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

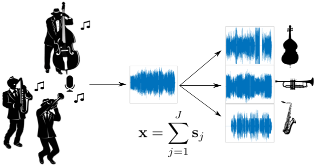
Audio source separation

- 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.

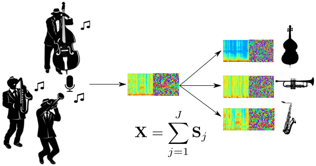


- Automatic speech recognition (clean speech vs. noise).
- Rhythm analysis (drums vs. harmonic instruments).
- Time-stretching (transients vs. partials).

Audio source separation

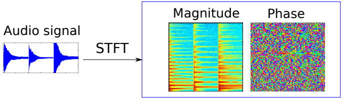
- ▷ Audio signals are composed of several constitutive sounds: multiple speakers, background noise, domestic sounds, musical instruments...

Source separation = recovering the sources from the mixture.

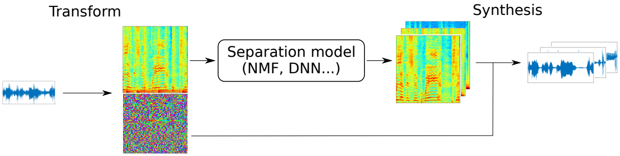


- ▷ Automatic speech recognition (clean speech vs. noise).
- ▷ Rhythm analysis (drums vs. harmonic instruments).
- ▷ Time-stretching (transients vs. partials).

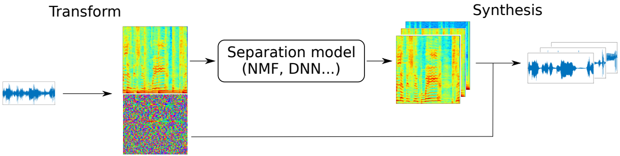
Time-frequency separation = acts on the short-time Fourier transform (STFT).



General framework

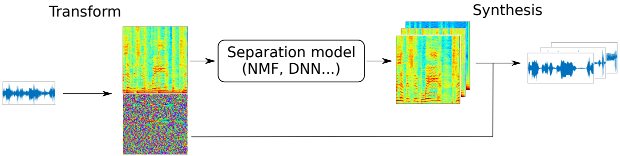


General framework

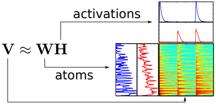


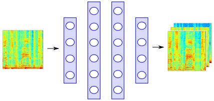
1. Nonnegative representation, e.g., $\mathbf{V} = |\text{STFT}(\mathbf{x})|^2$.

General framework

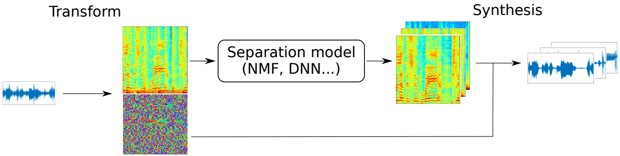


1. Nonnegative representation, e.g., $\mathbf{V} = |\mathsf{STFT}(\mathbf{x})|^2$.
2. Structured model, e.g., nonnegative matrix factorization, deep neural networks.

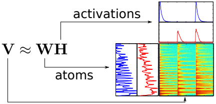




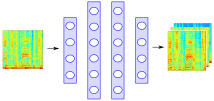
General framework



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



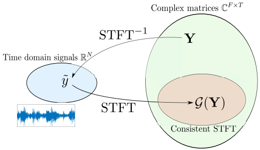
2. Structured model, e.g., nonnegative matrix factorization, deep neural networks.
3. Nonnegative masking and synthesis: $\tilde{\mathbf{s}}_j = \mathsf{STFT}^{-1}(\mathbf{M}_j \odot \mathbf{X})$.



The phase problem

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

- ✗ Issues in sound quality when sources overlap.
- ✗ *Inconsistency*: $\hat{\mathbf{S}}_j \notin \mathsf{STFT}(\mathbb{R}^N)$.

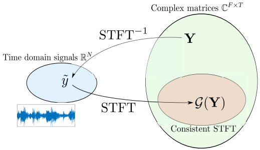


The phase problem

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

✗ Issues in sound quality when sources overlap.

✗ *Inconsistency*: $\hat{\mathbf{S}}_j \notin \mathsf{STFT}(\mathbb{R}^N)$.



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

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

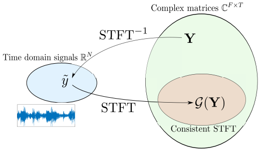
$$\min_{\mathbf{s}_j} \sum_{j=1}^{J} \|\mathbf{V}_j - |\mathsf{STFT}(\mathbf{s}_j)|\|^2 \text{ s.t. } \sum_{j=1}^{J} \mathbf{s}_j = \mathbf{x}.$$

Gunawan and Sen, **Iterative phase estimation for the synthesis of separated sources from single-channel mixtures**, *IEEE SPL*, 2010.

The phase problem

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

- ✗ Issues in sound quality when sources overlap.
- ✗ *Inconsistency*: $\hat{\mathbf{S}}_j \notin \mathsf{STFT}(\mathbb{R}^N)$.



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

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

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

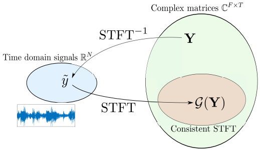
- ✓ Performance is improved over masking.

Gunawan and Sen, **Iterative phase estimation for the synthesis of separated sources from single-channel mixtures**, *IEEE SPL*, 2010.

The phase problem

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

- ✗ Issues in sound quality when sources overlap.
- ✗ *Inconsistency*: $\hat{\mathbf{S}}_j \notin \mathsf{STFT}(\mathbb{R}^N)$.



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

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

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

- ✓ Performance is improved over masking.
- ✗ Euclidean distance is not the most appropriate in audio.

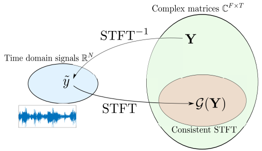
Gunawan and Sen, **Iterative phase estimation for the synthesis of separated sources from single-channel mixtures**, *IEEE SPL*, 2010.

The phase problem

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

✗ Issues in sound quality when sources overlap.

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



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

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

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

✓ Performance is improved over masking.

✗ Euclidean distance is not the most appropriate in audio.

Goal

Extend MISI to non-quadratic losses for source separation.

Gunawan and Sen, **Iterative phase estimation for the synthesis of separated sources from single-channel mixtures**, *IEEE SPL*, 2010.

Proposed method

Problem setting

Bregman divergences

$$\mathcal{D}_\psi(\mathbf{P} \,|\, \mathbf{Q}) = \sum_{f,t} \psi(p_{f,t}) - \psi(q_{f,t}) - \psi'(q_{f,t})(p_{f,t} - q_{f,t})$$

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

Hennequin et al., **Beta-divergence as a subclass of Bregman divergence**, *IEEE SPL*, 2011.

Problem setting

Bregman divergences

$$\mathcal{D}_\psi(\mathbf{P} \,|\, \mathbf{Q}) = \sum_{f,t} \psi(p_{f,t}) - \psi(q_{f,t}) - \psi'(q_{f,t})(p_{f,t} - q_{f,t})$$

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

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

- Accounting for the non-symmetry of Bregman divergences:

$$\mathcal{C}_j(\mathbf{s}_j) = \underbrace{\mathcal{D}_\psi(\mathbf{V}_j \,|\, |\text{STFT}(\mathbf{s}_j)|^d)}_{\text{“right” problem}} \quad \text{or} \quad \underbrace{\mathcal{D}_\psi(|\text{STFT}(\mathbf{s}_j)|^d \,|\, \mathbf{V}_j)}_{\text{“left” problem}}$$

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

Hennequin et al., **Beta-divergence as a subclass of Bregman divergence**, *IEEE SPL*, 2011.

Projected gradient descent

$$\min_{\mathbf{s}_j} \sum_{j=1}^{J} \underbrace{\mathcal{C}_j(\mathbf{s}_j)}_{\text{Data fitting}} \quad \text{s.t.} \quad \underbrace{\sum_{j=1}^{J} \mathbf{s}_j = \mathbf{x}}_{\text{Mixing constraint}}$$

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

$$\mathbf{y}_j \leftarrow \mathbf{s}_j - \mu \nabla \mathcal{C}_j(\mathbf{s}_j)$$

$$\mathbf{s}_j \leftarrow \mathbf{y}_j + \frac{1}{J}\left(\mathbf{x} - \sum_{i=1}^{J} \mathbf{y}_i\right)$$

- ▷ Compute the gradient $\nabla \mathcal{C}_j$ using the chain rule [Vial, 2021].

Vial et al., "Phase retrieval with Bregman divergences and application to audio signal recovery", *IEEE JSTSP*, 2021.

Algorithm overview

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

Algorithm overview

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

Update rules

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

Algorithm overview

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

Update rules

STFT	$\mathbf{S}_j = \mathsf{STFT}(\mathbf{s}_j)$
Compute the gradient	$\mathbf{G}_j = \psi''(\lvert\mathbf{S}_j\rvert^d) \odot (\lvert\mathbf{S}_j\rvert^d - \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \psi'(\lvert\mathbf{S}_j\rvert^d) - \psi'(\mathbf{V}_j)$ (left)

Algorithm overview

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

Update rules

STFT	$\mathbf{S}_j = \mathsf{STFT}(\mathbf{s}_j)$
Compute the gradient	$\mathbf{G}_j = \psi''(\lvert\mathbf{S}_j\rvert^d) \odot (\lvert\mathbf{S}_j\rvert^d - \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \psi'(\lvert\mathbf{S}_j\rvert^d) - \psi'(\mathbf{V}_j)$ (left)
Gradient descent	$\mathbf{Y}_j = \mathbf{S}_j - \mu d \times \mathbf{S}_j \odot \lvert\mathbf{S}_j\rvert^{d-2} \odot \mathbf{G}_j$

Algorithm overview

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

Update rules

STFT	$\mathbf{S}_j = \mathsf{STFT}(\mathbf{s}_j)$
Compute the gradient	$\mathbf{G}_j = \psi''(\lvert\mathbf{S}_j\rvert^d) \odot (\lvert\mathbf{S}_j\rvert^d - \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \psi'(\lvert\mathbf{S}_j\rvert^d) - \psi'(\mathbf{V}_j)$ (left)
Gradient descent	$\mathbf{Y}_j = \mathbf{S}_j - \mu d \times \mathbf{S}_j \odot \lvert\mathbf{S}_j\rvert^{d-2} \odot \mathbf{G}_j$
Inverse STFT	$\mathbf{y}_j = \mathsf{STFT}^{-1}(\mathbf{Y}_j)$

Algorithm overview

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

Update rules

STFT	$\mathbf{S}_j = \mathsf{STFT}(\mathbf{s}_j)$
Compute the gradient	$\mathbf{G}_j = \psi''(\lvert\mathbf{S}_j\rvert^d) \odot (\lvert\mathbf{S}_j\rvert^d - \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \psi'(\lvert\mathbf{S}_j\rvert^d) - \psi'(\mathbf{V}_j)$ (left)
Gradient descent	$\mathbf{Y}_j = \mathbf{S}_j - \mu d \times \mathbf{S}_j \odot \lvert\mathbf{S}_j\rvert^{d-2} \odot \mathbf{G}_j$
Inverse STFT	$\mathbf{y}_j = \mathsf{STFT}^{-1}(\mathbf{Y}_j)$
Mixing	$\mathbf{s}_j = \mathbf{y}_j + \frac{1}{J}\left(\mathbf{x} - \sum_{i=1}^{J} \mathbf{y}_i\right)$

Algorithm overview

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

Update rules

STFT	$\mathbf{S}_j = \mathsf{STFT}(\mathbf{s}_j)$
Compute the gradient	$\mathbf{G}_j = \psi''(\lvert\mathbf{S}_j\rvert^d) \odot (\lvert\mathbf{S}_j\rvert^d - \mathbf{V}_j)$ (right)
	$\mathbf{G}_j = \psi'(\lvert\mathbf{S}_j\rvert^d) - \psi'(\mathbf{V}_j)$ (left)
Gradient descent	$\mathbf{Y}_j = \mathbf{S}_j - \mu d \times \mathbf{S}_j \odot \lvert\mathbf{S}_j\rvert^{d-2} \odot \mathbf{G}_j$
Inverse STFT	$\mathbf{y}_j = \mathsf{STFT}^{-1}(\mathbf{Y}_j)$
Mixing	$\mathbf{s}_j = \mathbf{y}_j + \frac{1}{J}\left(\mathbf{x} - \sum_{i=1}^{J} \mathbf{y}_i\right)$

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

$$\mathbf{Y}_j = \mathbf{V}_j \odot \frac{\mathbf{S}_j}{|\mathbf{S}_j|}$$

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

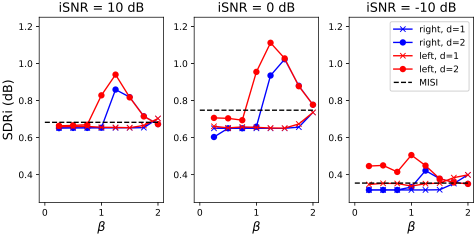
- Open-Unmix (a freely available pretrained Bi-LSTM network).
- The network is trained on different speakers and noises.

Split

- 50 mixtures for validation (tuning the step size μ).
- 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.

Conclusion

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`