Blind one-microphone speech separation: A spectral learning approach

Francis Bach
fbach@cs.berkeley.edu

Michael Jordan
jordan@cs.berkeley.edu

Computer Science, UC Berkeley

December, 2004

Summary

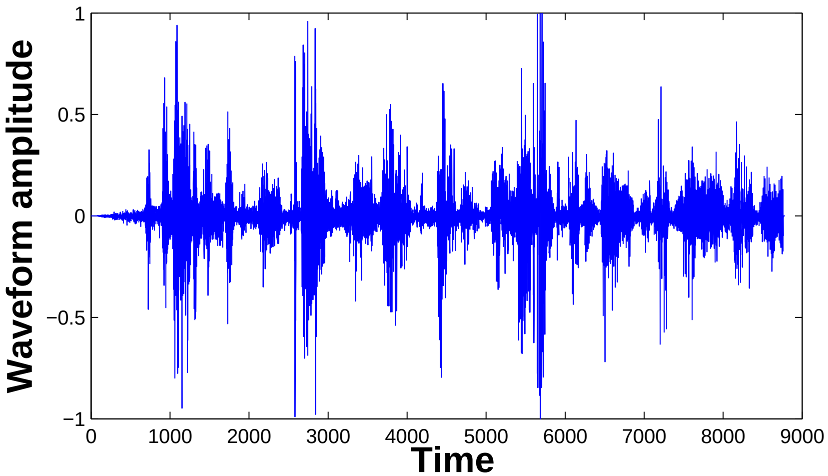
- Discriminative approach to blind one-microphone separation
- Reformulation as spectrogram segmentation
- Learning from artificially mixed data
- Machine learning algorithm for
  - segmenting
  - learning how to segment from training data

Blind one-microphone speech separation

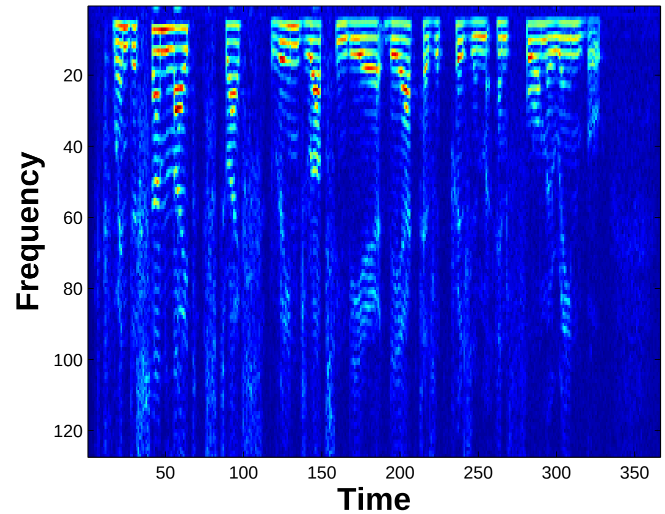
- Two or more speakers $s_1, \ldots, s_m$ - one microphone x
- Ideal acoustics $x = s_1 + s_2 + \cdots + s_m$
- **Goal**: recover $s_1, \ldots, s_m$ from x
- **Blind**: without knowing the speakers in advance
- Two types of approaches
  - **Generative**
    - Learn source model $p(s)$ ... then “simply” an inference problem
    - Model too simple : does not separate
    - Model too complex : inference intractable
    - Works for non blind situations (Roweis, 2001, Lee et al., 2002)
  - **Discriminative**: model of separation task, not of speakers

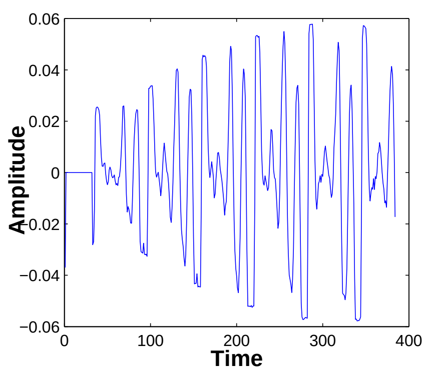
Spectrogram

- Spectrogram (a.k.a Gabor analysis, Windowed Fourier transforms)
  - cut the signals in overlapping frames
  - apply a window and compute the FFT



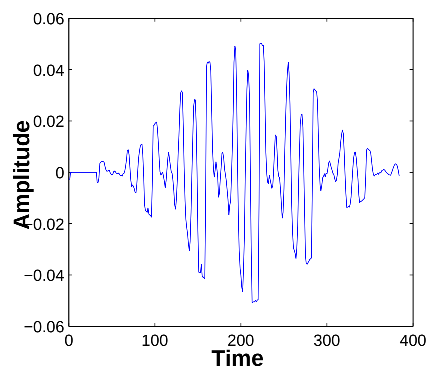
⇒





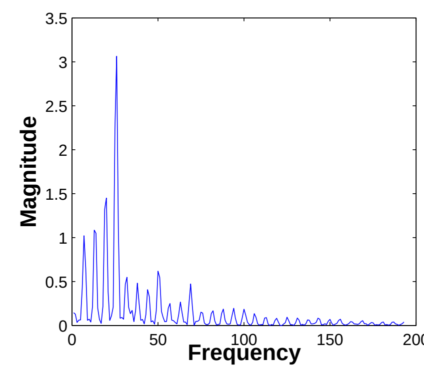
Windowing

⇒



Hamming window

⇒

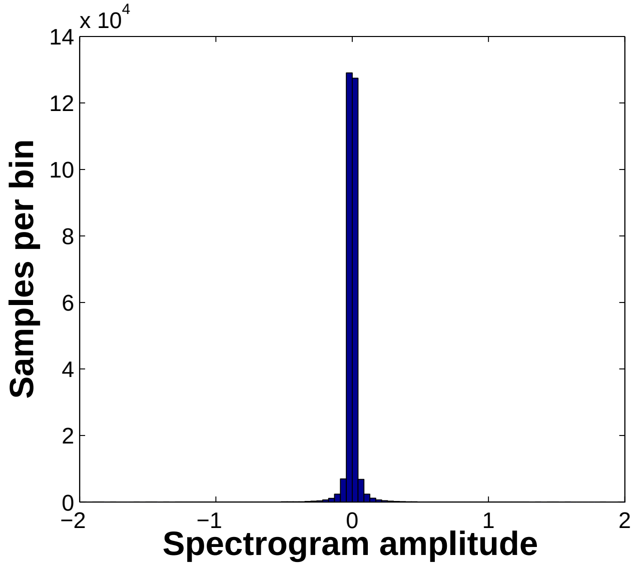


Fourier transform

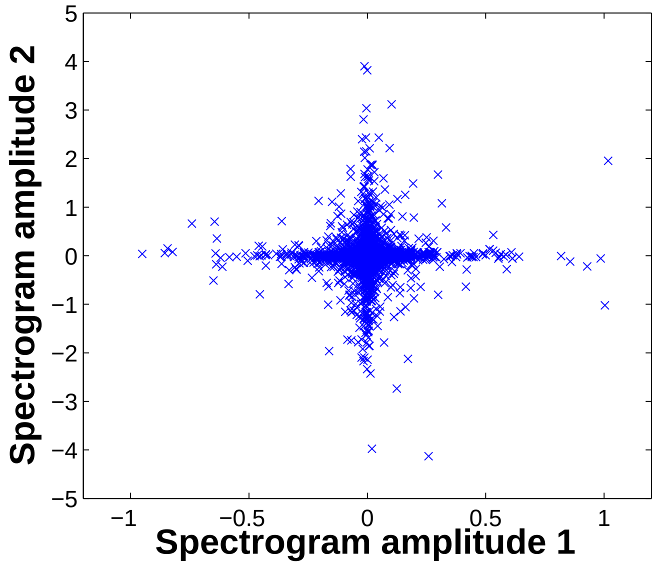
Sparsity of speech signals - spectrogram

- Disjoint support of spectrograms observed by several researchers (Cooke, 1994, Roweis, 2000, Yilmaz and Rickard, 2004)
- Sparsity of the spectrogram (all pixels taken together)

histogram of one signal

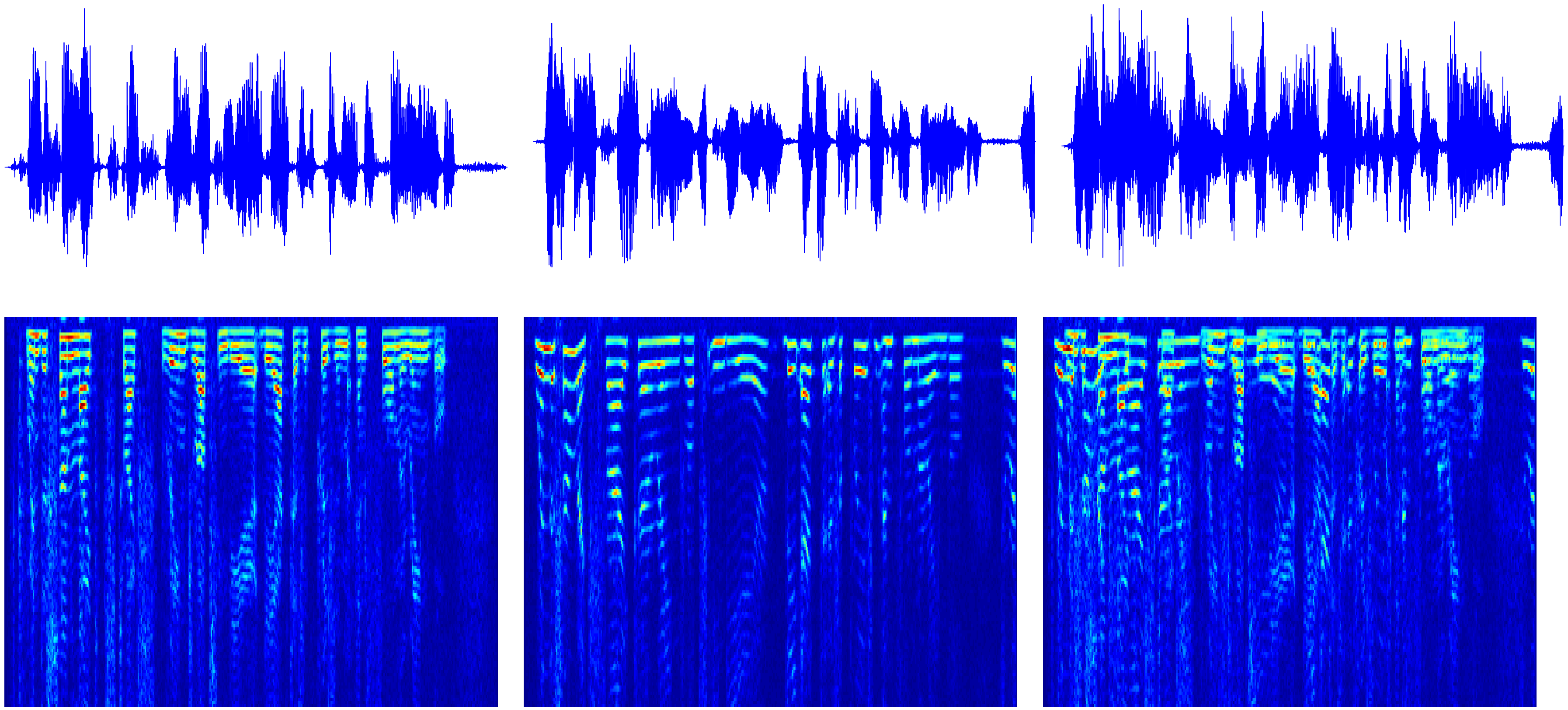


scatter plot of two signals



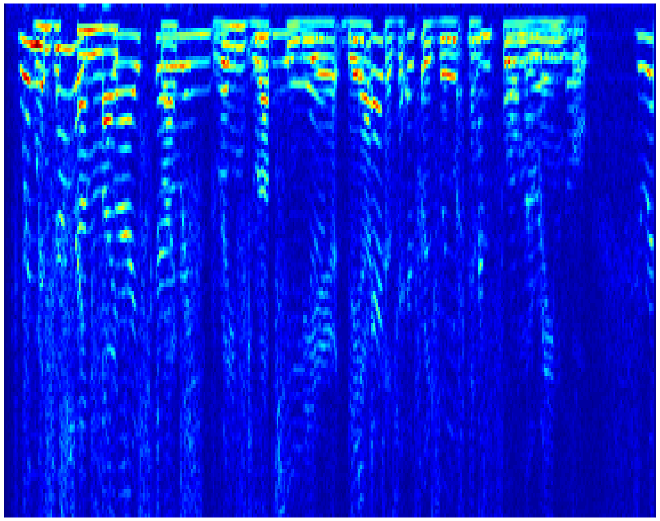
Sparsity and superposition

$$s_1 \qquad + \qquad s_2 \qquad = \qquad x$$

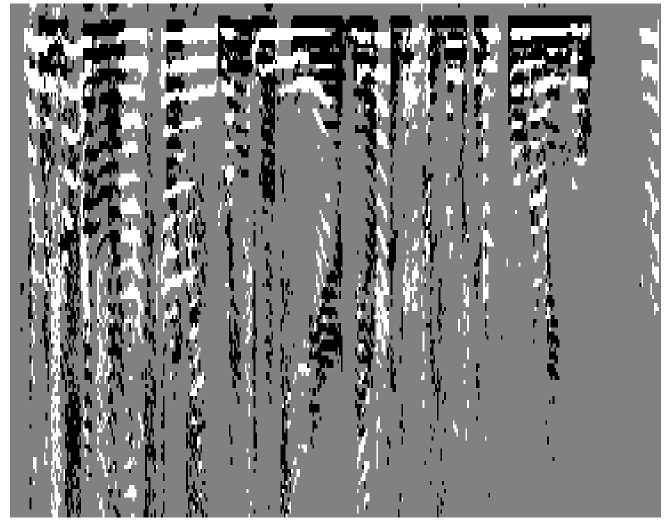


Building training set

Spectrogram of the mix



"Optimal" segmentation



- Empirical property: there exists a segmentation that leads to audibly acceptable signals (e.g., take $\arg\max(|S_1|, |S_2|)$)

- Work as possibly large training datasets

- Requires new way of segmenting images ...

- ... which can be learned from data

Summary of spectral clustering

Data: P elements $x_p \in \mathcal{X}$, $p = 1, \ldots, P$

$\Downarrow$

Step 1: build "affinity/similarity" matrix $W \in \mathbb{R}^{P \times P}$

$\Downarrow$

Step 2: normalize the affinity matrix: $\widetilde{W} = D^{-1/2} W D^{-1/2}$ where D is diagonal with sums of rows of W

$\Downarrow$

Step 3: compute the R largest eigenvectors $U(W) \in \mathbb{R}^{P \times R}$ of $\widetilde{W}$

$\Downarrow$

Step 4: considering $U(W)$ as P points in $\mathbb{R}^R$, cluster U using weighted K-means

$\Downarrow$

Output: partition E

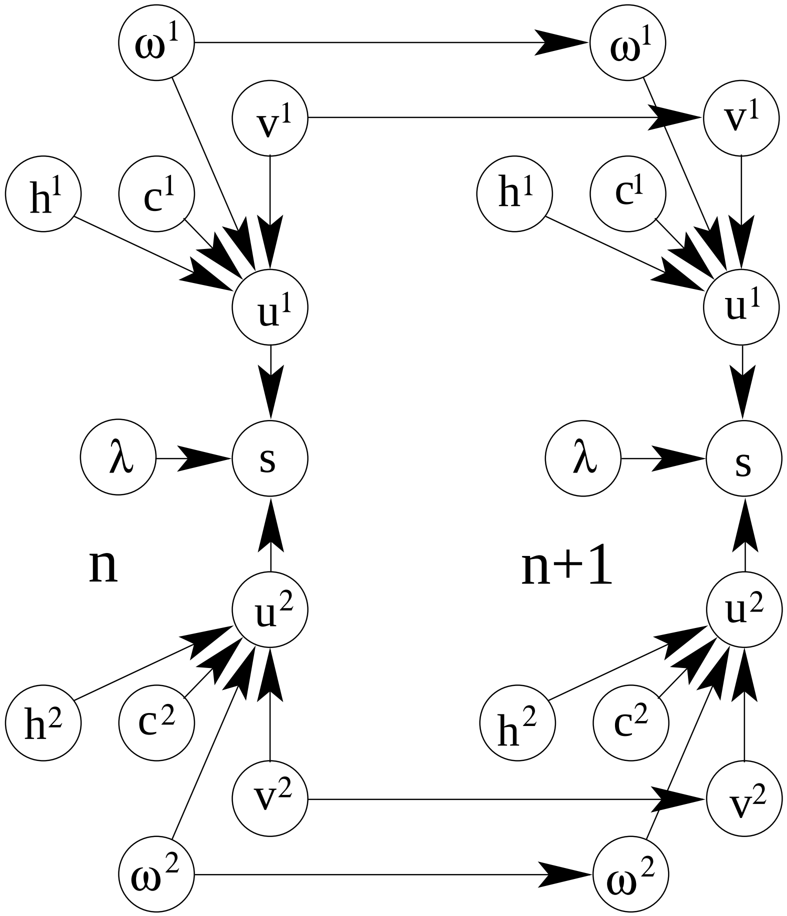
Learning problem

- **Input**:
  - spectrograms of mixed signals
  - “optimal” segmentations
- **Output**:
  - features for each spectrogram
  - Parameterized similarity matrix for spectral clustering
- **Challenges**:
  - Requires complex features
  - Large dimensionality of the spectrogram

Features for speech separation

- Classical cues from speech psychophysics
- Non-harmonic cues (similar to vision cues):
  - Continuity
  - Common fate cues
- Harmonic cues (requires different type of affinity matrices):
  - Pitch and potentially timbre
  - Requires multiple pitch estimation

Multiple pitch extraction

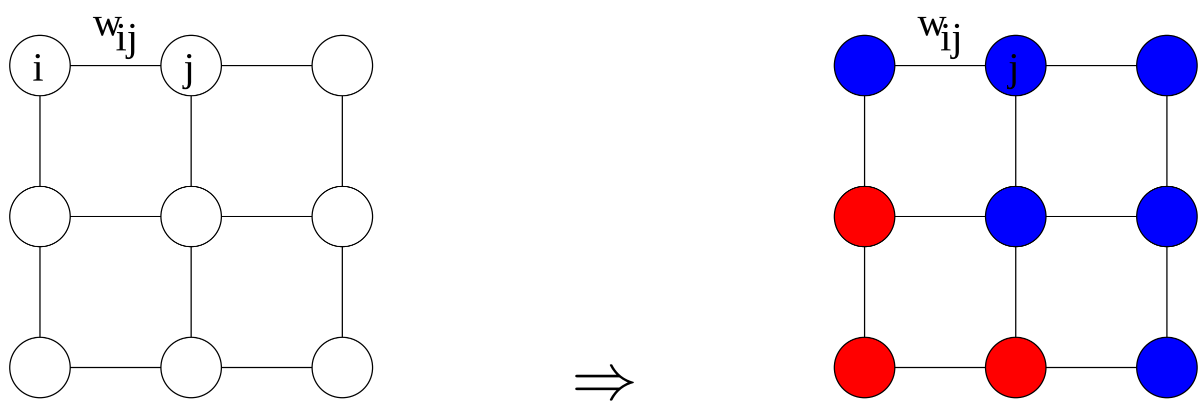


ω: pitch frequency
v: voicing decision
h: spectral envelope
c: constant unvoiced amplitude

- Additive model for the magnitude of the spectrogram
- Factorial HMM
- Smoothness prior on the spectral envelope
- Discriminative training
- Determination of number of speakers

Spectral graph partitioning

- P vertices of a weighted graph to partition into disjoint clusters



- Affinity matrix $W \in \mathbb{R}^{P \times P}$ ($W_{pp'}$ is large when points p and p' are likely to be in the same cluster)

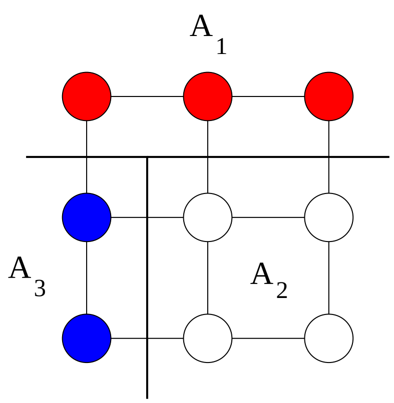
- **Goal**: find clusters with high intra-similarity and low inter-similarity

Normalized cuts

- Weight between two sets of vertices A and B, defined as:

$$W(A,B) = \sum_{i \in A, j \in B} W_{ij}$$

- (multi-way) normalized cut for partition $V = A_1 \cup \cdots \cup A_R$ (Shi and Malik, 2000, Zha et al, 2001):



$$J(A_1, \ldots, A_R, W) = \sum_{r=1}^{R} \frac{W(A_r, V \backslash A_r)}{W(A_r, V)}$$

$$J(A_1, A_2, W) = W(A_1, A_2) \left(\frac{1}{W(A_2,V)} + \frac{1}{W(A_1,V)} \right)$$

- Goal: minimize normalized cut

Learning spectral clustering

- Learning from fully segmented images (Bach & Jordan, NIPS 2004)
- Single cost function $J(W, E)$
  - Minimize with respect to the partition $E \Rightarrow$ spectral clustering
  - Minimize with respect to the matrix $W \Rightarrow$ learning similarities
- Uses the power method to approximate eigenvectors
- Requires parameterized affinity matrices

Very large similarity matrices

- Three different time scales $\Rightarrow W = \alpha_1 W_1 + \alpha_2 W_2 + \alpha_3 W_3$
- **Small**
  - Fine scale structure (continuity, harmonicity)
  - very sparse approximation
- **Medium**
  - Medium scale structure (common fate cues)
  - band-diagonal approximation, potentially reduced rank
- **Large**
  - Global structure (e.g., speaker identification)
  - low-rank approximation (rank is independent of duration)

Parameterized affinity matrices

- Non pitch-related features f_a, $a = 1, \ldots, P$.

$$W_{ab} = \exp(-||f_a - f_b||^{\beta})$$

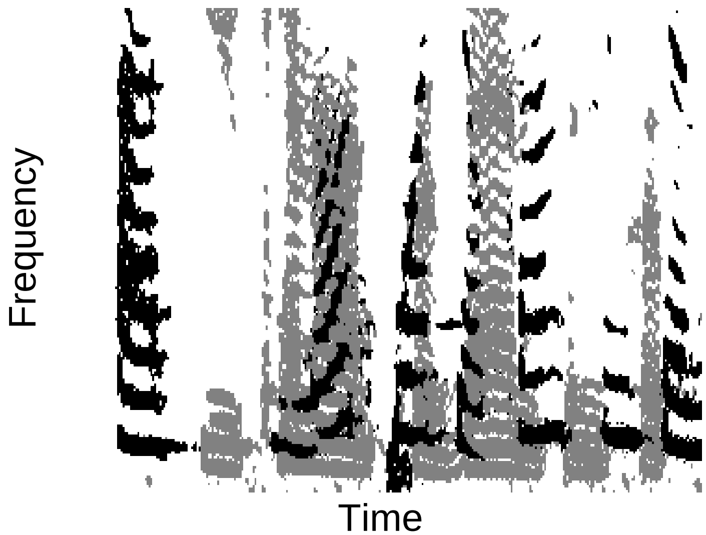
- Pitch related features
  - feature f_a, $a = 1, \ldots, P$
  - strength of pitch y_a:

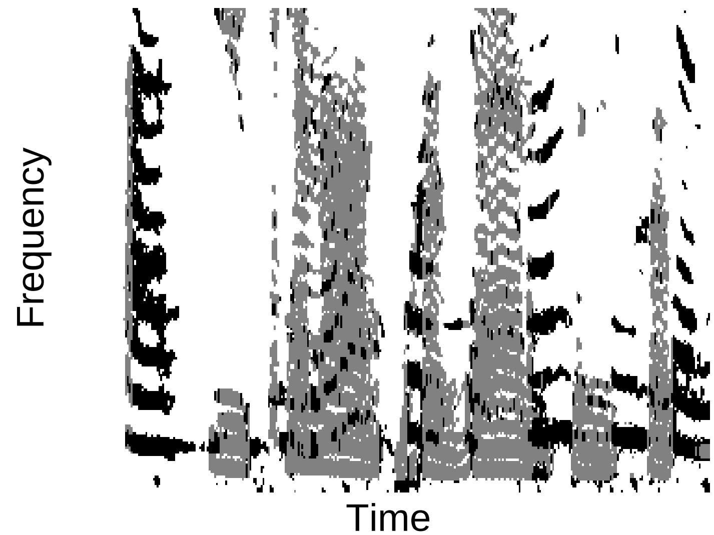
$$W_{ab} = \exp(-|g(y_a, y_b) + \beta_3|^{\beta_4} ||f_a - f_b||^{\beta_2})$$

where $g(u, v) = (ue^{\beta_5 u} + ve^{\beta_5 v})/(e^{\beta_5 u} + e^{\beta_5 v})$ ranges from the minimum of u and v for $\beta_5 = -\infty$ to their maximum for $\beta_5 = +\infty$.

Experiments

- Two datasets of speakers: one for testing, one for training
- Left: optimal segmentation - right: blind segmentation





- Testing time (linear in duration of signal): currently 30 minutes for 4 seconds of speech
- Speech samples on web site

Current work

- Mixing conditions: allow some form of delay or echo
- speaker vs. speaker $\Rightarrow$ speaker vs. non stationary noise
- Post processing of spectrogram segmentation
- Time and memory requirements