

Phase recovery with Bregman divergences for audio source separation

Paul Magron, Pierre-Hugo Vial, Thomas Oberlin, Cédric Févotte

CNRS, IRIT, Université de Toulouse, France

ICASSP2021



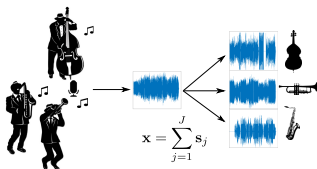
Audio source separation

- . Audio signals are composed of several constitutive sounds: multiple speakers, background noise, domestic sounds, musical instruments...

Audio source separation

- . Audio signals are composed of several constitutive sounds: multiple speakers, background noise, domestic sounds, musical instruments...

Source separation = recovering the sources from the mixture.

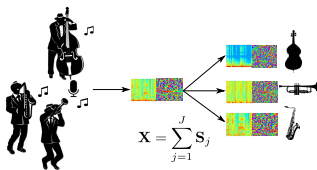


- . Automatic speech recognition (clean speech vs. noise).
- . Rhythm analysis (drums vs. harmonic instruments).
- . Time-stretching (transients vs. partials).

Audio source separation

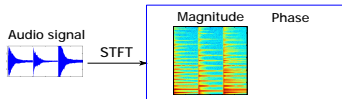
- . Audio signals are composed of several constitutive sounds: multiple speakers, background noise, domestic sounds, musical instruments...

Source separation = recovering the sources from the mixture.



- . Automatic speech recognition (clean speech vs. noise).
- . Rhythm analysis (drums vs. harmonic instruments).
- . Time-stretching (transients vs. partials).

Time-frequency separation = acts on the short-time Fourier transform (STFT).



General framework

1. Nonnegative representation, e.g., $\mathbf{V} = |STFT(\mathbf{x})|^2$.

General framework

1. Nonnegative representation, e.g., $\mathbf{V} = |\text{STFT}(\mathbf{x})|^2$.
2. Structured model, e.g., nonnegative matrix factorization, deep neural networks.

1. Nonnegative representation, e.g., $\mathbf{V} = |\text{STFT}(\mathbf{x})|^2$.
2. Structured model, e.g., nonnegative matrix factorization, deep neural networks.
3. Nonnegative masking and synthesis:
 $\mathbf{s}_j = \text{STFT}^{-1}(\mathbf{M}_j \mathbf{X})$.

The phase problem

Nonnegative masking: $\setminus S_j = \setminus X$.

7 Issues in sound quality when sources overlap.

7 *Inconsistency*: $\hat{S}_j \notin \text{STFT}(\mathbb{R}^N)$.

The phase problem

Nonnegative masking: $\setminus \mathbf{S}_j = \setminus \mathbf{X}$.

7 Issues in sound quality when sources overlap.

7 *Inconsistency*: $\hat{\mathbf{S}}_j \not\subseteq \text{STFT}(\mathbb{R}^N)$.

Multiple Input Spectrogram Inversion (MISI) [Gunawan, 2010]

. Extends the Griffin-Lim algorithm to multiple signals by solving:

$$\min_{\mathbf{s}_j} \sum_{j=1}^J \|\mathbf{V}_j^H \text{STFT}(\mathbf{s}_j)\|_2^2 \quad \text{s.t.} \quad \sum_{j=1}^J \mathbf{s}_j = \mathbf{x}$$

The phase problem

Nonnegative masking: $\setminus \mathbf{S}_j = \setminus \mathbf{X}$.

7 Issues in sound quality when sources overlap.

7 *Inconsistency*: $\hat{\mathbf{S}}_j \not\subseteq \text{STFT}(\mathbb{R}^N)$.

Multiple Input Spectrogram Inversion (MISI) [Gunawan, 2010]

. Extends the Griffin-Lim algorithm to multiple signals by solving:

$$\min_{\mathbf{s}_j} \sum_{j=1}^{\mathcal{J}} \|\mathbf{V}_j^{-1} \text{STFT}(\mathbf{s}_j)\|_2^2 \quad \text{s.t.} \quad \sum_{j=1}^{\mathcal{J}} \mathbf{s}_j = \mathbf{x}$$

3 Performance is improved over masking.

The phase problem

Nonnegative masking: $\setminus \mathbf{S}_j = \setminus \mathbf{X}$.

7 Issues in sound quality when sources overlap.

7 *Inconsistency*: $\hat{\mathbf{S}}_j \not\subseteq \text{STFT}(\mathbb{R}^N)$.

Multiple Input Spectrogram Inversion (MISI) [Gunawan, 2010]

. Extends the Griffin-Lim algorithm to multiple signals by solving:

$$\min_{\mathbf{s}_j} \sum_{j=1}^{\mathcal{J}} \|\mathbf{V}_j^H \text{STFT}(\mathbf{s}_j)\|_k^2 \text{ s.t. } \sum_{j=1}^{\mathcal{J}} \mathbf{s}_j = \mathbf{x}$$

3 Performance is improved over masking.

7 Euclidean distance is not the most appropriate in audio.

The phase problem

Nonnegative masking: $\setminus \mathbf{S}_j = \setminus \mathbf{X}$.

7 Issues in sound quality when sources overlap.

7 *Inconsistency*: $\hat{\mathbf{S}}_j \not\subseteq \text{STFT}(\mathbb{R}^N)$.

Multiple Input Spectrogram Inversion (MISI) [Gunawan, 2010]

. Extends the Griffin-Lim algorithm to multiple signals by solving:

$$\min_{\mathbf{s}_j} \sum_{j=1}^{\mathcal{J}} \|\mathbf{V}_j^T \text{STFT}(\mathbf{s}_j)\|_k^2 \quad \text{s.t.} \quad \sum_{j=1}^{\mathcal{J}} \mathbf{s}_j = \mathbf{x}$$

3 Performance is improved over masking.

7 Euclidean distance is not the most appropriate in audio.

Goal

Extend MISI to non-quadratic losses for source separation.

Proposed method

Problem setting

Bregman divergences

$$D(\mathbf{P} \parallel \mathbf{Q}) = \sum_{f;t} \left(p_{f;t} - q_{f;t} \right) \phi(q_{f;t})$$

- . The generating function ϕ determines the divergence.
- . Encompass the α -divergences, with particular cases: Euclidean ($\alpha = 2$), Kullback-Leibler ($\alpha = 1$) and Itakura-Saito ($\alpha = 0$) [Hennequin, 2011]

Problem setting

Bregman divergences

$$D(\mathbf{P} \parallel \mathbf{Q}) = \sum_{f;t} (p_{f;t} - q_{f;t}) \phi(q_{f;t})$$

- . The generating function ϕ determines the divergence.
- . Encompass the α -divergences, with particular cases: Euclidean ($\alpha = 2$), Kullback-Leibler ($\alpha = 1$) and Itakura-Saito ($\alpha = 0$) [Hennequin, 2011]

Problem formulation: $\min_{\mathbf{s}_j} \sum_{j=1}^J C_j(\mathbf{s}_j)$ s.t. $\sum_{j=1}^J \mathbf{s}_j = \mathbf{x}$

- . Accounting for the non-symmetry of Bregman divergences:

$$C_j(\mathbf{s}_j) = \underbrace{D(\mathbf{V}_j \parallel \{\text{STFT}(\mathbf{s}_j)\}^d)}_{\text{"right" problem}} \quad \text{or} \quad \underbrace{D(\{\text{STFT}(\mathbf{s}_j)\}^d \parallel \mathbf{V}_j)}_{\text{"left" problem}}$$

- . $d = 1$ (\mathbf{V}_j are magnitudes) or $d = 2$ (\mathbf{V}_j are power spectrograms).

Projected gradient descent

$$\min_{\mathbf{s}_j} \sum_{j=1}^J C_j(\mathbf{s}_j) \quad \text{s.t.} \quad \sum_{j=1}^J \mathbf{s}_j = \mathbf{x}$$

Data fitting Mixing constraint

- The set defined by the mixing constraint is convex.
- The data fitting terms are independent from each other.
- Projected gradient descent:

$$\mathbf{s}_j \leftarrow \mathbf{s}_j - r \nabla C_j(\mathbf{s}_j)$$

$$\mathbf{s}_j \leftarrow \mathbf{s}_j + \frac{1}{J} \sum_{i=1}^J \mathbf{y}_i$$

- Compute the gradient ∇C_j using the chain rule [Vial, 2021].

Algorithm overview

Initialization: Wiener-like mask: $s_j = \text{STFT}^{-1}(\mathbf{V}_j^{1=d} e^{i\angle \mathbf{x}})$

Algorithm overview

Initialization: Wiener-like mask: $\mathbf{s}_j = \text{STFT}^{-1}(\mathbf{V}_j^{1=d} e^{i\angle \mathbf{x}})$

Update rules

STFT

$$\mathbf{S}_j = \text{STFT}(\mathbf{s}_j)$$

Algorithm overview

Initialization: Wiener-like mask: $s_j = \text{STFT}^{-1}(\mathbf{V}_j^{1=d} \odot e^{i\angle \mathbf{X}})$

Update rules

STFT

$$\mathbf{S}_j = \text{STFT}(s_j)$$

Compute the gradient

$$\mathbf{G}_j = \nabla_{\mathbf{S}_j} (\mathbf{S}_j \mathbf{V}_j^{1=d}) \quad (\nabla_{\mathbf{S}_j \mathbf{V}_j^{1=d}} \mathbf{V}_j) \text{ (right)}$$

$$\mathbf{G}_j = \nabla_{\mathbf{S}_j} (\mathbf{S}_j \mathbf{V}_j^{1=d}) \quad \nabla_{\mathbf{V}_j} (\mathbf{V}_j) \text{ (left)}$$

Algorithm overview

Initialization: Wiener-like mask: $s_j = \text{STFT}^{-1}(\mathbf{V}_j^{1=d} \odot e^{i\mathbf{x}})$

Update rules

STFT	$\mathbf{S}_j = \text{STFT}(s_j)$
Compute the gradient	$\mathbf{G}_j = \nabla_{\mathbf{S}_j} (\mathbf{S}_j \mathbf{j}^d) \quad (\nabla_{\mathbf{S}_j \mathbf{j}^d} \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \nabla_{\mathbf{S}_j \mathbf{j}^d} (\mathbf{V}_j)$ (left)
Gradient descent	$\mathbf{Y}_j = \mathbf{S}_j \odot \mathbf{j}^d \odot \mathbf{S}_j \odot \mathbf{j}^d \odot \mathbf{G}_j$

Algorithm overview

Initialization: Wiener-like mask: $s_j = \text{STFT}^{-1}(\mathbf{V}_j^{1=d} \odot e^{i\mathbf{x}})$

Update rules

STFT	$\mathbf{S}_j = \text{STFT}(s_j)$
Compute the gradient	$\mathbf{G}_j = \nabla_{\mathbf{S}_j} (\mathbf{S}_j \mathbf{j}^d) \odot (\mathbf{S}_j \mathbf{j}^d \odot \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \nabla_{\mathbf{S}_j} (\mathbf{S}_j \mathbf{j}^d) \odot \mathbf{V}_j$ (left)
Gradient descent	$\mathbf{Y}_j = \mathbf{S}_j \odot d \odot \mathbf{S}_j \odot \mathbf{j} \mathbf{S}_j \mathbf{j}^d \odot \mathbf{G}_j$
Inverse STFT	$\mathbf{y}_j = \text{STFT}^{-1}(\mathbf{Y}_j)$

Algorithm overview

Initialization: Wiener-like mask: $s_j = \text{STFT}^{-1}(\mathbf{V}_j^{1=d} \odot e^{i\angle \mathbf{x}})$

Update rules

STFT	$\mathbf{S}_j = \text{STFT}(s_j)$
Compute the gradient	$\mathbf{G}_j = \nabla_{\mathbf{S}_j} (\mathbf{S}_j \mathbf{j}^d) \odot (\mathbf{S}_j \mathbf{j}^d - \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \nabla_{\mathbf{S}_j} (\mathbf{S}_j \mathbf{j}^d) \odot \mathbf{V}_j$ (left)
Gradient descent	$\mathbf{Y}_j = \mathbf{S}_j - d \mathbf{S}_j \mathbf{j}^d \odot \mathbf{G}_j$
Inverse STFT	$\mathbf{y}_j = \text{STFT}^{-1}(\mathbf{Y}_j)$
Mixing	$s_j = \mathbf{y}_j + \frac{1}{J} \sum_{i=1}^J \mathbf{y}_i$

Algorithm overview

Initialization: Wiener-like mask: $s_j = \text{STFT}^{-1}(\mathbf{V}_j^{1=d} \odot e^{i\angle \mathbf{x}})$

Update rules

STFT	$\mathbf{S}_j = \text{STFT}(s_j)$
Compute the gradient	$\mathbf{G}_j = \nabla_{\mathbf{S}_j} \ell(\mathbf{S}_j, \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \nabla_{\mathbf{V}_j} \ell(\mathbf{S}_j, \mathbf{V}_j)$ (left)
Gradient descent	$\mathbf{Y}_j = \mathbf{S}_j - d \nabla_{\mathbf{S}_j} \ell(\mathbf{S}_j, \mathbf{V}_j) \odot \mathbf{S}_j$
Inverse STFT	$\mathbf{y}_j = \text{STFT}^{-1}(\mathbf{Y}_j)$
Mixing	$s_j = \mathbf{y}_j + \frac{1}{J} \sum_{i=1}^J \mathbf{y}_i$

MISI is a particular case (quadratic loss, $d = 1$, and $\eta = 1$):

$$\mathbf{Y}_j = \mathbf{V}_j - \frac{\mathbf{S}_j}{J \|\mathbf{S}_j\|^2}$$

Experiments

Protocol

Task: speech enhancement ($J = 2$), 100 mixtures:

- . Clean speech from the VoiceBank dataset.
- . Real-life noises from the DEMAND dataset (living room, bus, and public square noises).
- . Mixtures at various input SNR (-10 , 0 , and 10 dB).

Magnitude estimation

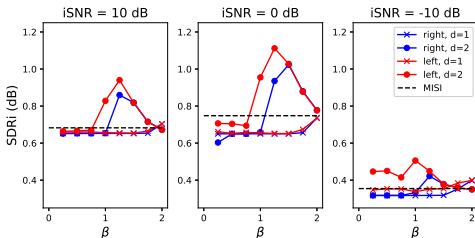
- . Open-Unmix (a freely available pretrained Bi-LSTM network).
- . The network is trained on different speakers and noises.

Split

- . 50 mixtures for validation (tuning the step size η).
- . 50 mixtures for testing (MISI and the proposed algorithm, 5 iterations).

Results

Signal-to-distortion ratio (improvement over the baseline amplitude mask):




- The proposed method outperforms MISI when $d = 2$:
 - At high/moderate input SNR when $\beta > 1$.
 - At low input SNR for all β and the "left" problem.
- Performance peak around $\beta = 1.25$, close to Kullback-Leibler ($\beta = 1$).
- Results depend on the type of noise.

Alternative divergences have some potential for phase retrieval in audio source separation from highly corrupted spectrograms

Perspectives

- . Alternative optimization schemes (majorization-minimization, ADMM).
- . Inclusion within deep learning (e.g., with deep unfolding) for end-to-end separation.

 <https://github.com/magronp/bregmisi>