Codebook-based, Single Channel Blind Source Separation of Audio Signals

Guy Rapaport

Electrical Engineering department
Technion - Israel Institute of Technology

Supervised by Prof. Israel Cohen
Outline

- Introduction
  - Blind Source Separation (BSS)
  - Single Channel BSS
- Codebook-based Single Channel BSS
  - NMF/GMM/AR
- Separation Cost function
  - Frequency Dependent Separation
  - Distant Power Spectral Densities (PSDs)
- Experimental Study
- Conclusions
Blind Source Separation (1)

- Problem definition -

Separating N sources from M observations

\[ \mathbf{x}_1, \mathbf{x}_2 \rightarrow s_1, s_2, s_3 \]
Blind Source Separation (2)

- The separation problem depends on –
  - Mixing model:
    - Instantaneous (linear mixtures)
    - Un-echoic (introducing delays)
    - Echoic (reverberant environment)
Blind Source Separation (3)

- The separation problem depends on -
  - Number of sources ($N$) and observations ($M$)
    - Over-determined ($M \geq N$) scenario
      - Many separation methods exist
        - [O'Grady at el., 2005]
          - A survey of methods for source separation
    - Under-determined ($M < N$) scenario
      - Prior knowledge on the sources is needed
Single Channel BSS (1)

- The most extreme case of the under-determined separation problem
- Objective: separating two (or more) sources from their mixture
- Assumptions:
  - Instantaneous mixing model
  - Main focus on audio signals
Single Channel BSS (2)

- Single channel BSS requires priors for successful separation
- These priors may originate from -
  - Computational Auditory Scene Analysis (CASA)
  - Independent Component Analysis
  - Predefined Codebook of the sources
Single Channel BSS (3)

- CASA-based separation methods –
  - Mimic psycho-acoustics characteristics of the Human Auditory System
  - Use perceptual cues as heuristics in the source separation scheme

- ICA-based separation methods –
  - Adjusting the ICA-based solution from the over-determined realm into the single channel scenario
Single Channel BSS (4)

- CASA-based separation methods –
  - Examples:
    - [Roweis, 2001]
      Assumes only one signal is dominant per T-F bin
    - [Duan, 2004]
      Music separation according to harmonic structure
    - [Bach & Jordan, 2003]
      Source separation via spectral clustering. The clustering cost function is using a CASA-driven features
Single Channel BSS (5)

- ICA-based separation methods –
  - Examples:
    - [Jang & Lee, 2003]
      Describing each source, in the time domain, as a mixture of statistically independent components
    - [Beierholm et al., 2004]
      Similar to Jang and Lee, only in the DCT domain
    - [Mijovic et al., 2010]
      ICA-based separation method in the wavelet domain or following a dedicated data-driven transform
Codebook-based
Single Channel BSS (1)

- Prior –
  A codebook (CB) is used for representing each source
    - The CB describes the source according to a selected representation model
    - Requires an offline learning stage
      - Train signals - similar to the source in the mixture
      - Clustering the train observations into CBs
Codebook-based
Single Channel BSS (2)

- In the context of audio signals
  - Time domain representation –
    \[ x[n] \approx s_1[n] + s_1[n] \]
  - In the STFT domain –
    \[ X(f, t) \approx S_1(f, t) + S_2(f, t) \]

- Separation is achieved by estimating the sources’ power spectral densities (PSDs) that will best match the mixture’s PSD –
  \[ P_x(f, t) \approx P_1(f, t) + P_2(f, t) \]
Codebook-based Single Channel BSS (3)

- CB of PSDs:
  - The Source’s PSD are represented by a codebook of PSDs
    - The gain factors are non-negative
  - Aim to separate a quasi-stationary mixture with a time-varying combination of stationary spectral shapes

\[
P_1(f, t) = \sum_{i=1}^{K_1} a_i^1(t) \cdot \varphi_1^i(f)
\]

\[
P_2(f, t) = \sum_{j=1}^{K_2} a_j^2(t) \cdot \varphi_2^j(f)
\]
Codebook-based Single Channel BSS (4)

- Separation stages – overview:
Codebook-based
Single Channel BSS (5)

- Separation stage:

\[ P_1(f,t) = \sum_{i=1}^{K_1} \hat{a}_1^{i}(t) \cdot \phi_1^{i}(f) \]

\[ P_2(f,t) = \sum_{j=1}^{K_2} \hat{a}_2^{j}(t) \cdot \phi_2^{j}(f) \]
Codebook-based
Single Channel BSS (6)

- Several types of CB-based separation algorithms -
  - Non-negative Matrix Factorization (NMF)
  - Gaussian Mixture Model (GMM)
  - Auto Regressive (AR) model

- All are eventually evolving to CBs of PSDs
NMF-based Separation (1)

[Lee & Seung, 2001]

- Non-negative Matrix Factorization
  - Efficient decomposition method
    \[ P = B \cdot G \]
  - \( P, G, B \) – non-negative matrices

- Two cost functions
  - Frobenious norm -
    \[ \| P - BG \|_F^2 = \sum_{i,j} (P_{i,j} - (BG)_{i,j})^2 \]
  - KL Divergence -
    \[ \sum_{i,j} \left( P_{i,j} \cdot \log \frac{P_{i,j}}{(BG)_{i,j}} - P_{i,j} + (BG)_{i,j} \right) \]
NMF-based Separation (2)

- Non-negative Matrix Factorization (cont.)
  - Using a multiplicative update rule
    - \( \otimes \) - represent element-wise multiplication

Frobenious norm:

\[
\begin{align*}
B &= B \otimes \left( \frac{PG^T}{BGG^T} \right) \\
G &= G \otimes \left( \frac{B^TP}{B^T B G} \right)
\end{align*}
\]

KL divergence:

\[
\begin{align*}
B &= B \otimes \left( \frac{P}{BG} \frac{G^T}{1 \cdot G^T} \right) \\
G &= G \otimes \left( \frac{B^T P}{BG} \frac{B^T \cdot 1}{B^T \cdot 1} \right)
\end{align*}
\]
NMF-based Separation (3)

- Separation algorithm

\[ P_x(f, t) \approx P_1(f, t) + P_2(f, t) \]

- Where -

\[
\begin{align*}
P_1(f, t) &= \sum_{i=1}^{K_1} a_i^1(t) \cdot \varphi_i^1(f) \\
P_2(f, t) &= \sum_{j=1}^{K_2} a_j^2(t) \cdot \varphi_j^2(f)
\end{align*}
\]

- Matrix Notation -

\[ P_x \approx P_1 + P_2 = B_1 \cdot G_1 + B_2 \cdot G_2 = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \cdot \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \]
NMF-based Separation (4)

- Separation algorithm (cont.)
  - Basis matrix: columns contains the CB entries \( \phi_1^i(f)_{i=1}^{K_1}, \phi_2^j(f)_{j=1}^{K_2} \)
  - Gain matrix: rows contains the time-varying gains \( a_1^i(t)_{i=1}^{K_1}, a_2^j(t)_{j=1}^{K_2} \)

- Learning stage [Wang & Plumbley, 2006]
  - Run NMF on the training data to extract \( B_1, B_2 \) as the PSD CB of each source
NMF-based Separation Algorithmic Flow (1)

- Run NMF
  - Only update the gain matrix
- Estimate Sources’ PSD

\[
\begin{align*}
P_1(f, t) &= (B_1 G_1)_{f,t} \\
P_2(f, t) &= (B_2 G_2)_{f,t}
\end{align*}
\]
NMF-based Separation Algorithmic Flow (2)

- Wiener filtering

\[
\hat{S}_1(f,t) = \frac{P_1(f,t)}{P_1(f,t) + P_2(f,t)} \cdot X(f,t)
\]
GMM-based Separation (1)

[Benaroya et al., 2006]

- The sources are statistically independent
- Each source is represented by a GMM

\[ p(s) = \sum_{i=1}^{K} p(s \mid \theta_i) \cdot \Pr(\theta_i) \]

- \( s \mid \theta_i \sim N(0, \Sigma_i) \)
- \( \Pr(\theta_i) \) - Prior probability of \( \theta_i \)
- EM is used for training
GMM-based Separation (2)

- **Task:**
  Separating the mixture - \( x[n] = s_1[n] + s_2[n] \)

- **Simple case**
  - The active Gaussian components \( (\theta^i_1, \theta^i_2) \), are known
    - Both sources are Gaussian
  - Wiener filtering can be used for separation

\[
s_1 = \sum^i_1 \cdot \left( \sum^i_1 + \sum^j_2 \right)^{-1} \cdot x
\]
GMM-based Separation (3)

Simple case (cont.)

- STFT domain solution:
  - Assume: the signals are quasi-stationary and approximately circular

\[
\Sigma_1^i \rightarrow \varphi_1^i(f) , \Sigma_2^j \rightarrow \varphi_2^j(f) \\
\downarrow \\
S_1(f,t) = \frac{\varphi_1^i(f)}{\varphi_1^i(f) + \varphi_2^j(f)} \cdot X(f,t)
\]
GMM-based Separation (4)

- General case
  - The active components are unknown
    - Bayesian formalism
      \[
p(\theta_1^i, \theta_2^j | x) \propto p(x | \theta_1^i, \theta_2^j) \cdot \Pr(\theta_1^i) \cdot \Pr(\theta_2^j)
\]
  - MAP criterion
    \[
    (i^*, j^*) = \arg\max_{i,j} \{p(\theta_1^i, \theta_2^j | x)\}
    \]
  - We are back in the simple case scenario...
GMM-based Separation (5)

- General case (cont.)
  - STFT domain interpretation
    - Covariance matrices turns to CB of PSDs
      \[
      \{\Sigma_1^i\}_{i=1}^{K_1} \rightarrow \{\varphi_1^i(f)\}_{i=1}^{K_1}, \quad \{\Sigma_2^j\}_{j=1}^{K_2} \rightarrow \{\varphi_2^j(f)\}_{j=1}^{K_2}
      \]
    - MAP criterion
      \[
      \left(i^*(t), j^*(t)\right) = \arg\max_{i,j} \left\{ p\left(\mathbf{X}(f,t) | \theta_1^i, \theta_2^j\right) \cdot \Pr(\theta_1^i) \cdot \Pr(\theta_2^j) \right\}
      \]
    - No attention for gain estimation
GMM-based Separation (6)

- GSMM approach
  - Gaussian **Scaled** Mixture Model
  - Introducing a gain for each Gaussian component
    - If the gains \( \{a_i^j\}_{i=1}^K \geq 0 \) are known: GSMM \( \rightarrow \) GMM
    - With covariance matrices - \( \{a_i^j \cdot \Sigma_i^j\}_{i=1}^K \)
  - GSMM requires an additional gain estimation stage (prior to the pair selection)
    - ML criterion
      \[
      (\hat{a}_1^i(t), \hat{a}_2^j(t)) = \arg\max_{(a_1^i, a_2^j) \geq 0} \left\{ p\left( X(f, t) \mid \theta_1^i, \theta_2^j, a_1^i, a_2^j \right) \right\}
      \]
GMM-based Separation
Algorithmic Flow (1)

- **ML criterion**

\[
(\hat{a}_1(t), \hat{a}_2(t)) = \arg\max_{(a_1, a_2) \geq 0} \left\{ p(X(f, t) | \theta_1^i, \theta_2^j, a_1^i, a_2^j) \right\}
\]

- Solved via multiplicative update rule (NMF-like)
GMM-based Separation Algorithmic Flow (2)

- Choosing the optimal pair

\[
\left(i^*(t), j^*(t)\right) = \arg\max_{i,j} \left\{ p\left(X(f,t) | \theta^i, \theta^j, \hat{a}_1(t), \hat{a}_2(t)\right) \cdot \Pr(\theta^i) \cdot \Pr(\theta^j) \right\}
\]
GMM-based Separation
Algorithmic Flow (3)

- Wiener filtering

\[
\hat{S}_1(f,t) = \frac{\hat{a}_1^*(t) \cdot \varphi_1^*(f)}{\hat{a}_1^*(t) \cdot \varphi_1^*(f) + \hat{a}_2^*(t) \cdot \varphi_2^*(f)} \cdot X(f,t)
\]
AR-based Separation (1)

[Srinivasan et al., 2006]

- Originated from speech enhancement methods
  - Traditionally – noise is slowly changing in comparison to the speech signal
  - What if the undesired signal is changing rapidly?
    - A different prior is needed…

- Proposition:
  - At each time frame, model the sources as AR processes
  - A CB of AR processes is available for each source
AR-based Separation (2)

- **AR process**
  - Used for representing speech spectral shapes
  - **Definition:**
    
    AR of order P -
    \[
    s[n] = \sum_{i=1}^{P} \alpha_i \cdot s[n-i] + u[n]
    \]
    
    - \( \theta = \{\alpha_i\}_{i=1}^{P} \) - Linear Prediction Coefficients (LPC)
    - \( u[n] \sim N(0, \sigma^2) \)
    - \( \sigma^2 \) - Excitation variance
AR-based Separation (3)

- **AR process (cont.)**
  - **Spectral envelope**
    
    \[ P(f) = \frac{\sigma^2}{|A(f)|^2}, \quad A(f) = 1 + \sum_{i=n}^{P} \alpha_i \cdot e^{-2\pi j fn} \]

    - \( \varphi(f) = \frac{1}{|A(f)|^2} \) - Spectral shape

    - \( \sigma^2 \) - Gain factor (amplitude)
AR-based Separation (4)

- **Task:**
  
  Separating the mixture – \( x[n] = s_1[n] + s_2[n] \)

- **Solution:**
  
  - CB of AR processes for each source –
    \[
    \{\theta_i\}_{i=1}^{K_1}, \{\theta_j\}_{j=1}^{K_2} \rightarrow \{\phi_1(f)\}_{i=1}^{K_1}, \{\phi_2(f)\}_{j=1}^{K_2}
    \]

  - Learning stage
    
    - Clustering LPCs via Max Lloyd algorithm
AR-based Separation (5)

Solution (cont.)

ML criterion

\[
(i^*, j^*) = \arg\max_{(i,j)} \left\{ \max_{(\sigma_1^2, \sigma_2^2) \geq 0} \left\{ p(x | \theta_1^i, \theta_2^j, \sigma_1^2, \sigma_2^2) \right\} \right\}
\]

Find CB Representatives

Estimating gains
AR-based Separation (5)

Solution (cont.)

- STFT representation

\[
(i^*, j^*) = \arg \min_{(i,j)} \left\{ \min_{\{\sigma_1^2, \sigma_2^2\} \geq 0} \{D_{IS}\left(P_x(f,t), P_1(f,t) + P_2(f,t)\right)\}\right\}
\]

- Itakura-Saito distortion measure (\(D_{IS}\))
  - Widely used for measuring distance between PSDs

\[
D_{IS}(P_x, P_y) = \frac{1}{F} \sum_{f=0}^{F-1} \left[ \frac{P_x(f)}{P_y(f)} - \log\left(\frac{P_x(f)}{P_y(f)}\right) - 1 \right]
\]
AR-based Separation
Algorithmic Flow (1)

- ML criterion –

\[
\left( \hat{a}_1(t), \hat{a}_2(t) \right) = \arg \min \left\{ D_{IS} \left( P_x(f,t), P_1(f,t) + P_2(f,t) \right) \right\}
\]

\[
\left( a_1^i, a_2^j \right) \geq 0
\]

- Solved by linearization
Choosing the optimal pair

\[
(i^*, j^*) = \arg\min_{(i,j)} \{ D_{IS}(P_x(f,t), P_1(f,t) + P_2(f,t)) \}
\]

Where

\[
\begin{align*}
P_1(f,t) &= \hat{a}_1(t) \cdot \varphi_1(f) \\
P_2(f,t) &= \hat{a}_2(t) \cdot \varphi_2(f)
\end{align*}
\]
AR-based Separation
Algorithmic Flow (3)

- Wiener filtering

\[ \hat{S}_1(f,t) = \frac{\hat{a}_1^*(t) \cdot \varphi_1^*(f)}{\hat{a}_1^*(t) \cdot \varphi_1^*(f) + \hat{a}_2^*(t) \cdot \varphi_2^*(f)} \cdot X(f,t) \]
AR-based BSS
Extensions (1)

- Several suggested alterations TODO!!!
  - MMSE approach [Benaroya et al., 2006]
    - Similar to the MAP estimator (but more complicated)
    - All pairs are participating in the separation stage
      - Using a weighted combination of Wiener filters

\[
S_1(f, t) = \sum_{i,j} p(\theta_1^i, \theta_2^j \mid X(f, t)) \cdot \frac{\hat{a}_1^i(t) \cdot \varphi_1^i(f)}{\hat{a}_1^i(t) \cdot \varphi_1^i(f) + \hat{a}_2^j(t) \cdot \varphi_2^j(f)} \cdot X(f, t)
\]

- Separation quality – ~identical to MAP
AR-based BSS

Extensions (2)

- [Benaroya et al., 2003]
  Using Hidden Markov Model (HMM) in order to describe time-correlation between adjacent frames

- [Ozerov et al., 2005, 2007]
  On-line update of the sources’ CBs using EM for voice/music separation (Requires VAD)

- [Abramson & Cohen, 2008]
  Introducing a classification and estimation approach on-top the GMM-based separation method

- [Emiya et al., 2009]
  Learning a CB of the mixture instead of the sources

- [Litvin & Cohen, 2010]
  Working in the Bark-scale wavelet domain instead of STFT
CB Separation Methods - Observations (1)

- Separation in the STFT domain
  - CBs of PSDs
- CB entries selection
  - GSMM/AR methods seeks for the optimal pair
  - NMF allows all entries to be active
- Cost function
  - GSMM/AR both use the Itakura-Saito distortion measure (SHOW HOW)

\[
\arg\min_{(i,j)} \{ D_{IS} (P_x(f,t), P_1(f,t) + P_2(f,t)) \} = \arg\max_{(i,j)} \{ p(x \mid \theta_1^i, \theta_2^i, a_1^i, a_2^i) \}
\]
CB Separation Methods - Observations (2)

- Current separation results – not good enough!

- Improvements?
  - Diving into the separation cost function
Separation Cost Function (1)

- AR/GMM cost function

\[ D_{IS}\left( P_x(f,t), \frac{P_{1+2}(f,t)}{P_1(f,t) + P_2(f,t)} \right) = \frac{1}{F} \sum_{f=0}^{F-1} \left[ \frac{P_x(f)}{P_{1+2}(f)} - \log\left( \frac{P_x(f)}{P_{1+2}(f)} \right) - 1 \right] \]

- Observation
  - Each frequency bin is treated **identically**
Separation Cost Function (2)

- But -
  - Frequency bins with sufficient energy should be more “important” than noisy bins
  - Wiener filtering – accurate PSD estimation is not important where $|X(f,t)| \approx 0$
  - What if a signal is band limited?
  - CASA-motivated frequency differentiation
Frequency Dependent
Cost Function (1)

- Generalizing the cost function

\[ \tilde{D}_{IS}(P_x(f,t), P_{1+2}(f,t)) = \frac{1}{F} \sum_{f=0}^{F-1} \lambda_f \left[ \frac{P_x(f)}{P_{1+2}(f)} - \log \left( \frac{P_x(f)}{P_{1+2}(f)} \right) - 1 \right] \]

- \( \lambda_f \) \( \{ \lambda_f \}_{f=0}^{F-1} \) - frequency weights
- If \( \lambda_f = 1 \) , \( \forall f \) - we are back to the regular IS distortion measure
Frequency Dependent Cost Function (2)

- Probability function interpretation
  - Following the connection

\[
\text{argmin}_{(i,j)} \{ D_{IS}(P_x(f,t), P_{1+2}(f,t)) \} = \text{argmax}_{(i,j)} \left\{ p\left(X(f,t) \mid \theta_1^i, \theta_2^j, a_1^i, a_2^j \right) \right\}
\]

- Where -

\[
p\left(X(f,t) \mid \theta_1^i, \theta_2^j, a_1^i, a_2^j \right) \propto \prod_{f=0}^{F-1} \left\{ \left[ P_{1+2}(f,t) \right]^{-\frac{1}{2}} \cdot \exp \left( -\frac{P_x(f,t)}{2P_{1+2}(f,t)} \right) \right\}
\]

- Turns to -

\[
\tilde{p}\left(X(f,t) \mid \theta_1^i, \theta_2^j, a_1^i, a_2^j \right) \propto \prod_{f=0}^{F-1} \left\{ \left[ P_{1+2}(f,t) \right]^{-\frac{1}{2}} \cdot \exp \left( -\frac{P_x(f,t)}{2P_{1+2}(f,t)} \right) \right\}^{2f}
\]
Frequency Dependent
Cost Function (3)

- Probability function interpretation (cont.)

\[
\tilde{p}(X(f,t) | \theta_1^i, \theta_2^j, a_1^i, a_2^j) \propto \prod_{f=0}^{F-1} \left[ P_{1+2}(f,t) \right]^{-\frac{1}{2}} \cdot \exp \left(-\frac{P_x(f,t)}{2P_{1+2}(f,t)}\right)^{\lambda_f}
\]

- Each Gaussian component is weighted according to \(\{\lambda_f\}_{f=0}^{F-1}\)
Frequency Dependent Cost Function (4)

- How to choose $\{\lambda_f\}_{f=0}^{F-1}$?
  - According to the observed PSD of the mixture
    - Example
  - According to the learning stage
    - Identifying the spectral content of each source
Following the cost function alteration

- New separation algorithm evolves
  - Based on GSMM-MAP separation method
  - Can also be applied to AR-based separation methods
- Estimates the sources’ PSDs while giving different attention to each frequency bin
Frequency Dependent Separation Algorithm (2)

- ML criterion

\[
\left( \hat{a}_1^i(t), \hat{a}_2^j(t) \right) = \arg\max_{\left( a_1^i, a_2^j \right) \geq 0} \{ \tilde{p}(X(f,t) | \theta_1^i, \theta_2^j, a_1^i, a_2^j) \}
\]

- Solved via multiplicative update rule
  - Similarly to GSMM-MAP
Frequency Dependent Separation Algorithm (3)

Choosing the optimal pair

\[ (i^*(t), j^*(t)) = \arg\max_{i,j} \{ \tilde{p}(X(f, t) | \theta_1^i, \theta_2^j, \hat{a}_1^i(t), \hat{a}_2^j(t)) \cdot \Pr(\theta_1^i) \cdot \Pr(\theta_2^j) \} \]
**Frequency Dependent Separation Algorithm (4)**

- Wiener filtering

\[
\hat{S}_1(f, t) = \frac{\hat{a}_1^*(t) \cdot \varphi_1^*(f)}{\hat{a}_1^*(t) \cdot \varphi_1^*(f) + \hat{a}_2^*(t) \cdot \varphi_2^*(f)} \cdot X(f, t)
\]
Separation Cost Function (1)

- **GMM MAP criterion**
  \[
  (i^*, j^*) = \arg\max_{i, j} \{ p(x | \theta^i_1, \theta^j_2, \hat{a}^i_1, \hat{a}^j_2) \cdot \Pr(\theta^i_1) \cdot \Pr(\theta^j_2) \}
  \]

- **ML term**
  - minimize - \( D_{IS}(P_x(f, t), P_{1+2}(f, t)) \)

- **Priors**
  - Prior probability for each CB entry
  - Sources are statistically independent
Separation Cost Function (2)

- Are these priors sufficient?
  - Hint - [Benaroya & Bimbot, 2003]
    Using de-correlation as post-processing for improved separation result

- Actual separation

\[ D_{IS}(P_x(f,t), P_{1+2}(f,t)) \]

- Aims to match the observed PSD - \[ P_x(f,t) \]
  with the sources’ estimated PSD - \[ P_{1+2}(f,t) \]
Separation Cost Function (3)

- Which is better separated?
Separation with Distant PSDs Prior (1)

- Introducing an additional prior -

\[
(i^*, j^*) = \underset{i, j}{\operatorname{argmax}} \left\{ p(x | \theta_1^i, \theta_2^j, \hat{a}_1^i, \hat{a}_2^j) \cdot \frac{p(P_1(f, t), P_2(f, t))}{\Pr(\theta_1^i) \cdot \Pr(\theta_2^j)} \right\}
\]

- Intention -
  - The separated signals should be as ‘distant’ as possible
  - Compare the estimated PSDs of the sources
    - Disregard similar PSDs
Separation with Distant PSDs Prior (2)

\[ p(P_1(f, t), P_2(f, t)) : \]

- High probability for distant PSDs
- Low probability for similar PSDs

Examples –

- \( L_2 \) prior:
  \[ p(P_1(f, t), P_2(f, t)) \propto \exp \left[ \frac{\gamma}{2} \| P_1(f, t) - P_2(f, t) \|_2^2 \right] \]

- Itakura-Saito prior:
  \[ p(P_1(f, t), P_2(f, t)) \propto \exp \left[ \gamma \cdot \frac{F}{2} \cdot D_{IS}(P_1(f, t), P_2(f, t)) \right] \]
Separation with Distant PSDs Prior (3)

- **Cost function alteration (Gain estimation)**
  - **L₂** prior:
    \[
    \left( \hat{a}_1(t), \hat{a}_2(t) \right) = \arg\min_{(a_1^i, a_2^i) \geq 0} \left\{ D_{IS}(P_x(f, t), P_{1+2}(f, t)) - \frac{\gamma}{F} \|P_1(f, t) - P_2(f, t)\|_2^2 \right\}
    \]
  - **Itakura-Saito** prior:
    \[
    \left( \hat{a}_1(t), \hat{a}_2(t) \right) = \arg\min_{(a_1^i, a_2^i) \geq 0} \left\{ D_{IS}(P_x(f, t), P_{1+2}(f, t)) - \gamma \cdot D_{IS}(P_1(f, t), P_2(f, t)) \right\}
    \]
    - \( \gamma \) - Lagrange multiplier
    - \( \gamma = 0 \) - back to regular GSMM
Distant PSDs Prior
Separation Algorithm (1)

Following the cost function alteration

- New separation algorithm evolves
  - With the Itakura-Saito prior
  - Based on GSMM-MAP separation method
  - Can also be applied to AR-based separation methods

- Cost function
  - Match Mixture’s PSD to the sources’ PSDs
  - Favor distant sources’ PSDs
Distant PSDs Prior Separation Algorithm (2)

- **ML criterion**

\[
\left( \hat{a}_1(t), \hat{a}_2(t) \right) = \arg\min_{\left( a_1, a_2 \right) \geq 0} \left\{ D_{IS} \left( P_x (f, t), P_{1+2} (f, t) \right) - \gamma \cdot D_{IS} \left( P_1 (f, t), P_2 (f, t) \right) \right\}
\]

- Solved via gradient descent algorithm
Distant PSDs Prior Separation Algorithm (3)

- Choosing the optimal pair

\[
(i^*, j^*) = \arg\max_{i,j} \left\{ p(x \mid \theta_1^i, \theta_2^j, \hat{a}_1^i, \hat{a}_2^j) \cdot p(P_1(f, t), P_2(f, t)) \cdot \Pr(\theta_1^i) \cdot \Pr(\theta_2^j) \right\}
\]
Distant PSDs Prior Separation Algorithm (4)

- Wiener filtering

\[
\hat{S}_1(f,t) = \frac{\hat{a}_1^*(t) \cdot \varphi_1^*(f)}{\hat{a}_1^*(t) \cdot \varphi_1^*(f) + \hat{a}_2^*(t) \cdot \varphi_2^*(f)} \cdot X(f,t)
\]
Experimental Study

- Simulation on real audio signals
- Two separation experiments
  - Speech (TIMIT) & piano (from the web)
  - Speech (TIMIT) & drums (from the web)
Experimental Study

- Distortion measures
- Scenario
- Comparing GMM/AR/NMF
- Comparing GMM to our extensions
Conclusion

- Summary
- Future directions
NMF-based BSS
Extensions (1)

- Continuity priors
  - [Smargadis, 2007]
    Introducing convolutive NMF for incorporating time dependencies into the NMF framework
  - [Virtanen, 2007]
    Introducing continuity constraints on the gain factors into the NMF cost function

- Sparsity priors
  - [Virtanen, 2007]
    Introducing Sparsity priors on the gain matrix into the NMF cost function
  - [Virtanen, 2008]
    Using the Sparse NMF (SNMF) for single channel source separation
NMF-based BSS

Extensions (2)

- Complex NMF
  - [Kameoka et al., 2009; King & Atlas 2010]
    Working on the STFT domain, instead of directly on the PSDs

- CASA-driven NMF
  - [Virtanen, 2007]
    Weighting frequency bins according to the loudness perception
  - [Kirbiz & Gunsel 2010]
    Pre-emphasizing frequency bands that are important for the Human Auditory System (HAS)
GMM-based BSS Extensions (1)

- Several suggested alterations
  - MMSE approach [Benaroya et al., 2006]
    - Similar to the MAP estimator (but more complicated)
    - All pairs are participating in the separation stage
      - Using a weighted combination of Wiener filters

\[
S_1(f, t) = \sum_{i,j} p(\theta_1^i, \theta_2^j | X(f, t)) \cdot \frac{\hat{a}_1^i(t) \cdot \varphi_1^i(f)}{\hat{a}_1^i(t) \cdot \varphi_1^i(f) + \hat{a}_2^j(t) \cdot \varphi_2^j(f)} \cdot X(f, t)
\]

- Separation quality – \(~\)identical to MAP
GMM-based BSS

Extensions (2)

- [Benaroya et al., 2003]
  Using Hidden Markov Model (HMM) in order to describe time-correlation between adjacent frames

- [Ozerov et al., 2005, 2007]
  On-line update of the sources’ CBs using EM for voice/music separation (Requires VAD)

- [Abramson & Cohen, 2008]
  Introducing a classification and estimation approach on-top the GMM-based separation method

- [Emiya et al., 2009]
  Learning a CB of the mixture instead of the sources

- [Litvin & Cohen, 2010]
  Working in the Bark-scale wavelet domain instead of STFT
Learning Stage

Off-line

Train Signals

\[
\begin{align*}
\{s_1[n]\} & \rightarrow \text{STFT} & S_1(f,t) \\
\{s_2[n]\} & \rightarrow \text{STFT} & S_2(f,t)
\end{align*}
\]

\[\{\varphi^i(f)\}_{i=1}^{K_1}, \{\varphi^j(f)\}_{j=1}^{K_2}\]

\[\text{Learning Stage}\]

\[\{\varphi^i(f)\}_{i=1}^{K_1}, \{\varphi^j(f)\}_{j=1}^{K_2}\]

CBs of PSDs

On-line

Mixture Signal

\[x[n]\]

\[X(f,t)\]

\[\text{STFT}\]

\[\hat{S}_1(f,t)\]

\[\hat{S}_2(f,t)\]

\[\text{Separation Stage}\]

\[\{\varphi^i(f)\}_{i=1}^{K_1}, \{\varphi^j(f)\}_{j=1}^{K_2}\]

\[\text{ISTFT}\]

\[\hat{s}_1[n]\]

\[\hat{s}_2[n]\]

\[\text{Estimated Signals}\]
Calculate Gains

$X(f, t)$

Identify Active CB Entries

Extract Sources

$S_1(f, t)$

$S_2(f, t)$

$P_1(f, t)$

$P_2(f, t)$
Calculate Gains

Identify Active CB Entries

Extract Sources

$X(f,t)$

$\{\hat{a}_i(t)\}_{i=1}^{K_1}$

$\{\hat{a}_j(t)\}_{j=1}^{K_2}$

$P_1(f,t)$

$P_2(f,t)$

$\hat{S}_1(f,t)$

$\hat{S}_2(f,t)$
Calculate Gains

Identify Active CB Entries

Extract Sources

$X(f,t)$

$\{\hat{a}_1^i(t)\}_{i=1}^{K_1}$

$\{\hat{a}_2^j(t)\}_{j=1}^{K_2}$

$P_1(f,t)$

$P_2(f,t)$

$\hat{S}_1(f,t)$

$\hat{S}_2(f,t)$
\[ P_{\min} \leq P \leq P_{\max} \]

\[ \lambda_f \]