Blind Source Separation using Non-negative Matrix Factorization: FASST

Jordi Pons i Puig

May, 2015
Outline

Background
- NMF
- Cost function
- PSA

FASST
- Objectives and source model
- Optimization problem: decomposition

Implementation

Evaluation

Try it!
Outline

Background
- NMF
- Cost function
- PSA

FASST
- Objectives and source model
- Optimization problem: decomposition

Implementation

Evaluation

Try it!

Thanks
Non-negative Matrix Factorization *(Lee and Seung, 2001)*

\[ V \approx \hat{V} = WH \]

- \( \hat{V} \) and \( V \) are spectrograms \( \in \mathbb{R}_{\geq 0}^{M \times N} \)
- \( W \) are the bases where \( \in \mathbb{R}_{\geq 0}^{M \times K} \)
- \( H \) is the activation matrix where \( H \in \mathbb{R}_{\geq 0}^{K \times N} \)
- Positive \( \rightarrow \) Meaningful

Example:
Considering $\beta$-divergence as cost function

Definition:

$$d_\beta(x|y) = \begin{cases} \frac{1}{\beta(\beta-1)} (x^\beta + (\beta - 1)y^\beta - \beta xy^{\beta-1}) & \beta \in \mathbb{R}\{0, 1\} \\ x (\log x - \log y) + (y - x) & \beta = 1 \\ \frac{x}{y} - \log \frac{x}{y} - 1 & \beta = 0 \end{cases}$$

Observations:

▶ Note that for different $\beta$ some common cost functions are defined:

▶ $\beta = 2$: Euclidean distance
▶ $\beta = 1$: generalised Kullback-Leibler divergence
▶ $\beta = 0$: Itakura-Saito divergence

Gradient Descend and Multiplicative Updates:

▶ $b = a - \gamma \nabla F(a)$
▶ for $\gamma$ small enough, then $F(a) \geq F(b)$
Prior Subspace Analysis: composing $W$

$W$ is formed by two parts:

- **Fixed Trained Patterns** ($W_{tar}$) where $W_t \in \mathbb{R}^{M \times K_{tar}}_{\geq 0}$
- **Adaptive Patterns** ($W_{bgnd}$) where $W_t \in \mathbb{R}^{M \times K_{bgnd}}_{\geq 0}$

(Note that $K_{tar} + K_{bgnd} = K$)

Example:
Outline

Background
- NMF
- Cost function
- PSA

FASST
- Objectives and source model
- Optimization problem: decomposition

Implementation

Evaluation

Try it!
Objectives:

- Flexible Audio Source Separation Toolbox (FASST) (Ozerov et al. 2012) is a general purpose framework able to model SS problems.

FASST: source model (I). Based on an excitation-filter model:

\[ \mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{W} \mathbf{H} = \sum_j \mathbf{V}_j; \mathbf{V}_j = \mathbf{V}_j^{\text{ex}} \odot \mathbf{V}_j^{\text{ft}} \]
FASST: source model (II). Based on an excitation-filter model:

\[ V_j = V_j^{ex} \odot V_j^{ft} = E_j^{ex} P_j^{ex} \odot E_j^{ft} P_j^{ft} = \]

\[ = (W_j^{ex} U_j^{ex} G_j^{ex} H_j^{ex}) \odot (W_j^{ft} U_j^{ft} G_j^{ft} H_j^{ft}) \]

\[ V_j^{ex} \] is further decomposed as:

- **Spectral patterns** \( E_j^{ex} \) are modulated by **time activations** \( P_j^{ex} \).

**Spectral patterns** \( E_j^{ex} \) are linear combinations of:

- **Narrow band spectral patterns** \( W_j^{ex} \) with weights \( U_j^{ex} \).

**Time activations** \( P_j^{ex} \) are a combination of:

- time-localized patterns \( H_j^{ex} \) with weights \( G_j^{ex} \).

The filter spectral power \( V_j^{ft} \) is similarly expressed.

**EXTREMELLY VERSATILE!** Fixed and learnt basis.
FASST: source model (III). Based on an excitation-filter model:

\[
V_j = V_j^{ex} \odot V_j^{ft} = E_j^{ex} P_j^{ex} \odot E_j^{ft} P_j^{ft} = \\
= (W_j^{ex} U_j^{ex} G_j^{ex} H_j^{ex}) \odot (W_j^{ft} U_j^{ft} G_j^{ft} H_j^{ft})
\]
FASST estimation of the sources is achieved in two steps:

- **Step 1.** Given initial parameter values $\theta_{\text{init}}$, the model parameters $\theta$ are estimated from the mixture $\mathbf{V}$ using an iterative Generalized Expectation Maximization (GEM) algorithm; where each iteration is achieved by means of Multiplicative Update (MU) rules inspired from the NMF methodology.

- **Step 2.** Given the mixture $\mathbf{V}$ and the estimated model parameters $\theta$, source estimates $\mathbf{V}_j$ are computed using Wiener filtering.
Blind Source Separation using Non-negative Matrix Factorization: FASST

Jordi Pons i Puig

Background

NMF

Cost function

PSA

FASST

Objectives and source model

Optimization problem: decomposition

Implementation

Evaluation

Try it!

Thanks

Decomposition block diagram
Outline

Background
  NMF
  Cost function
  PSA

FASST
  Objectives and source model
  Optimization problem: decomposition

Implementation

Evaluation

Try it!

Thanks
Implementation details (I)

- FASST is set for separating the bass, drums, melody and the remaining instruments from a stereo music recording.
- Is available online written in Matlab.

Relies on two main concepts:
- Having prior knowledge about the sources to label them automatically (PSA).
- Joint modelling of all sources instead of a cascading approach (separating the drums, then the melody, etc.).

The sound scene is modelled with 12 sources ($V_j$):
- 4 sources representing the bass.
- 4 sources drums.
- 4 sources modelling other instruments.
Implementation details (II)

The bass and drums sources are constrained to \( V_j = W^e_j G^e_j \) with:

- \( G^e_j \) being adaptive
- \( W^e_j \) being fixed and pre-trained (with the same framework) from isolated bass and drum samples from the RWC music database.

The melody and to the remaining instruments are constrained to \( V_j = W^e_j U^e_j G^e_j \) where:

- \( U^e_j \) and \( G^e_j \) are adaptive.
- \( W^e_j \) is a fixed dictionary representing the narrow band spectral patterns modelling the harmonic pitch.

Considering:

- ERB spectrogram (providing higher low-frequency resolution than the STFT desired for modelling drums and bass).
- With random initialization to all the adaptive parameters.

The melody is reconstructed by choosing the most energetic source of the 4 sources modelling other instruments.
Outline

Background
- NMF
- Cost function
- PSA

FASST
- Objectives and source model
- Optimization problem: decomposition

Implementation

Evaluation

Try it!
BSS Evaluation (I)

- Emmanuel Vincent, Remi Gribonval, Cedric Fevotte. *Performance measurement in blind audio source separation*. IEEE Transactions on Audio, Speech and Language Processing, Institute of Electrical and Electronics Engineers (IEEE), 2006.

- Objective measures accepted for the community.

Formalization of the problem:

\[
\hat{s}_{\text{target}}(n) = s_{\text{target}}(n) + e_{\text{interf}}(n) + e_{\text{noise}}(n) + e_{\text{artif}}(n)
\]
BSS Evaluation (II)

Formalization of the objective measures:

- **Source to Distortion Ratio (SDR):**
  
  $$SDR = 10 \log_{10} \frac{\|s_{target}(n)\|^2}{\|e_{interf}(n) + e_{noise}(n) + e_{artif}(n)\|^2}$$

- **Source to Interference Ratio (SIR):**
  
  $$SIR = 10 \log_{10} \frac{\|s_{target}(n)\|^2}{\|e_{interf}(n)\|^2}$$

- **Sources to Noise Ratio (SNR):**
  
  $$SNR = 10 \log_{10} \frac{\|s_{target}(n) + e_{interf}(n)\|^2}{\|e_{noise}(n)\|^2}$$

- **Sources to Artifacts Ratio (SAR):**
  
  $$SAR = 10 \log_{10} \frac{\|s_{target}(n) + e_{interf}(n) + e_{noise}(n)\|^2}{\|e_{artif}(n)\|^2}$$
Outline

Background
- NMF
- Cost function
- PSA

FASST
- Objectives and source model
- Optimization problem: decomposition

Implementation

Evaluation

Try it!
Download and use FASST

- http://bass-db.gforge.inria.fr/fasst/
- Download Version 1 (Matlab)
- Uncompress
- Run:
  ```
  EXAMPLE_prof_rec_sep_drums_bass_melody_FULL('Shannon_Hurley_Sunrise_prof_prod_mix.wav', 'example_data',
  'example_data')
  ```
- And... HAVE FUN! :)
Thanks for your attention!