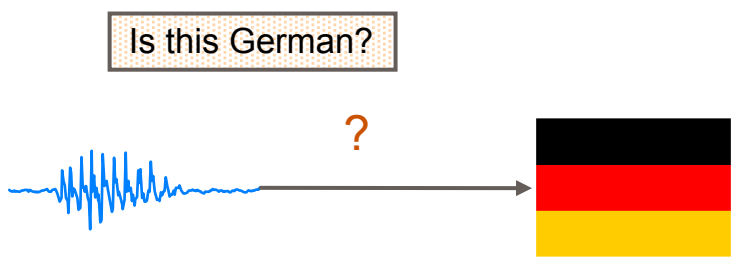




Verification/Authentication/Detection

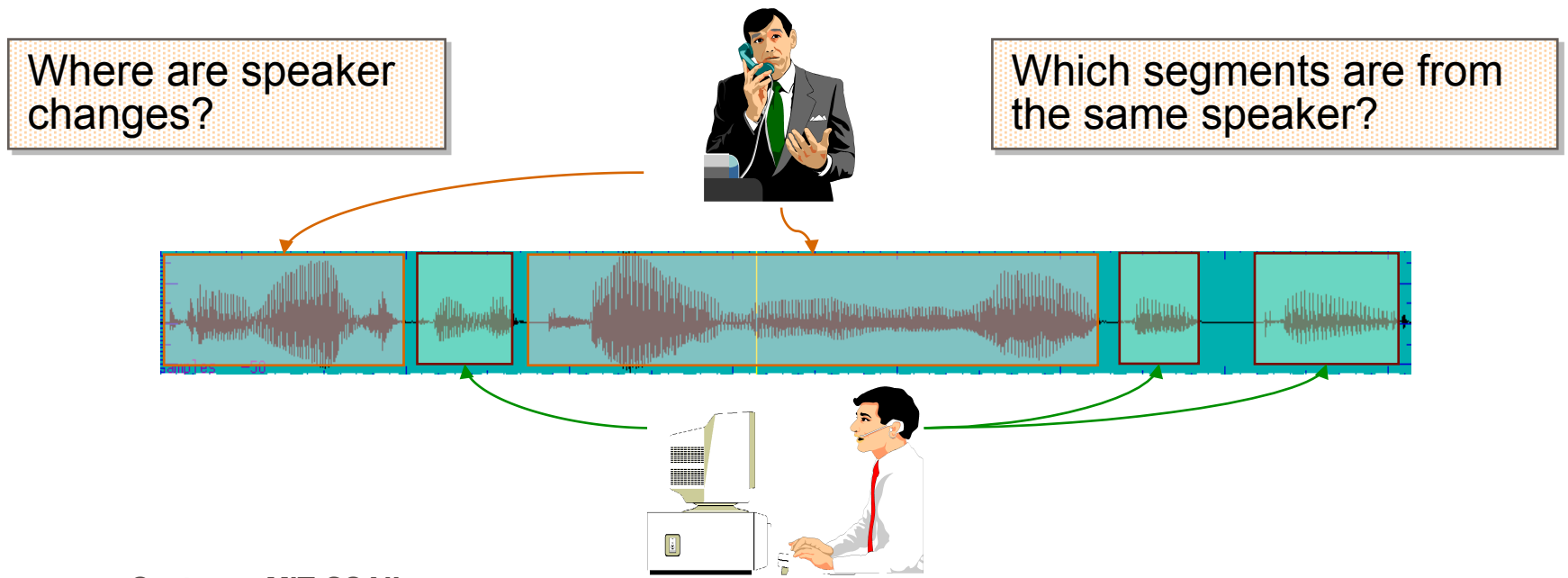
- Determine whether unknown speaker (language) matches a specific speaker (language)
- One-to-one mapping
- Unknown speech could come from a large set of unknown speakers (languages) – referred to as **open-set** verification
- Adding "none of the above" option to closed-set identification gives open-set identification





Diarization
Segmentation and Clustering

- **Determine when a speaker change has occurred in the speech signal (segmentation)**
- **Group together speech segments corresponding to the same speaker (clustering)**
- **Prior speaker information may or may not be available**



Speech Modalities

Application dictates different speech modalities:

Text-dependent	Text-independent

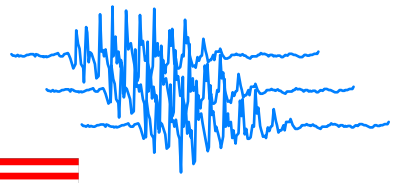
- Recognition system knows text spoken by person
- Examples: fixed phrase, prompted phrase
- Used for applications with strong control over user input
- Knowledge of spoken text can improve system performance

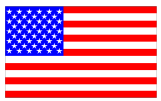
- Recognition system does not know text spoken by person
- Examples: User selected phrase, conversational speech
- Used for applications with less control over user input
- More flexible system but also more difficult problem
- Speech recognition can provide knowledge of spoken text

Framework for Speaker/Language Recognition Systems

Training Phase

Known train

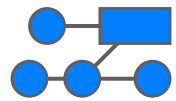




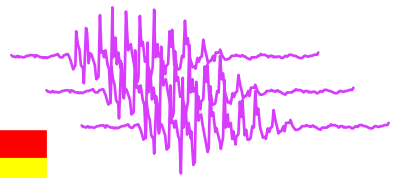
Feature extraction

Training algorithm

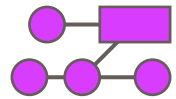
Model for each speaker (language)



Bob (English)



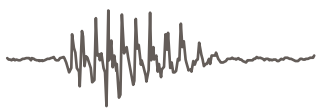
Algorithm parameters



Sally (Spanish)

Recognition Phase

?



Feature extraction

Recognition algorithm

Decision

Unknown test

Speaker/language set

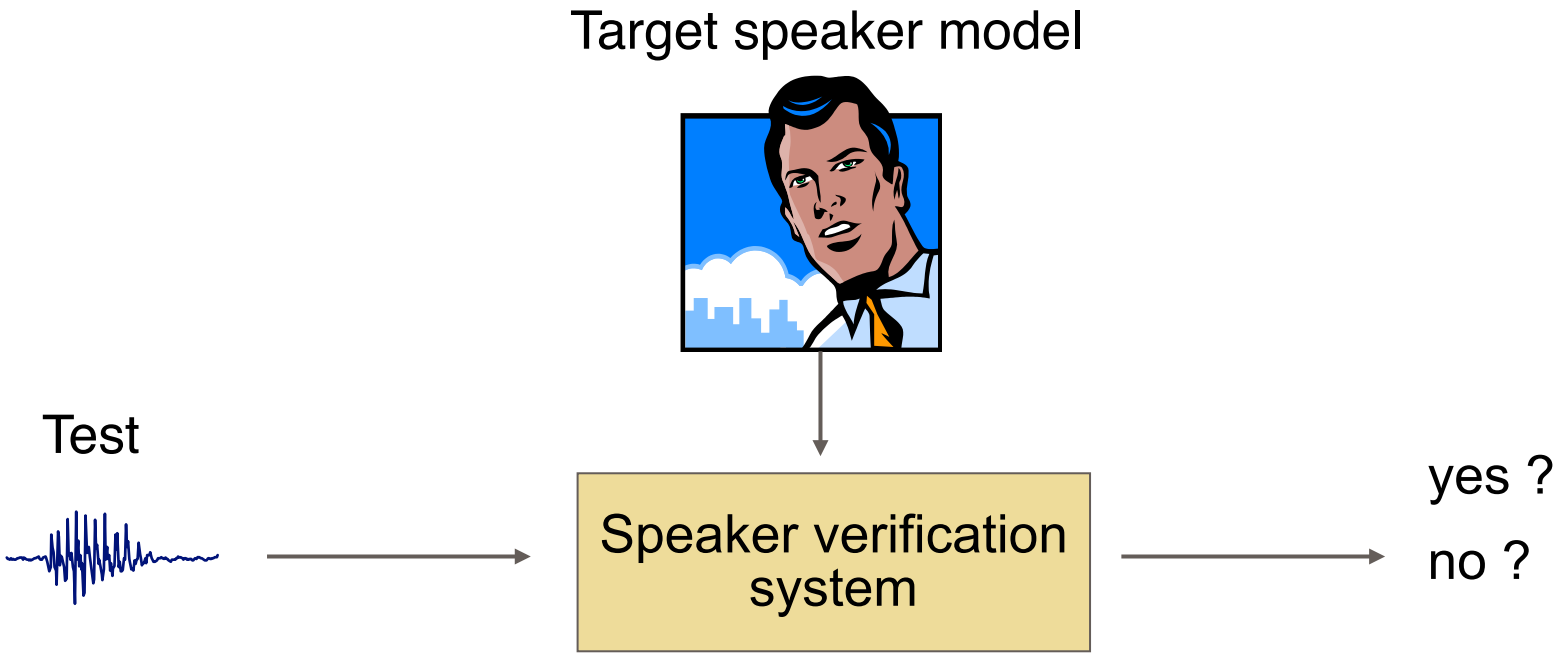
Roadmap

- Introduction
  - Terminology, tasks, and framework

- **Low-Dimensional Representation**
  - Speaker Verification
  - Sequence of features: GMM
  - Super-vectors: JFA
  - Low-dimensional vectors: i-vectors
  - Processing i-vectors: compensation and scoring
  - Data visualization

- Other Applications
  - Speaker diarization
  - Language recognition

Speaker Verification Problem

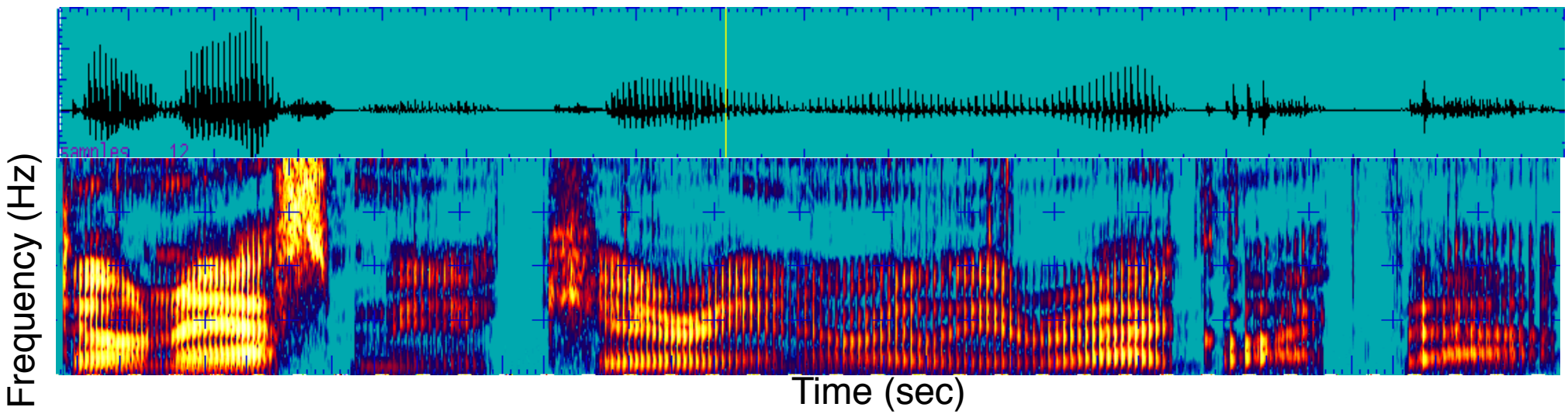


- **Feature extraction**
  - Cepstral features
  - Prosodic features
  - High level features

- **Speaker modeling**
  - Gaussian Mixture Models
  - Hidden Markov Models
  - Support Vector Machines
  - Neural Networks

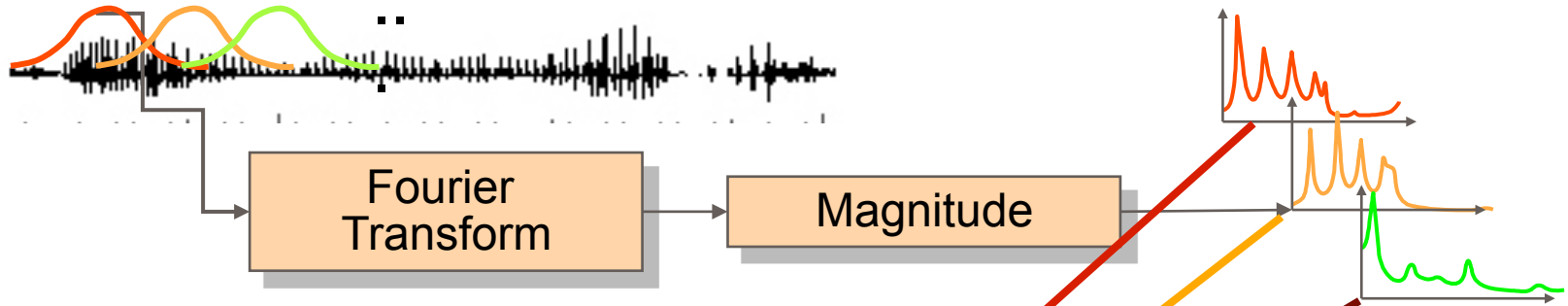
Information in Speech

- **Speech is a time-varying signal conveying multiple layers of information**
  - Words
  - Speaker
  - Language
  - Emotion
- **Information in speech is observed in the time and frequency domains**

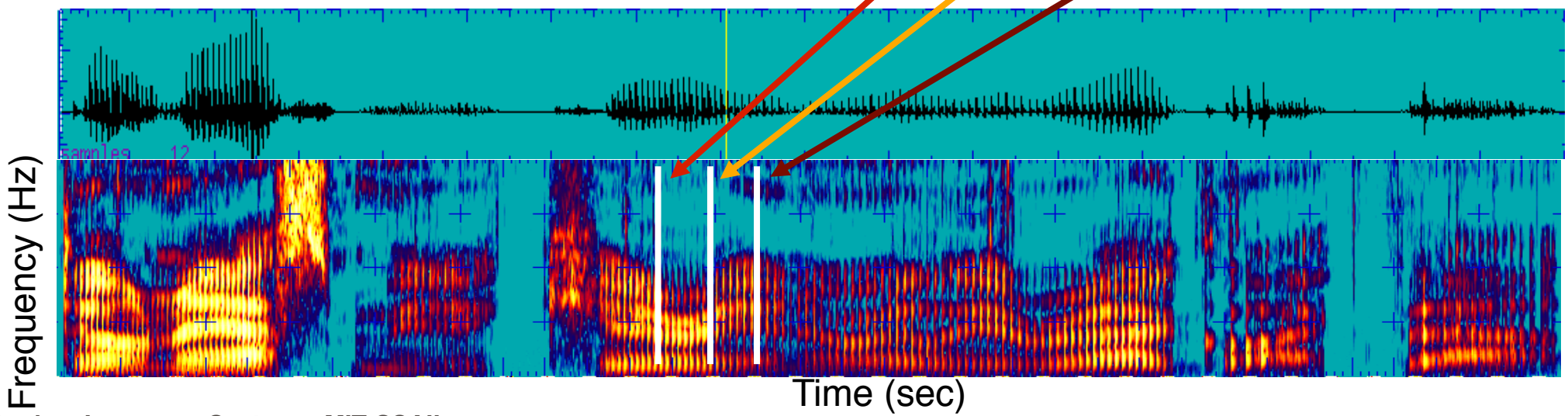


Feature Extraction from Speech

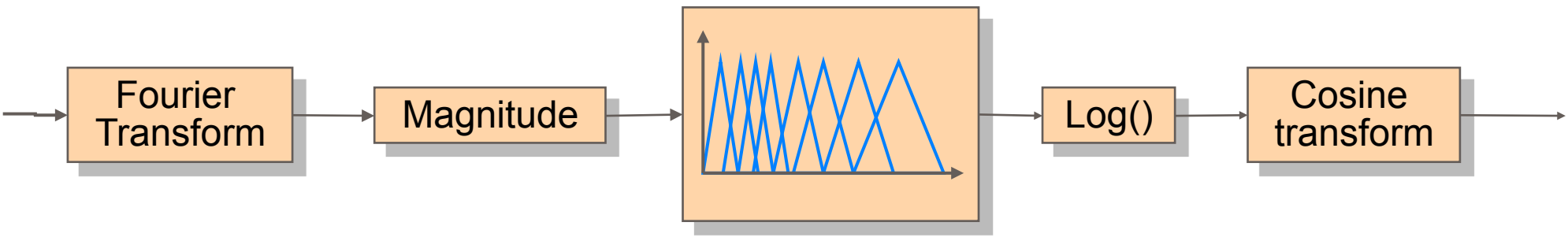
- **A time sequence of features is needed to capture speech information**
  - Typically some spectral-based features are extracted using sliding window - 20 ms window, 10 ms shift



- Produces time-frequency evolution of the spectrum

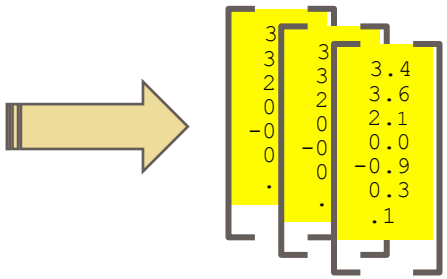


Cepstral Features



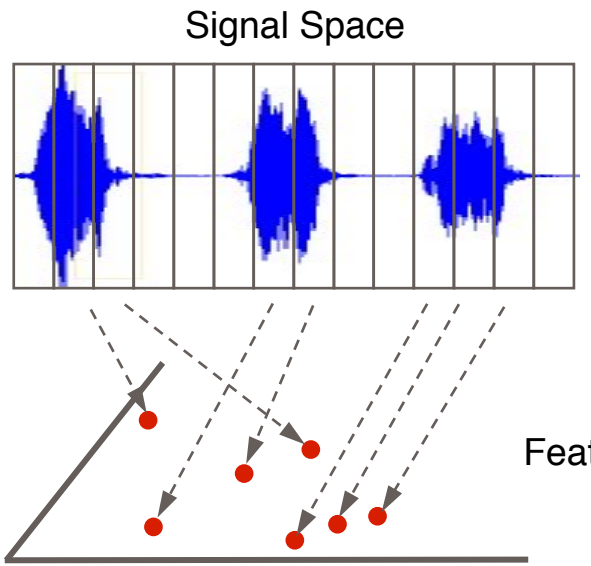
Modeling Sequence of Features
Gaussian Mixture Models

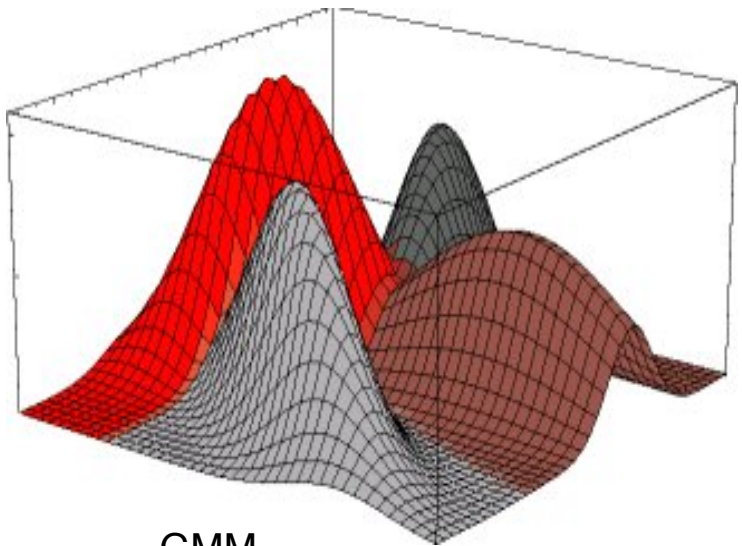
- For most recognition tasks, we need to model the distribution of feature vector sequences



100 vec/sec

- In practice, we often use Gaussian Mixture Models (GMMs).





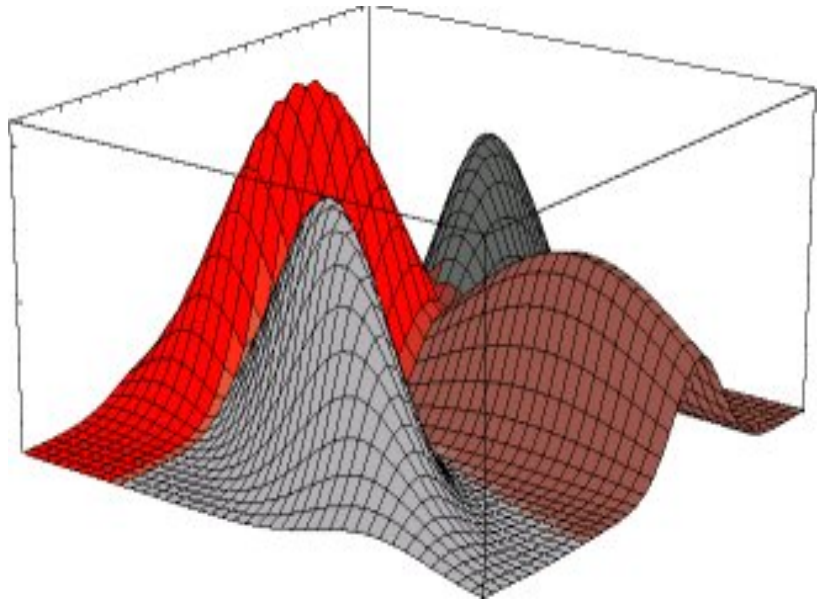
GMM

Gaussian Mixture Models

- **A GMM is a weighted sum of Gaussian distributions**

$$p(\vec{x} \mid \lambda_s) = \sum_{i=1}^{M} p_i b_i(\vec{x})$$

$$\lambda_s = (p_i, \vec{\mu}_i, \Sigma_i)$$



p_i = mixture weight (Gaussian prior proability)

$\vec{\mu}_i$ = mixture mean vector

Σ_i = mixture covariance matrix

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} \left|\Sigma_i\right|^{1/2}} \exp(-\tfrac{1}{2}(\vec{x} - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i))$$

Gaussian Mixture Models
Log Likelihood

- **To build a GMM, we need to do two things**
  - 1 – Compute the likelihood of a sequence of features given a GMM
  - 2 – Estimate the parameters of a GMM given a set of feature vectors
- **If we assume independence between feature vectors in a sequence, then we can compute the likelihood as**

$$p(\vec{x}_1,\ldots,\vec{x}_N \mid \lambda) = \prod_{n=1}^{N} p(\vec{x}_n \mid \lambda)$$

- **Usually written as log likelihood**

$$\log p(\vec{x}_1,\ldots,\vec{x}_N \mid \lambda) = \sum_{n=1}^{N} \log p(\vec{x}_n \mid \lambda)$$

$$= \sum_{n=1}^{N} \log\left(\sum_{i=1}^{M} p_i b_i(\vec{x}_n) \right)$$

Gaussian Mixture Models
Parameter Estimation

- GMM parameters are estimated by maximizing the likelihood of on a set of training vectors

$$\lambda^* = \arg\max_{\lambda} \sum_{n=1}^{N} \log p(\vec{x}_n \mid \lambda)$$

- Setting the derivatives with respect to model parameters to zero and solving

$$\Pr(i \mid \vec{x}) = \frac{p_i b_i(\vec{x})}{\sum_{j=1}^{M} p_j b_j(\vec{x})}$$

$$p_i = \frac{1}{N} \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n)$$

$$\vec{\mu}_i = \frac{1}{n_i} \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n) \vec{x}_n$$

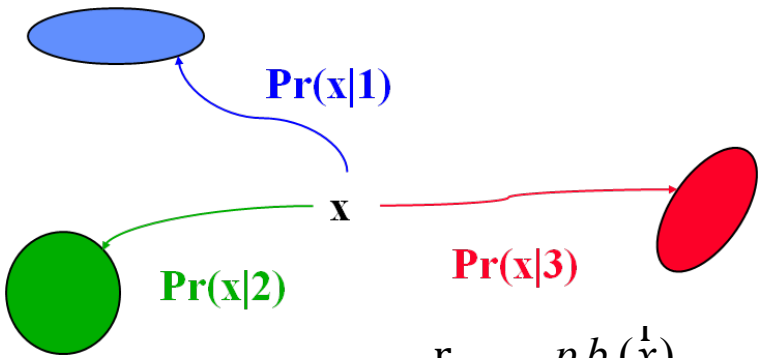
$$n_i = \sum_{n=1}^{N} \Pr(i \mid \vec{x}_t)$$

$$\Sigma_i = \frac{1}{n_i} \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n) \vec{x}_i \vec{x}_i' - \vec{\mu}_i \vec{\mu}_i'$$

Gaussian Mixture Models
Expectation Maximization (EM)

E-Step

Probabilistically align vectors to model



$$\Pr(i \mid \vec{x}) = \frac{p_i b_i(\vec{x})}{\sum_{j=1}^{M} p_j b_j(\vec{x})}$$

Accumulate sufficient statistics

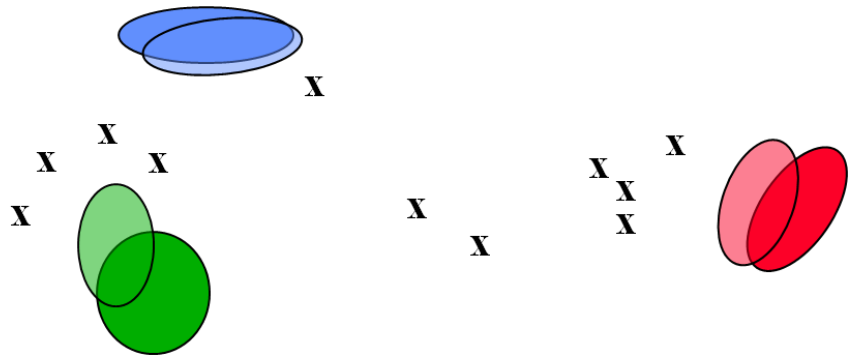
$$n_i = \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n)$$

$$E_i(\vec{x}) = \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n)\, \vec{x}_n$$

$$E_i(\vec{x}\vec{x}') = \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n)\, \vec{x}_n \vec{x}_n'$$

M-Step

Update model parameters



$$p_i = \frac{1}{N} n_i$$

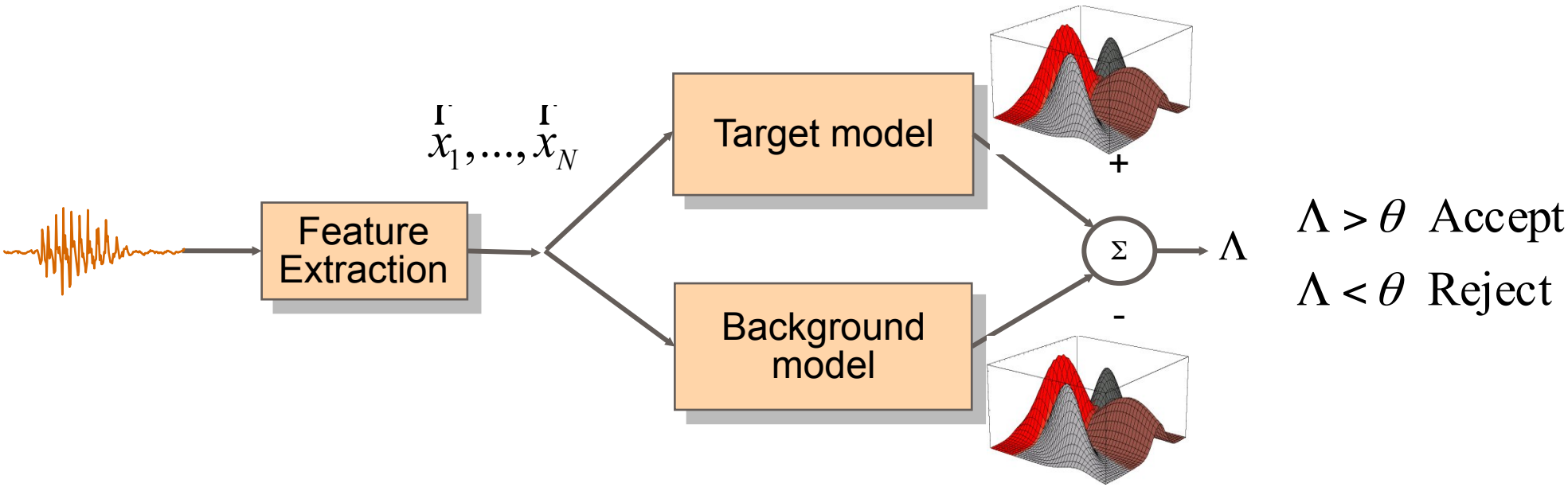
$$\vec{\mu}_i = \frac{1}{n_i} E_i(\vec{x})$$

$$\Sigma_i = \frac{1}{n_i} E_i(\vec{x}\vec{x}') - \vec{\mu}_i \vec{\mu}_i'$$

Detection System
GMM-UBM

- Realization of log-likelihood ratio test from signal detection theory

$$LLR = \Lambda = \log p(X \mid \text{target}) - \log p(X \mid \overline{\text{target}})$$



- GMMs used for both target and background model
  - Target model trained using enrollment speech
  - Background model trained using speech from many speakers (often referred to as Universal Background Model – UBM)

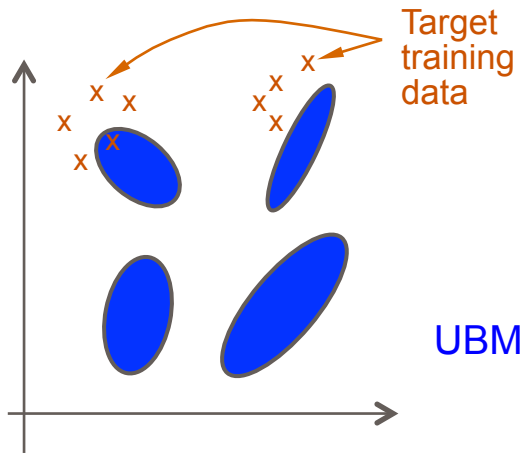
MAP Adaptation

- **Target model is often trained by adapting from background model**
  - Couples models together and helps with limited target training data
- **Maximum A Posteriori (MAP) Adaptation (similar to EM)**
  - Align target training vectors to UBM
  - Accumulate sufficient statistics
  - Update target model parameters with smoothing to UBM parameters
- **Adaptation only updates parameters representing acoustic events seen in target training data**
  - Sparse regions of feature space filled in by UBM parameters
- **Side benefits**
  - Keeps correspondence between target and UBM mixtures (important later)

Adapted GMMs
Mean-only adaptation

- Probabilistically align target training data into UBM mixture states

$$\Pr(i \mid \vec{x}) = \frac{p_i b_i(\vec{x})}{\sum_{j=1}^{M} p_j b_j(\vec{x})}$$



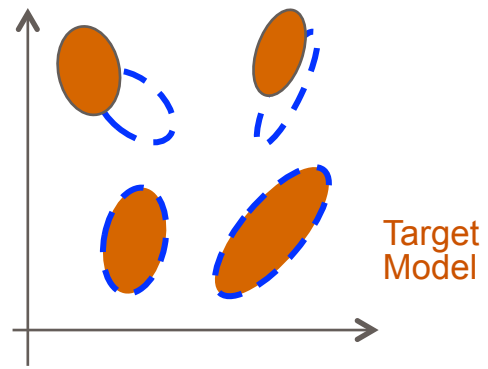
- Accumulate sufficient statistics from probabilistic alignment
  - Mean-only adaptation empirically found to be better

$$n_i = \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n)$$

$$E_i(\vec{x}) = \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n)\, \vec{x}_n$$

- Update target model parameters using sufficient statistics and adapt parameter (α)
  - *Relevance factor r controls rate of adaptation*
  - *r → 0, MAP → EM*
  - *r → ∞. No adaptation*

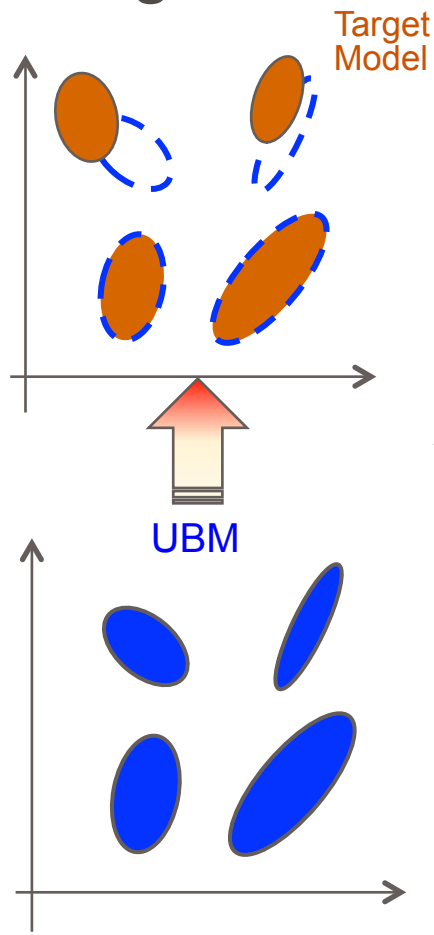
$$\alpha_i = \frac{n_i}{n_i + r}$$



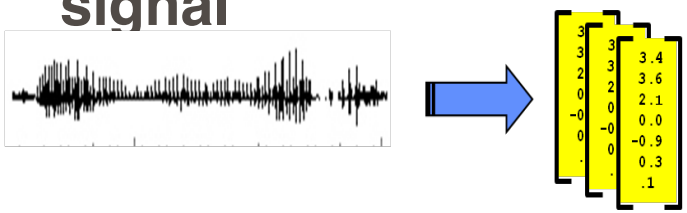
$$\vec{\mu}_i = \alpha_I E_i(\vec{x}) + (1 - \alpha_I)\vec{\mu}_i^{ubm}$$

GMM-UBM Recap

(3) Adapt target model from UBM



(1) Extract feature vector sequence from speech signal



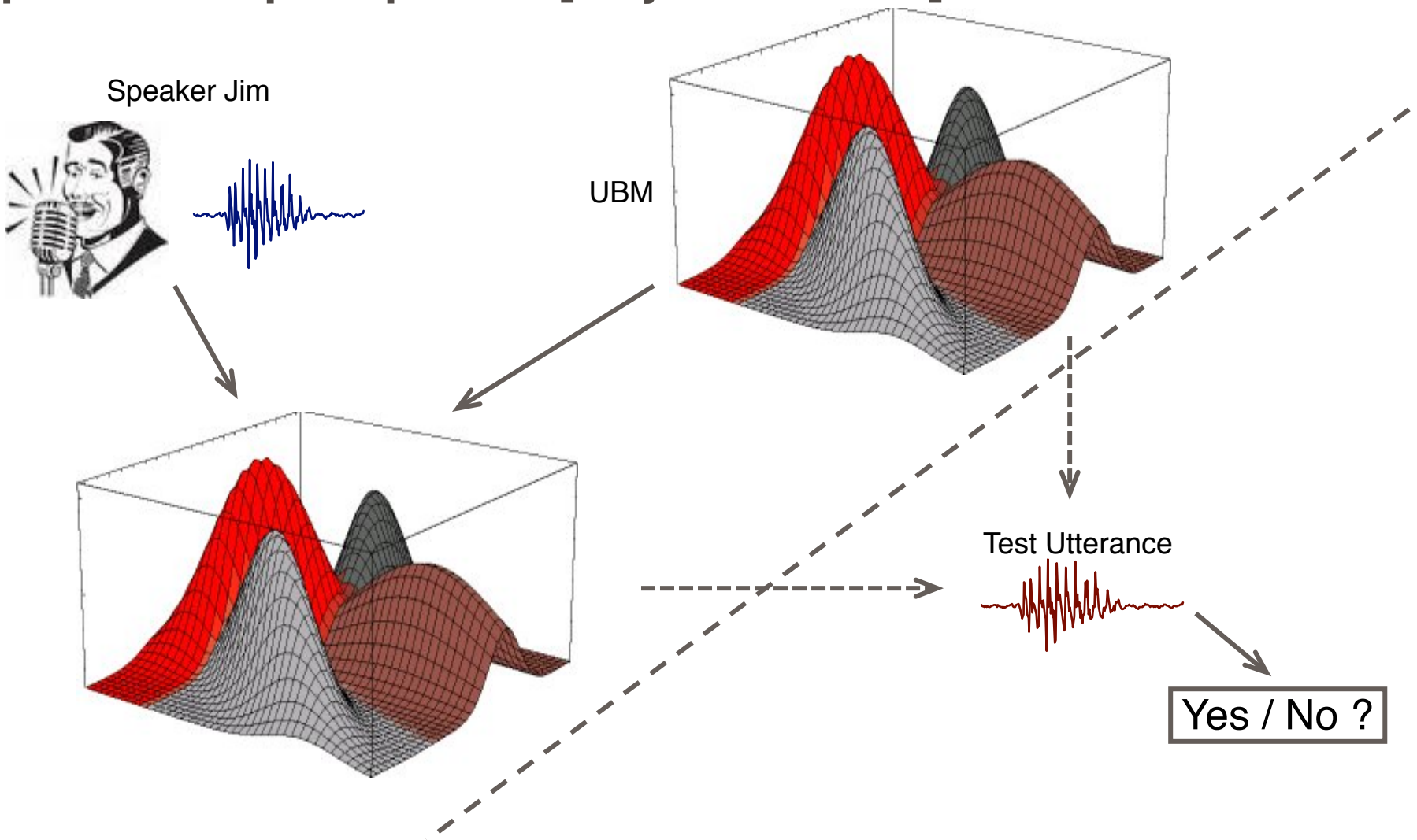
(4) Compute likelihood ratio of test data

$$LLR(X) = \log p(X \mid \lambda_{target}) - \log p(X \mid \lambda_{ubm})$$

(2) Train UBM with speech from many speakers using EM

The GMM-UBM Approach

- We enroll a given speaker by adapting the UBM using the speaker's input speech. [Reynolds 2000]



Another View of Log-Likelihoods

- We can use sufficient statistics to score a GMM…

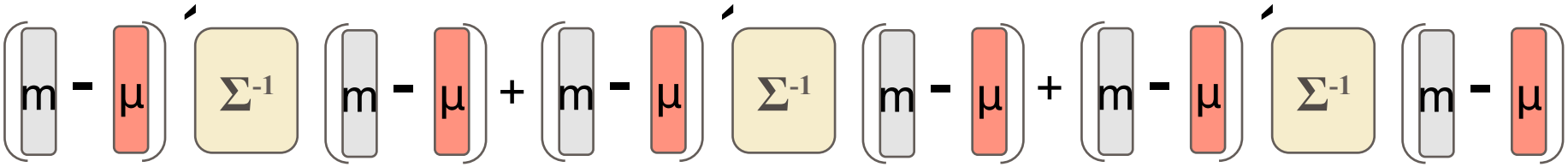
$$n_i = \sum_{n=1}^{N} \Pr(i \mid \vec{x}_n) \qquad \vec{m}_i = E_i\left[\vec{x}\right]/n_i \qquad S_i = E_i(\vec{x}\vec{x}')/n_i$$

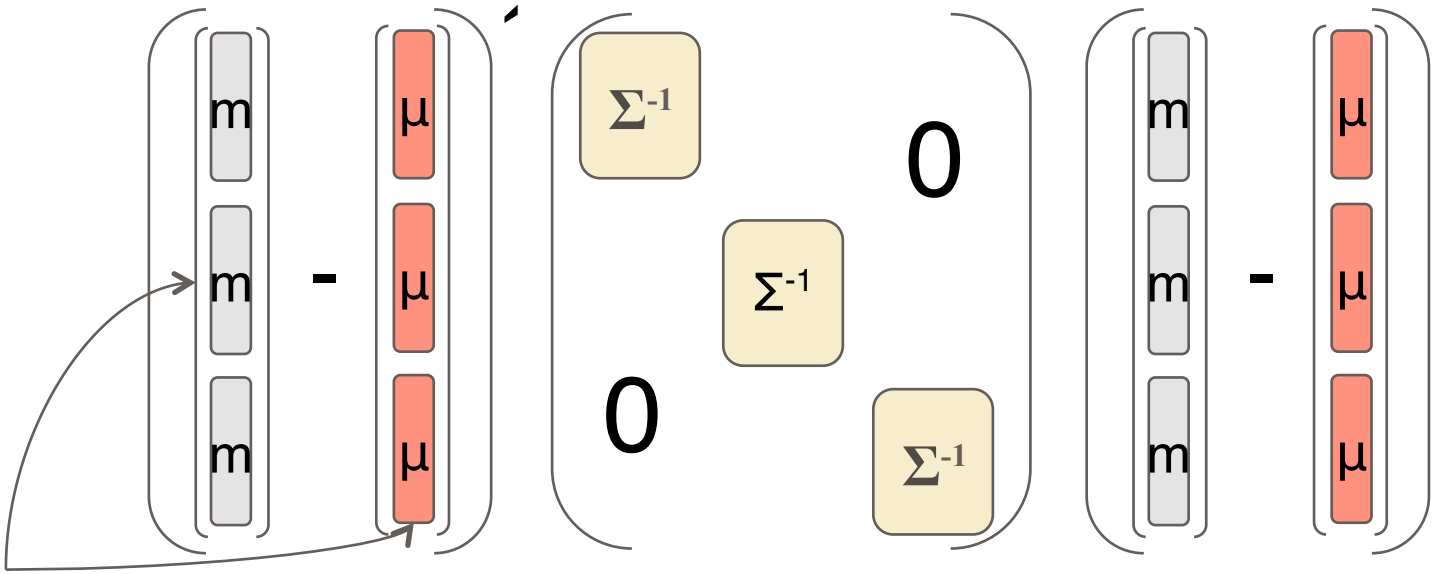
- And since we adapt only the UBM means,

$$E\left[\log p(X, I \mid \lambda)\right] = -\frac{1}{2}\sum_{i=1}^{M}\left(\vec{m}_i - \vec{\mu}_i\right)'\left(\frac{\Sigma_i^{ubm}}{n_i}\right)^{-1}\left(\vec{m}_i - \vec{\mu}_i\right) + C_i$$

Supervectors

- By stacking vectors and matrices, we can work directly with vector-matrix manipulations





Super-vectors

MAP Reformulated

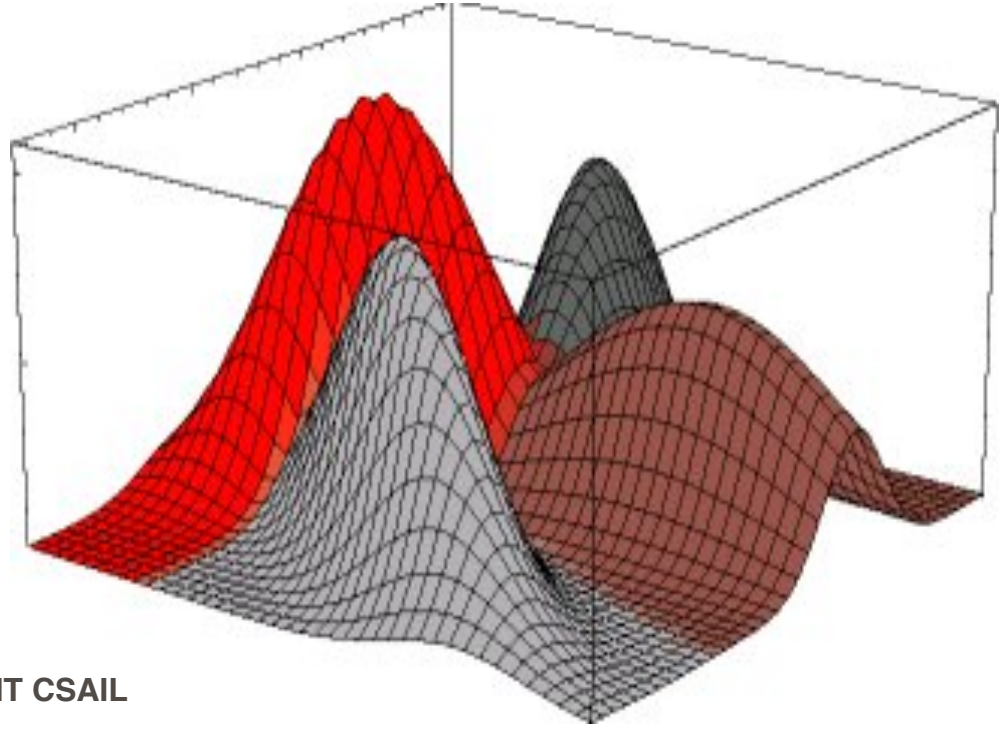
- **New design for MAP adaptation based on Factor Analysis**
- $M = m + Dz$
  - M : speaker and channel dependent supervector
  - m : speaker and channel independent supervector (UBM)
  - d : diagonal matrix
  - z **:** random vectors with a standard normal prior

- **M is normally distributed with mean m and covariance D^2.**
- **Matrix D can be trained via maximum likelihood.**

- **If we let** $D^2 = {}^{1}\!/_{r} \cdot \Sigma$
  - then we have the equivalent of Relevance MAP adaptation.

Intuition

- **The way the UBM adapts to a given speaker ought to be somewhat constrained**
  - There should exist some relationship in the way the mean parameters move relative to speaker to another
  - The Joint Factor Analysis [Kenny 2008] explored this relationship
    * **Jointly model between- and within-speaker variabilities**
  - Support Vector Machine GMM supervector [Campbell 2006]



Roadmap

- **Introduction**
  - Terminology, tasks, and framework
- **Low-Dimensional Representation**
  - Speaker Verification
  - Sequence of features: GMM
  - Super-vectors: JFA
  - Low-dimensional vectors: i-vectors
  - Processing i-vectors: compensation and scoring
  - Data visualization
- **Other Applications**
  - Speaker diarization
  - Language recognition

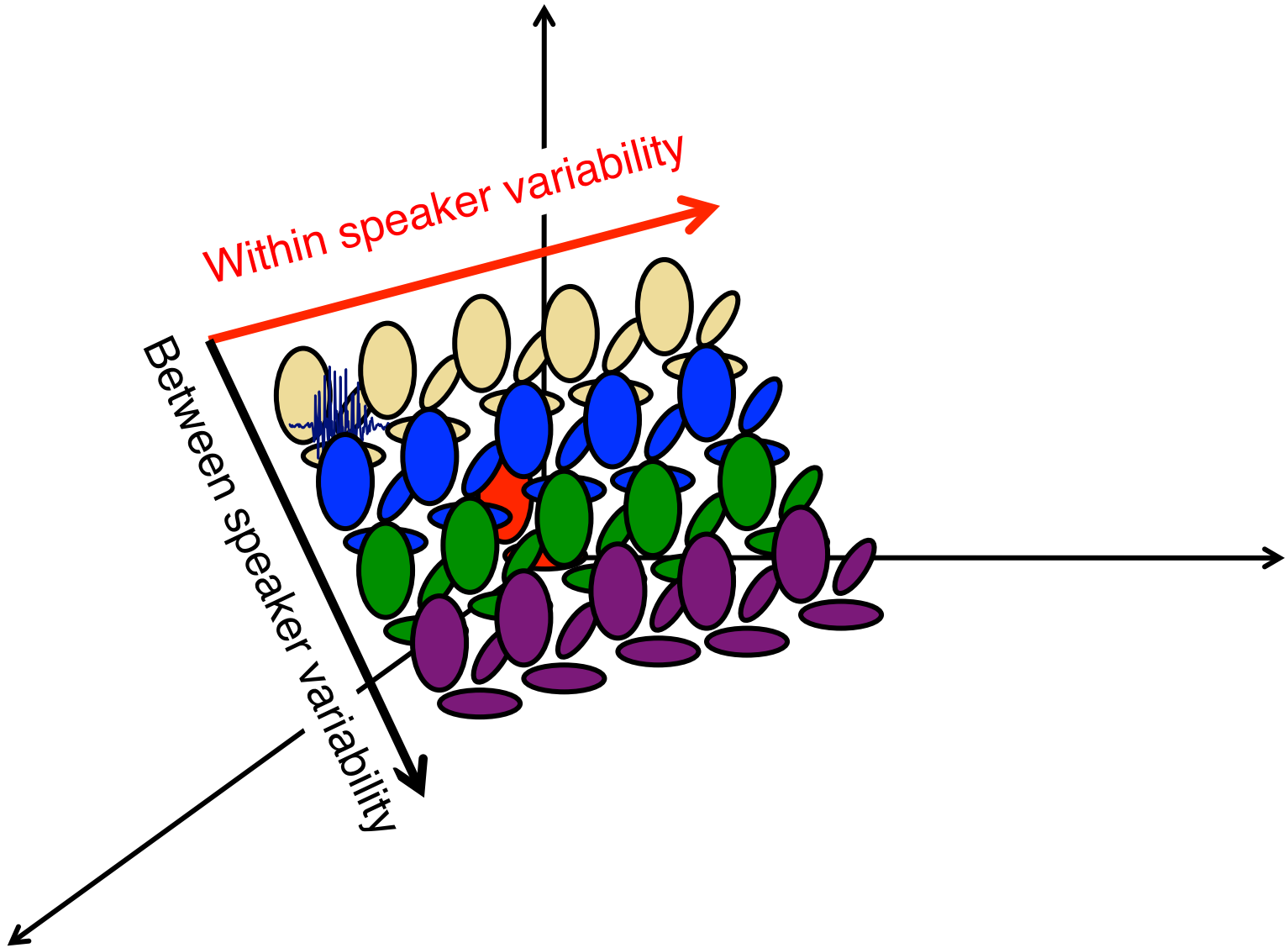
Handling Variabilities

- **Distinguishing between good and bad speaker variabilities**

- **Useful variability**
  - Between speaker variability
    - **Modeling the difference between speakers**

- **Bad variability**
  - Within speaker variability (intersession variability)
    - **Channel variability**
    - **Emotional state**
    - **Physical state**

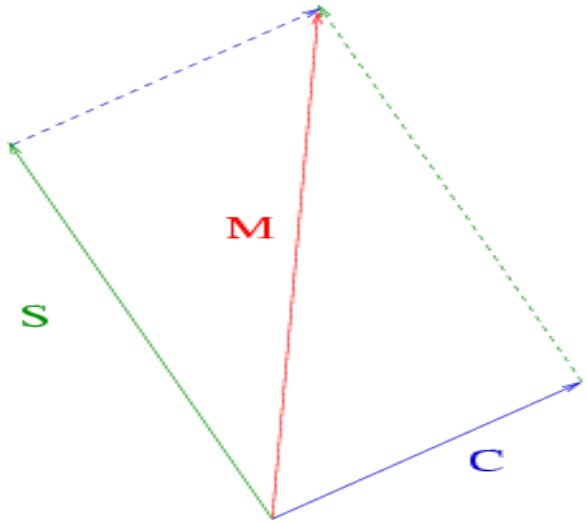
Joint Factor Analysis



Joint Factor Analysis

- Proposed for the GMM frameworks
- Assumption [Kenny2008]

$$\mathbf{M=s+c}$$



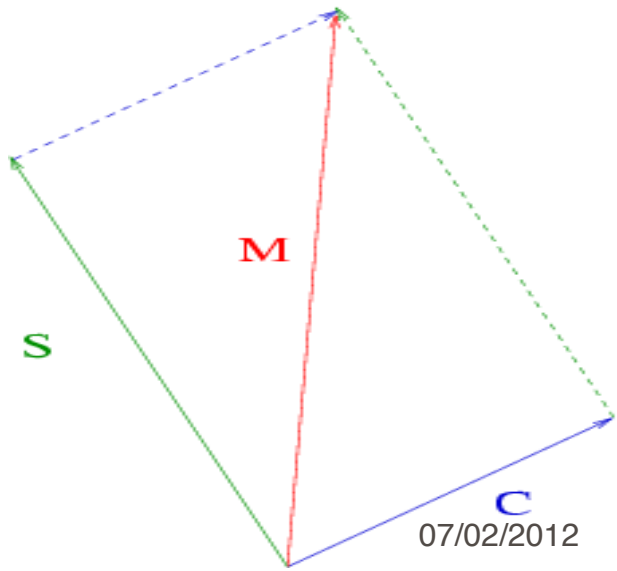
- M : speaker and channel dependent supervector
- s : speaker dependent supervector
- c : channel dependent supervector
  - GMM supervector : concatenation of the means components

Joint Factor Analysis – Details

- $s = m + Vy + Dz$
  - m : speaker- and channel-independent supervector
  - V : rectangular matrix of low-rank (eigenvoices- speaker space)
  - D : diagonal matrix
  - y, z : random vectors with a standard normal prior

- $c = Ux$
  - U : rectangular matrix of low rank (eigenchannels- channel space)
  - x : random vector with standard normal prior

$$M = m + Vy + Dz + Ux$$



Joint Factor Analysis: scoring

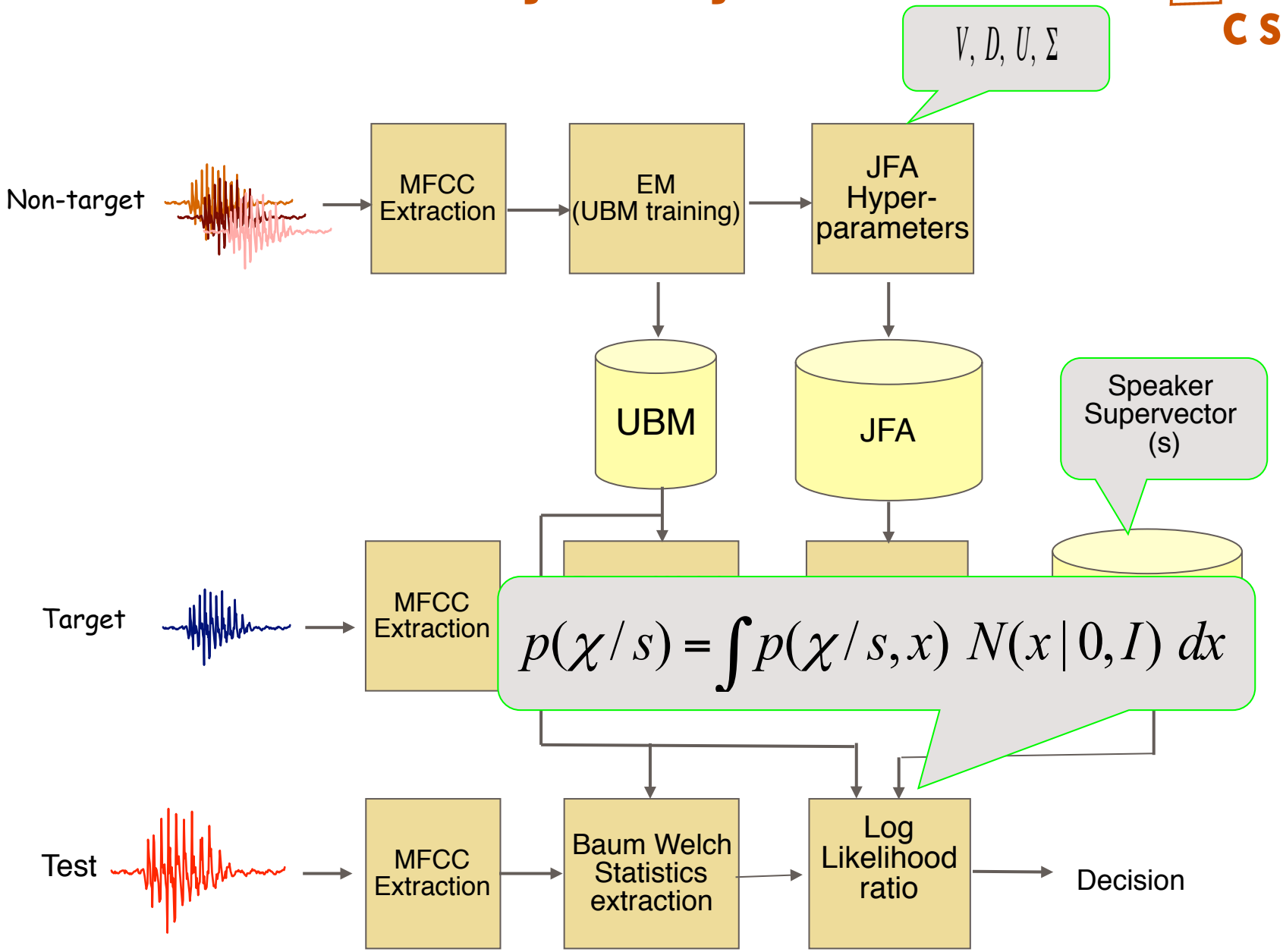
- Likelihood [Kenny2008]

$$p(\chi / s) = \int p(\chi / s, x)\ N(x \mid 0, I)\ dx$$

- Log likelihood ratio

$$score = \ln \frac{p(\chi / s)}{p(\chi / \Omega)} \begin{matrix} \geq \\ < \end{matrix} \theta$$

Joint Factor Analysis System



Roadmap

- Introduction
  - Terminology, tasks, and framework
- **Low-Dimensional Representation**
  - Speaker Verification
  - Sequence of features: GMM
  - Super-vectors: JFA
  - Low-dimensional vectors: i-vectors
  - Processing i-vectors: compensation and scoring
  - Data visualization
- Other Applications
  - Speaker diarization
  - Language recognition

The story begins…

- **John Hopkins University**
  - The Center for Language and Speech Processing Summer Workshop 2008

- **Robust Speaker Recognition Over Varying Channels**

- **The importance of each factor vect**

Channel factors
Contain speaker
information

$$\mathbf{M} = m + V\mathbf{y} + D\mathbf{z} + U\mathbf{x}$$

Total variability

- Factor analysis as feature extractor
- Joint factor analysis

$$\mathbf{M} = \boldsymbol{m} + \boldsymbol{Vy} + \boldsymbol{Dz} + \boldsymbol{Ux}$$

- Speaker and channel dependent supervector

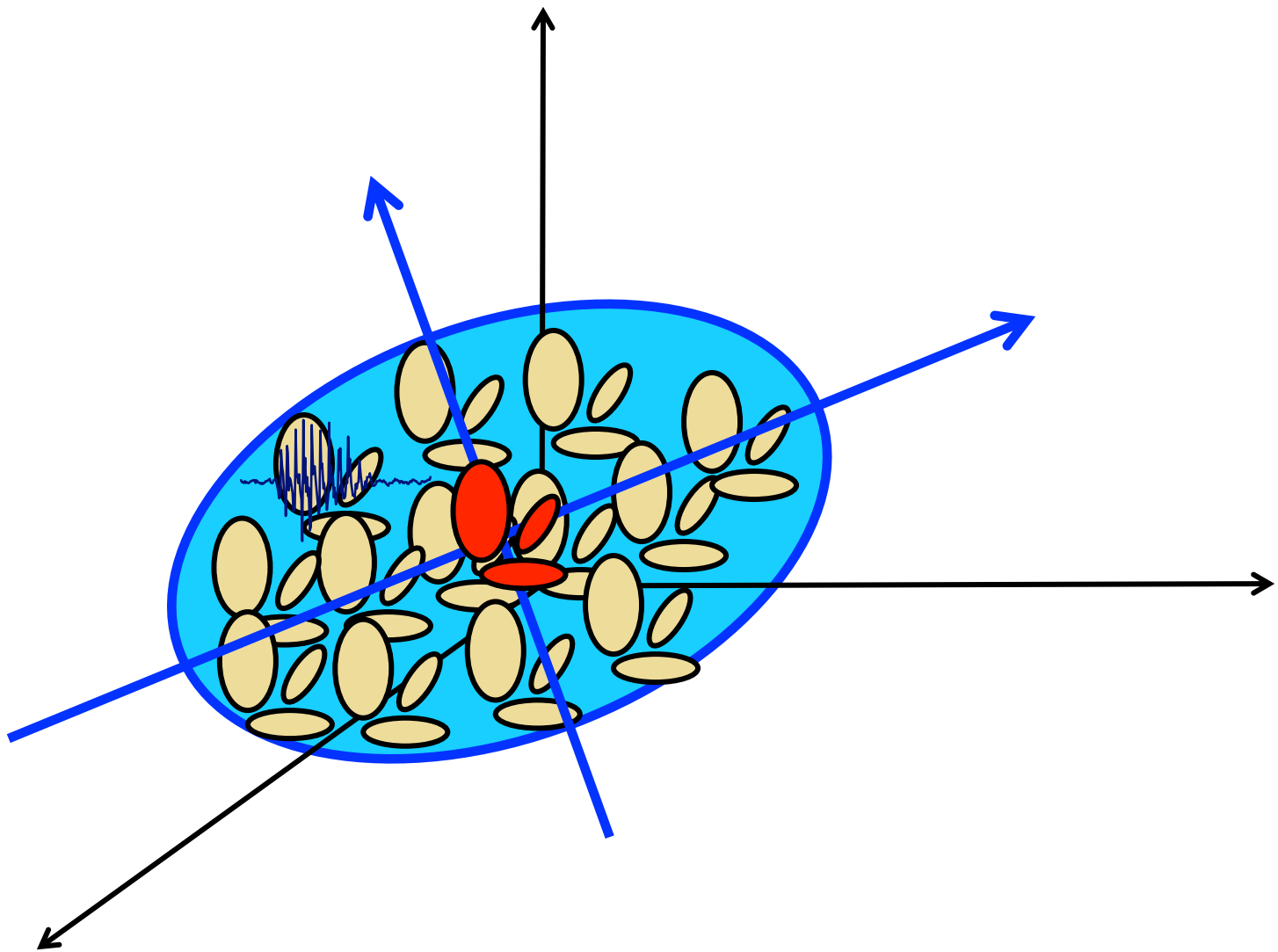
$$\mathbf{M} = \boldsymbol{m} + \boldsymbol{Tw}$$

  - T is rectangular, low rank (total variability matrix)
  - w standard Normal random (total factors – intermediate vector or i-vector)
- Estimate the i-vector: contain both variabilities
- **No distinction between speaker V and channel U variabilities**
- We will apply channel compensation later in the i-vector space.

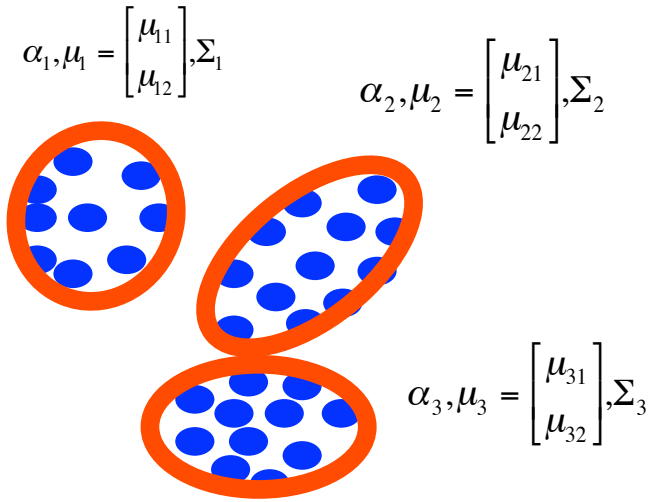
Najim Dehak, Patrick Kenny, Pierre Dumouchel, Reda Dehak, Pierre Ouellet, «Front-end factor analysis for speaker verification » in IEEE Transactions on Audio, speech and Language Processing 2011.

Najim Dehak, Réda Dehak, Patrick Kenny, Niko Brummer, Pierre Ouellet and Pierre Dumouchel, Support Vector Machine versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification. In Proc INTERSPEECH 2009, Brighton, UK, September 2009.

Total variability space



Why call it an i-vector?



GMM components: 2048
Feature dimension: 60

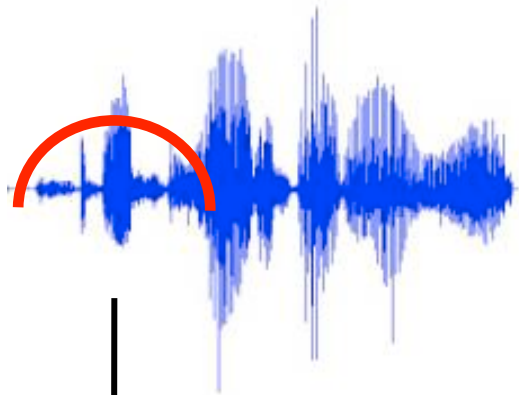
GMM-SV :
60*2048=122880

$$\begin{bmatrix} \mu_{11} \\ \mu_{12} \\ \mu_{21} \\ \mu_{22} \\ \mu_{31} \\ \mu_{32} \end{bmatrix}$$

I- for Intermediate representation

IVECTOR

Actually between 100 to 1000



MFCC

Feature dimension 60

Extracting the Hyperparameters

- Speaker and channel dependent supervector

$$\mathbf{M} = \boldsymbol{m} + \boldsymbol{Tw}$$

- The i-vector extractor is characterized by
  - m : A supervector mean (can be the UBM)
  - T : low rank Total variability matrix
  - Σ : diagonal covariance matrix
- **Preliminaries**
  - Acoustic observations $u = \{\vec{x}_1, \ldots, \vec{x}_L\}$
    - * **Each y_t has dimension F**
  - Universal Background Model θ_{UBM}
    - * **Number of Gaussian components C, indexed by c**
    - * **→ Supervector dimension = CF**

Baum-Welch (Sufficient) Statistics

- **Zeroth Order**

$$N_c(u) = \sum_{t=1}^{L} P(c \mid \vec{x}_t, \theta_{\text{UBM}}) = \sum_t \gamma_t(c)$$

- **First Order**

$$F_c(u) = \sum_{t=1}^{L} P(c \mid \vec{x}_t, \theta_{\text{UBM}}) \cdot \vec{x}_t = \sum_t \gamma_t(c) \cdot \vec{x}_t$$

- **Second Order**

$$S_c(u) = \text{diag}\left(\sum_t \gamma_t(c) \cdot \vec{x}_t \vec{x}_t^{\,t} \right)$$

where $c = 1, \ldots, C$ for each UBM component

Simplified Notation I

- **Recall**

$$\gamma_t(c) = P(c \mid \vec{x}_t, \theta_{\text{UBM}}) = \frac{\pi_c P_c(\vec{x}_t \mid \mu_c, \Sigma_c)}{\sum_{i=1}^{C} \pi_i P_i(\vec{x}_t \mid \mu_i, \Sigma_i)}$$

- **Centralized First- / Second-Order Statistics**

$$\tilde{F}_c(u) = \sum_t \gamma_t(c) \cdot (\vec{x}_t - m_c)$$

$$\tilde{S}_c(u) = \text{diag}\left(\sum_t \gamma_t(c) \cdot (\vec{x}_t - m_c)(\vec{x}_t - m_c)^t \right)$$

$$m = \begin{bmatrix} m_1 & m_2 & \cdots & m_C \end{bmatrix}^t$$

Simplified Notation II

$$N(u) = \begin{bmatrix} N_1(u) \cdot I_{F \times F} & 0 & \cdots & 0 \\ 0 & N_2(u) \cdot I_{F \times F} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & N_C(u) \cdot I_{F \times F} \end{bmatrix}$$

$$\widetilde{F}(u) = \begin{bmatrix} \widetilde{F}_1(u) \\ \widetilde{F}_2(u) \\ \vdots \\ \widetilde{F}_C(u) \end{bmatrix} \qquad \widetilde{S}(u) = \begin{bmatrix} \widetilde{S}_1(u) & 0 & \cdots & 0 \\ 0 & \widetilde{S}_2(u) & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \widetilde{S}_C(u) \end{bmatrix}$$

The EM Algorithm

- **Initialize m and Σ as defined by our UBM covariance matrices**
- **Pick a desired rank R for the Total Variability Matrix T and initialize this CF x R matrix randomly.**

- **E-step:**
  - For each utterance u, calculate the parameters of the posterior distribution of w(u) using the current estimates of m, T, Σ

- **M-step:**
  - Update T and by Σ solving a set of linear equations in which the w(u)'s play the role of explanatory variables

- **Iterate until parameters / data likelihood converges…**

Kenny, P., Boulianne, G. and P. Dumouchel. Eigenvoice Modeling with Sparse Training Data. IEEE Transactions on Speech and Audio Processing, 13 May (3) 2005 : 345-359.

E-step: The Posterior Distribution of w(u)

- For each utterance u, let l(u) be the matrix defined by

$$l(u) = I + T^t \Sigma^{-1} N(u) T$$

- Then the posterior distribution of w(u) conditioned on the acoustic observations of an utterance u is Gaussian with mean

$$E[w(u)] = l^{-1}(u) T^t \Sigma^{-1} \tilde{F}(u)$$

and covariance matrix

$$\text{cov}(w(u), w(u)) = l^{-1}(u)$$

Kenny, P., Boulianne, G. and P. Dumouchel. Eigenvoice Modeling with Sparse Training Data. IEEE Transactions on Speech and Audio Processing, 13 May (3) 2005 : 345-359.

Proof Sketch

- To show this, let $E(u) = l^{-1}(u)T^t\Sigma^{-1}\tilde{F}(u)$
- Then it suffices to show that

$$P_{T,\Sigma}(w \mid u) \propto \exp\left(-\frac{1}{2}(w - E(u))^t l(u)(w - E(u))\right)$$

- First, just apply Bayes' Rule

$$P_{T,\Sigma}(w \mid u) \propto P_{T,\Sigma}(u \mid w) \cdot N(w \mid 0, I)$$
$$= P_{T,\Sigma}(\{\vec{x}_1, \ldots, \vec{x}_L\} \mid w) \cdot N(w \mid 0, I)$$

M-step: Maximum Likelihood Re-estimation 1/2

$$N_c = \sum_u N_c(u)$$

$$A_c = \sum_u N_c(u) E[w(u)w^t(u)]$$

$$C = \sum_u \tilde{F}(u) E[w^t(u)]$$

$$N = \sum_u N(u)$$

$u = 1,\ldots,$ number of utterances

$c = 1,\ldots,$ number of GMM components

$$E[w(u)w^t(u)] = Cov(w(u),w(u)) + E[w(u)].E[w(u)]^t$$

M-step: Maximum Likelihood Re-estimation 2/2

- Update matrix T

$$T(i,:)A_c = C_i$$
$$\text{where } i = (c-1) * D\_F + f$$

- Update the diagonal covariance matrix

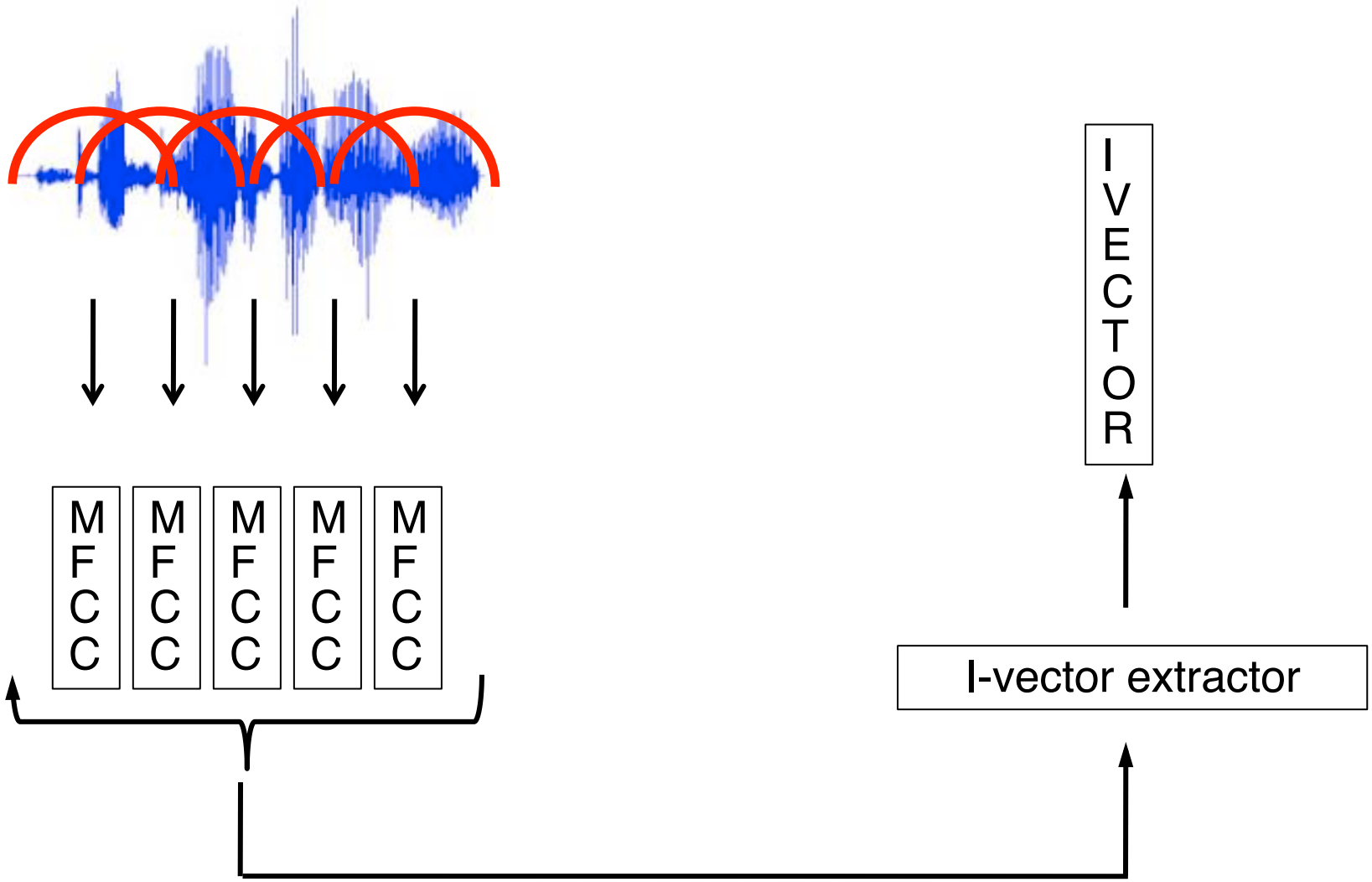
$$\Sigma = N^{-1}(\sum_u \tilde{S}(u) - diag(CT^t)$$

$$f = 1,\ldots, \text{D\_F}$$
$$\text{D\_F} = \text{dimensionality of features vector}$$
$$\text{c} = 1,\ldots, \text{number of GMM components}$$

Roadmap

- Introduction
  - Terminology, tasks, and framework
- **Low-Dimensional Representation**
  - Speaker Verification
  - Sequence of features: GMM
  - Super-vectors: JFA
  - Low-dimensional vectors: i-vectors
  - Processing i-vectors: compensation and scoring
  - Data visualization
- Other Applications
  - Speaker diarization
  - Language recognition

I-vector Extraction



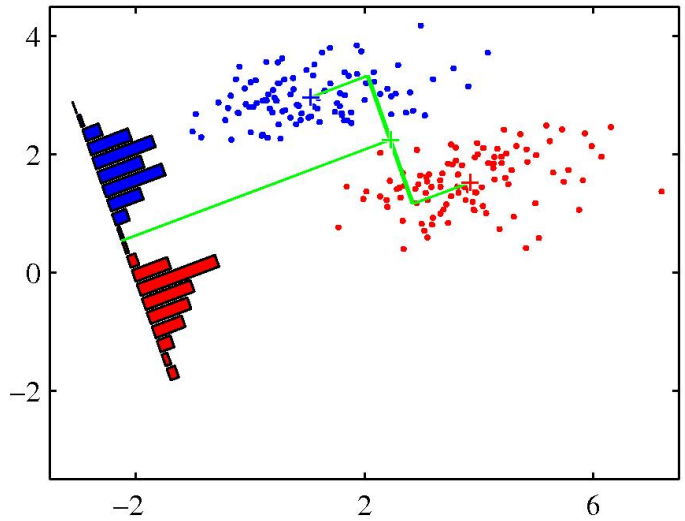
Intersession compensation

- **LDA [Dehak 2009,2011]**

A is matrix of eigenvectors from $S_b . v = \lambda . S_w . v$

$$S_b = \sum_{j=1}^{S} (w_j - \overline{w})(w_j - \overline{w})^t$$

$$S_w = \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} (w_i^s - w_s)(w_i^s - w_s)^t$$



- **LDA+ WCCN [Hatch2006] , [Dehak 2009,2011]**

$$W = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} (A^t w_i^s - w_s)(A^t w_i^s - w_s)^t$$

$w_s = \frac{1}{n_s} \sum_{i=1}^{n_s} A^t w_i^s$ mean of utterances of each speaker

S number of speakers
n_s number of utterances for each speaker (s)
$\overline{w}$ the mean of the entire population

Modified Cosine Scoring

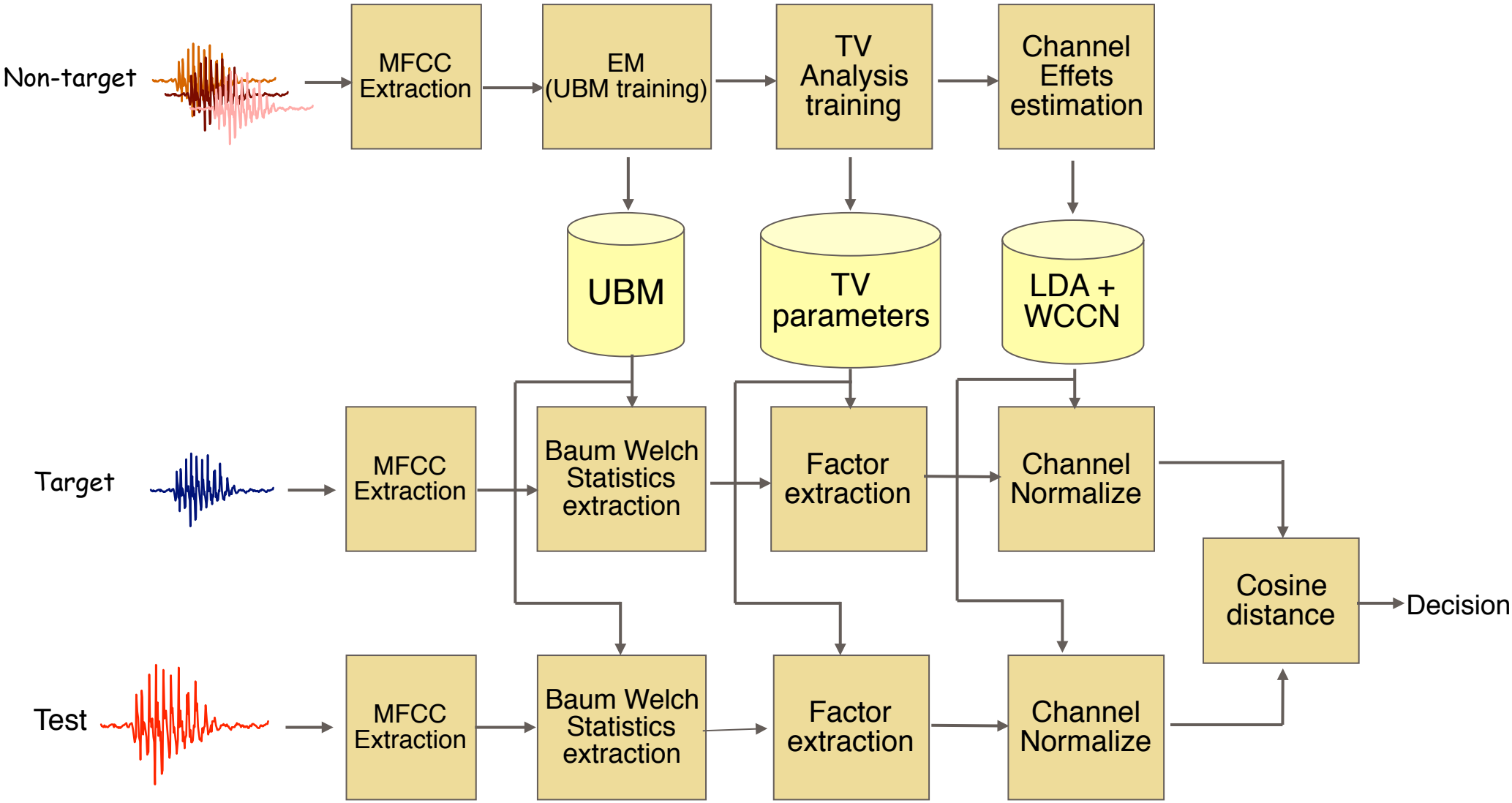
- **LDA and WCCN combination [Dehak 09,11]**

$$score(w_{target}, w_{test}) = \frac{(A^t w_{target})^t W^{-1} (A^t w_{test})}{\sqrt{(A^t w_{target})^t W^{-1} (A^t w_{target})} \cdot \sqrt{(A^t w_{test})^t W^{-1} (A^t w_{test})}} \gtrless \theta$$

A : Linear Discriminant Analysis

W : Within Class Covariance Normalization

I-vector System for Speaker Verification

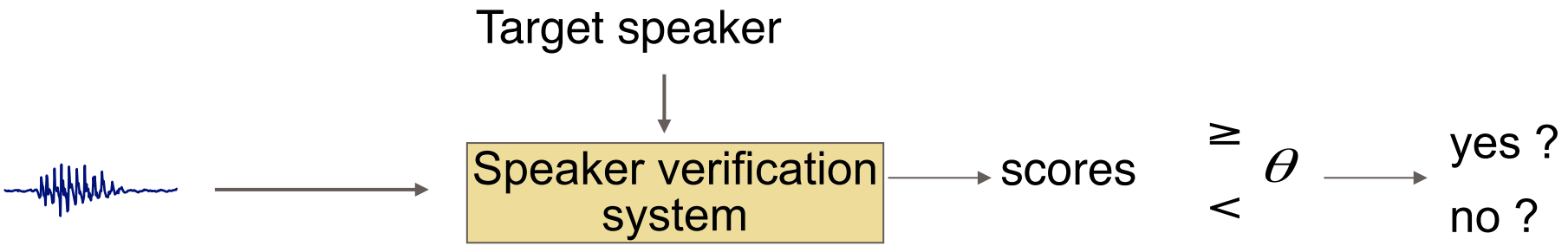


NIST Speaker Recognition evaluation

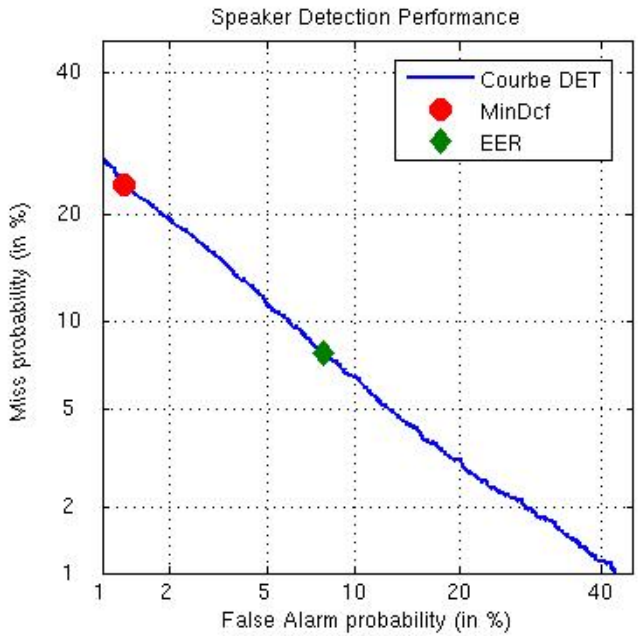
- Several training and testing conditions (dependent on speech duration: 10sec, 1conv , 3conv,…)
- Telephone conversation and microphone data

		Test			
		10sec	1conv	1conv sum	1conv aux mic
Training	10sec	**Opt**			
	1conv	Opt	**Core**	Opt	Opt
	3conv	Opt	Opt	Opt	Opt
	8conv		Opt	Opt	Opt
	3conv sum		Opt	Opt	

Speaker verification system performances



- **Detcurve**



- **False acceptance and rejection Rates**

$$R_{FA} = \frac{\text{Number of False Acceptance}}{\text{Number of impostors accesses}}$$

$$R_{FR} = \frac{\text{Number of False Rejection}}{\text{Number of target accesses}}$$

- **EER**

$$R_{FA} = R_{FR}$$

- **MinDCF**

$$DCF = C_{FR}.P_{target}.R_{FR} + C_{FA}.P_{imposteur}.R_{FA}$$

Total Variability – I-vector [Dehak 09,11]

- Factor analysis as feature extractor

- $\mathbf{M} = \boldsymbol{m} + \boldsymbol{Tw}$
  - T is rectangular, low rank (total variability matrix)
  - w variable with standard Normal prior (i-vectors)

$$w(u) = (I + T^t \Sigma^{-1} N(u) T)^{-1} T^* \Sigma^{-1} \tilde{F}(u)$$

- Cosine scoring

$$score = \frac{< w_{target}, w_{test} >}{\left\| w_{target} \right\| . \left\| w_{test} \right\|}$$

JFA/TV Comparison experiments

- **Gender dependent UBM**
  - 2048 Gaussians
  - 60 dimensional features : 19 Gaussianized MFCC's + energy + delta + double delta

- **JFA**
  - 300 speaker factors, 100 channel factors, common factors.
  - 1000 z-norm utterances and around 200 t-norm impostor models

- **Cosine distance scoring**
  - i-vector dim=400
  - LDA (dim=200) +WCCN
  - 1000 z-norm utterances and around 200 t-norm impostor models

Results on core condition
NIST 2008 SRE- JFA/TV comparison

- NIST 2008 SRE : female trials

	English trials		All trials	
	EER	M		MinDCF
JFA scoring	3.17%	0.0[illegible]	6.15%	0.032
Cosine distance scoring	**2.90%**	**0.012**	**5.76%**	**0.032**

9.5% relative improvement

- NIST 2008 SRE : male trials

	English trials		All trials	
	EER	M		MinDCF
JFA scoring	2.64%	0.0[illegible]	5.15%	0.027
Cosine distance scoring	**1.12%**	**0.009**	**4.48%**	**0.024**

57% relative improvement

Results on 10sec-10sec
NIST 2008 SRE- JFA/TV comparison

- NIST 2008 SRE : female trials

	English trials		All trials	
	EER	M		inDCF
JFA scoring	16.01%	0.0	17.99%	0.075
Cosine distance scoring	**12.19%**	**0.057**	**16.59%**	**0.072**

25% relative improvement

- NIST 2008 SRE : male trials

	English trials		All trials	
	EER	M		MinDCF
JFA scoring	15.20%	0.0	15.45%	0.068
Cosine distance scoring	**11.09%**	**0.047**	**14.44%**	**0.063**

26% relative improvement

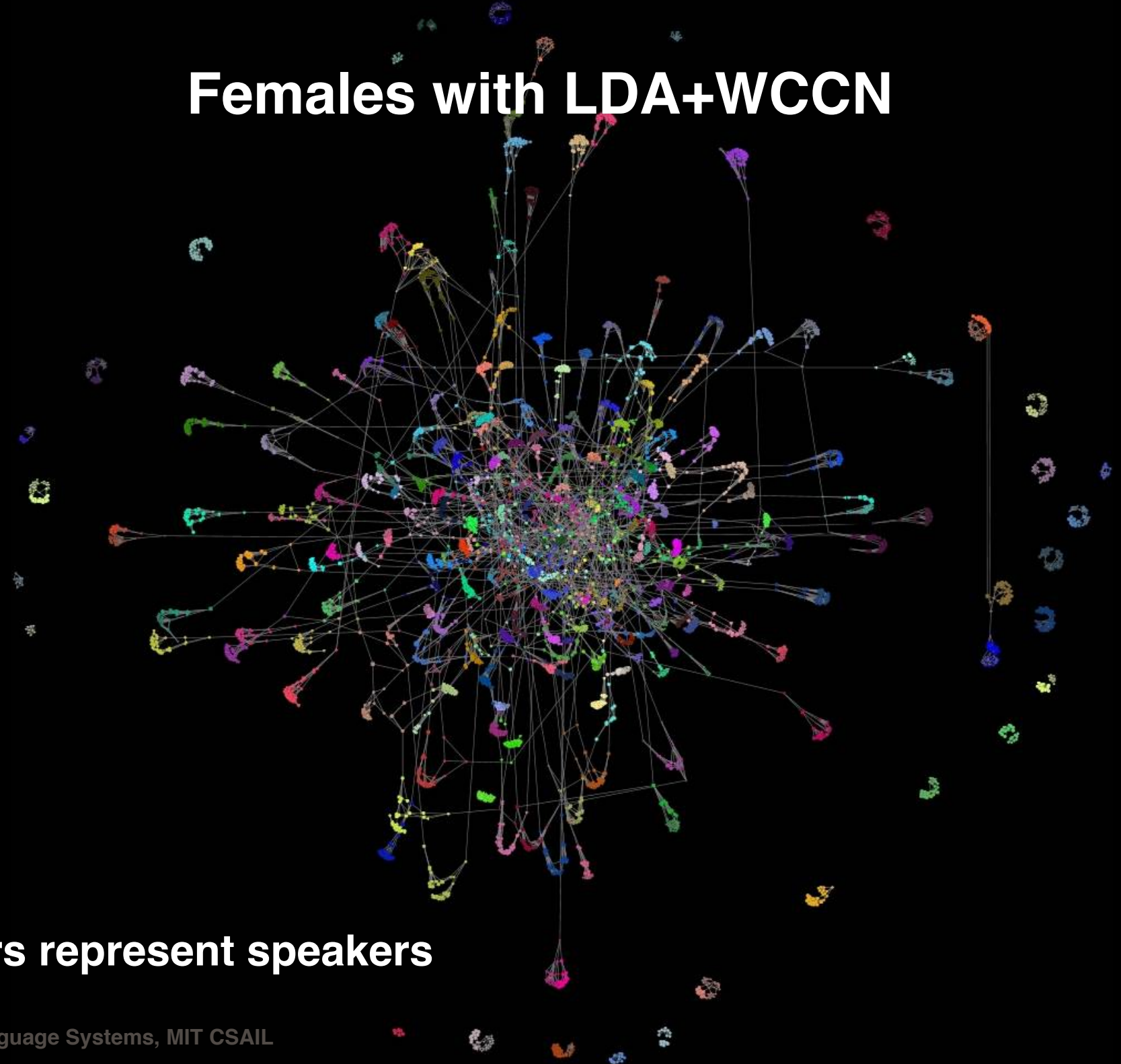
Roadmap

- Introduction
  - Terminology, tasks, and framework
- **Low-Dimensional Representation**
  - Speaker Verification
  - Sequence of features: GMM
  - Super-vectors: JFA
  - Low-dimensional vectors: i-vectors
  - Processing i-vectors: compensation and scoring
  - Data visualization
- Other Applications
  - Speaker diarization
  - Language recognition

Graph Visualization

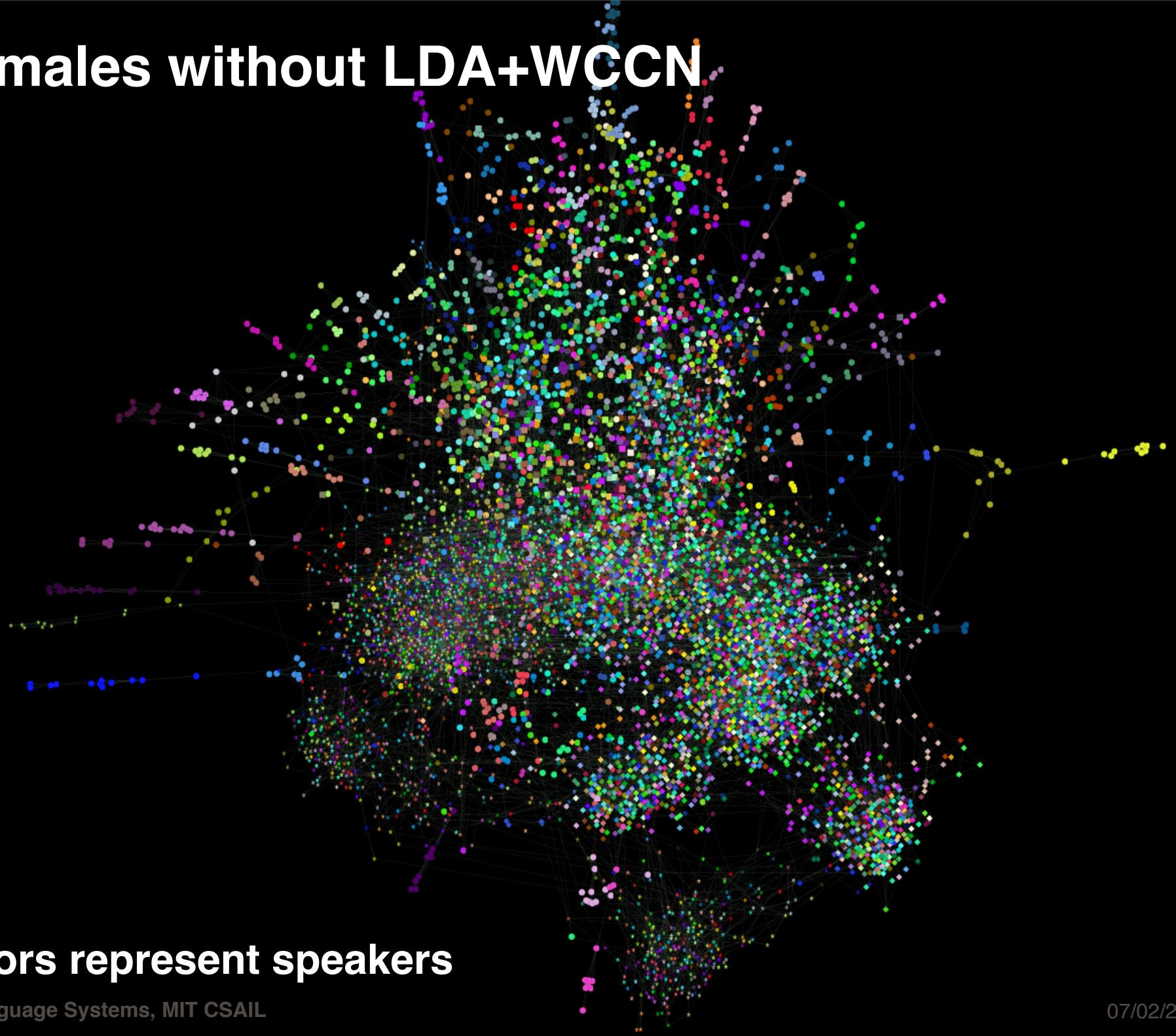
- **Visualization using the Graph Exploration System (GUESS) [Eytan 06]**
- **Represent segment as a node with connections (edges) to nearest neighbors (3 NN used)**
  - NN computed using i-vector system (with and without channel normalization)
- **Applied to 5438 utterances from the NIST SRE10 core**
  - Multiple telephone and microphone channels
- **Absolute locations of nodes not important**
- **Relative locations of nodes to one another is important:**
  - The visualization clusters nodes that are highly connected together
- **Meta data (speaker ID, channel info) not used in layout**
- **Colors and shapes of nodes used to highlight interesting phenomena**

Females with LDA+WCCN



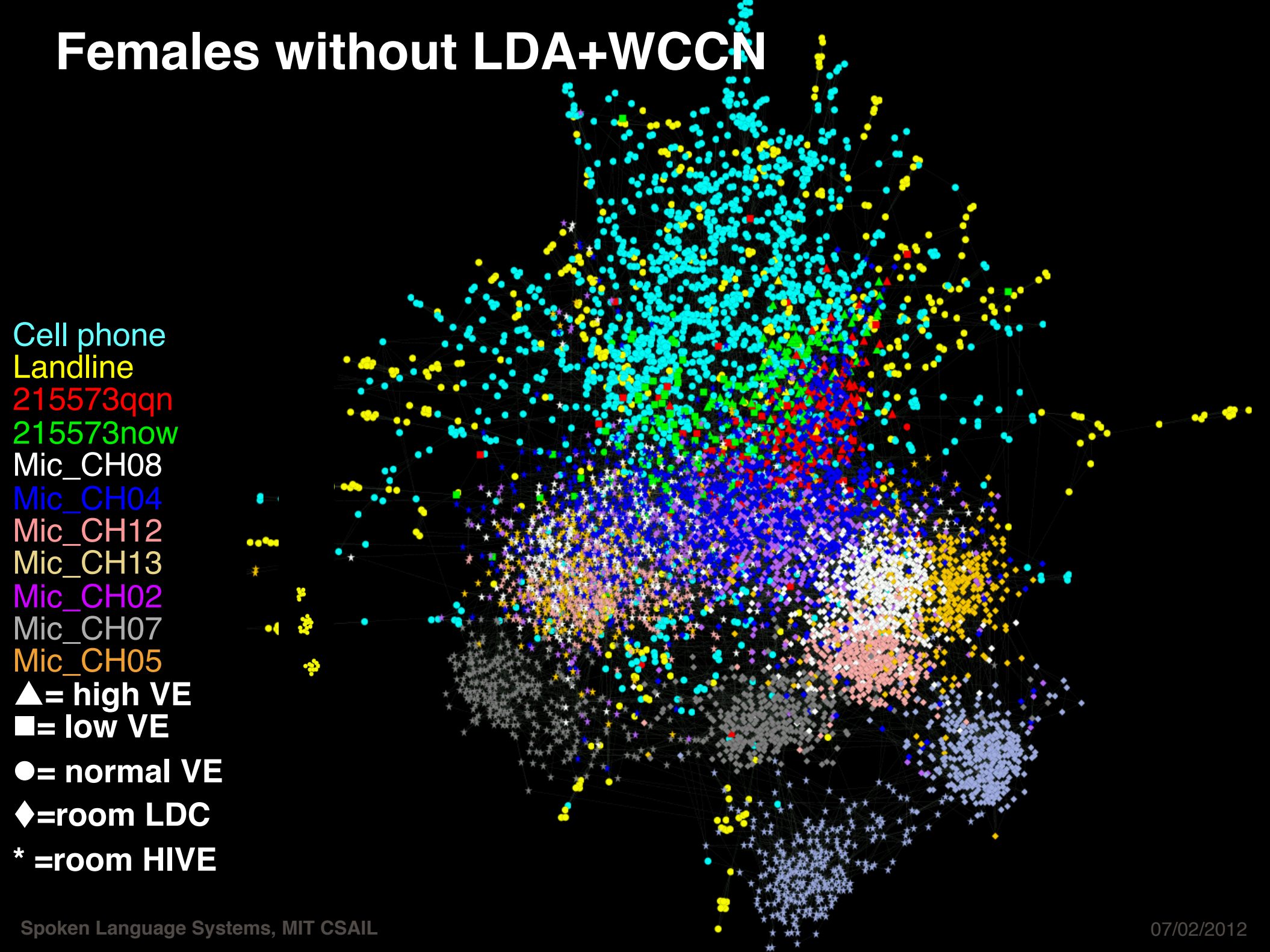
Colors represent speakers

Females without LDA+WCCN

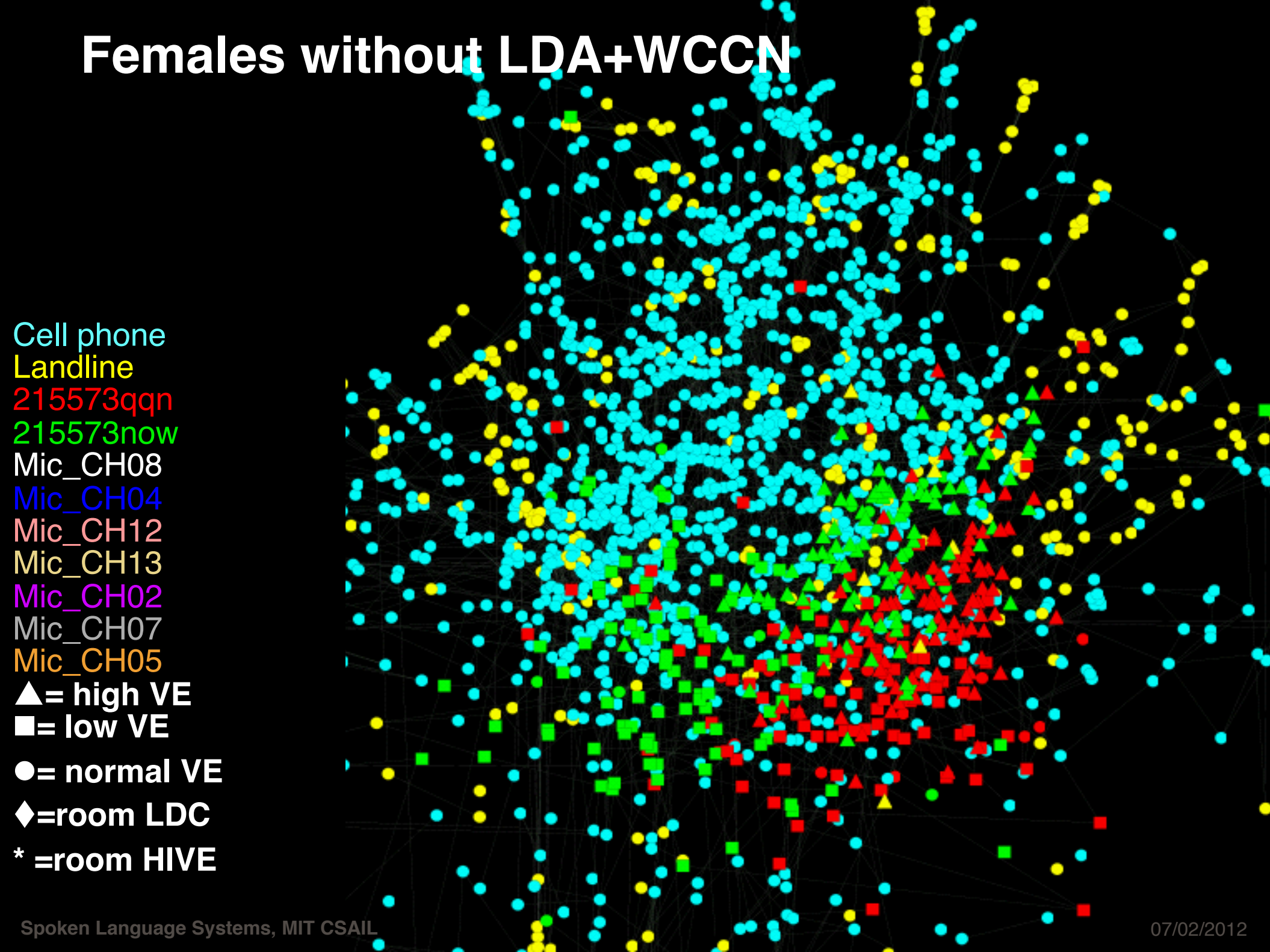


Colors represent speakers

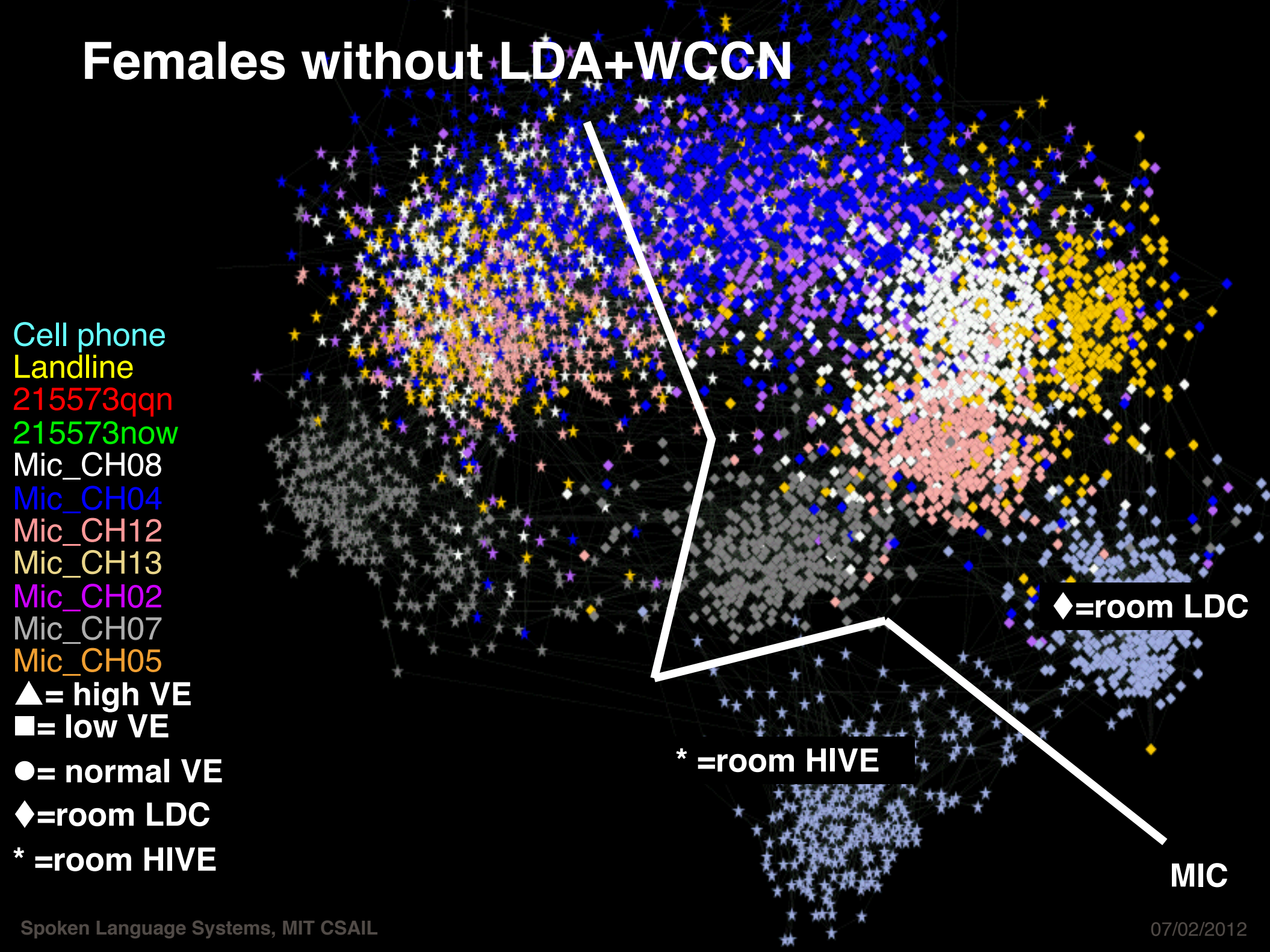
Females without LDA+WCCN



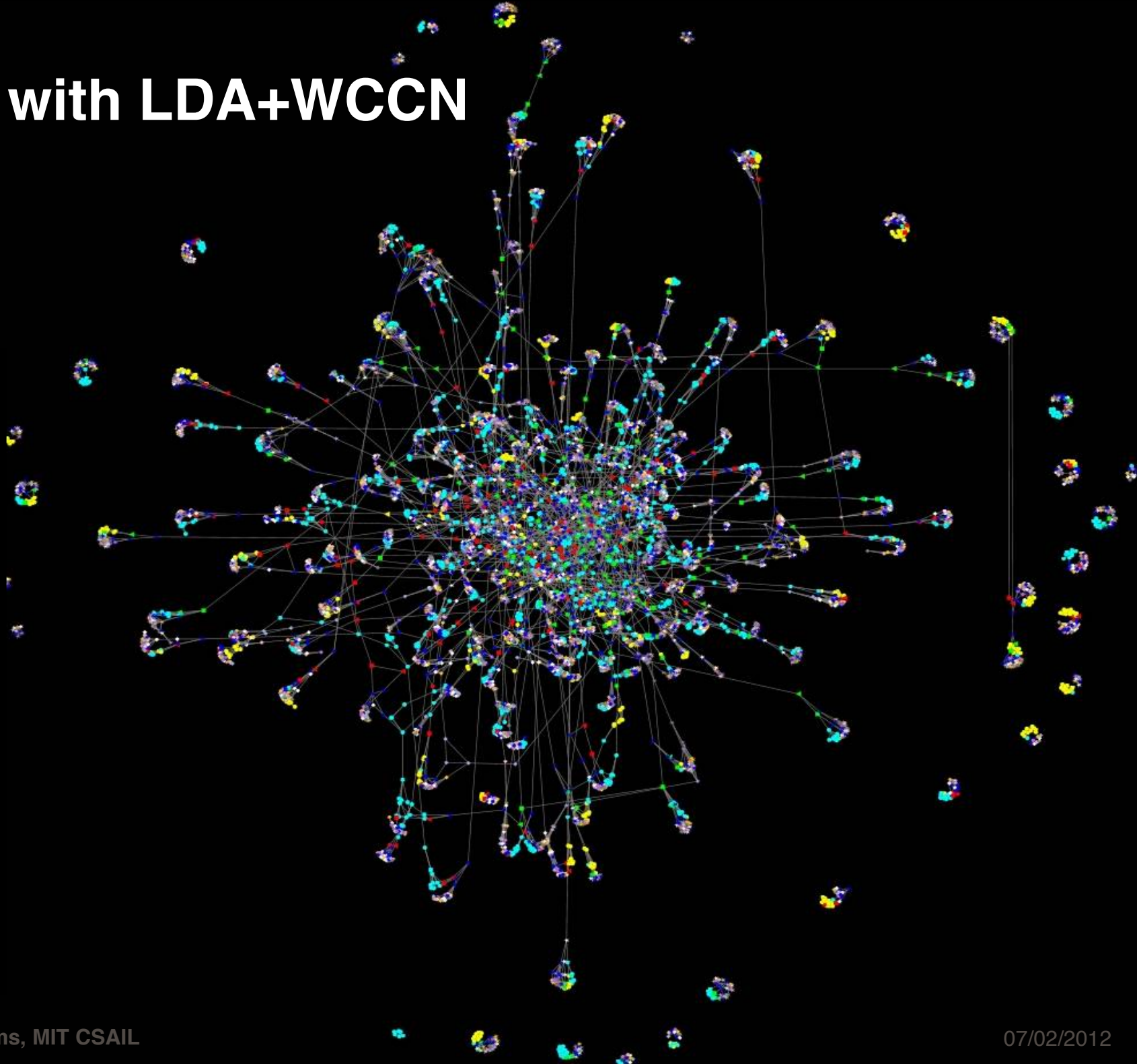
Females without LDA+WCCN



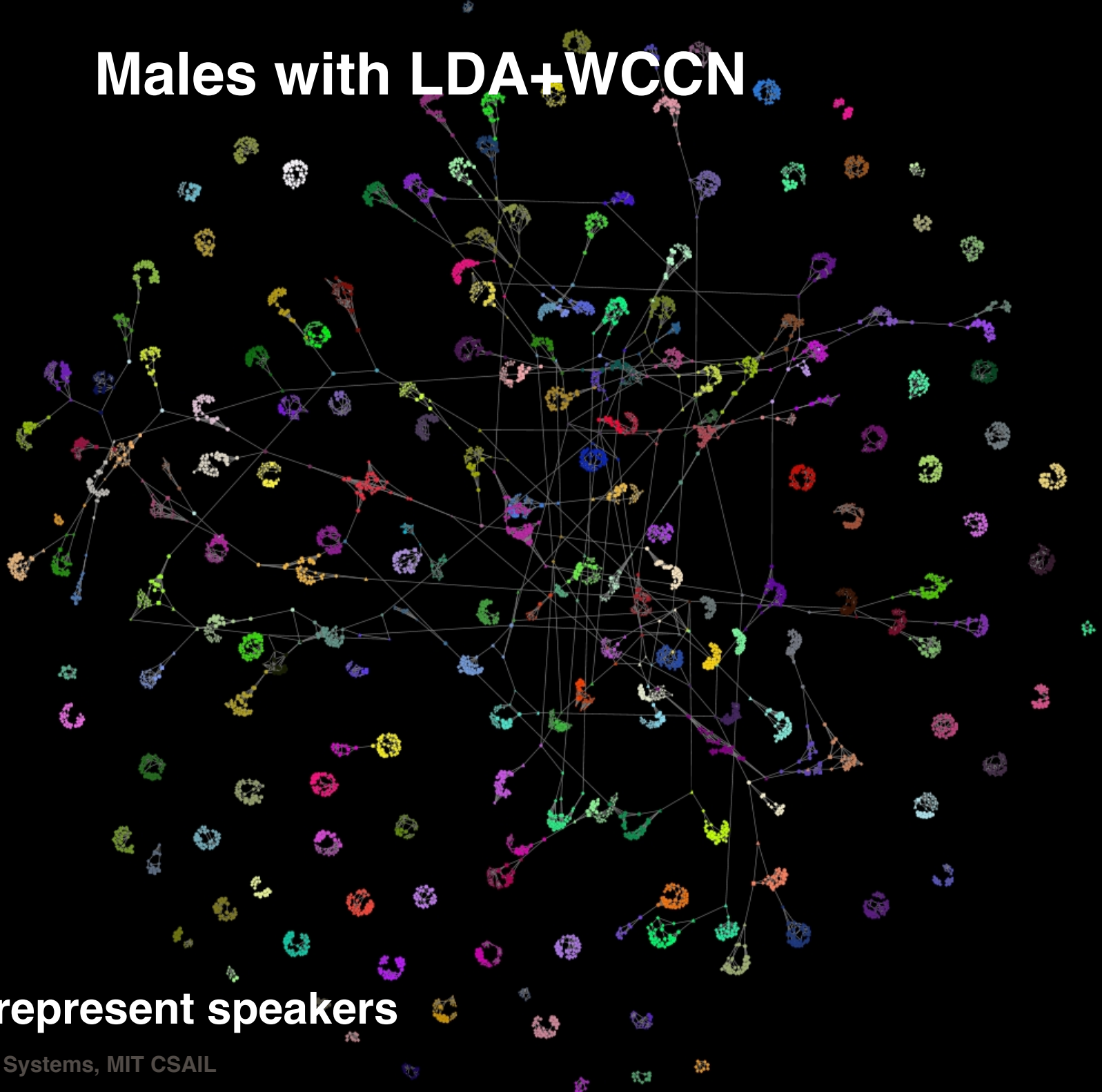
Females without LDA+WCCN



Females with LDA+WCCN

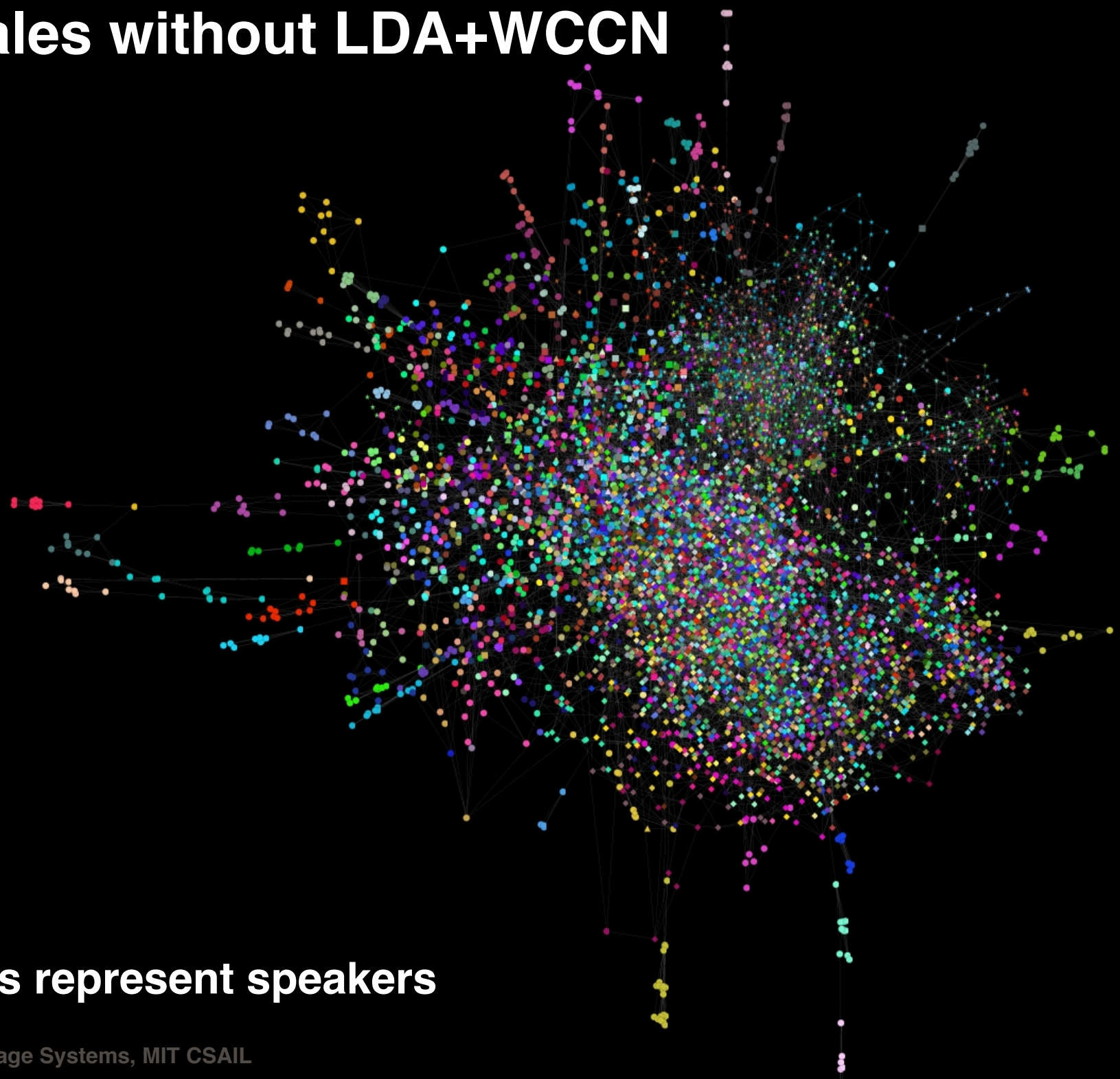


Males with LDA+WCCN



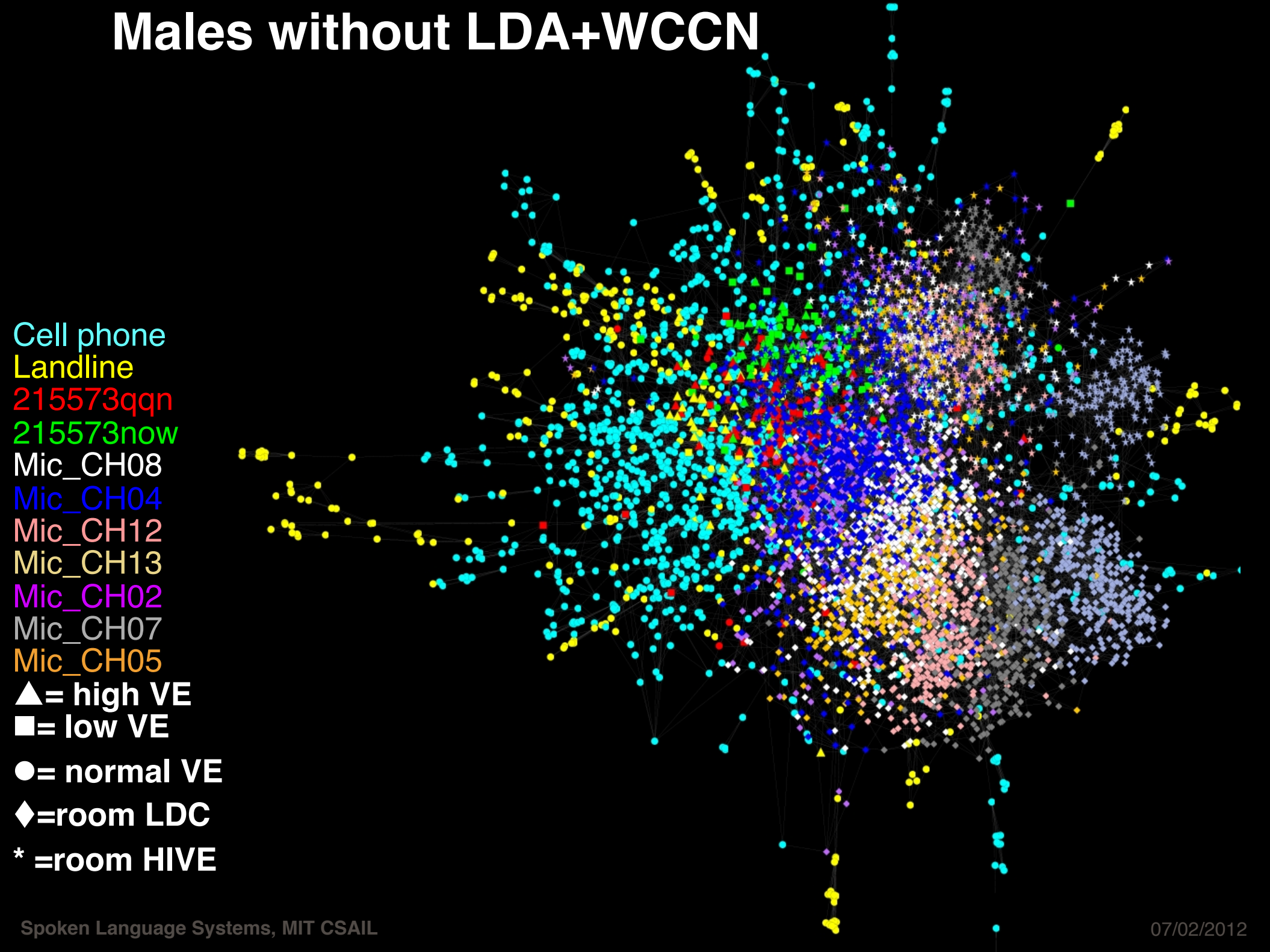
Colors represent speakers

Males without LDA+WCCN

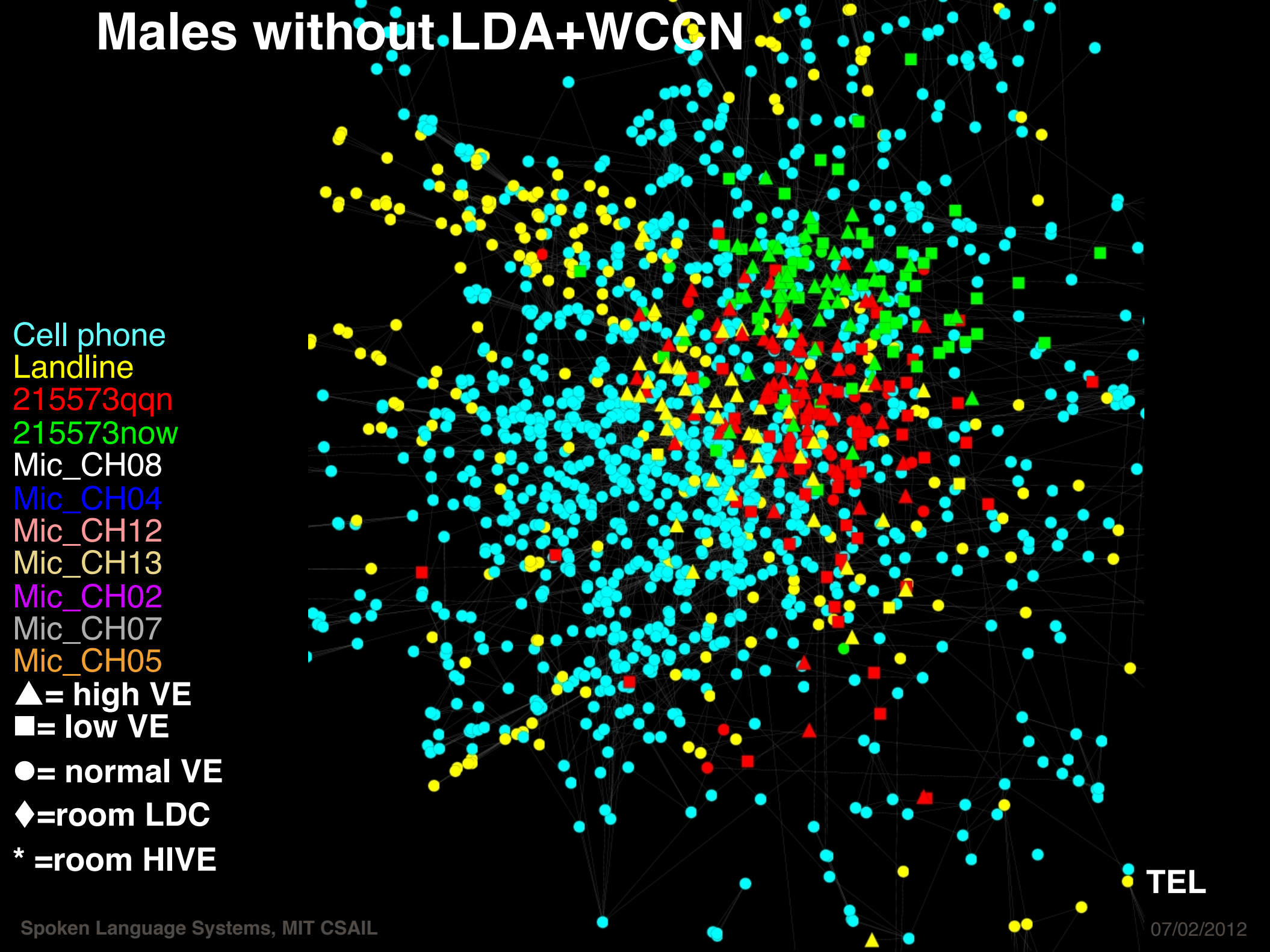


Colors represent speakers

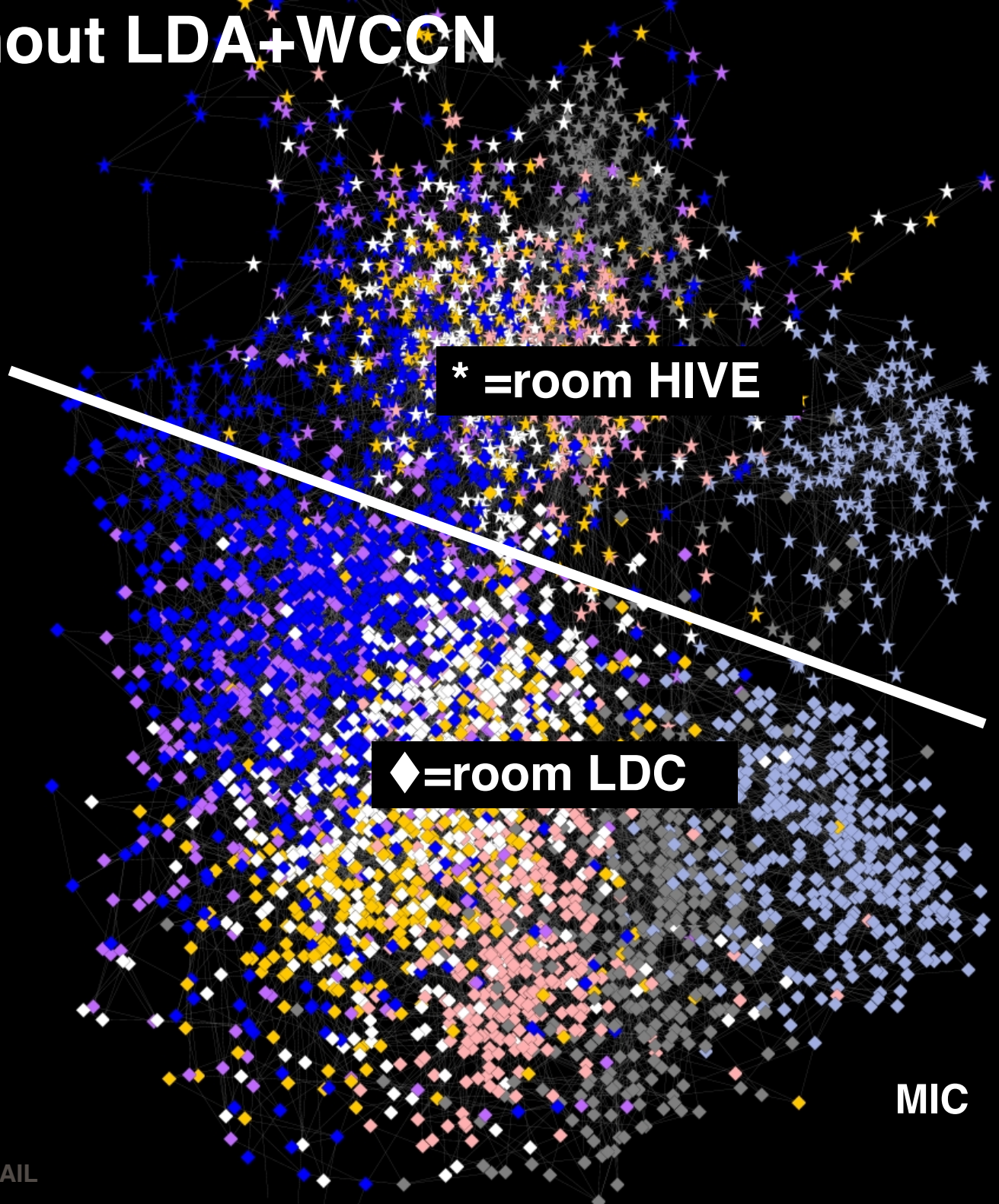
Males without LDA+WCCN



Males without LDA+WCCN



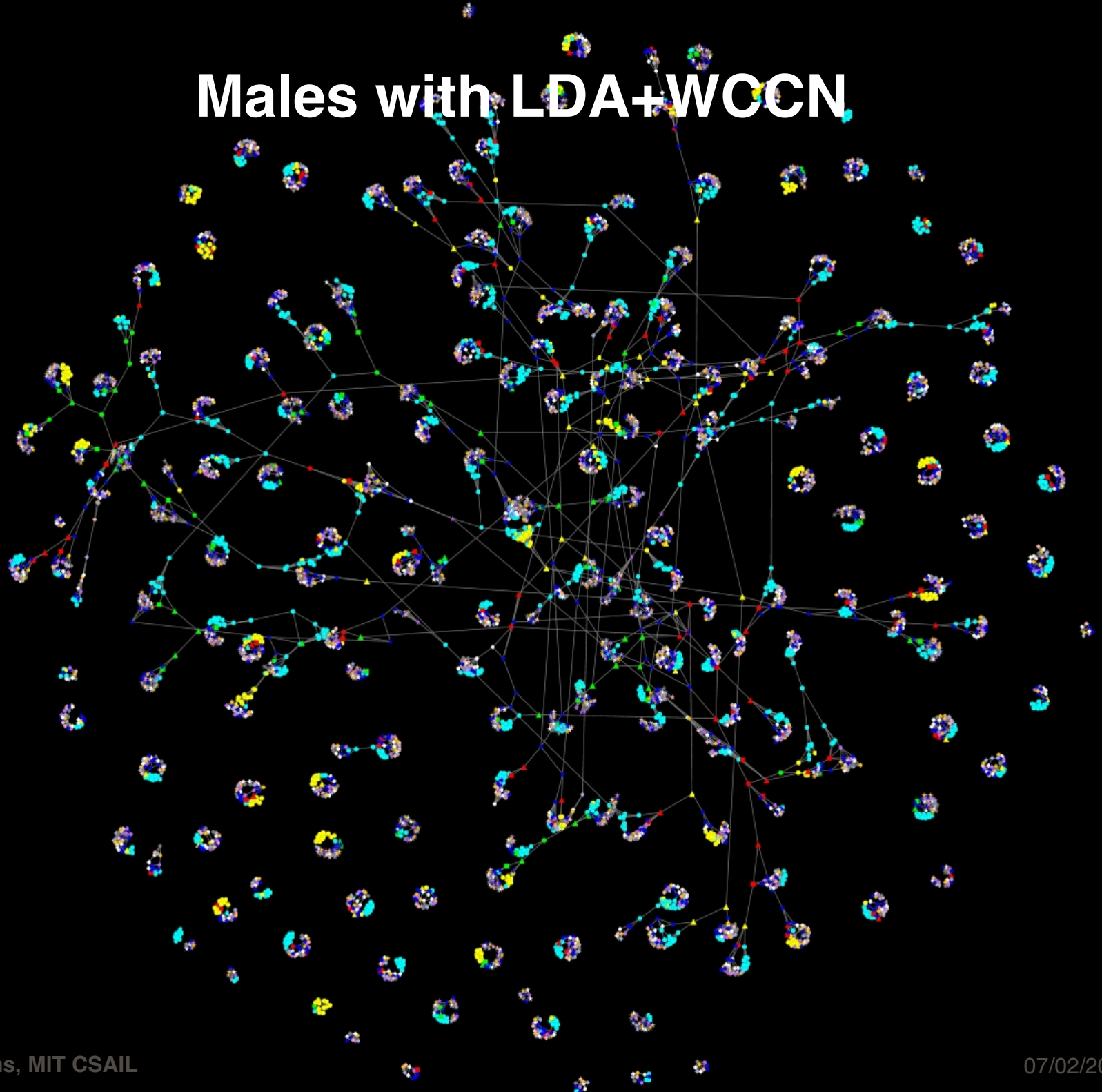
Males without LDA+WCCN



Cell phone
Landline
215573qqn
215573now
Mic_CH08
Mic_CH04
Mic_CH12
Mic_CH13
Mic_CH02
Mic_CH07
Mic_CH05

▲= high VE
■= low VE
●= normal VE
♦=room LDC
* =room HIVE

Males with LDA+WCCN



Cell phone
Landline
215573qqn
215573now
Mic_CH08
Mic_CH04
Mic_CH12
Mic_CH13
Mic_CH02
Mic_CH07
Mic_CH05
▲= high VE
■= low VE
●= normal VE
♦=room LDC
* =room HIVE

Conclusions

- **New powerful speaker representation:**
  - Low dimensional feature space (i-vectors)
  - Factor analysis as feature extractor

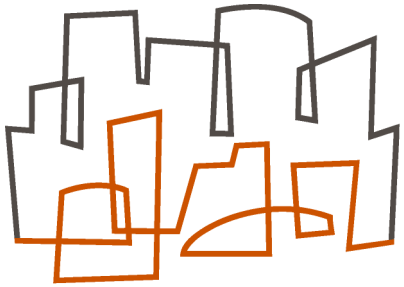
- **Since the i-vector space is a low dimensional**
  - Classical Linear Discriminant Analysis can be applied to maximize the discrimination between speakers
  - Graph analysis provides new data exploration techniques
  - Is there any non-linearity affect?

Roadmap

- Introduction
  - Terminology, tasks, and framework

- Low-Dimensional Representation
  - Speaker Verification
  - Sequence of features: GMM
  - Super-vectors: JFA
  - Low-dimensional vectors: i-vectors
  - Processing i-vectors: compensation and scoring
  - Data visualization

- **Other Applications**
  - Speaker diarization
  - Language recognition



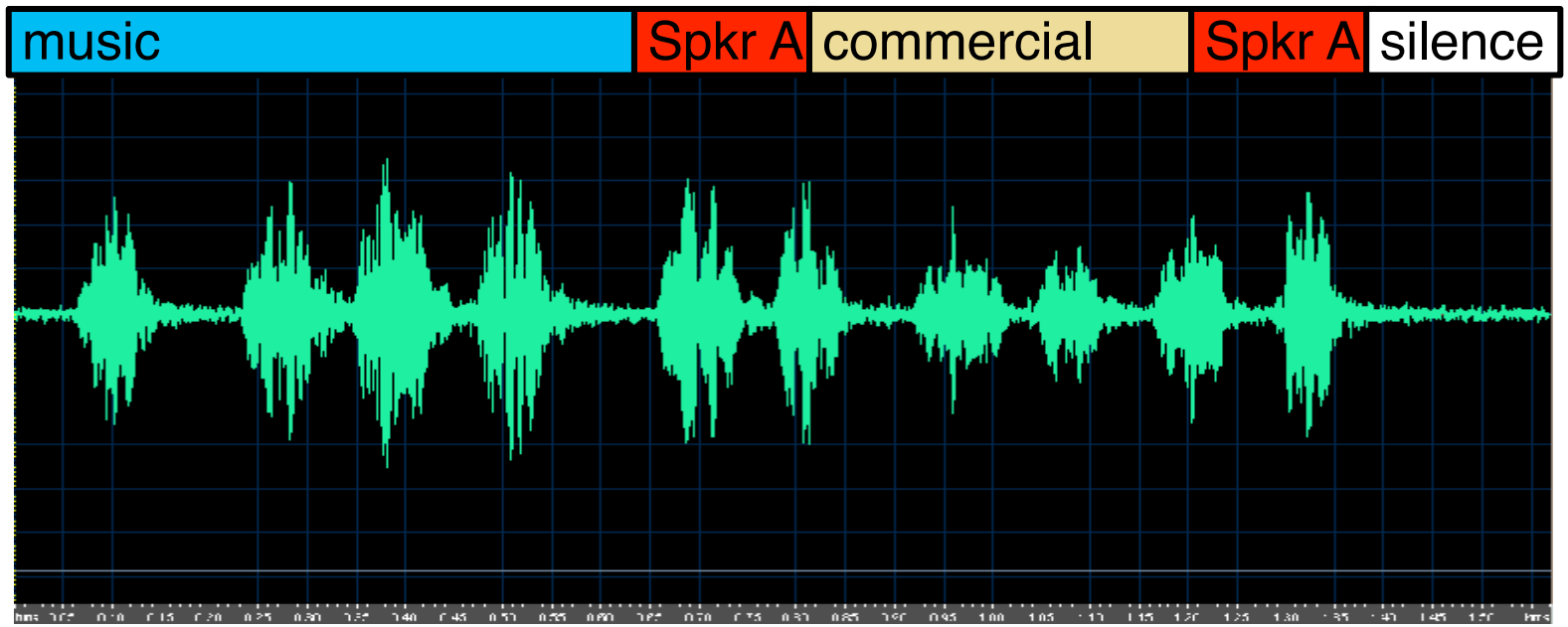
MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

Unsupervised Methods for Speaker Diarization:

An Integrated and Iterative Approach

Audio Diarization

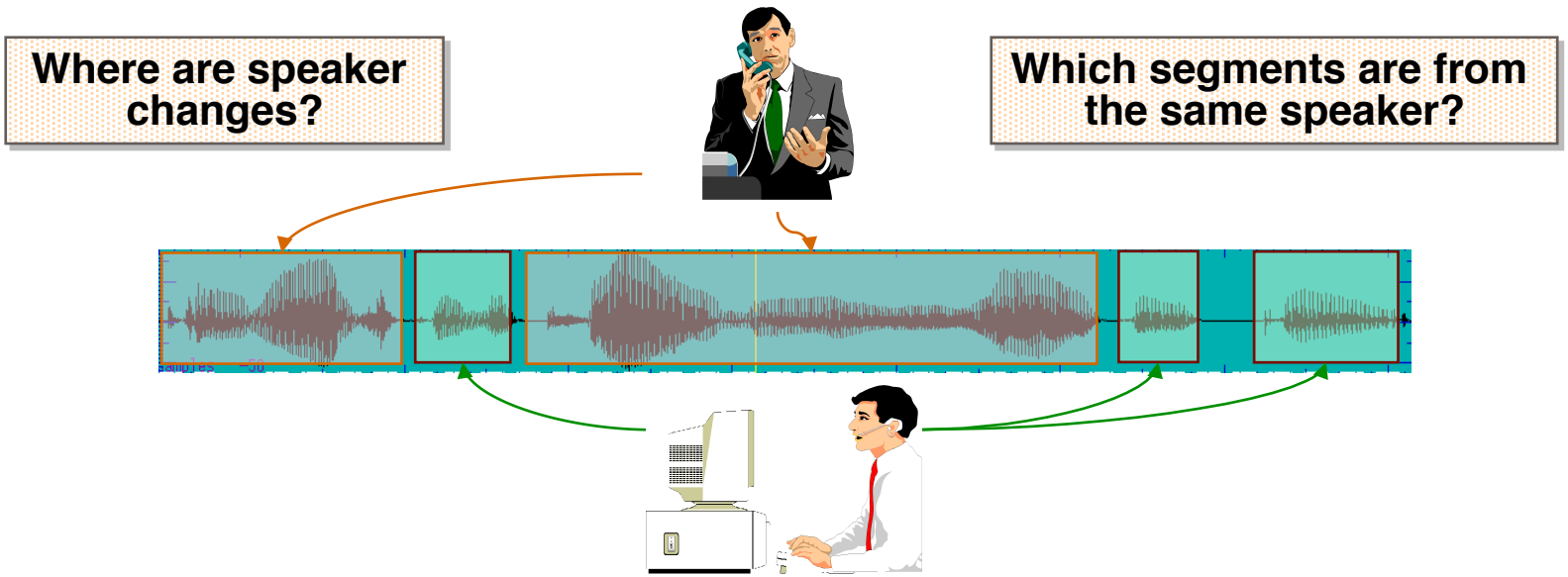
The task of marking and categorizing the different audio sources within an unmarked audio sequence.



Speaker Diarization

- **"Who is speaking when?"**

- **Segmentation**
  - Determine when speaker change has occurred in the speech signal
- **Clustering**
  - Group together speech segments from the same speaker



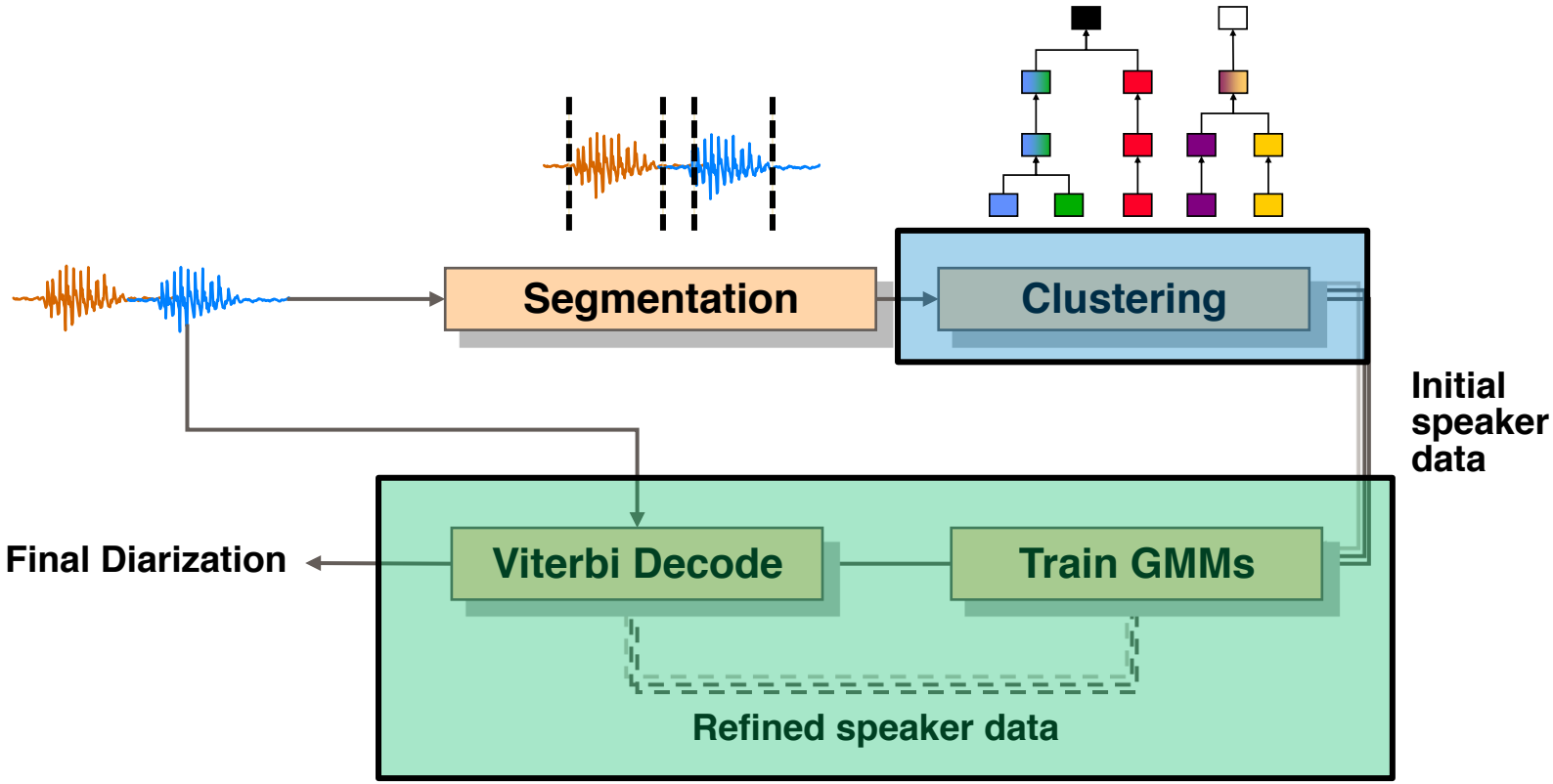
Summary of Contributions

- **Extended previous work in applying factor analysis-based speaker modeling to speaker diarization**
  - Interspeech 2011-2012, SM Thesis 2011
- **Integrated variational inference into speaker clustering**
  - SM Thesis 2011
- **Validated an iterative optimization procedure to refine clustering and segmentation hypotheses**
  - Interspeech 2012
- **Proposed a duration-proportional sampling scheme to combat issues of i-vector underrepresentation**
  - SM Thesis 2011

Roadmap

- **Introduction**
  - Summary of Contributions
- **Background**
  - Diarization System Overview
  - Speaker Modeling with Factor Analysis
- **Our Incremental Approach**
  - K-means and Spectral Clustering (Interspeech 2011, 2012)
  - Towards Probabilistic Clustering Methods
  - Iterative System Optimization (Re-segmentation/Clustering)
  - Duration-Proportional Sampling
- **Analysis and Discussion**
  - Benchmark Comparison (Castaldo 2008)
- **Conclusion**

Standard Diarization Setup



- **Agglomerative Hierarchical Clustering**
  - Requires methods for model selection
- **Iterative re-segmentation**

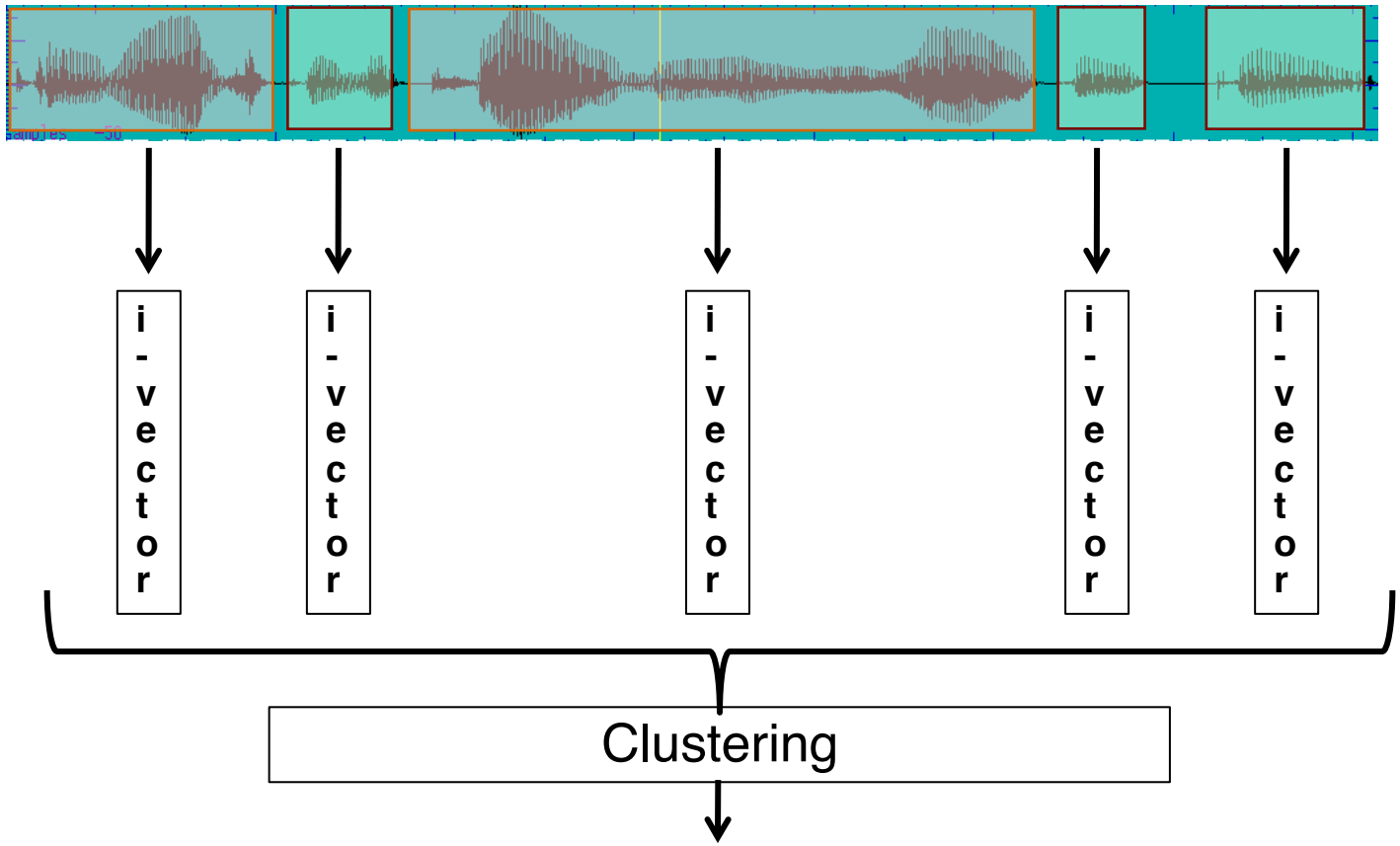
Recalling i-vectors

- **"For some speech segment s, its associated i-vector w_s can be seen as a low-dimensional summary of that segment's distribution of acoustic features with respect to a UBM."**
- **Low-dimensional random vector (100 << 20,000)**
  - Standard normal prior distribution, $\mathcal{N}(0, I)$
- **Given some speech data,**
  - Posterior mean → i-vector
  - Posterior covariance → i-vector covariance
- **Cosine similarity metric**
  - Can also length-normalize i-vectors onto the unit hypersphere

Roadmap

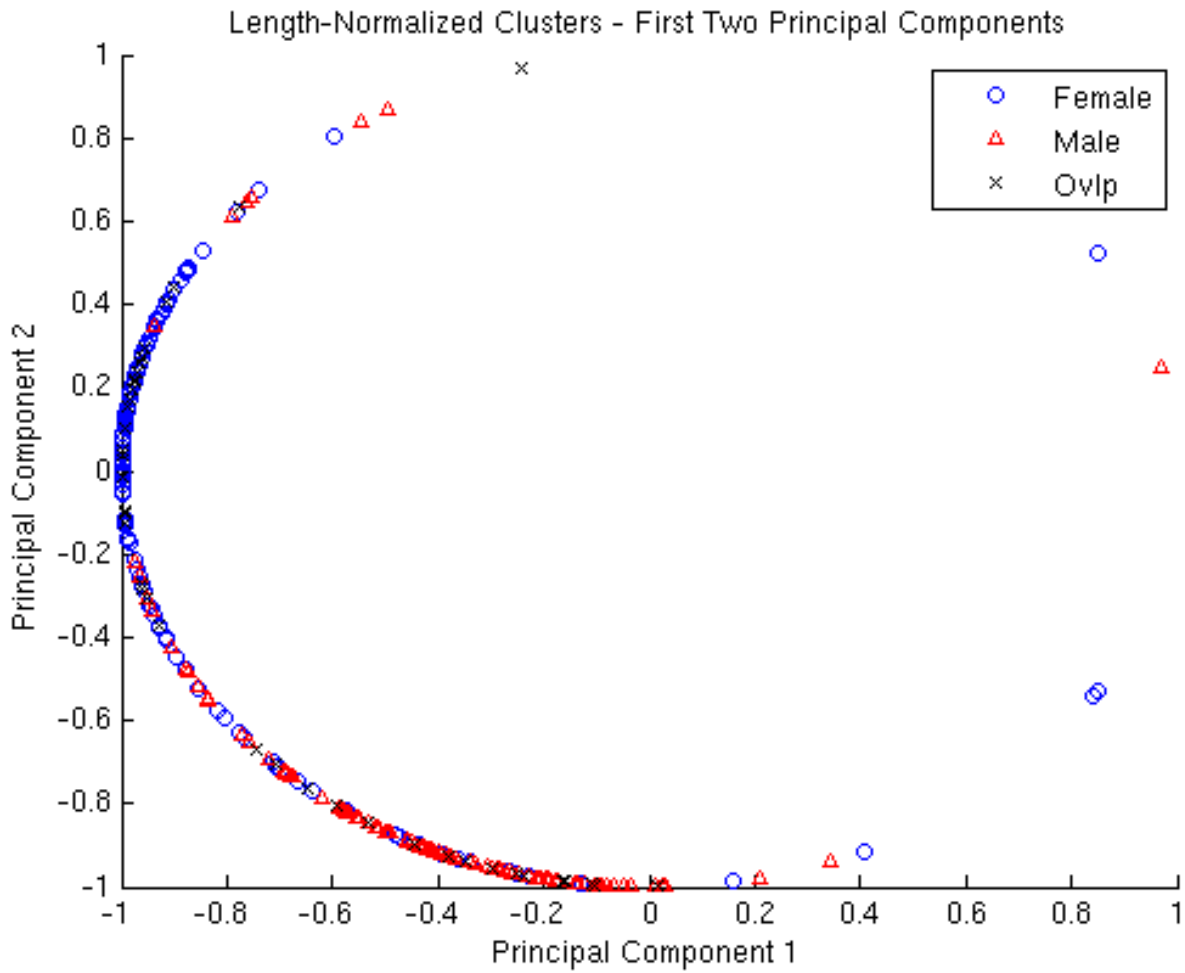
- Introduction
  - Summary of Contributions
- Background
  - Diarization System Overview
  - Speaker Modeling with Factor Analysis
- **Our Incremental Approach**
  - K-means and Spectral Clustering (Interspeech 2011, 2012)
  - Towards Probabilistic Clustering Methods
  - Iterative System Optimization (Re-segmentation/Clustering)
  - Duration-Proportional Sampling
- Analysis and Discussion
  - Benchmark Comparison (Castaldo 2008)
- Conclusion

Initialization



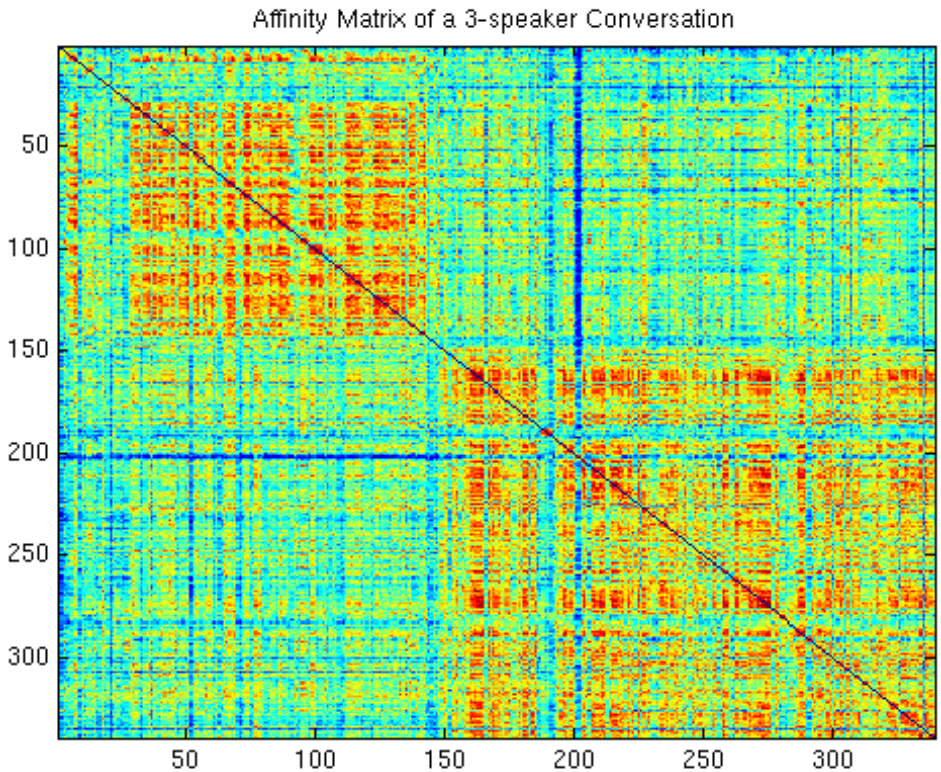
Clustering History

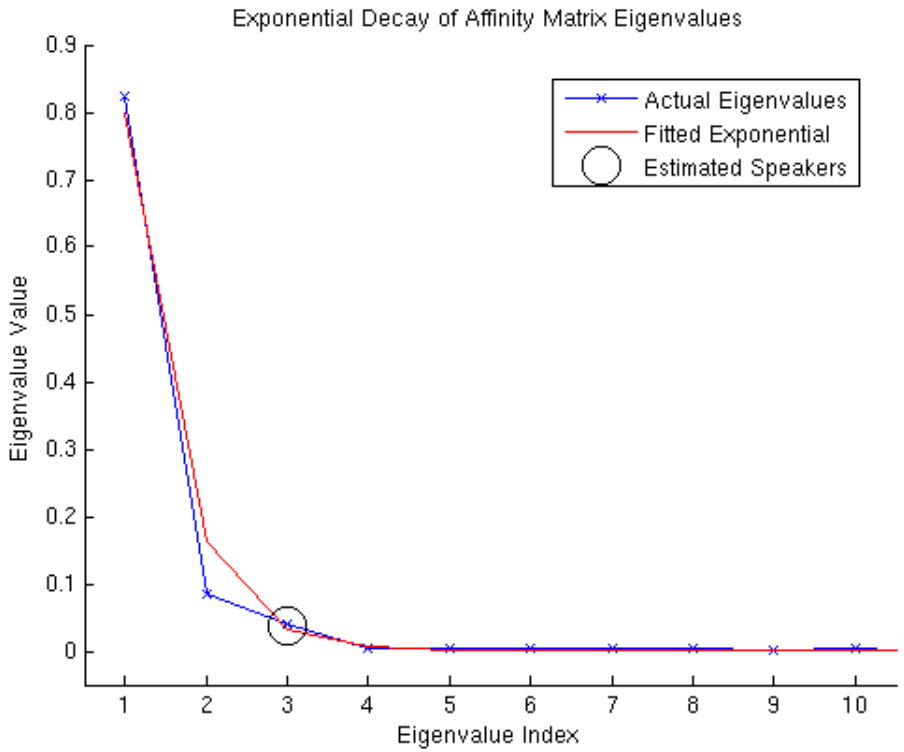
- **K-means on 2-speaker conversations (K = 2 known)**
  - Interspeech 2011, SM Thesis 2011



Clustering History

- **K-means on 2-speaker conversations (K = 2 known)**
  - Interspeech 2011, SM Thesis 2011
- **K-means and Spectral Clustering on K-speaker telephone conversations (K both known and unknown)**
  - Interspeech 2012





Clustering History

- **K-means on 2-speaker conversations (K = 2 known)**
  - Interspeech 2011, SM Thesis 2011
- **K-means and Spectral Clustering on K-speaker telephone conversations (K both known and unknown)**
  - Interspeech 2012

- **Probabilistic Methods (SM Thesis 2011)**
  - K-means → Gaussian Mixture Models
    - * **Bayesian model selection via variational inference**

The Need for Approximate Inference

- Consider some data Y, hidden variable set X, parameters θ
- For model selection m, we want to <u>maximize</u>

$$\log P(Y|m) = \log \int P(Y,X,\theta|m)\, dX\, d\theta$$

→ exact computation is intractable in general

- Introduce $q(X,\theta)=q(X)\cdot q(\theta)$ to approximate $P(X,\theta|Y,m)$

$$\log P(Y|m) = F(m)(q(X,\theta)) + KL(q(X,\theta)||P(X,\theta|Y,m))$$

* Maximizing the Free Energy minimizes the KL-divergence between the variational posterior and the true posterior distributions

Variational Free Energy

- $F_{\downarrow m}(q(X)q(\theta)) = \int q(X)q(\theta) \cdot \log P_{Y,X\theta,m}\, dX\, d\theta$

Expectation, under $q(X,\theta)$, of complete data log-likelihood

$+ H_{\downarrow q}(X) - KL(q(\theta)||P_{\theta m})$

Entropy of X

KL-divergence between variational parameters and actual priors

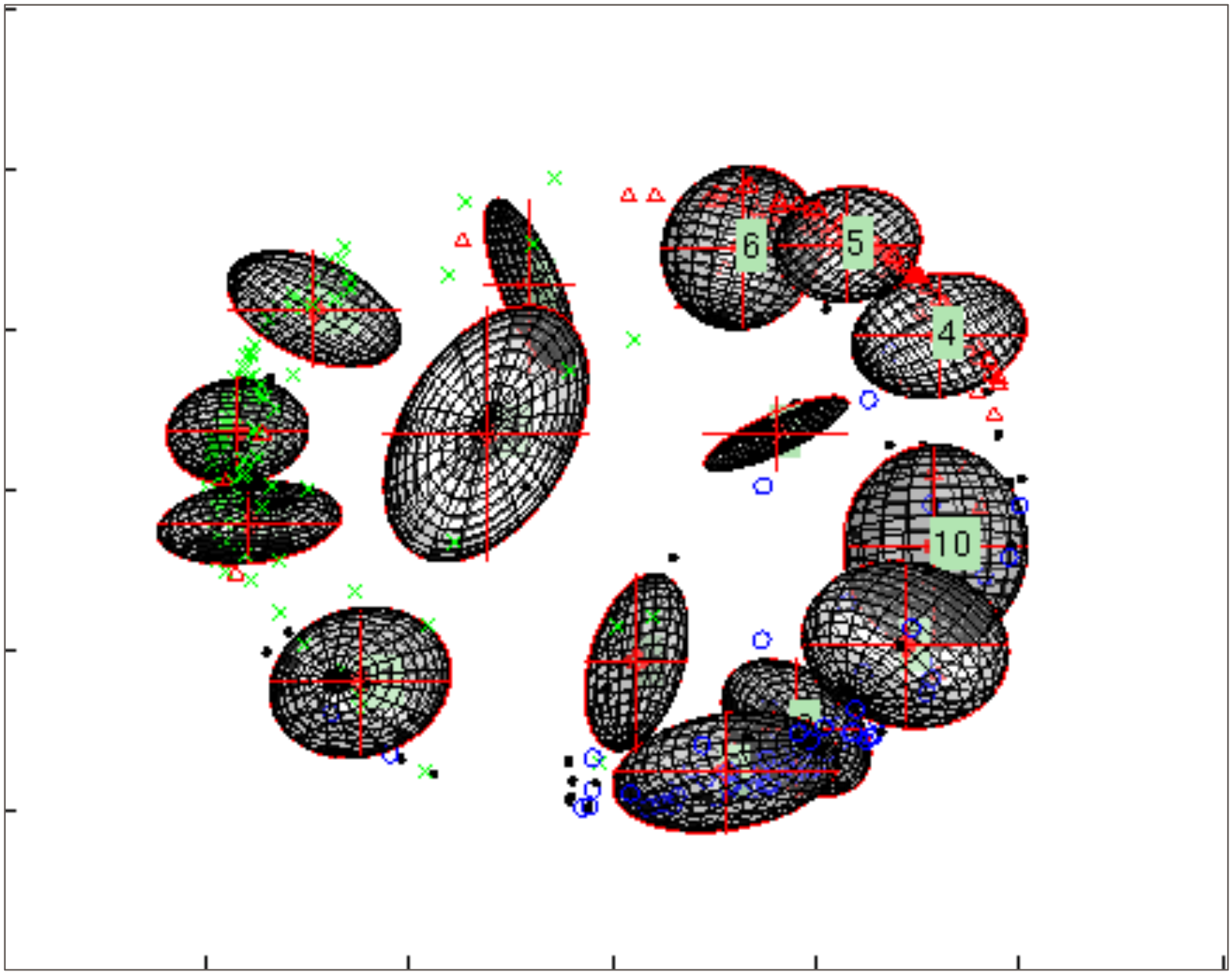
- The act of maximizing $F_{\downarrow m}(q(X)q(\theta))$ yields an EM Algorithm
  - VBEM-GMM

Clustering History

- **K-means on 2-speaker telephone conversations (K known)**
  - Interspeech 2011, SM Thesis 2011
- **K-means and Spectral Clustering on K-speaker telephone conversations (K both known and unknown)**
  - Interspeech 2012

- **Probabilistic Methods (SM Thesis 2011)**
  - K-means → Gaussian Mixture Models
    - * **Bayesian model selection via variational inference**
  - Rote application of VBEM-GMM

VBEM-GMM Visualization



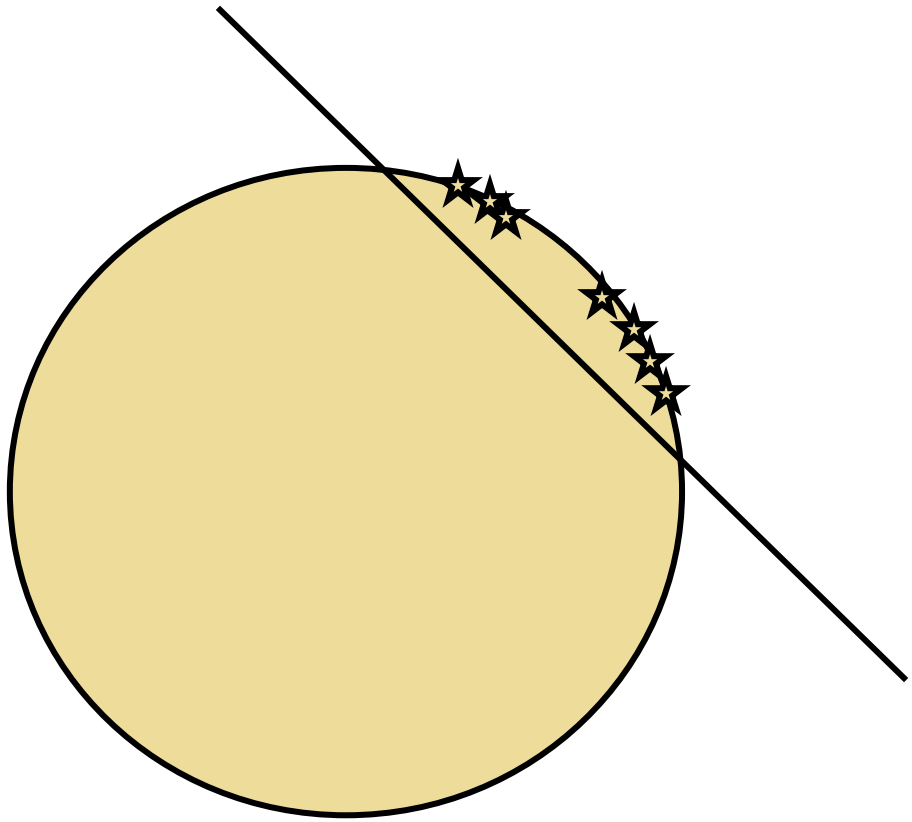
Clustering History

- **K-means on 2-speaker telephone conversations (K known)**
  - Interspeech 2011, SM Thesis 2011
- **K-means and Spectral Clustering on K-speaker telephone conversations (K both known and unknown)**
  - Interspeech 2012

- **Probabilistic Methods (SM Thesis 2011)**
  - K-means → Gaussian Mixture Models
    - **Bayesian model selection via the variational approximation**
  - Rote application of VBEM-GMM
    - **GMMs are a poor way to model data living on a unit hypersphere.**

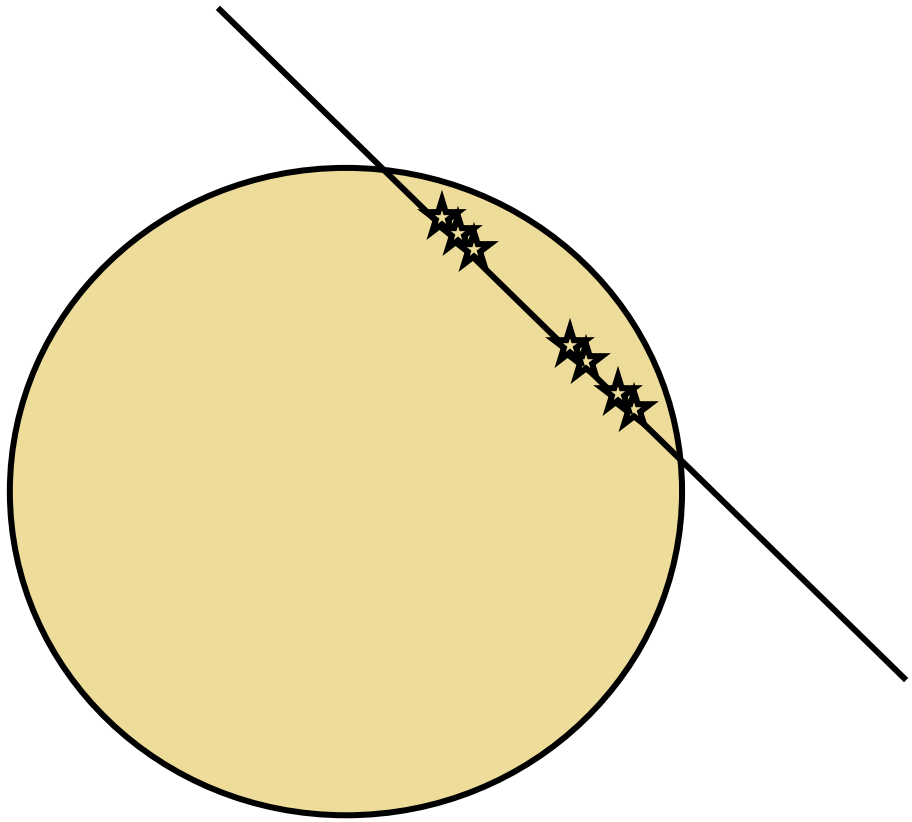
Dimensionality Reduction

- **i-vectors are both speaker- and channel-dependent**
  - Channel effect localizes all i-vectors onto one small region on the unit hypersphere
  - Consider a projection (PCA) onto a lower-dimensional plane

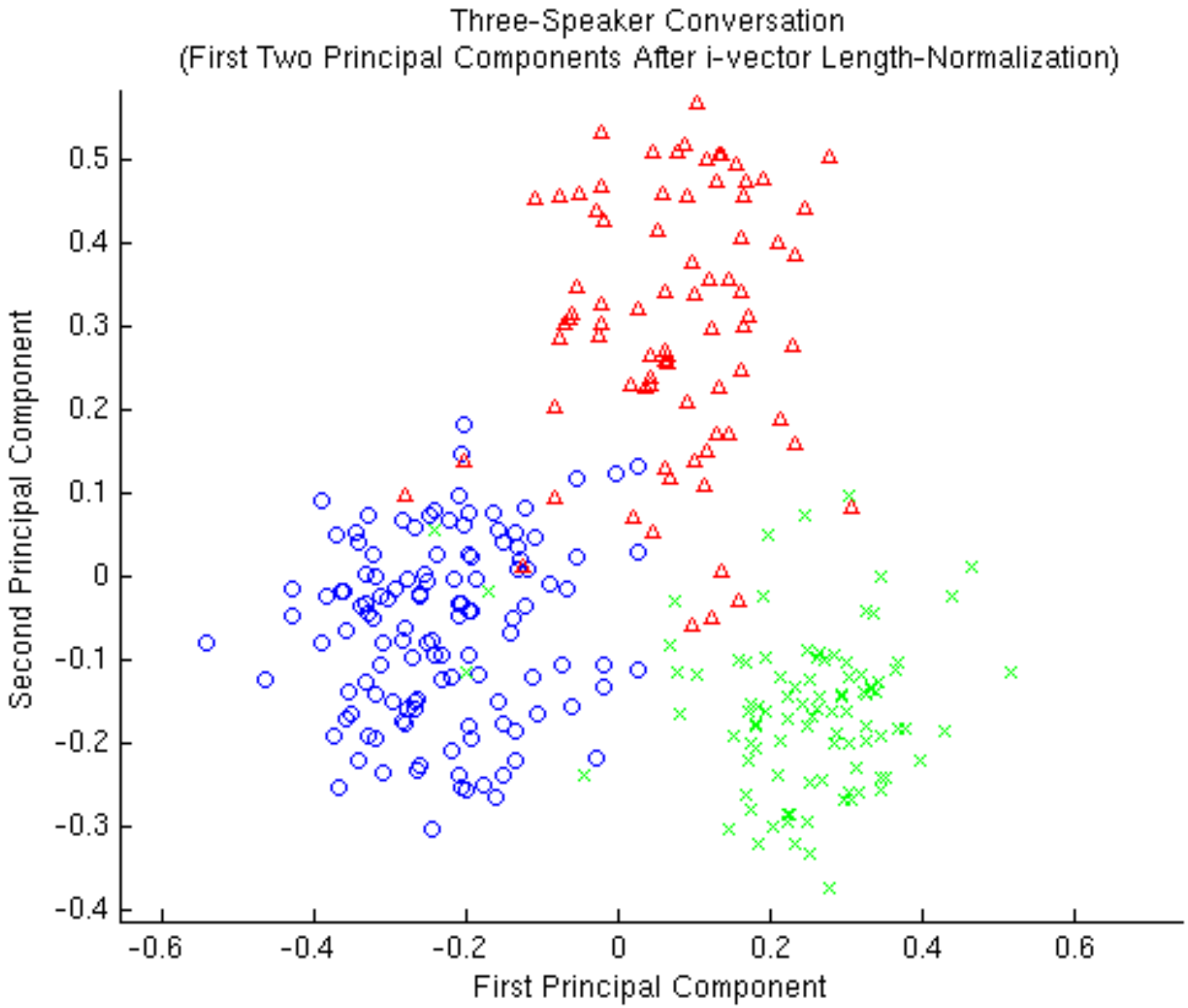


Dimensionality Reduction

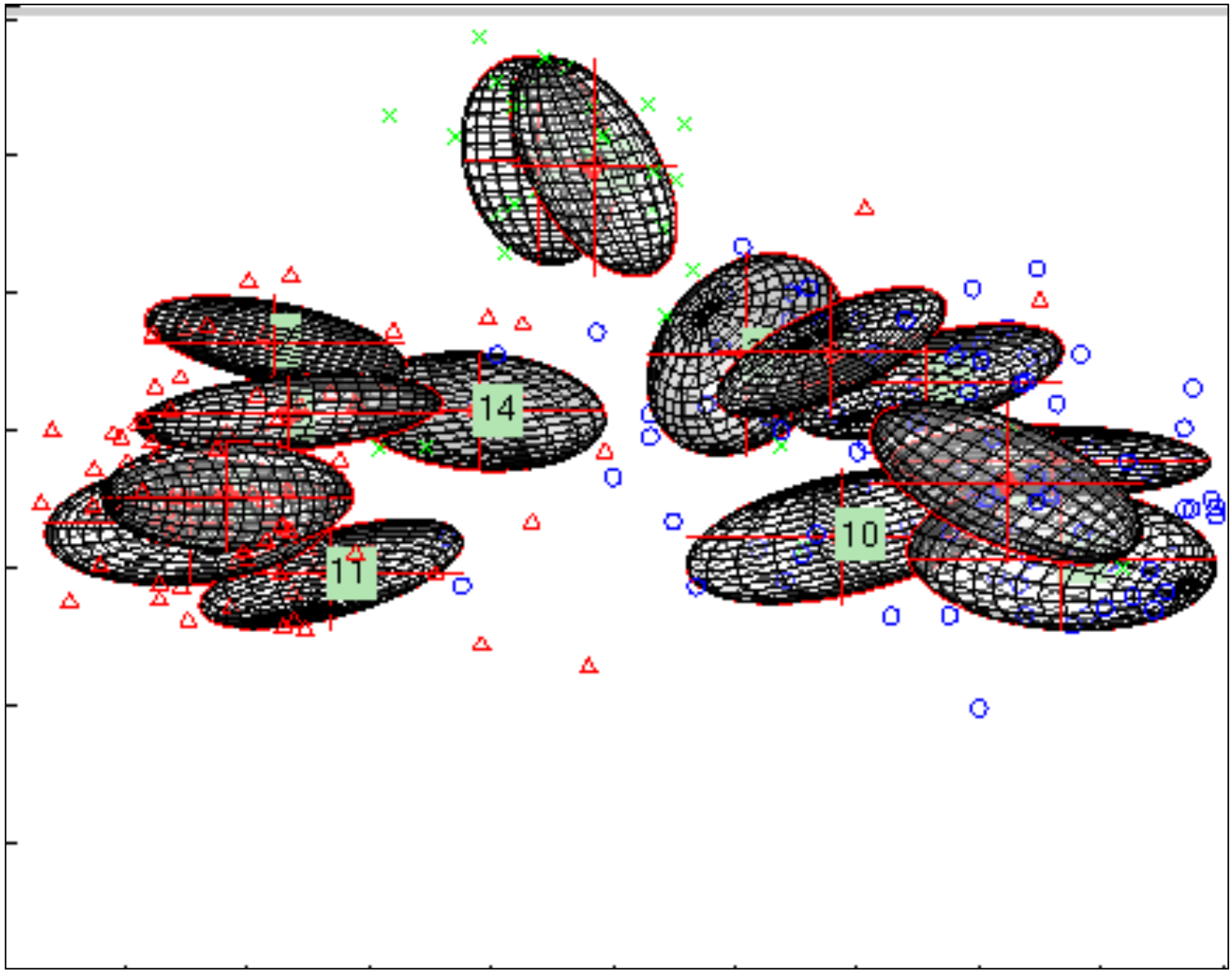
- **i-vectors are both speaker- and channel-dependent**
  - Channel effect localizes all i-vectors onto one small region on the unit hypersphere
  - Consider a projection (PCA) onto a lower-dimensional plane



PCA Visualization



VBEM-GMM Clustering (after PCA)

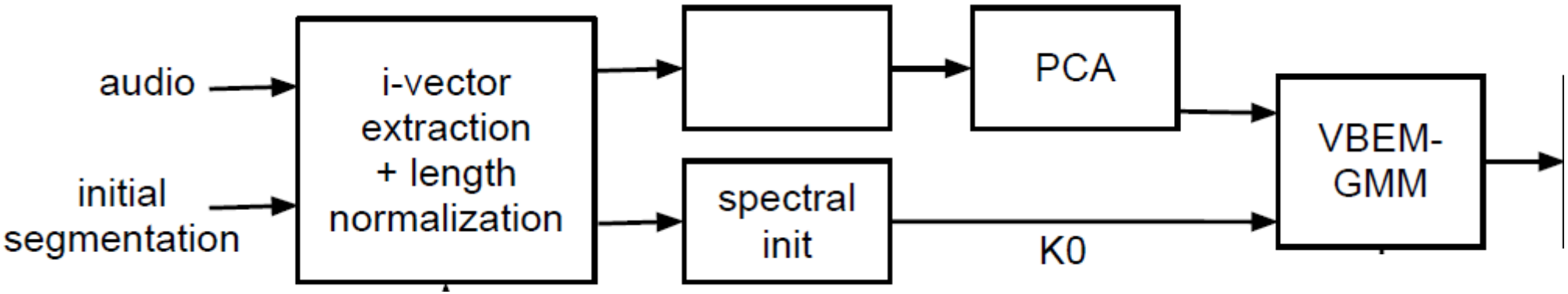


Cluster Initialization

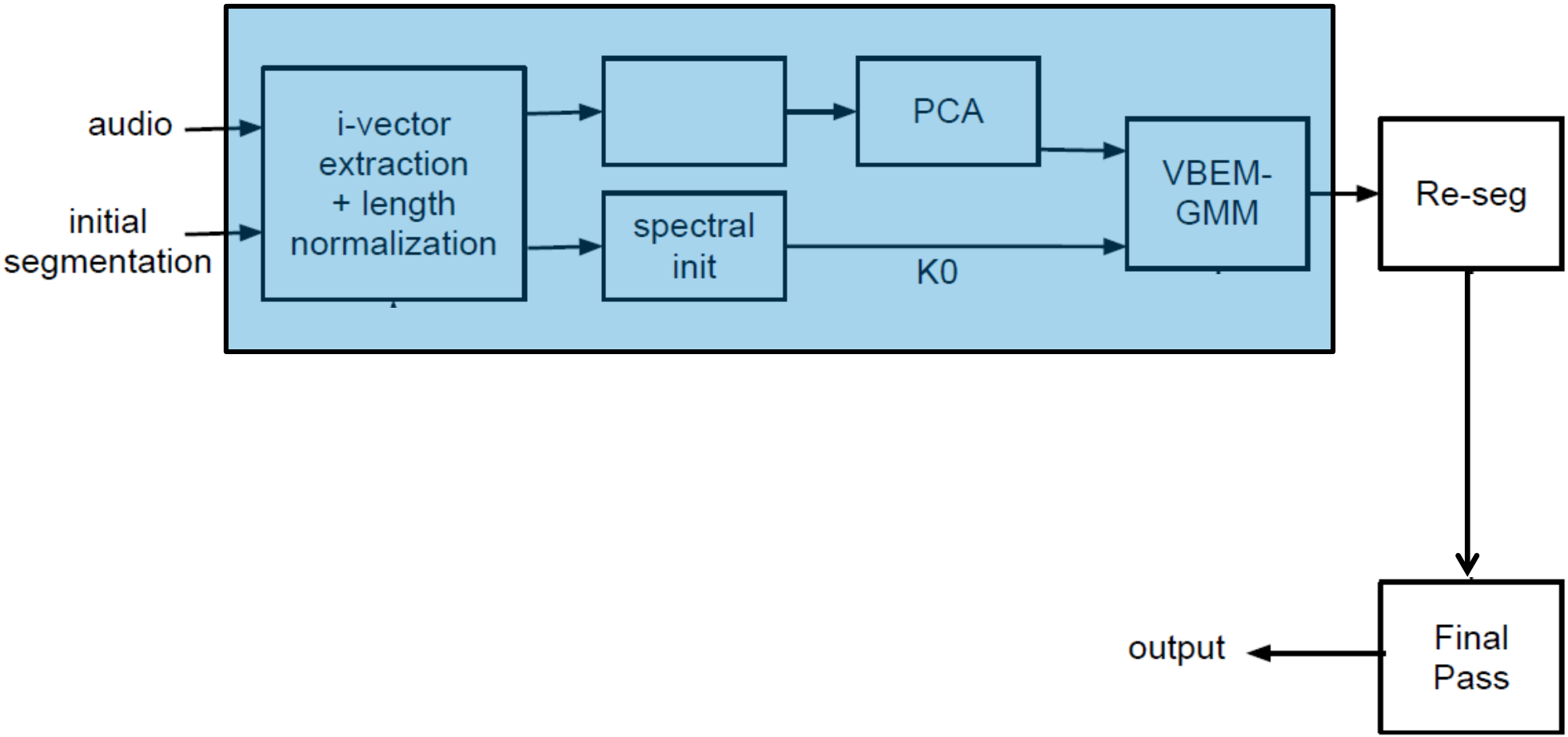
- **Baseline Approach**
  - Over-initialize the number of clusters
    - $K_0 = 15$
  - Remove components iteratively

- **Proposed Refinement**
  - Initialize using eigenvalue roll-off from the affinity matrix generated by the spectral clustering algorithm.
    - $K_0 = K + \lceil 3\sigma_K \rceil$
  - Still want to over-initialize clusters, but in a more informed manner

System Diagram (Clustering)



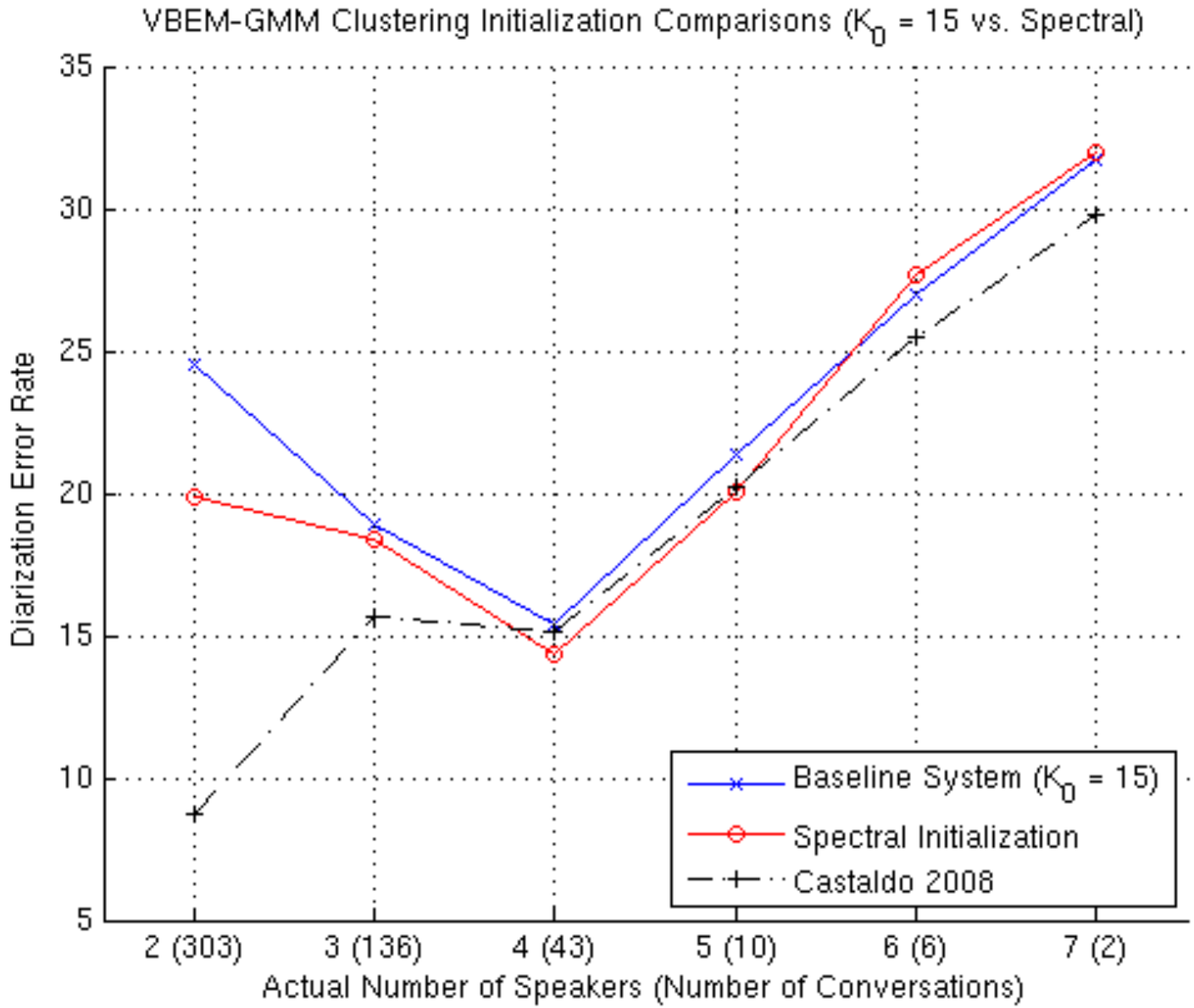
System Diagram (Baseline)



Experiment Details

- **Evaluation Data**
  - Multi-lingual CallHome corpus
    - * **500 recordings, 2-5 minutes each, containing 2-7 speakers**
- **Total Variability**
  - 20-dimensional MFCC acoustic feature vectors
  - UBM of 1024 Gaussians
  - Rank of Total Variability matrix = 100
    - * **i.e. 100-dimensional i-vectors**
- **Diarization Error Rate (DER)**
  - Amount of time spent confusing one speaker's speech as from another

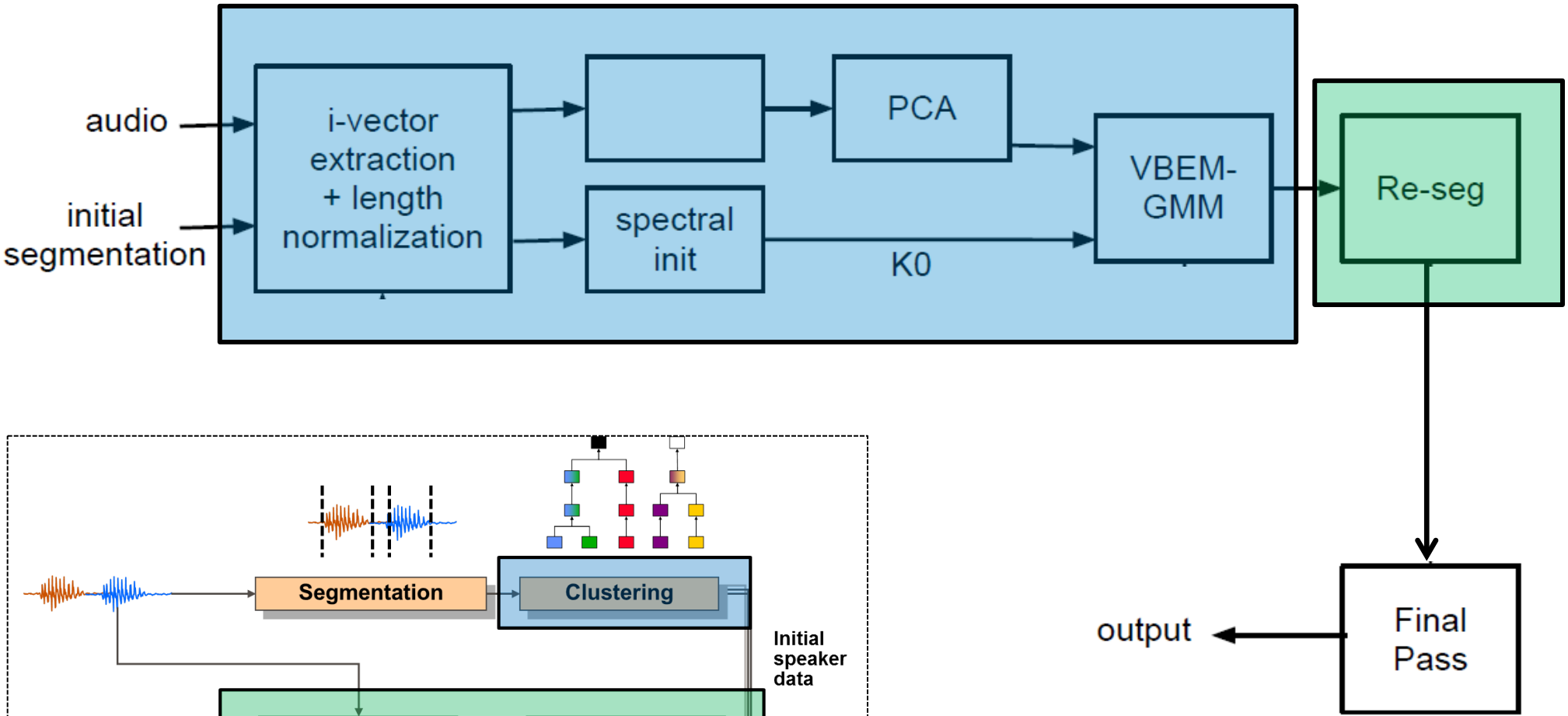
Initial Results



Roadmap

- Introduction
  - Summary of Contributions
- Background
  - Diarization System Overview
  - Speaker Modeling with Factor Analysis
- **Our Incremental Approach**
  - K-means and Spectral Clustering (Interspeech 2011, 2012)
  - Towards Probabilistic Clustering Methods
  - Iterative System Optimization (Re-segmentation/Clustering)
  - Duration-Proportional Sampling
- Analysis and Discussion
  - Benchmark Comparison (Castaldo 2008)
- Conclusion

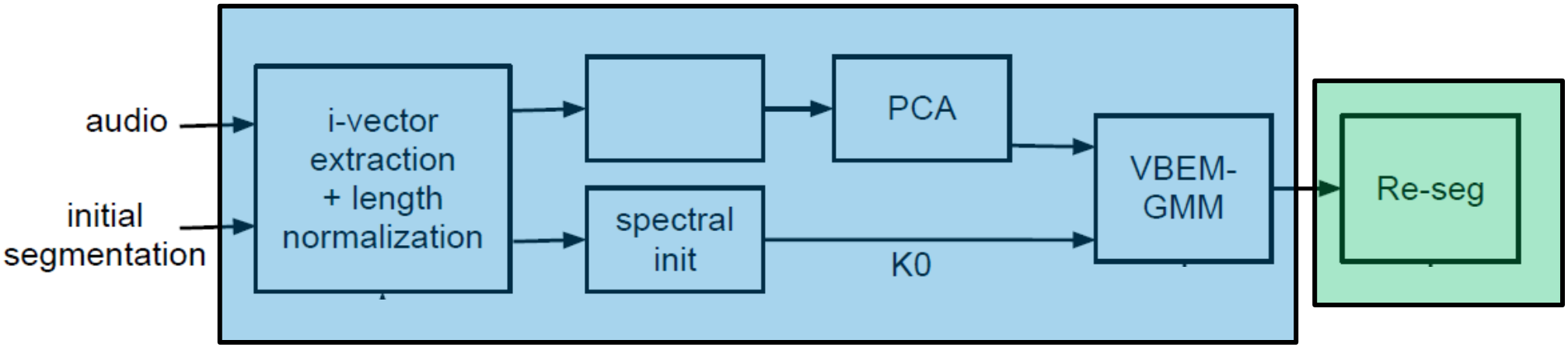
System Diagram (Baseline)



Iterative Re-segmentation

- Initialize a GMM for each cluster.
  - **Speaker 1, Speaker 2, …, Non-speech N**
- Obtain a posterior probability for each cluster given each feature vector.
  - $P(S_1|x_t)$, $P(S_2|x_t)$, …, $P(N|x_t)$
- Pool these probabilities across the entire conversation ($t = 1,\ldots,T$) and use them to re-estimate each respective speaker's GMM.
  - **The Non-speech GMM is never re-trained.**
- The Viterbi algorithm re-assigns each frame to the speaker/non-speech model with highest posterior probability.

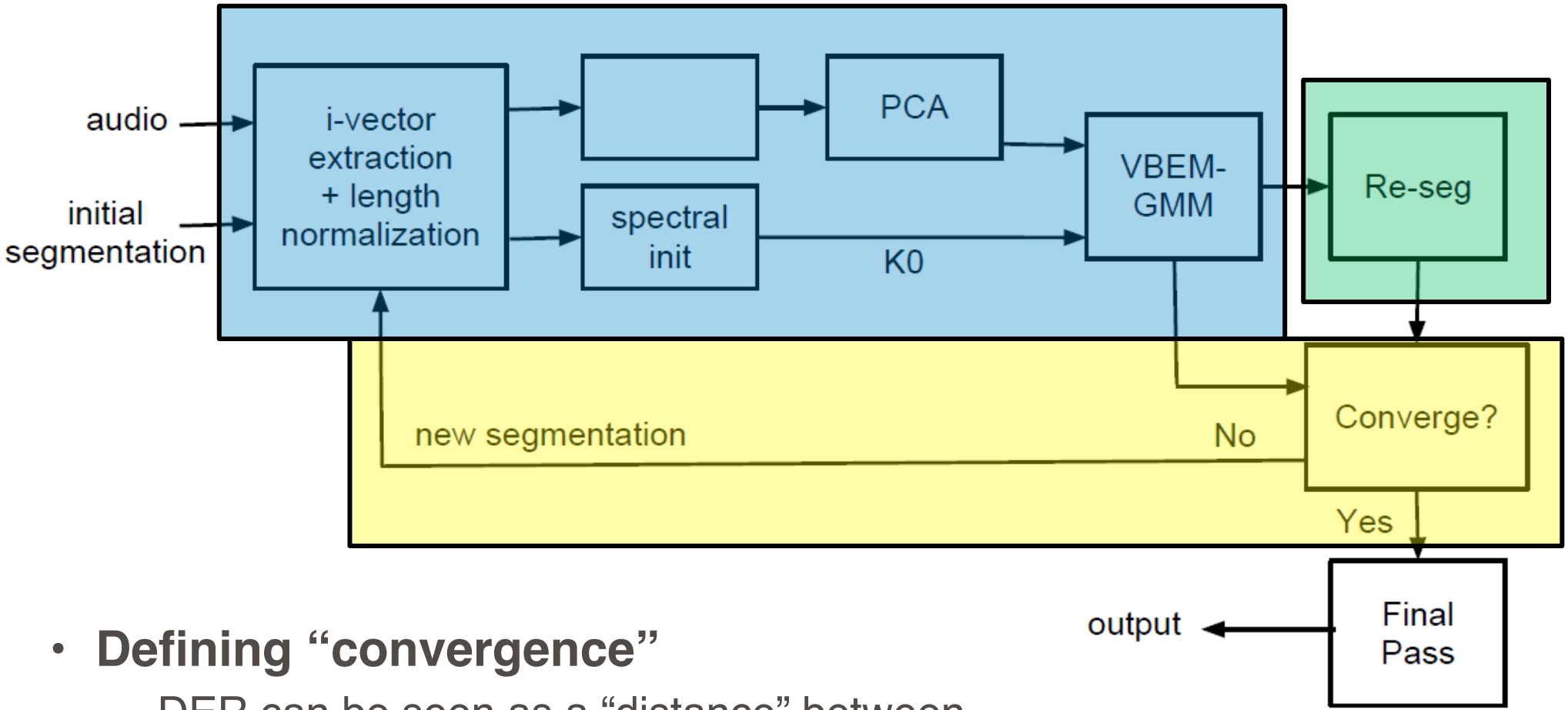
A Symbiotic Relationship



- **Clustering assumes some initial segmentation and clusters at the i-vector level**
  - Better speaker representation

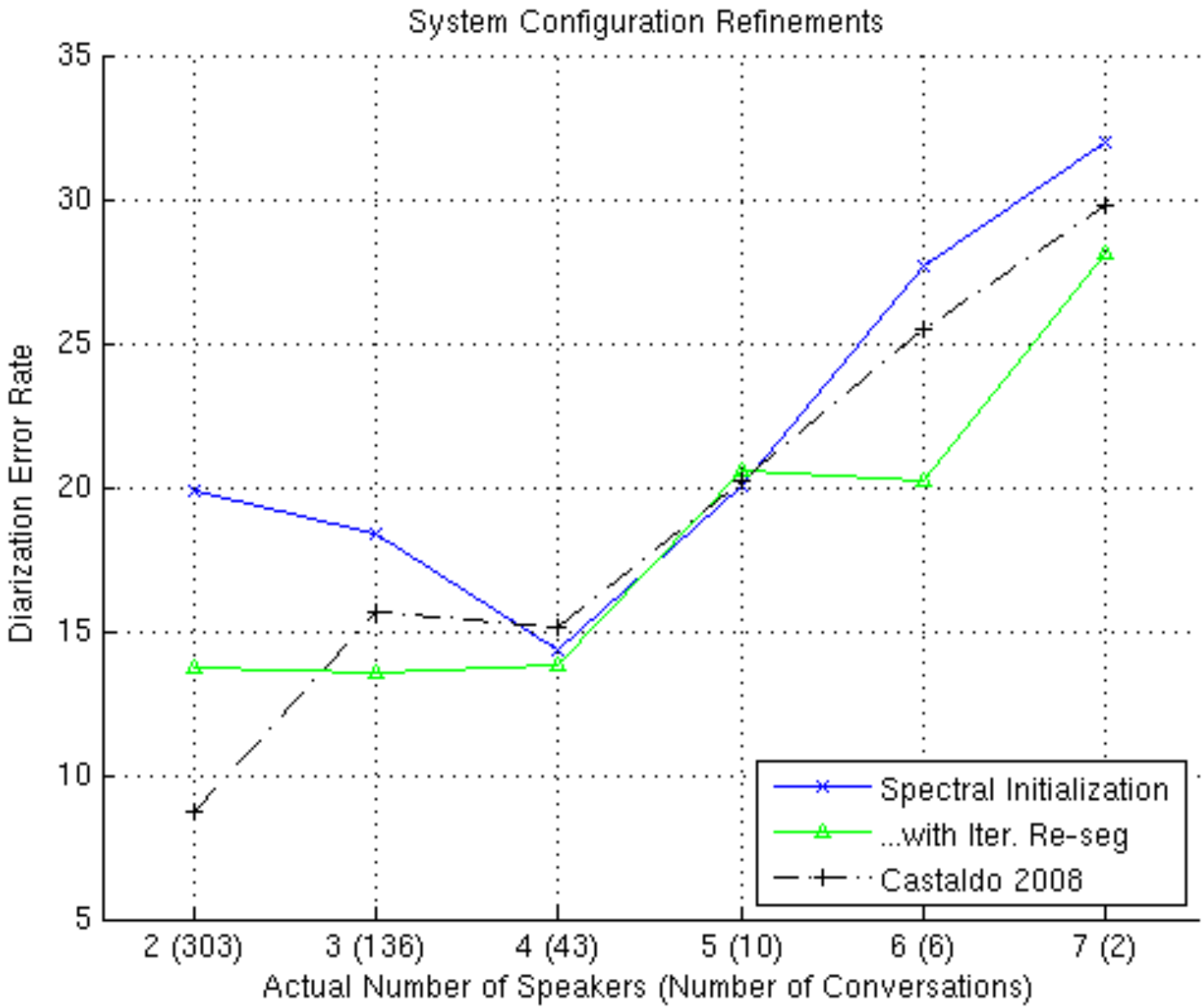
- **Re-segmentation operates at level of acoustic features**
  - Finer temporal resolution

Iterative System Optimization

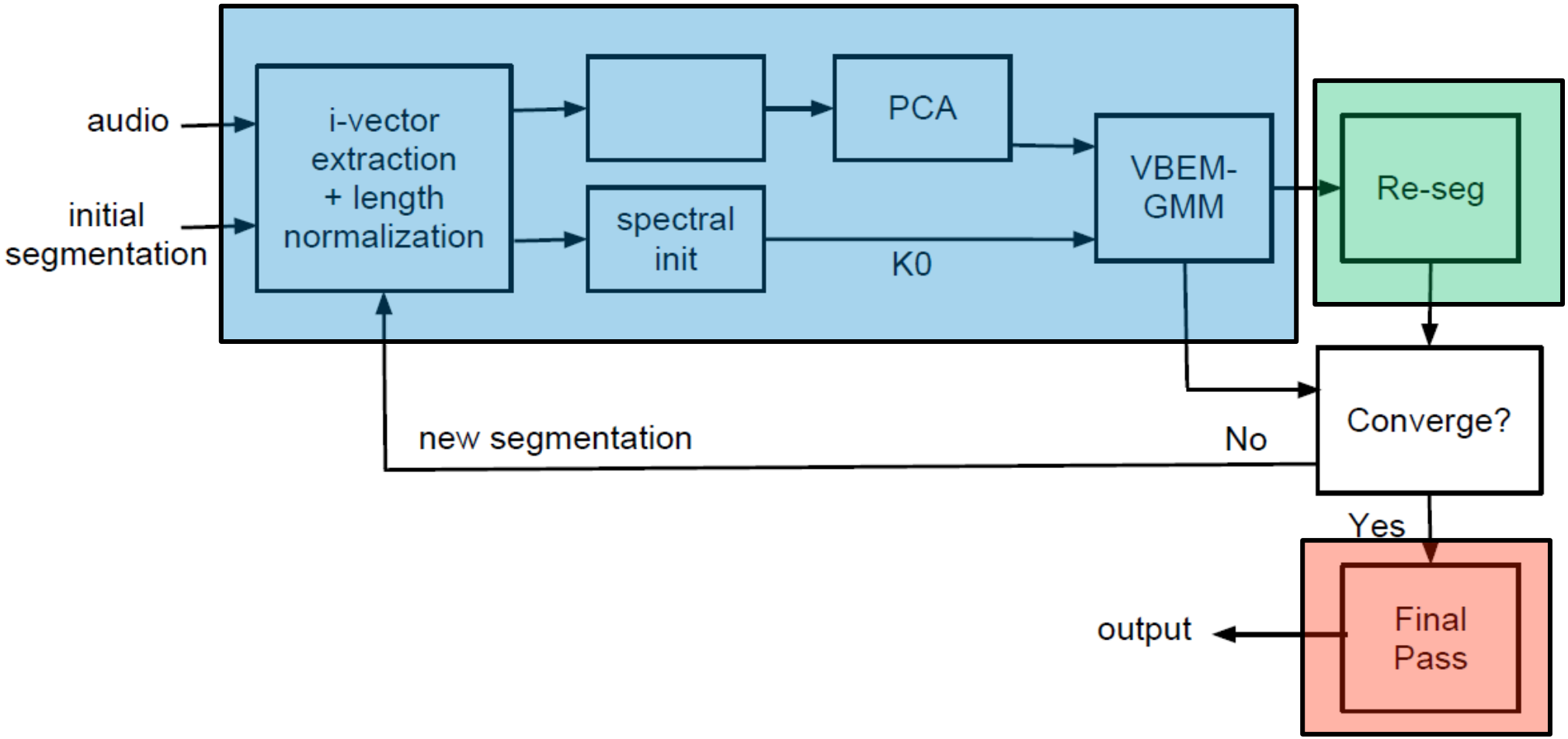


- **Defining "convergence"**
  - DER can be seen as a "distance" between two diarization hypotheses.

Iterative System Optimization Results



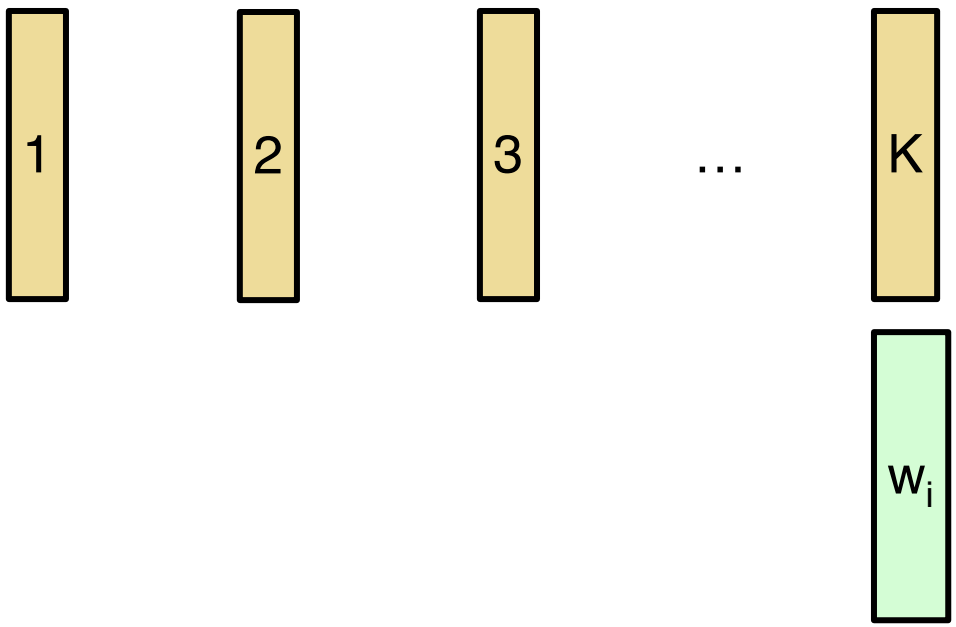
Diarization System So Far



Final Pass Refinements
(Interspeech 2011)

– Extract a single i-vector for each respective speaker.
  * **Using the newly defined re-segmentation assignments**

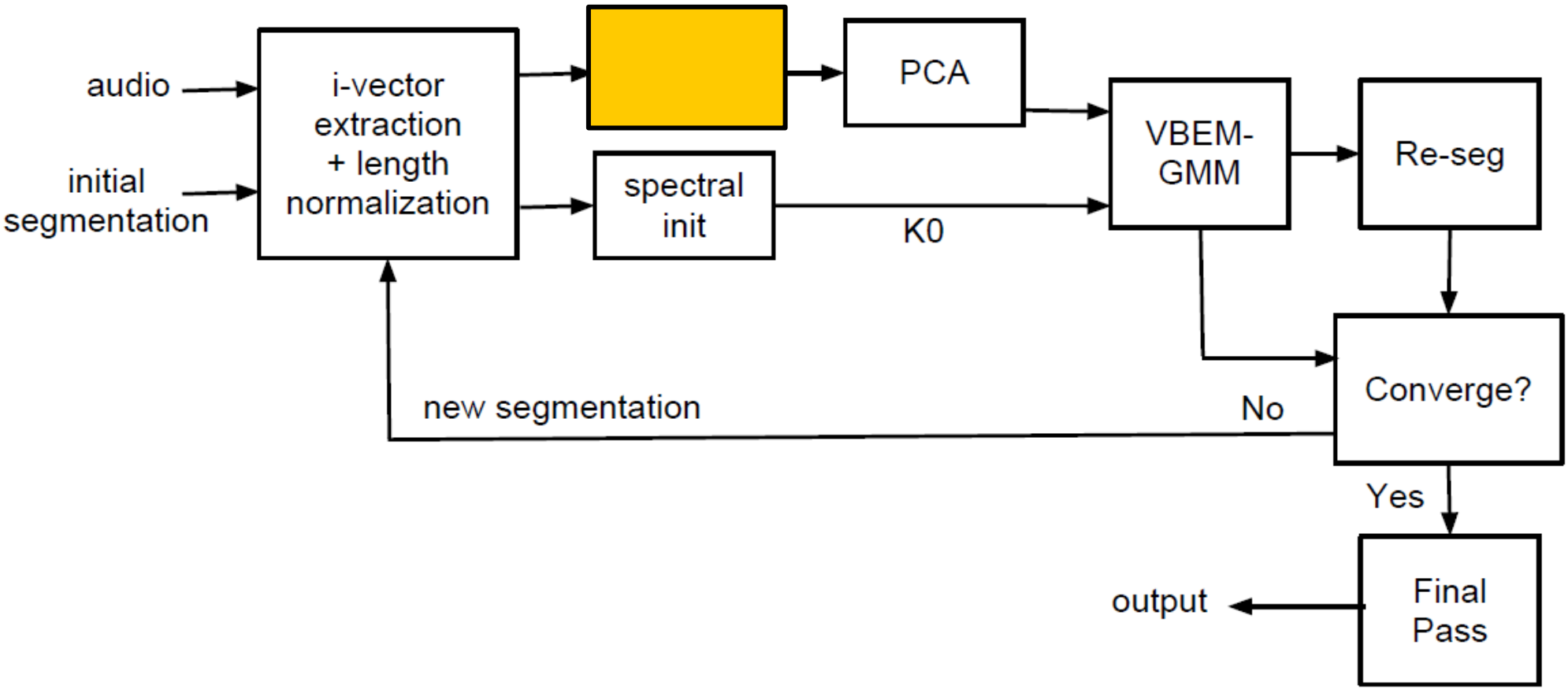
– Re-assign each newly-extracted segment i-vector w_i to the speaker i-vector $\{w_1, w_2, \ldots, w_K\}$ that is closer in cosine similarity.
  * **"Winner Takes All"**



Final Pass Refinements
(Interspeech 2011)

- Extract a single i-vector for each respective speaker.
  - **Using the newly defined re-segmentation assignments**
- Re-assign each newly-extracted segment i-vector w_i to the speaker i-vector $\{w_1, w_2, \ldots, w_K\}$ that is closer in cosine similarity.
  - **"Winner Takes All"**
- Iterate until convergence.
  - **i.e. when segment-speaker assignments no longer change**
- Essentially a K-means algorithm
  - **Except determine "means" $\{w_1, w_2, \ldots, w_K\}$ via i-vector extraction**

Diarization System So Far



i-vector Underrepresentation

- **i-vectors have been used as point estimates.**
  - During clustering, we treat them as independent and identically distributed samples from some underlying GMM.

- **However, some i-vectors may be more equal than others.**
  - i-vector from a 5-second speech segment versus 0.5-second segment

- **Recall: Given some speech,**
  - The i-vector is a posterior mean of a Gaussian distribution…
  - With an associated posterior covariance

$$\mathrm{cov}(w) = \left(I + T^{*}\Sigma^{-1}N(u)T\right)^{-1}$$

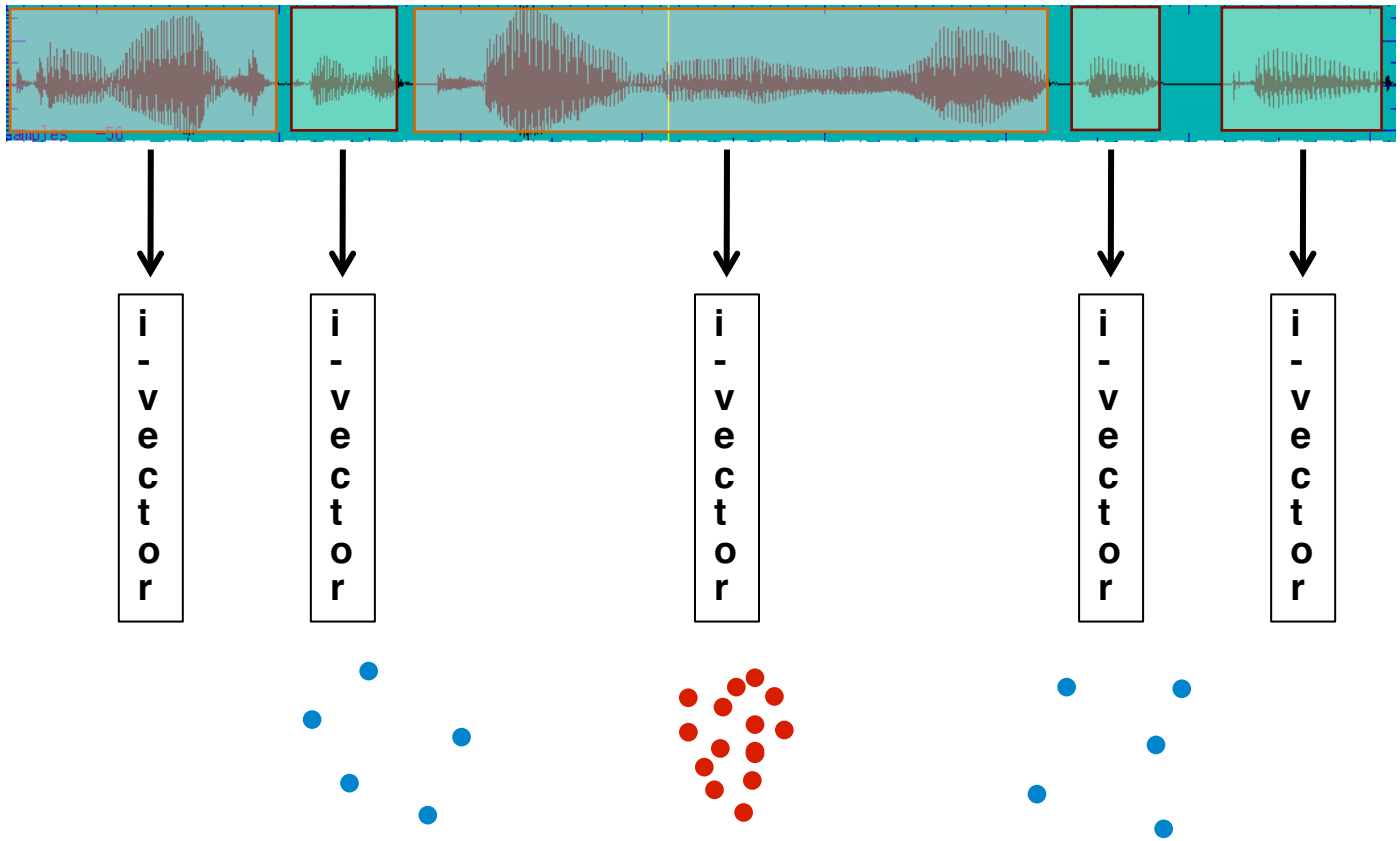
Overcoming Underrepresentation
– A Sampling Approach

- **"Size" of covariance is inversely proportional to number of frames N(u) in utterance u.**
  - More frames used to extract i-vector → "smaller" covariance

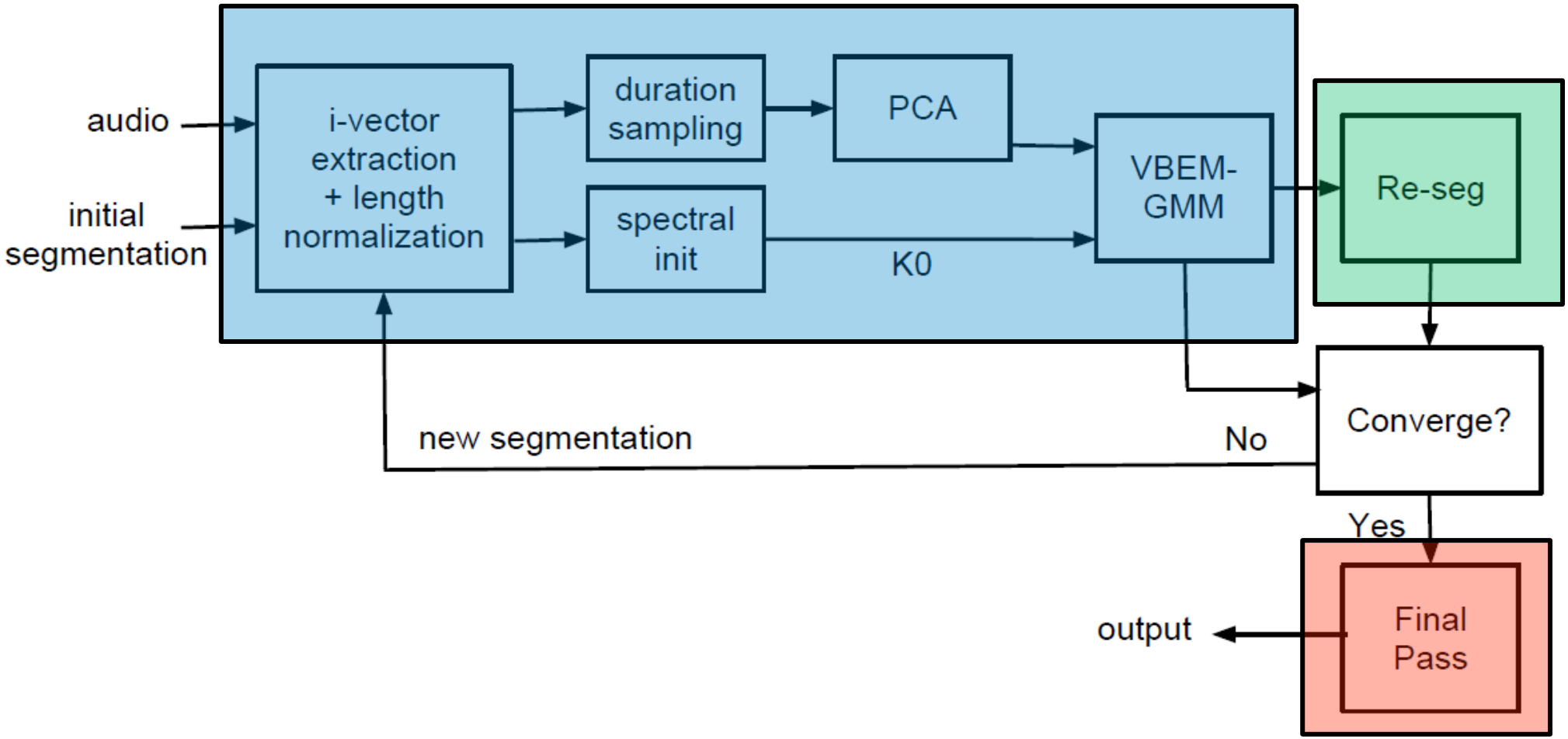
$$\mathrm{cov}(w) = \left(I + T^{*}\Sigma^{-1}N(u)T\right)^{-1}$$

- **Consider sampling the i-vector distribution**
  - Let the number of samples drawn be proportional to the number of frames used to extract the i-vector.
    - * **Shorter segments → larger covariance and fewer samples**
    - * **Longer segments → smaller covariance and more samples**

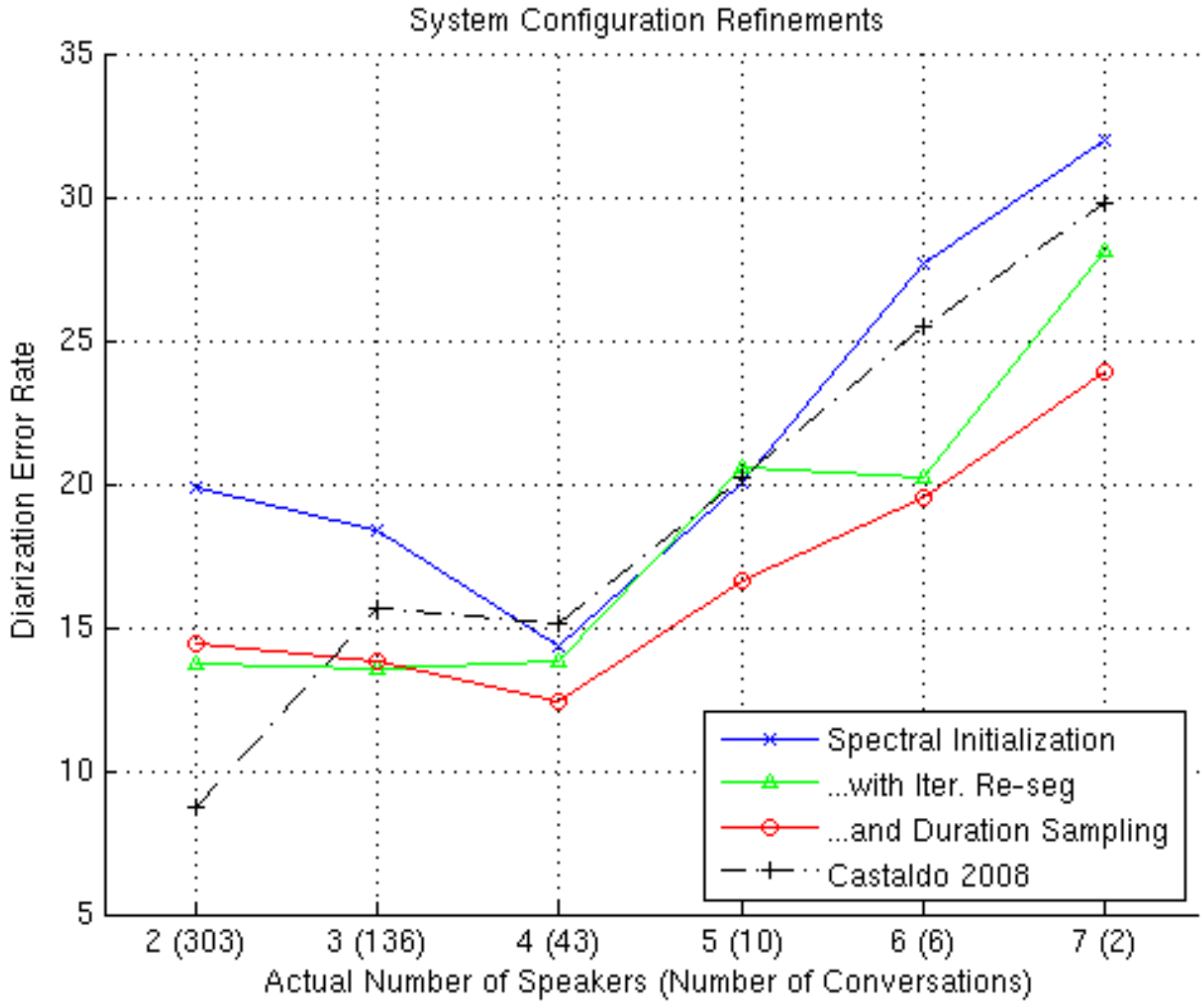
A Simplified Cartoon



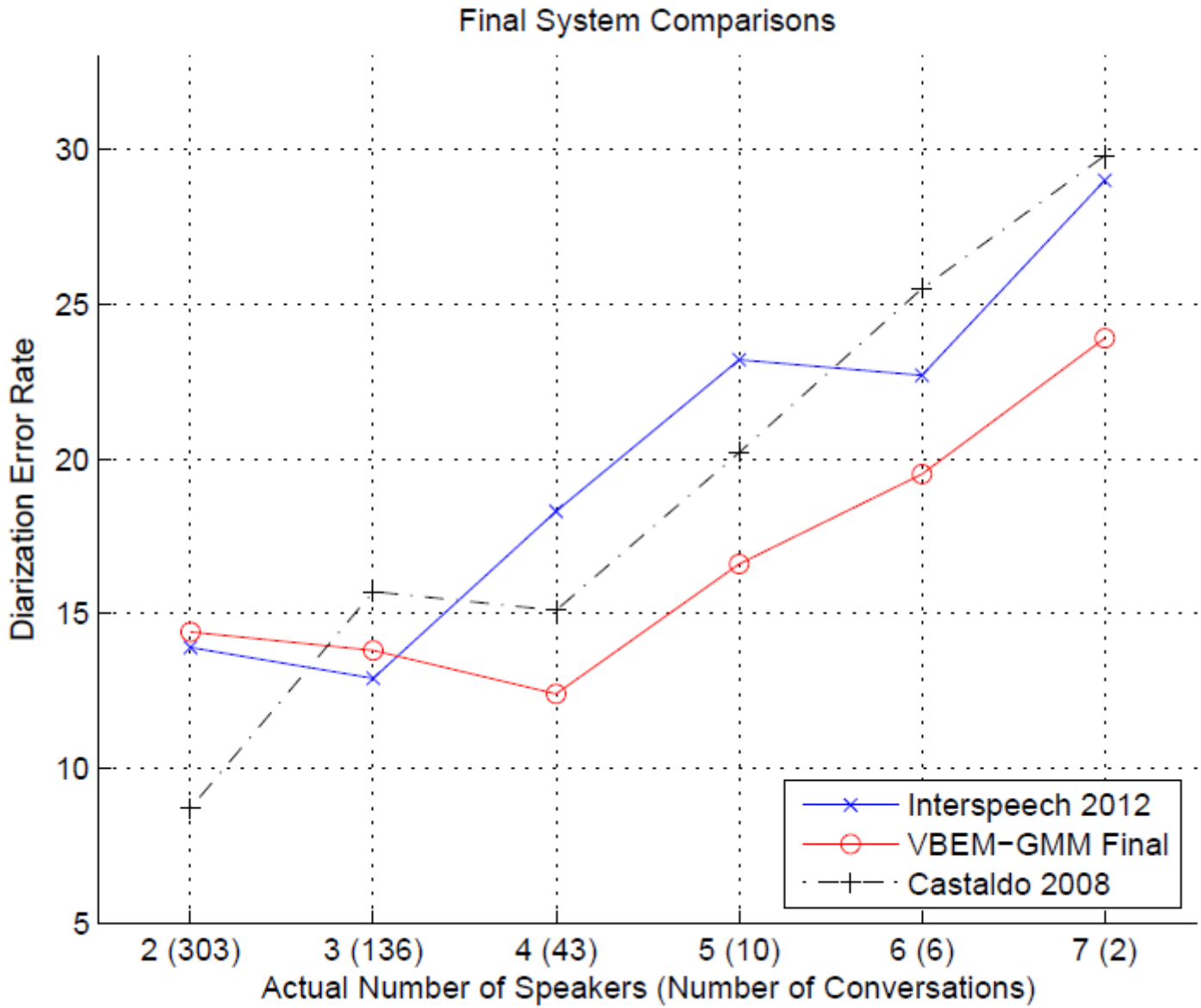
Final System Diagram



Proposed System Refinements



Final System Comparisons



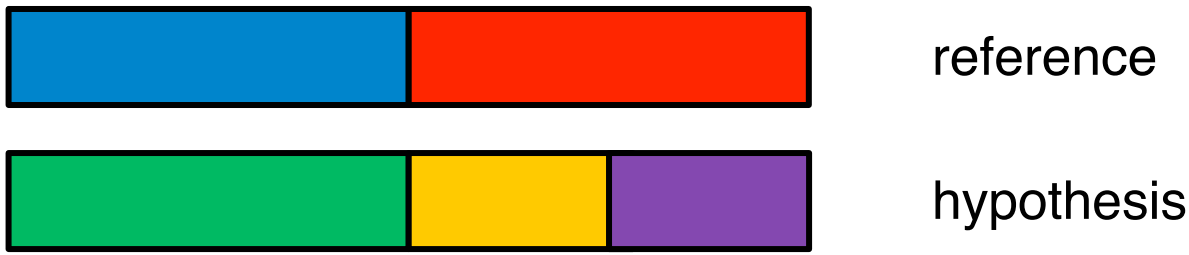
Reconciling Our 2-Speaker Results

- **Interspeech 2011 vs. Kenny 2010 vs. Castaldo 2008**
  - State-of-the-art results on diarization on two-speaker telephone calls (number of speakers given)

- **Interspeech 2012**
  - On the CallHome corpus, when it is known that the conversation contains only two participants
    - **DER = 4.3% vs. 8.7% (Castaldo 2008)**

DER Observations

- **Over-detecting the number of speakers**
  - In the conversations where we correctly detect two speakers (136/303),
    - **DER = 6.5% vs. 8.7% (Castaldo 2008)**
  - But DER is unforgiving towards overestimation



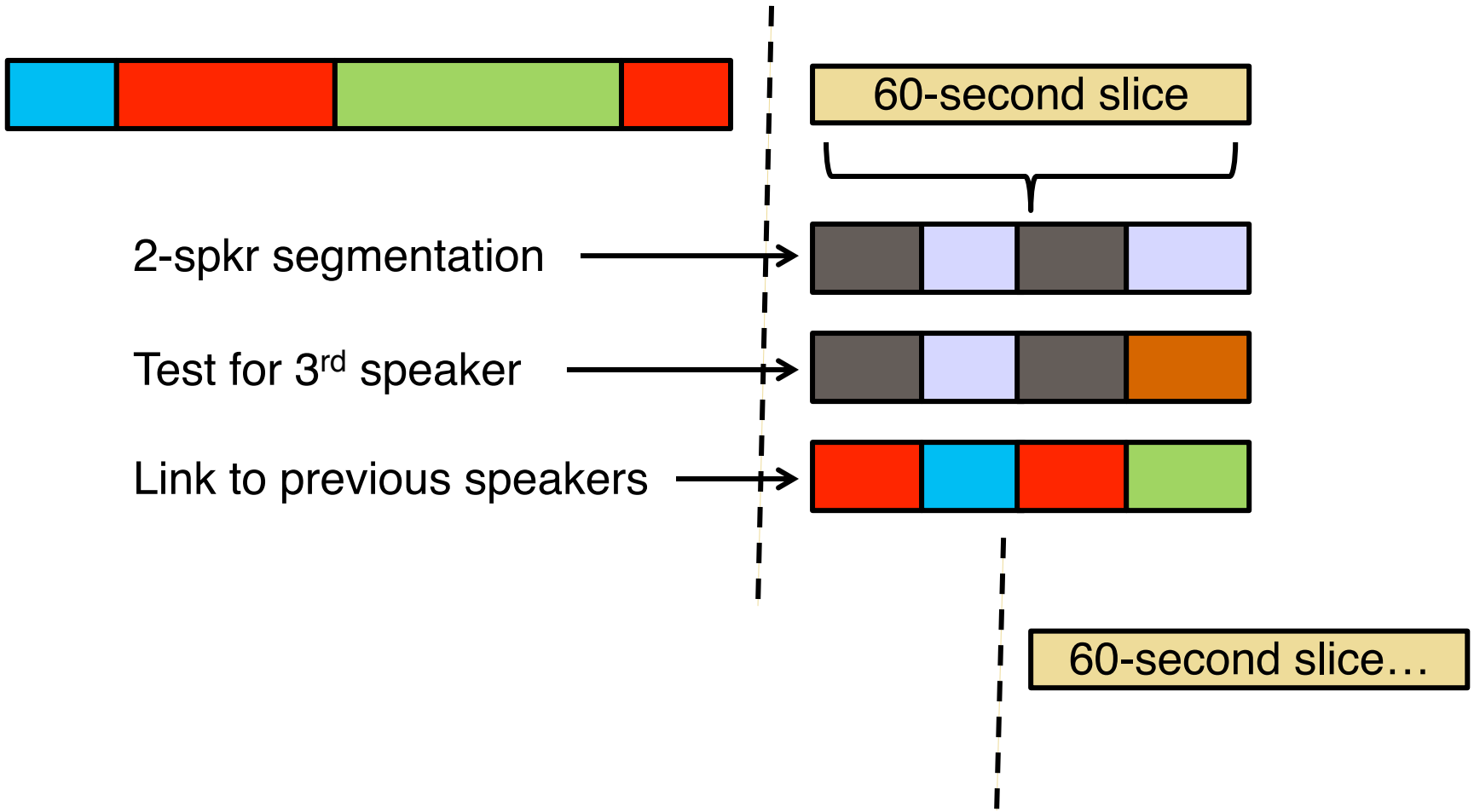
- **Conversely, underestimation**

Roadmap

- **Introduction**
  - Summary of Contributions
- **Background**
  - Diarization System Overview
  - Speaker Modeling with Factor Analysis
- **Our Incremental Approach**
  - K-means and Spectral Clustering (Interspeech 2011, 2012)
  - Towards Probabilistic Clustering Methods
  - Iterative System Optimization (Re-segmentation/Clustering)
  - Duration-Proportional Sampling
- **Analysis and Discussion**
  - Benchmark Comparison (Castaldo 2008)
- **Conclusion**

Explaining (Castaldo 2008)

- Causal system with fixed output delay
- Stream of factor analysis-based features (every 10ms)



Summary of Differences

- **Castaldo 2008**
  - Exploits structure of telephone conversations
    - **Assumes no more than 3 speakers exist in any 60-second slice**
  - Explicit use of speaker recognition system
    - **Links speakers from current slice to previous slices**

- **Our "bag of i-vectors"**
  - More general approach to clustering
    - **Can handle any number of speakers, regardless of temporal conversation dynamics**
    - **Prone to missing speakers that seldom participate**

Future Work

- **Dimensionality Reduction**
  - So far, only using first 3 principal components
  - High-dimensional, Robust PCA
    - * **Xu 2010**
  - Lower-dimensional Embeddings
    - * **Johnson-Lindenstrauss Lemma (1984)**

- **Within-utterance Factor Analysis**
  - Is there some way to directly exploit variabilities within the acoustic features of a particular conversation?

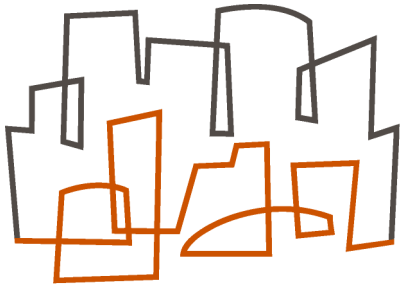
- **Temporal Modeling and Non-parametric Bayesian Models**
  - Hierarchical Dirichlet Process – Hidden Markov Model (HDP-HMM)
    - * **Fox 2008, Johnson 2010**

Roadmap

- **Introduction**
  - Terminology, tasks, and framework

- **Low-Dimensional Representation**
  - Speaker Verification
  - Sequence of features: GMM
  - Super-vectors: JFA
  - Low-dimensional vectors: i-vectors
  - Processing i-vectors: compensation and scoring
  - Data visualization

- **Other Applications**
  - Speaker diarization
  - Language recognition



MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

Language Identification

Language Identification Outline

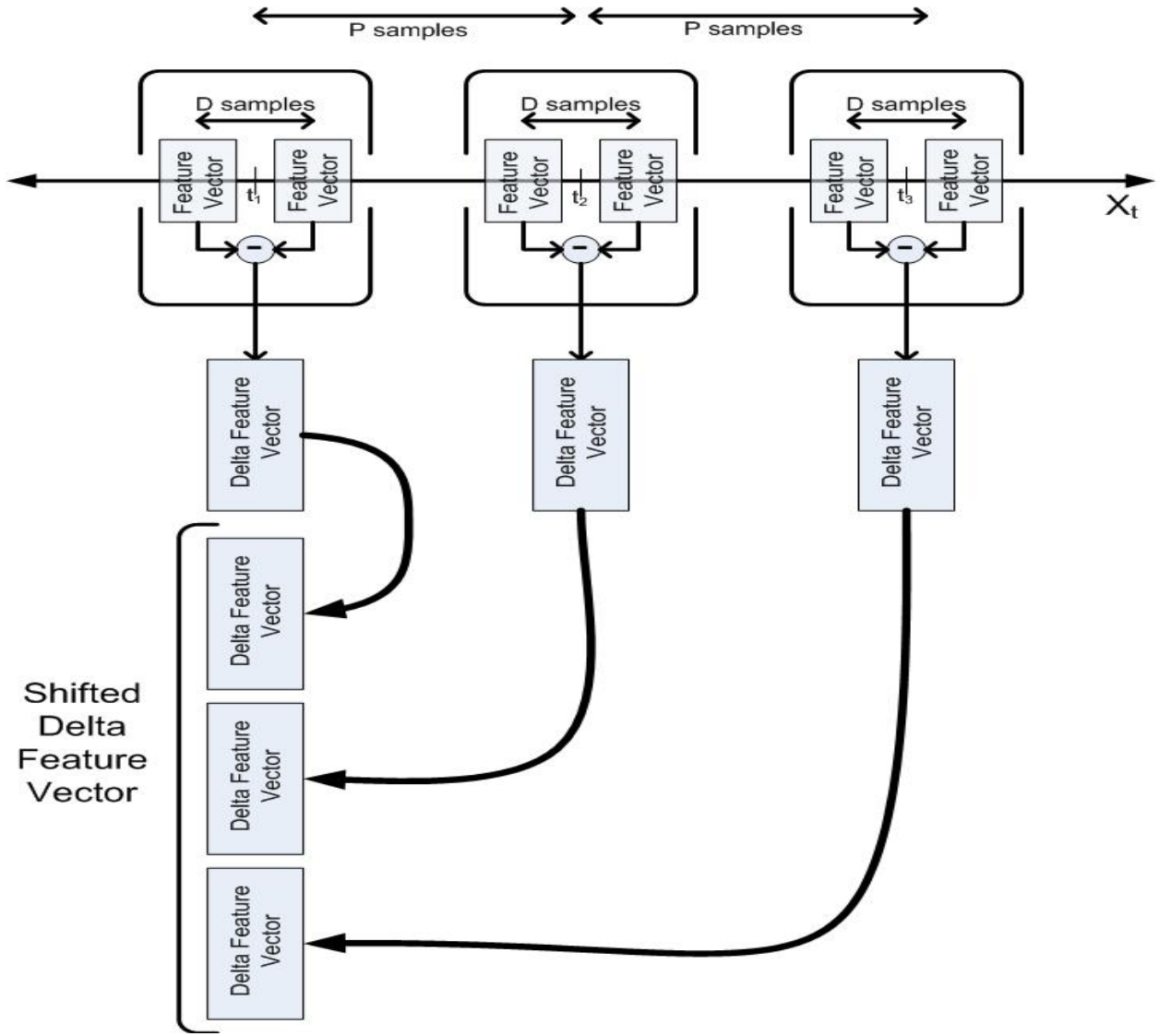
- Motivation
- Features extraction
- Intersession compensation and scoring
- NIST Language Recognition Evaluation
- Experiments and Results
- Interesting data visualization

Motivation

- **Low dimensional speech representation based on the Factor analysis**
  - Each speech recording is mapped on low dimensional vector (400)
- **Factor analysis as feature extractor**
  - Modeling the inter-language variability between different language classes
- **Score decision based on the cosine distance**
  - Simplicity of the system
- **Graph visualization to model connection between different languages**

Feature extraction for language Identification

- Shifted Delta Cepstral



Intersession compensation

- **Linear Discriminant Analysis to maximize the variability between the different language classes [Dehak 2009,2011]**

A is matrix of eigenvectors from $S_b \,.v = \lambda \,.S_w \,.v$

$$S_b = \sum_{i=1}^{L} (w_i - \overline{w})(w_i - \overline{w})^t$$

$\overline{w}$: the mean of the entire population

$$S_w = \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} (w_i^l - w_l)(w_i^l - w_l)^t$$

- **Within Class Covariance Normalization is used to scale the component [Hatch2006] , [Dehak 2009,2011]**

$$W = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{n_l} \sum_{i=1}^{n_l} (A^t w_i^l - w_l)(A^t w_i^l - w_l)^t$$

$$w_l = \frac{1}{n_l} \sum_{i=1}^{n_l} A^t w_i^l \quad \text{mean for language class l}$$

n_l = number of files for each language class l

L = total number of language classes

Language Identification Scoring

- **The scoring is based on a dot product**
  - Normalizing the length of the i-vectors
- **Training**
  - Project the i-vectors with LDA A and WCCN $BB^t = W^{-1}$

$$w' = \frac{B^t A^t w}{\left\| B^t A^t w \right\|}$$

  - For each class *i* compute the mean and than normalize the length

$$m_i = \frac{\frac{1}{N}\sum_{j=1}^{N} w'}{\left\| \frac{1}{N}\sum_{j=1}^{N} w' \right\|}$$

- **Test**
  - Project the test i-vector with LDA and WCCN $w'_{test} = \frac{B^t A^t w_{test}}{\left\| B^t A^t w_{test} \right\|}$
  - Compute the dot product of the test i-vector with the normalized mean of each class

$$score_i = w'_{test} * m_i$$

NIST 2009 Language Recognition Evaluation

- Current work
  - 23 languages

	languages		languages
1	amharic	13	hindi
2	bosnian	14	korean
3	cantonese	15	mandarin
4	creole	16	pashto
5	croatian	17	portuguese
6	dari	18	russian
7	english_american	19	spanish
8	english_indian	20	turkish
9	farsi	21	ukrainian
10	french	22	urdu
11	georgian	23	vietnamese
12	hausa		

Experimental setup

- **Features**
  - 7-1-3-7 SDC + static ceptral vector
  - Feature normalization to N(0,1)
  - SAD using GMMSAD
- **UBM 2048 Gaussian Components**
- **Ivector of dimension 400 (the best performances)**
- Development set consists of both CTS + VOA Data
- GMM – MMI (2048 mixtures + feature-based FA)
- SVM-GSV (1024 GMM + feature-based NAP)

Results

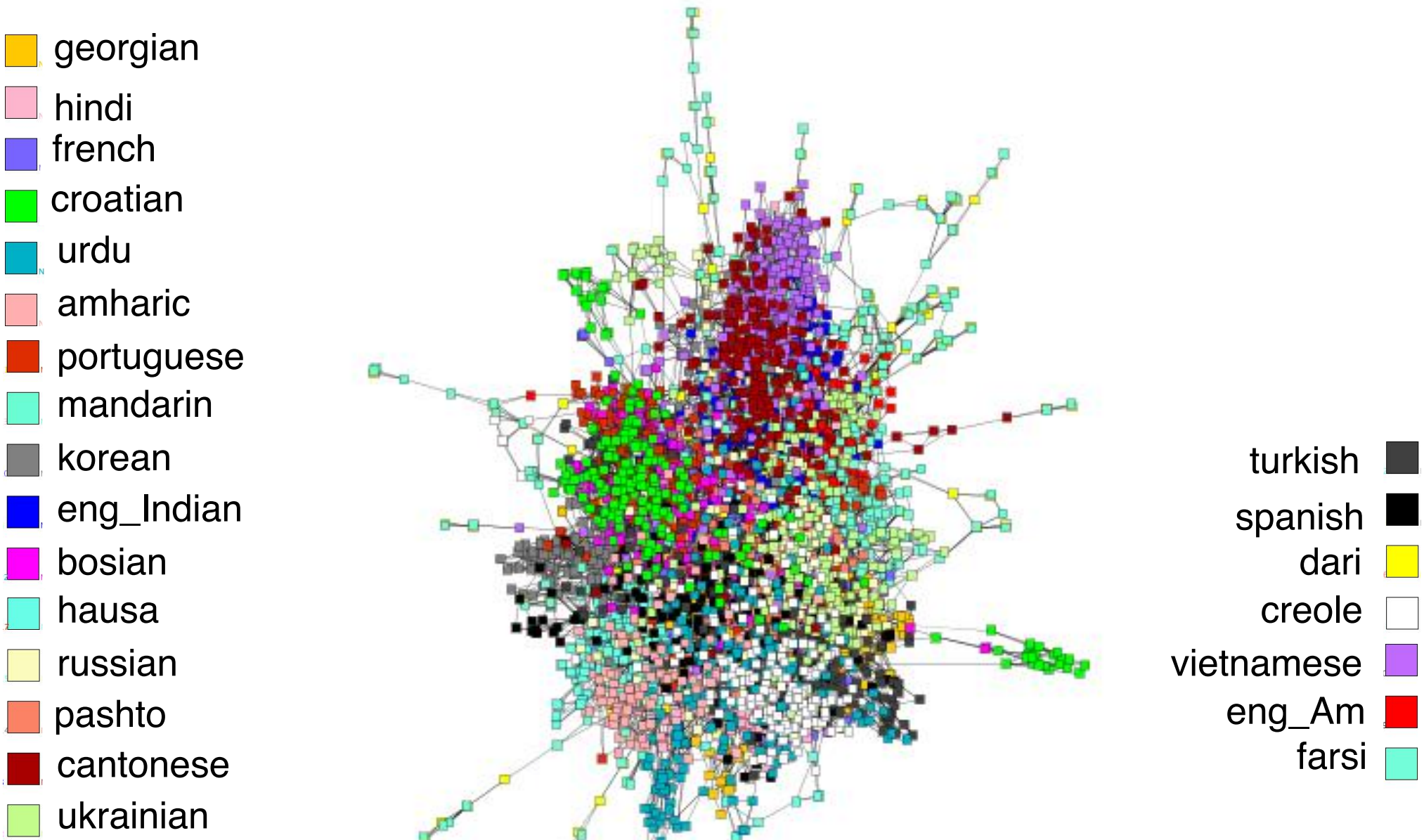
	30s		10s		3s	
	BB	AB	BB	AB	BB	AB
I-vector	2.2%	-	4.8%	-	13.8%	-
GMM-MMI	7.9%	2.3%	10.8%	4.4%	17.9%	12.9%
SVM-GSV	7.5%	2.3%	11.2%	5.0%	20.4%	15.4%

- **BB : Before Backend**
- **AB : After Backend**
- **Results on Equal Error Rate**

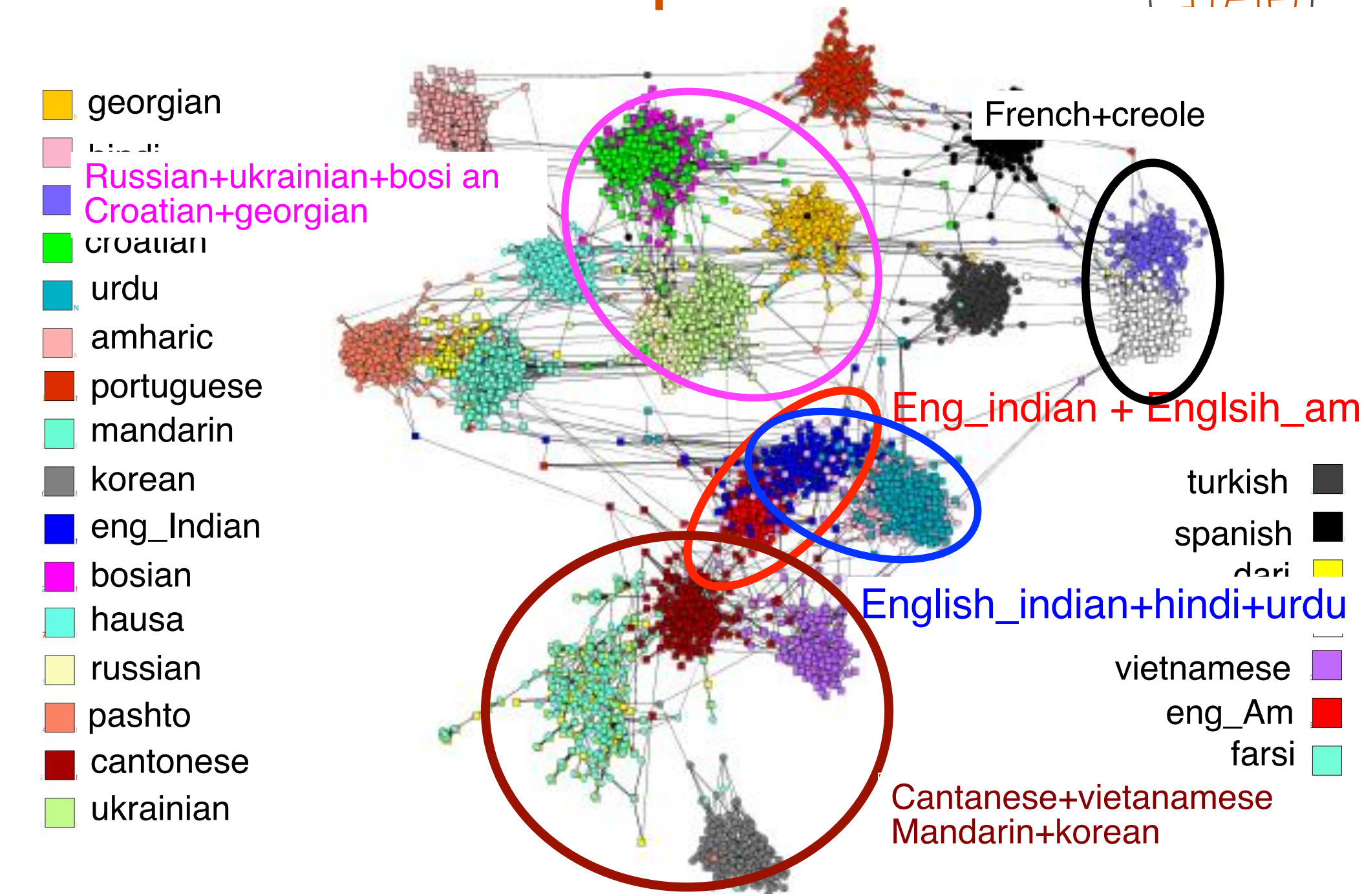
Graph Visualization

- **Work at Exploring the variability between different languages.**
  - Visualization using the Graph Exploration System (GUESS) [Eytan 06]
- **Represent segment as a node with connections (edges) to nearest neighbors (3 NN used)**
  - Euclidean distance after i-vectors length normalization.
  - NN computed using TV system (with and without intersession compensation normalization)
  - Intersession compensation :
    - **Linear Discriminant Analysis + Within Class Covariance Normalization**
- **Applied to 4600 utterances from 30s condition of the NIST LRE09**
  - 200 utterances for Language class
- **Absolute locations of nodes not important**
- **Relative locations of nodes to one another is important:**
  - The visualization clusters nodes that are highly connected together
- **Colors represent Language Classes**

No intersession Compensation



With intersession compensation

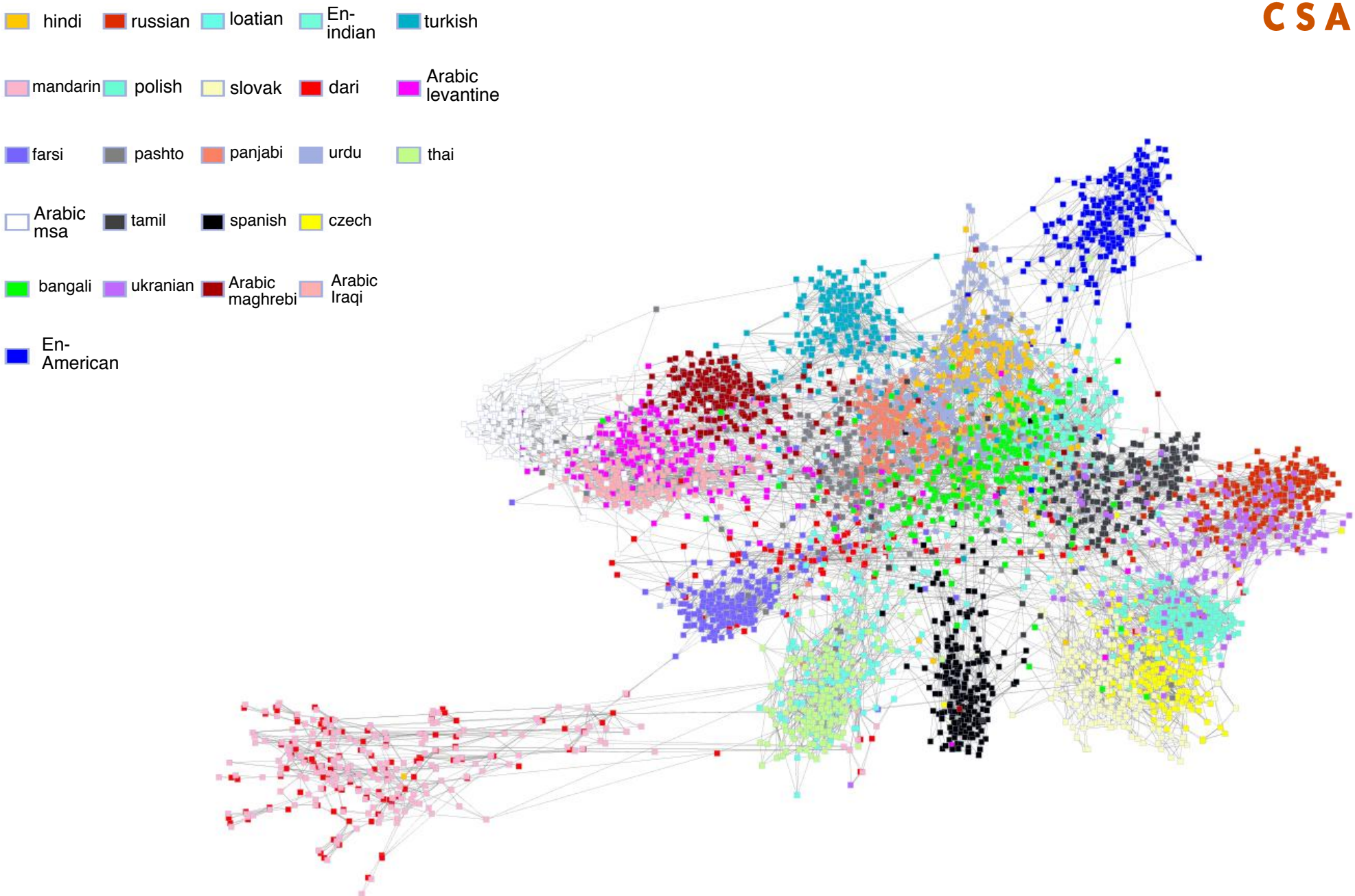


NIST 2011 Language Recognition evaluation

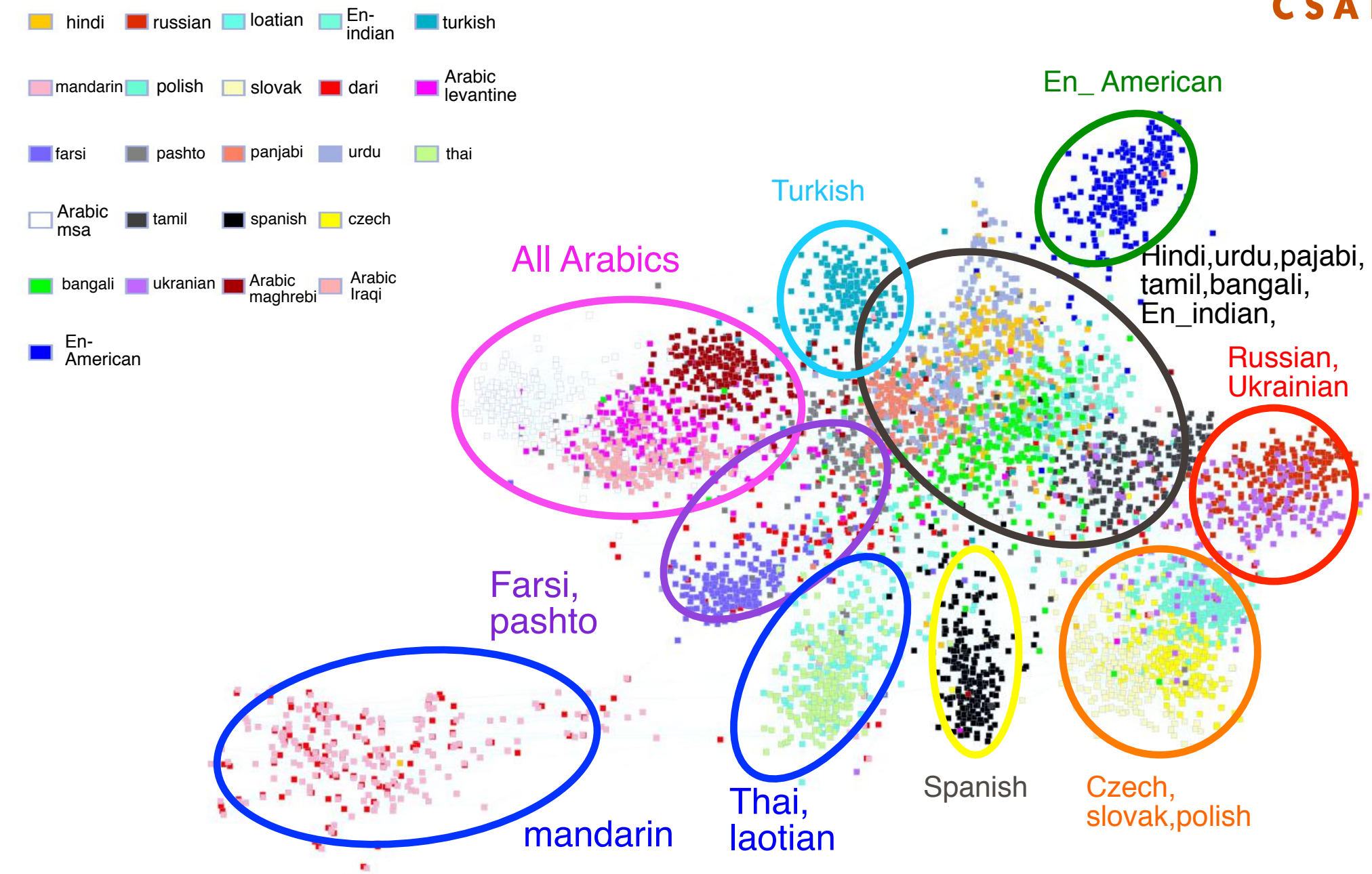
- Recent work
  - 20 languages
- Focus in the dialects as well
  - Arabic
  - English

Arabic Iraqi Arabic Levantine Arabic Maghrebi Arabic MSA	Polish
Bengali	Punjabi, Western
Czech	Russian
Dari	Slovak
English (American) English (Indian)	Spanish
Farsi/Persian	Tamil
Hindi	Thai
Lao	Turkish
Mandarin	Ukrainian
Pashto	Urdu

Visualization on NIST 2011 LRE dataset



Visualization on NIST 2011 LRE dataset



Final Words

- **What can you do with the I-vector modeling?**
- **I-vector framework can be used in any audio or sequential data classification problems**
  - One condition : you have enough data to estimate the space of the all different variabilities.
    - **Data need not be labeled a priori.**
  - Other sequential data classification problems: Perhaps video?

- **I-vector can be an appropriate tool to extract useful information from a large audio or sequential data corpora**
  - Useful for Big data problems (information retrieval)
  - Map all recordings into low dimensional space

References

- Dehak, N " Discriminative and Generative Approaches for Long- and Short-Term Speaker Characteristic Modeling : Application to Speaker Verification " PhD thesis, ETS, Montreal 2009.
- Dehak, N., Kenny, P., Dumouchel. P., Dehak, R., Ouellet. P., «Front-end factor analysis for speaker verification » in IEEE Transactions on Audio, speech and Language Processing 2011.
- Najim Dehak, Réda Dehak, Patrick Kenny, Niko Brummer, Pierre Ouellet and Pierre Dumouchel, Support Vector Machine versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification. In Proc INTERSPEECH 2009, Brighton, UK, September 2009.
- Kenny, P., Ouellet, P., Dehak, N., Gupta, V. and Dumouchel, P. "A Study of Inter-Speaker Variability in Speaker Verification" IEEE Transactions on Audio, Speech and Language Processing, 16 (5) July 2008 : 980-988.
- D. Reynolds, T. Quatieri, and R. Dunn, “Speaker verification using adapted Gaussian mixture models,” Digital Signal Processing, vol. 10, pp. 19–41, 2000.

Reference

- Adar, Eytan, "GUESS: A Language and Interface for Graph Exploration," CHI 2006
- Z. N. Karam, W. M. Campbell “Graph-Embedding for Speaker Recognition”, Submitted to Interspeech 2010