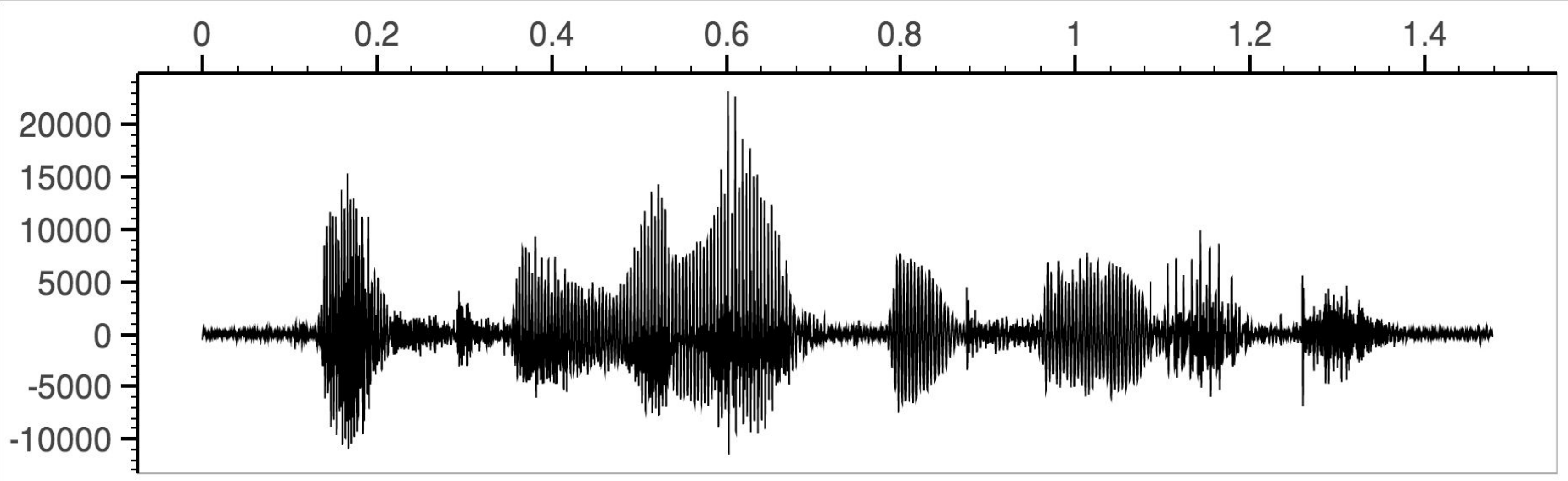
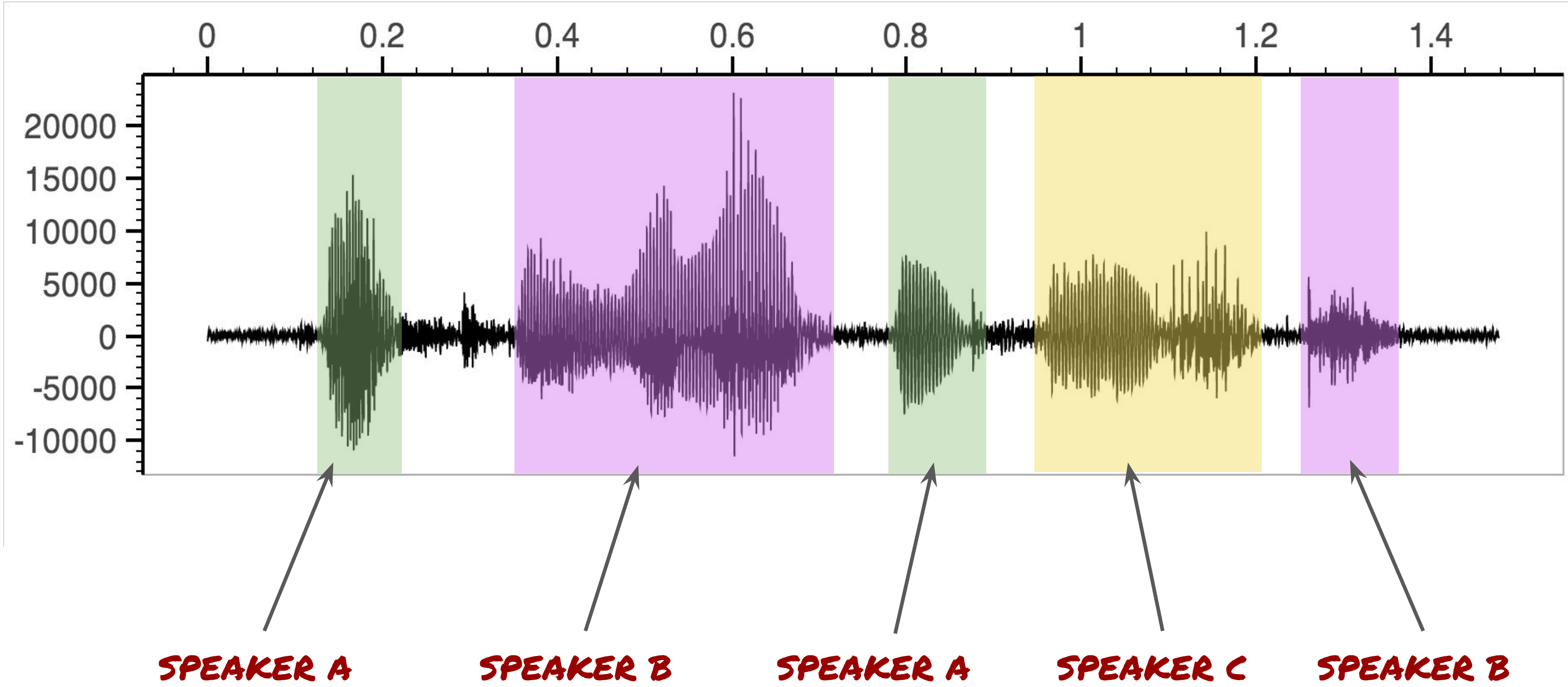
Automatic Annotation of Speakers in Phone Conversations

Yuriy Guts
DataRobot



0
0.2
0.4
0.6
0.8
1
1.2
1.4
20000
15000
10000
5000
0
-5000
-10000

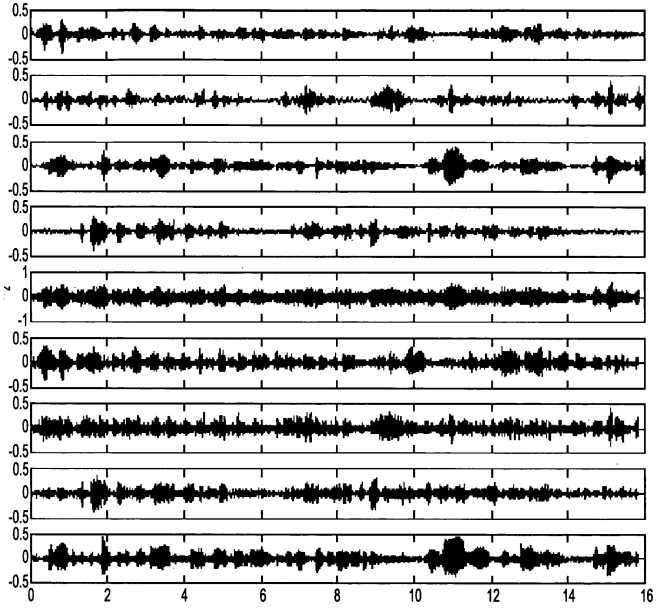


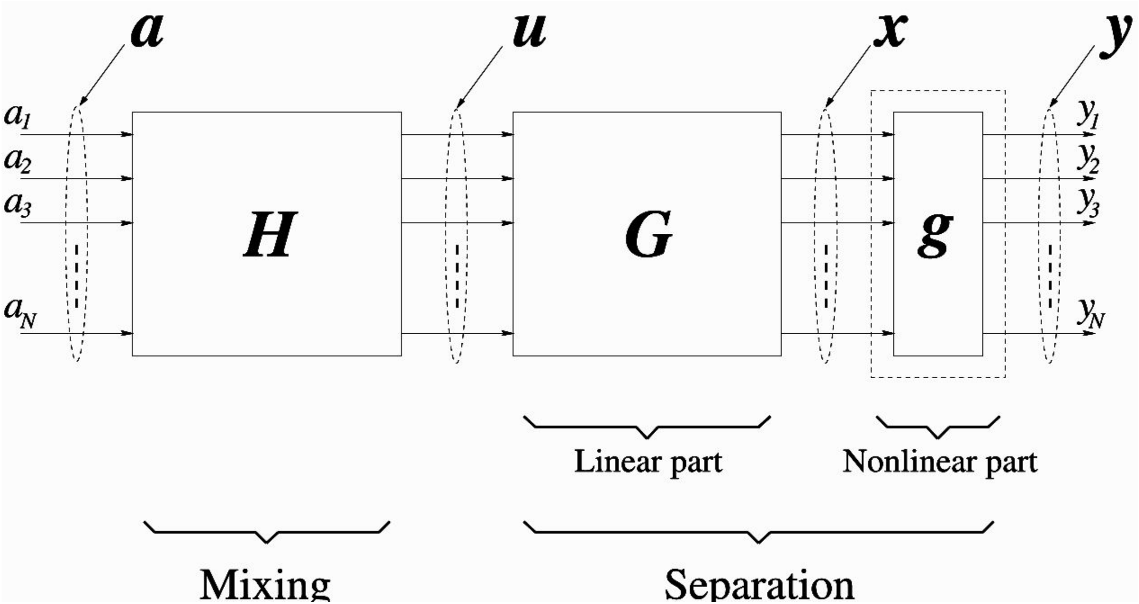
No prior knowledge about the speakers whatsoever.

Speaker Diarization: Why Do It?

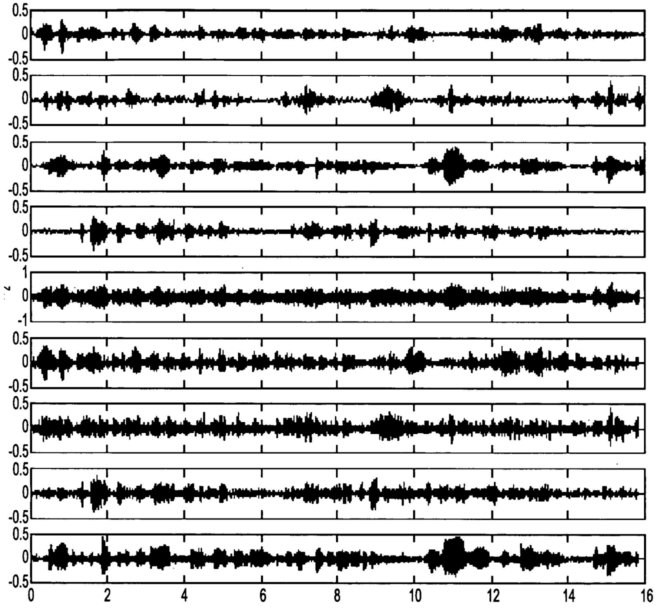
1. Extract compact metadata from bulky multimedia sources.
2. Enable information retrieval queries for audio content.
3. Provide training data for upstream modeling projects:
    - Speaker identification
    - Speech recognition
    - Emotion analysis

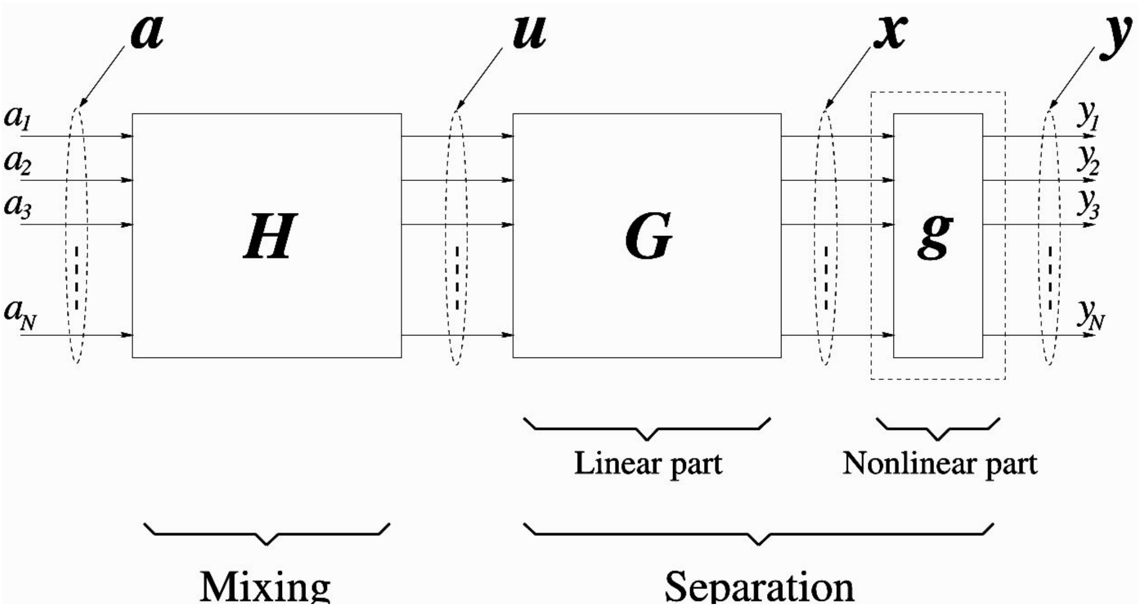
Cocktail Party Problem? ICA To The Rescue?





Cocktail Party Problem? ICA To The Rescue?





Nope. We have mono signal, so **# input mixtures < # desired separated outputs**

Source Types & Characteristics

	Broadcast News	Meetings	Phone Calls
Uninterrupted speech	Longer segments	Shorter segments	Shorter segments
Speaker overlap	Negligible	High	Moderate
Background conditions	Diverse: music, jingles, background events	Uniform	Uniform (but can be noisy)
Dominant speaker	Yes (anchor)	Unknown	Unknown
Number of speakers	Unknown	Unknown	Unknown (but usually 2)

Original publication: http://www.slideshare.net/ParthePandit/parthepandit10d070009ddpthesis-53767213

Input Data

100 hrs of recorded **customer support** calls.

- Mono WAV files, 8 kHz.
- Some files (20 hrs) are human-labeled:

```
0000.00,0005.63,agent,Female
0005.63,0017.13,customer,Female
0017.13,0027.76,agent,Female
0027.76,0034.94,customer,Female
0034.94,0035.89,agent,Female
0035.89,0044.50,silence,None
0044.50,0046.20,unrelated,None
0046.20,0049.10,customer,Female
0049.10,0050.14,agent,Female
0050.14,0060.99,silence,None
0060.99,0066.60,agent,Female
0066.60,0080.12,customer,Female
```

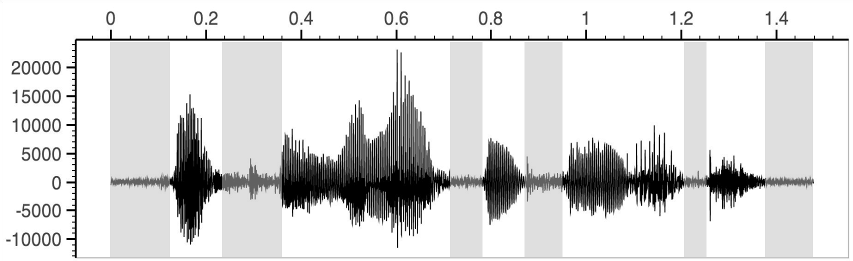
Particular Challenges of Support Calls

1. Audio durations are huge.
   20 sec min, **12 min** avg, **2 hours** max.
2. Lots of waiting on the line.
   Meaningful speech takes only ~**55%** of the time.
3. Inconsistent connection quality, lots of crackling and background noise.

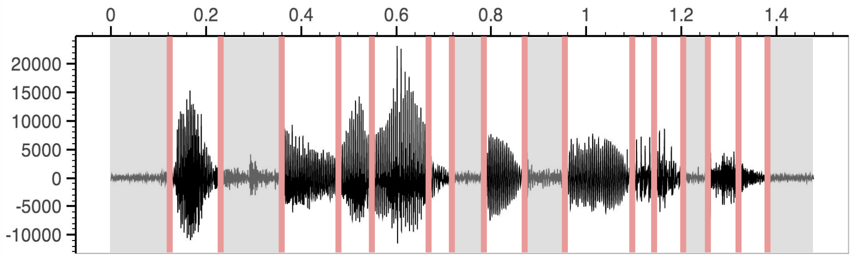
Overall Approach for Speaker Diarization

1. Non-speech activity cutoff.

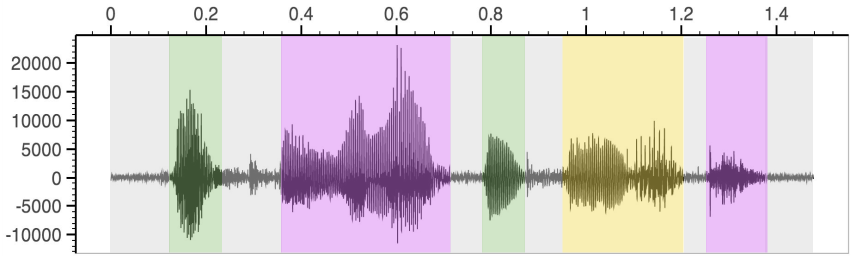
   For phone conversations, a simple RMSE threshold works fine. Otherwise, build a supervised VAD.



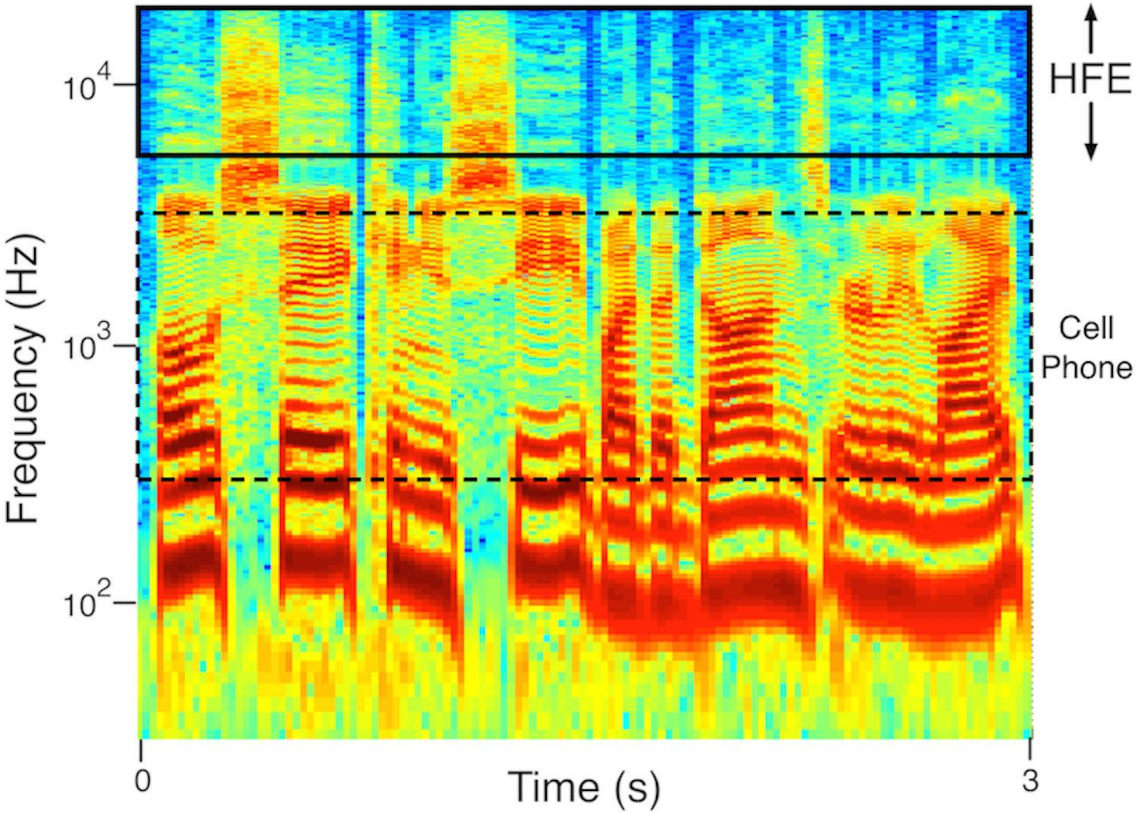
2. Changepoint candidate detection.



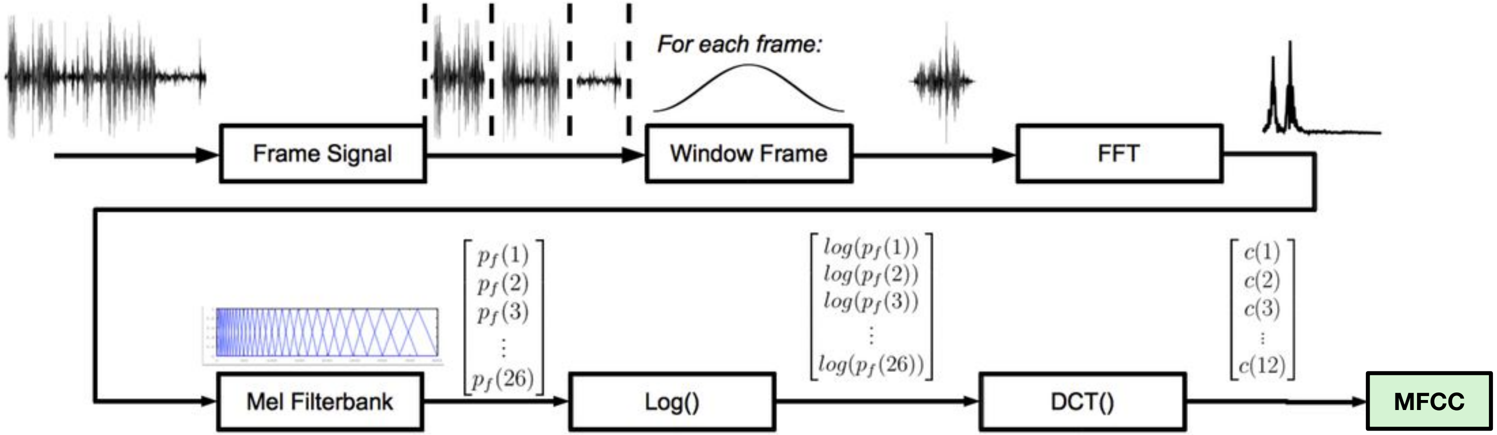
3. Segment recombination (clustering).

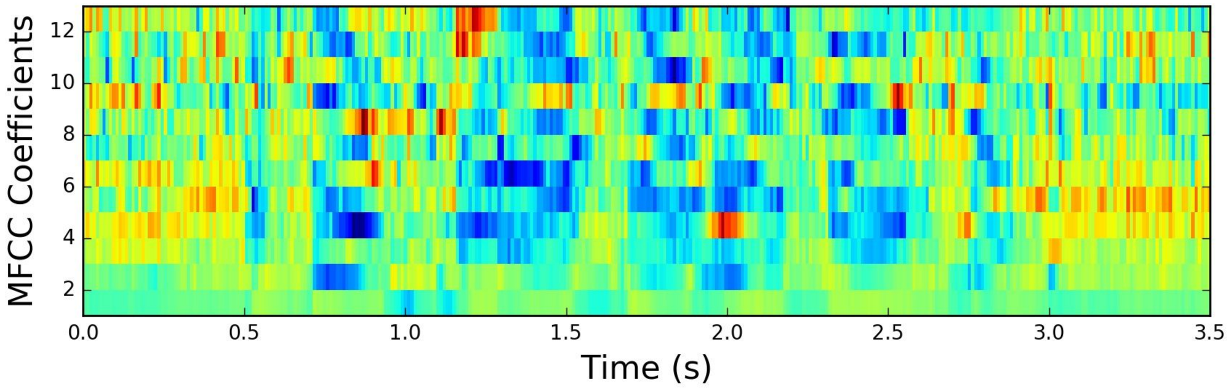


Process Raw Spectrogram?



MFCC (Mel-Frequency Cepstral Coefficients)



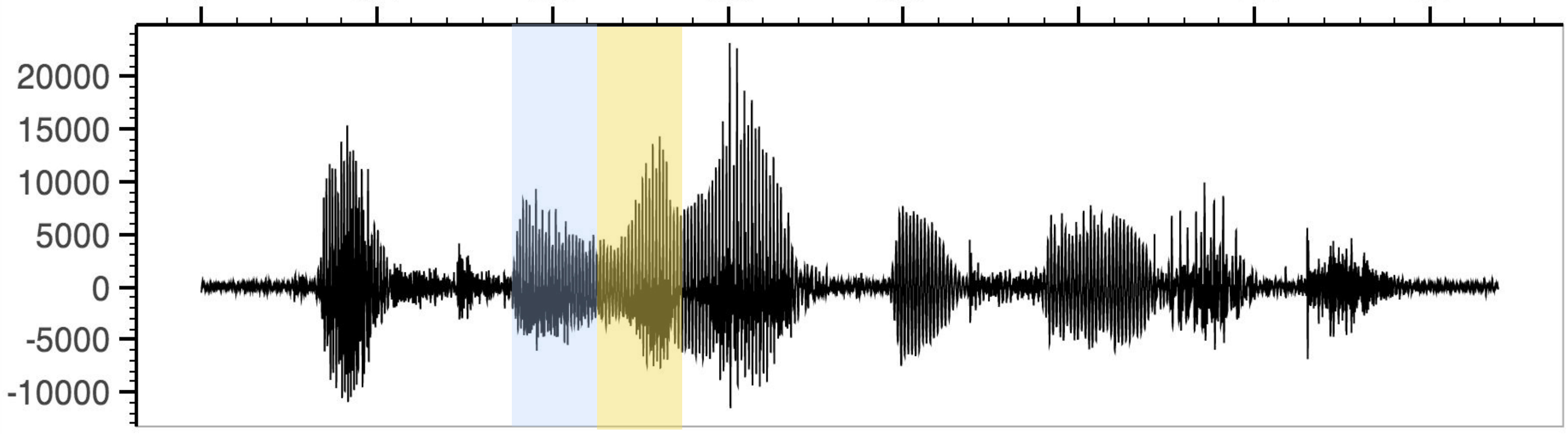


Changepoint Detection

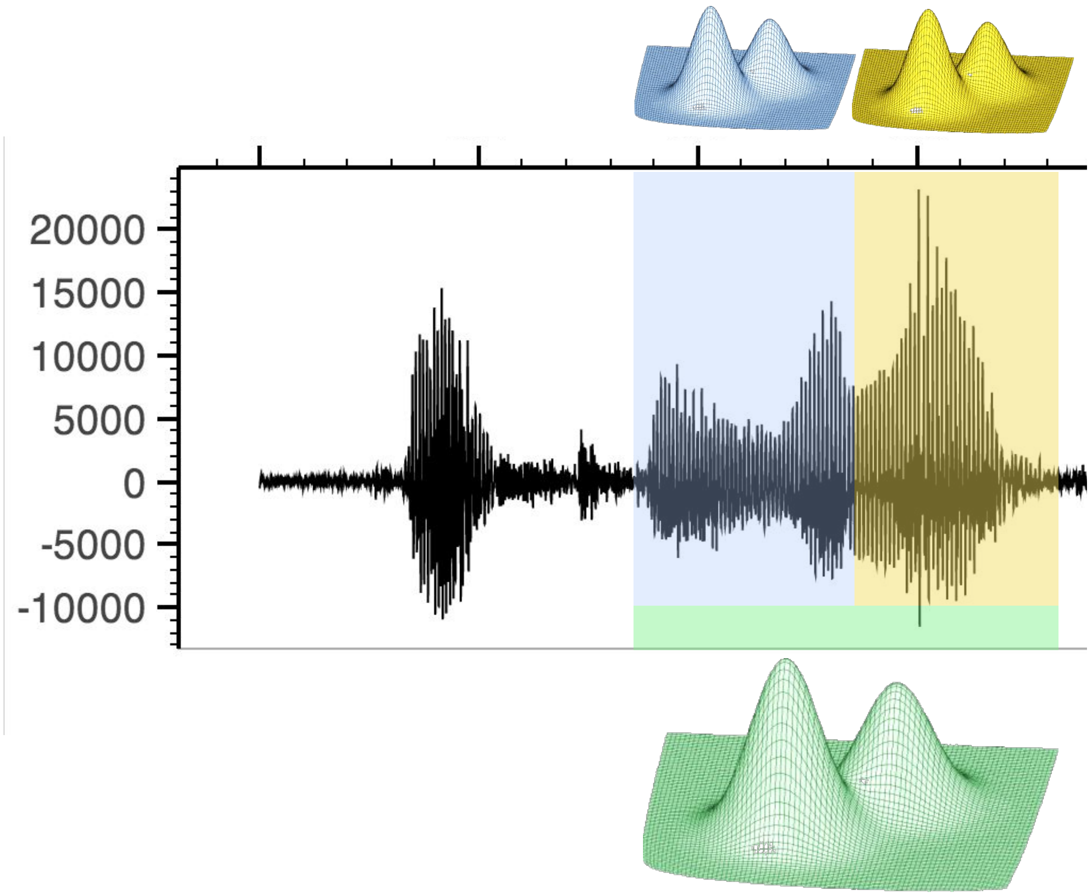
Sliding window (**0.5–3 sec** duration)

Statistical hypothesis testing:

Are these windows better modeled by one distribution or two?



Models: GMMs over MFCCs



Bayesian Information Criterion (BIC)

$$BIC(X, M) = \log P(X|M) - \lambda k_M \log N$$

$\log P(X\|M)$	Log-likelihood of the data points given the model
λ	Penalty term
k_M	Number of parameters of the model
N	Number of data points the model was trained on

Delta-BIC Distance Metric

$$\Delta BIC = BIC(M_{1,2}) - BIC(M_1) - BIC(M_2)$$

A positive value indicates dissimilarity between the audio windows.

Can use a threshold on ΔBIC to detect changepoints.

```python
def gmm_delta_bic(gmm1, gmm2, X1, X2):

    gmm_combined = sklearn.mixture.GaussianMixture(
        n_components=NUM_GAUSSIANS_PER_GMM,
        covariance_type="full",
        random_state=42,
        max_iter=200,
        verbose=0
    )

    X_combined = np.vstack([X1, X2])
    gmm_combined.fit(X_combined)

    bic_combined = gmm_combined.bic(X_combined)
    bic_gmm1 = gmm1.bic(X1)
    bic_gmm2 = gmm2.bic(X2)

    return bic_combined - bic_gmm1 - bic_gmm2
```

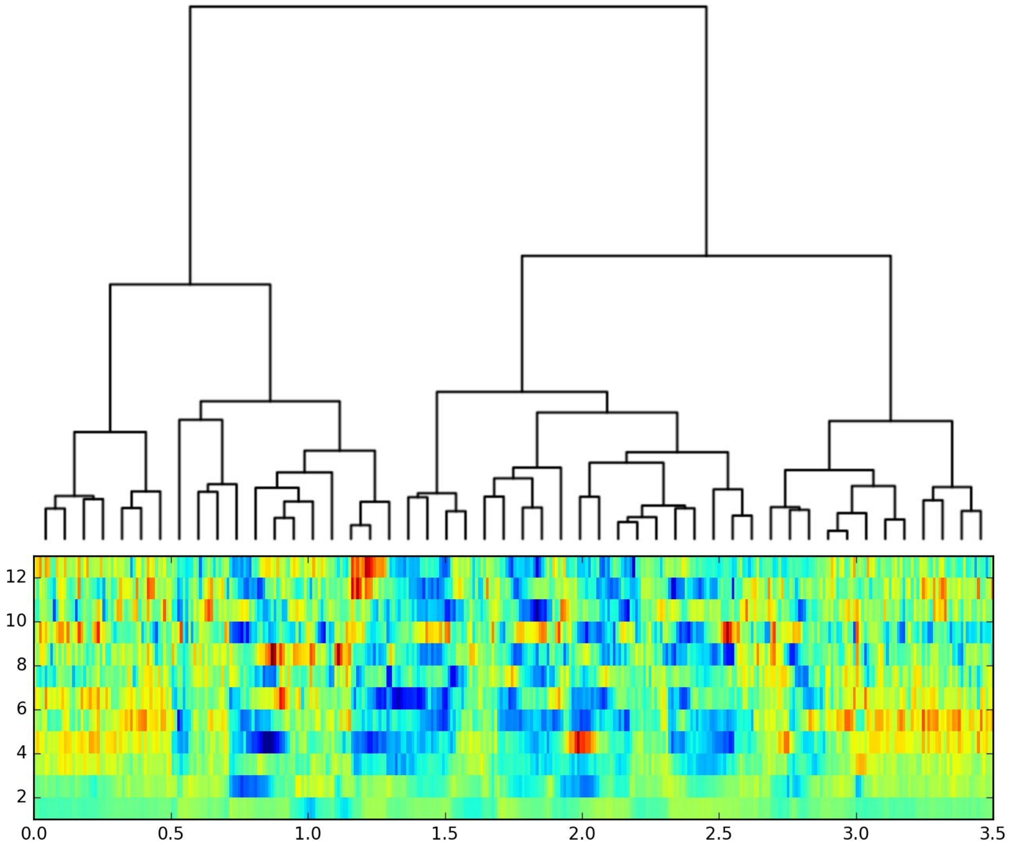
KL-Divergence

Can also use KL-Divergence for single-Gaussian GMMs:

$$D_{\mathrm{KL}}(\mathcal{N}_0 \| \mathcal{N}_1) = \frac{1}{2}\left\{ \mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\right) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - k + \ln\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} \right\}$$

Will be faster, but can produce more noisy changepoints.

Clustering



Perform hierarchical agglomerative clustering (HAC) based on ΔBIC until the stopping criterion is met.

Final System

1. Non-speech activity cutoff by RMSE threshold, analysis length = **0.5 sec**.
2. Modeling **0.5 sec** segments with **single-Gaussian GMMs** over **19**-dimensional MFCCs. MFCC analysis length = **30 ms**, frame hop length = **10 ms**.
3. Potential changepoint detection based on **KL-Divergence**.
4. HAC based on **ΔBIC** with a stopping threshold (or prior number of speakers).

Evaluation

DER (Diarization Error Rate)

$$DER = \frac{\sum_{s=1}^{S} dur(s) \cdot (max(N_{ref}, N_{hyp}) - N_{correct})}{\sum_{s=1}^{S} dur(s) \cdot N_{ref}}$$

DER = **Missed Speech %** + **False Alarm Speech %** + **Speaker Error %**

Achieved an average DER of **16.3%** (2.7% missed, 5% false alarm, 8.6% speaker) (state of the art: **14.9–16.1%** depending on source type)

Performance: **2x-5x RT** wall clock time (HAC has quadratic complexity)