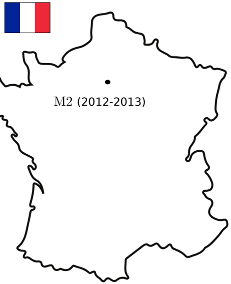
Phase recovery for audio demixing: contributions and perspectives

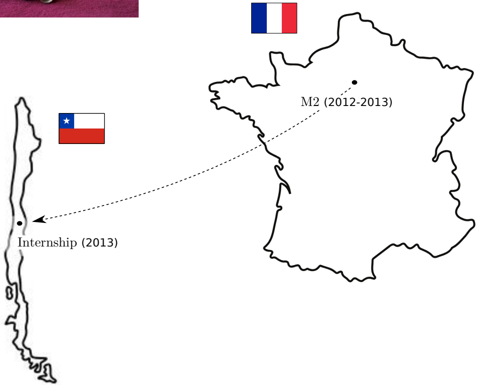
Talk at Neural DSP - Helsinki, August 18th, 2022

Paul Magron, Researcher - INRIA Nancy Grand Est

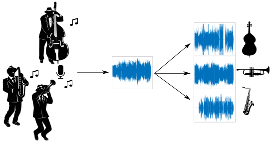
Background in a nutshell

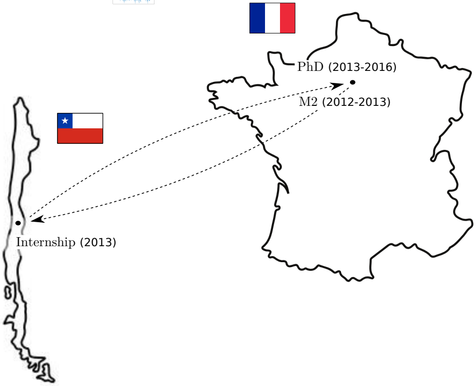


Background in a nutshell

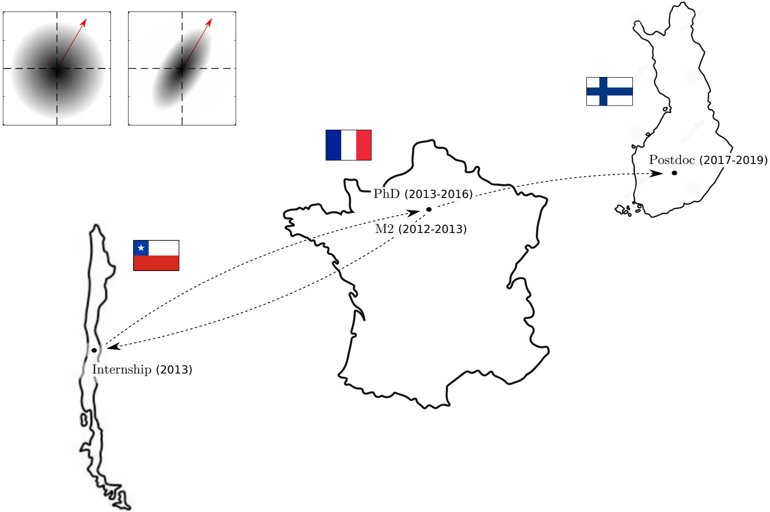


Background in a nutshell

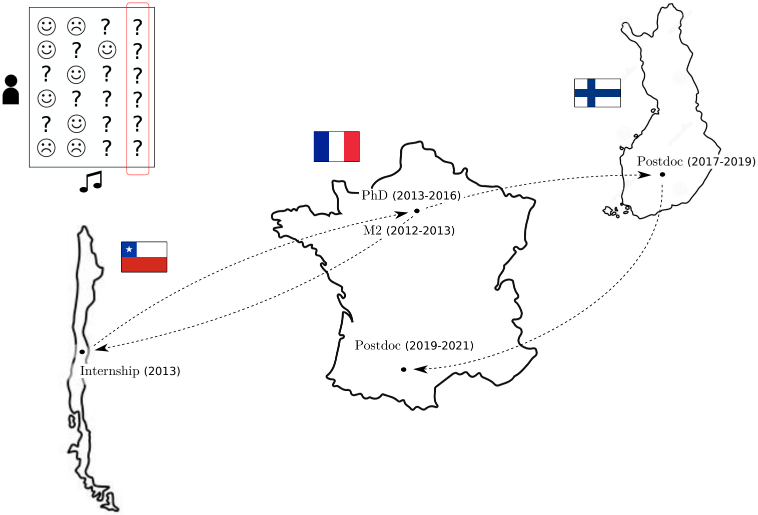




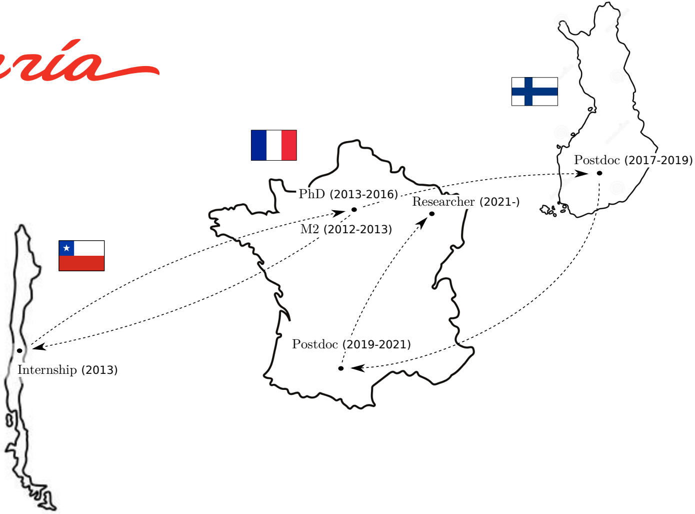
Background in a nutshell



Background in a nutshell



Background in a nutshell



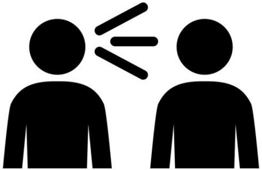
The audio realm

The audio realm

Speech

The audio realm

Speech



Ambient sounds

The audio realm

Speech

Ambient sounds

Music signals

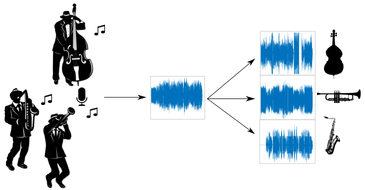
Audio demixing

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

Audio demixing

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

Source separation or Demixing = recovering the sources from the mixture.

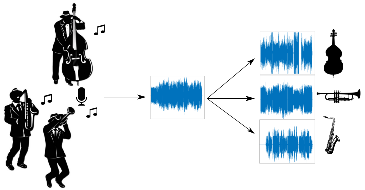


Audio demixing

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

Source separation or Demixing = recovering the sources from the mixture.

- A useful task *per se* (e.g., augmented mixing from mono to stereo).
- An important preprocessing for many analysis tasks (e.g., polyphonic music transcription).

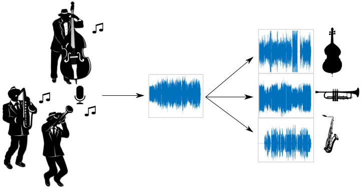


Audio demixing

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

Source separation or Demixing = recovering the sources from the mixture.

- A useful task *per se* (e.g., augmented mixing from mono to stereo).
- An important preprocessing for many analysis tasks (e.g., polyphonic music transcription).



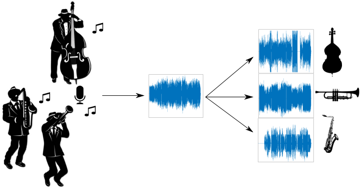
An example: Backing track generation

Audio demixing

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

Source separation or Demixing = recovering the sources from the mixture.

- ▷ A useful task *per se* (e.g., augmented mixing from mono to stereo).
- ▷ An important preprocessing for many analysis tasks (e.g., polyphonic music transcription).

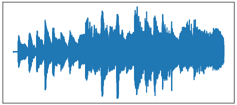


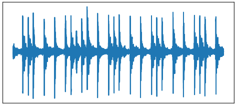
An example: Backing track generation

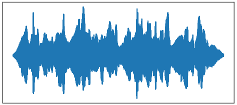
- ▷ Consider a mixture
- ▷ Demix the instruments and create a backing track

Setting the stage

- The raw material: audio signals.

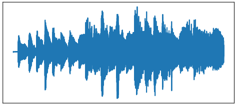


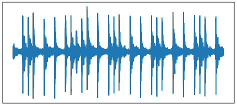


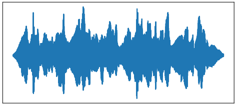


Setting the stage

- The raw material: audio signals.



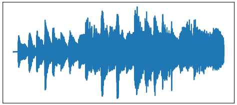
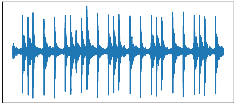
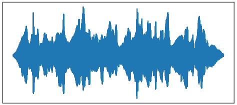




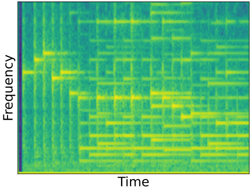
- It's hard to see structure there...

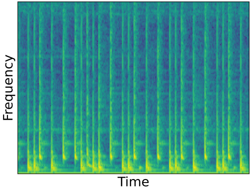
Setting the stage

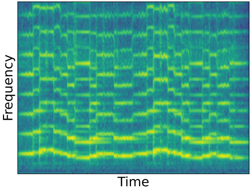
- ▷ The raw material: audio signals.

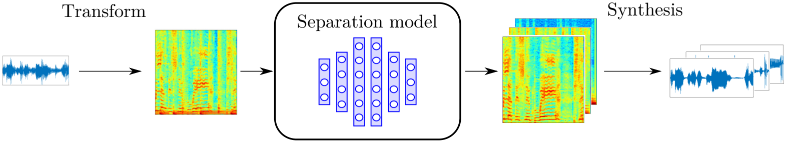
- ▷ It's hard to see structure there...
- ▷ We rather transform them into a time-frequency representation, e.g., a spectrogram.



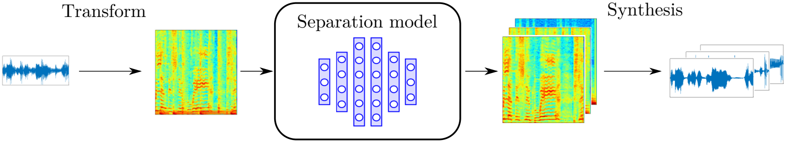




The demixing pipeline

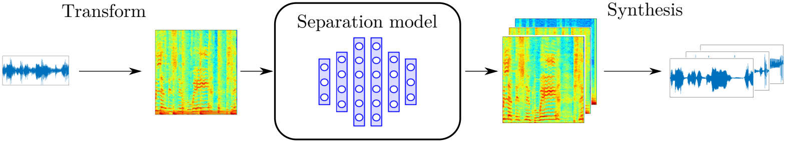


The demixing pipeline



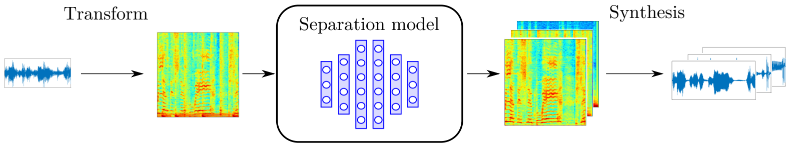
- The transform is usually the short-time Fourier transform (STFT).

The demixing pipeline



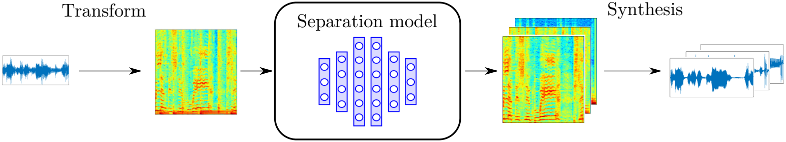
- The transform is usually the short-time Fourier transform (STFT).
- The separator is a deep neural network, trained using a (large) dataset of isolated tracks.

The demixing pipeline



- The transform is usually the short-time Fourier transform (STFT).
- The separator is a deep neural network, trained using a (large) dataset of isolated tracks.
- The synthesis is performed through inverse STFT.

The demixing pipeline



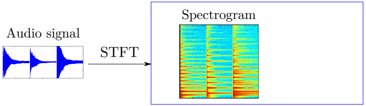
- The transform is usually the short-time Fourier transform (STFT).
- The separator is a deep neural network, trained using a (large) dataset of isolated tracks.
- The synthesis is performed through inverse STFT.

Nowadays demixing performance:

The phase catch

$$\mathbf{x} \in \mathbb{R}^N \xrightarrow{\textsf{STFT}} \mathbf{X} \in \mathbb{C}^{F \times T}$$

The STFT produces a spectrogram $|X|$

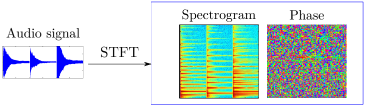


The phase catch

$$\mathbf{x} \in \mathbb{R}^N \xrightarrow{\mathsf{STFT}} \mathbf{X} \in \mathbb{C}^{F \times T}$$

The STFT produces a spectrogram $|X|$

... but also a phase $\angle X$.

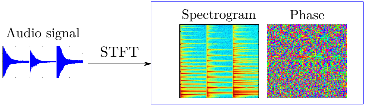


The phase catch

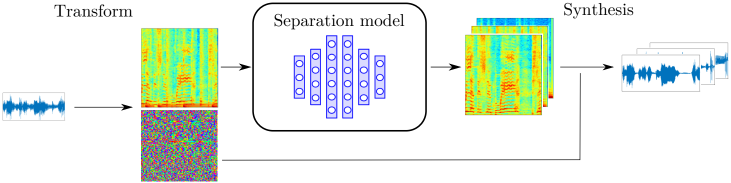
$$\mathbf{x} \in \mathbb{R}^N \xrightarrow{\text{STFT}} \mathbf{X} \in \mathbb{C}^{F \times T}$$

The STFT produces a spectrogram $|X|$

... but also a phase $\angle X$.



The actual demixing pipeline:



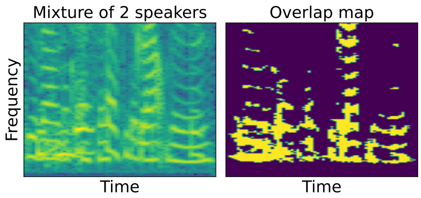
▷ The mixture's phase is assigned to each source using a Wiener-like filter.

The potential of phase recovery

✗ Wiener-like filter: Issues in sound quality when sources *overlap* in the TF domain.

When sources overlap:

$$|X| \neq |S_1| + |S_2|$$
$$\angle X \neq \angle S_1 \text{ or } \angle S_2$$



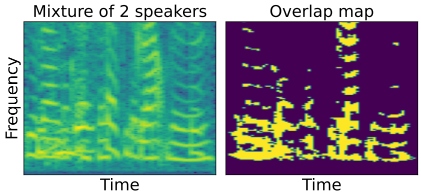
The potential of phase recovery

✗ Wiener-like filter: Issues in sound quality when sources *overlap* in the TF domain.

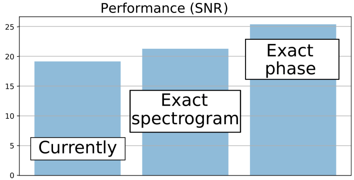
When sources overlap:

$$|X| \neq |S_1| + |S_2|$$

$$\angle X \neq \angle S_1 \text{ or } \angle S_2$$



Given the current state of the art:



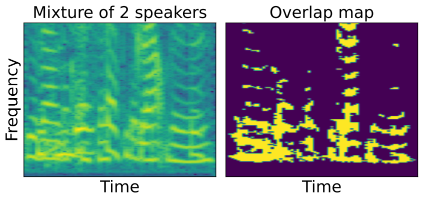
The potential of phase recovery

✗ Wiener-like filter: Issues in sound quality when sources *overlap* in the TF domain.

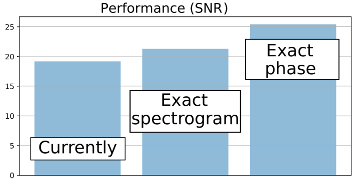
When sources overlap:

$$|X| \neq |S_1| + |S_2|$$

$$\angle X \neq \angle S_1 \text{ or } \angle S_2$$



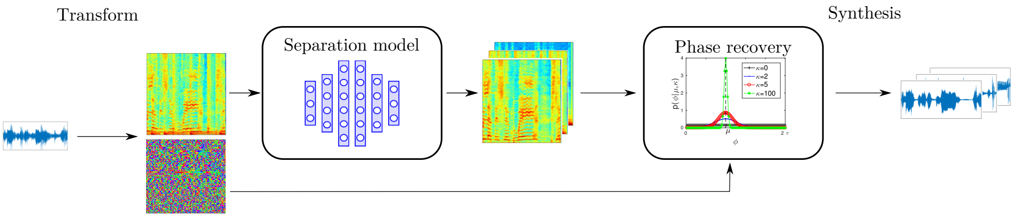
Given the current state of the art:



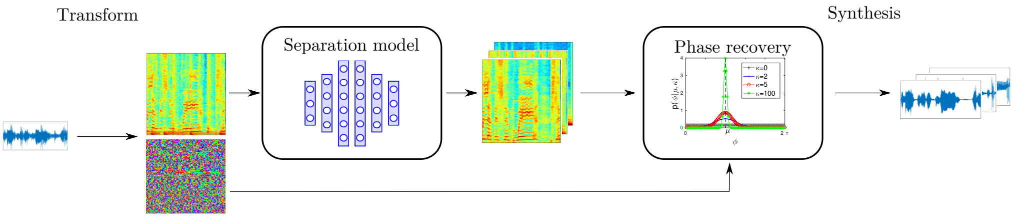
Main message

More potential gain in phase recovery than in magnitude estimation.

Phase recovery for audio demixing

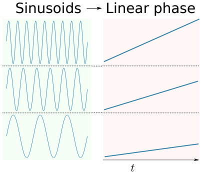


Phase recovery for audio demixing

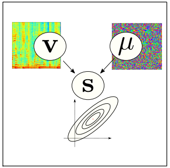


Main contributions

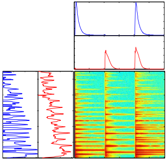
Phase models



Statistical framework



Factorization methods



Introduction

Model-based phase recovery

Probabilistic phase modeling

Factorization methods

Conclusion

Model-based phase recovery

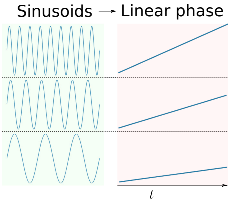
Sinusoidal phase model

Consider a mixture of sinusoids: $x(n) = \sum_{p=1}^{P} A_p \sin(2\pi \underbrace{\nu_p}_{\text{normalized frequency}} n + \phi_{0,p})$.

Sinusoidal phase model

Consider a mixture of sinusoids: $x(n) = \sum_{p=1}^{P} A_p \sin(2\pi \underbrace{\nu_p}_{\text{normalized frequency}} n + \phi_{0,p})$.

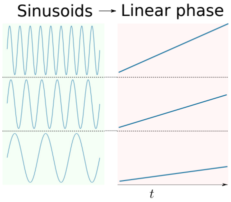
The STFT phase follows: $\mu_{f,t} = \mu_{f,t-1} + l\nu_{f,t}$

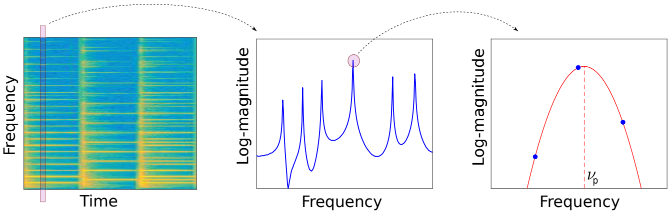


Sinusoidal phase model

Consider a mixture of sinusoids: $x(n) = \sum_{p=1}^{P} A_p \sin(2\pi \underbrace{\nu_p}_{\text{normalized frequency}} n + \phi_{0,p})$.

The STFT phase follows: $\mu_{f,t} = \mu_{f,t-1} + l\nu_{f,t}$

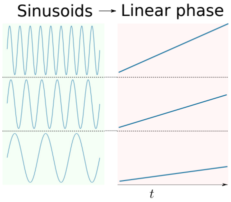


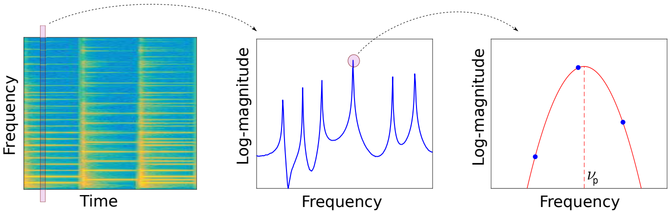


Sinusoidal phase model

Consider a mixture of sinusoids: $x(n) = \sum_{p=1}^{P} A_p \sin(2\pi \underbrace{\nu_p}_{\text{normalized frequency}} n + \phi_{0,p})$.

The STFT phase follows: $\mu_{f,t} = \mu_{f,t-1} + l\nu_{f,t}$





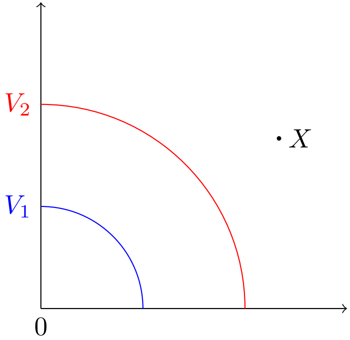
✓ Accounts for non-stationary signals, suitable for real-time processing.

✗ Bad performance for “pure” phase recovery: need to use an additional information.

An iterative source separation algorithm

Problem Given target magnitude spectrograms $\mathbf{V}_j$, solve:

$$\min_{\{\hat{\mathbf{S}}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \hat{\mathbf{S}}_j||^2 \quad \text{s.t.} \quad |\hat{\mathbf{S}}_j| = \mathbf{V}_j$$



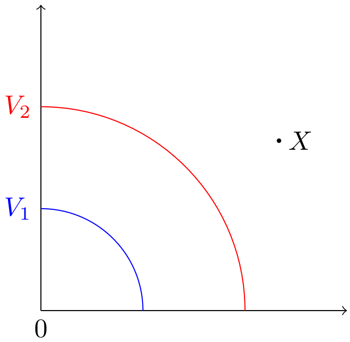
An iterative source separation algorithm

Problem Given target magnitude spectrograms $\mathbf{V}_j$, solve:

$$\min_{\{\hat{\mathbf{S}}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \hat{\mathbf{S}}_j||^2 \quad \text{s.t.} \quad |\hat{\mathbf{S}}_j| = \mathbf{V}_j$$

Strategy

- Obtain an iterative procedure using some optimization framework (e.g., majorization-minimization).
- Initialize the procedure using the sinusoidal phase model.



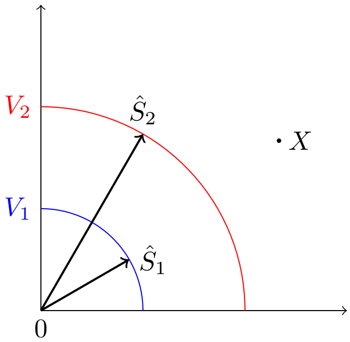
An iterative source separation algorithm

Problem Given target magnitude spectrograms $\mathbf{V}_j$, solve:

$$\min_{\{\hat{\mathbf{S}}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \hat{\mathbf{S}}_j||^2 \quad \text{s.t.} \quad |\hat{\mathbf{S}}_j| = \mathbf{V}_j$$

Strategy

- ▷ Obtain an iterative procedure using some optimization framework (e.g., majorization-minimization).
- ▷ Initialize the procedure using the sinusoidal phase model.



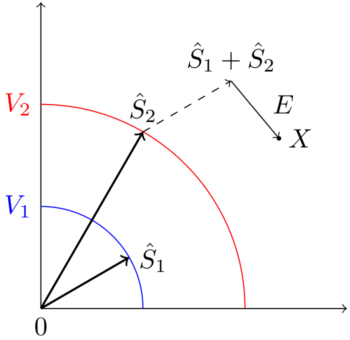
An iterative source separation algorithm

Problem Given target magnitude spectrograms $\mathbf{V}_j$, solve:

$$\min_{\{\hat{\mathbf{S}}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \hat{\mathbf{S}}_j||^2 \quad \text{s.t.} \quad |\hat{\mathbf{S}}_j| = \mathbf{V}_j$$

Strategy

- Obtain an iterative procedure using some optimization framework (e.g., majorization-minimization).
- Initialize the procedure using the sinusoidal phase model.



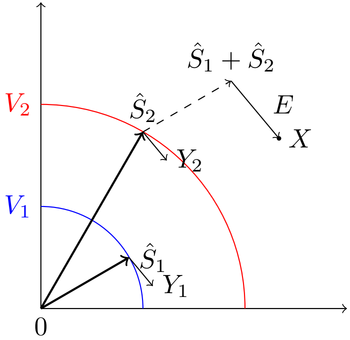
An iterative source separation algorithm

Problem Given target magnitude spectrograms $\mathbf{V}_j$, solve:

$$\min_{\{\hat{\mathbf{S}}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \hat{\mathbf{S}}_j||^2 \quad \text{s.t.} \quad |\hat{\mathbf{S}}_j| = \mathbf{V}_j$$

Strategy

- ▷ Obtain an iterative procedure using some optimization framework (e.g., majorization-minimization).
- ▷ Initialize the procedure using the sinusoidal phase model.



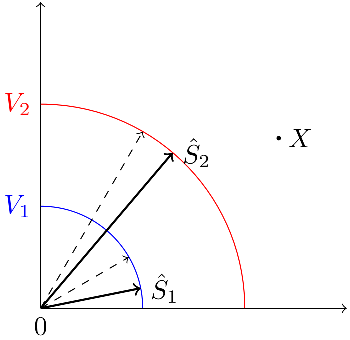
An iterative source separation algorithm

Problem Given target magnitude spectrograms $\mathbf{V}_j$, solve:

$$\min_{\{\hat{\mathbf{S}}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \hat{\mathbf{S}}_j||^2 \quad \text{s.t.} \quad |\hat{\mathbf{S}}_j| = \mathbf{V}_j$$

Strategy

- Obtain an iterative procedure using some optimization framework (e.g., majorization-minimization).
- Initialize the procedure using the sinusoidal phase model.



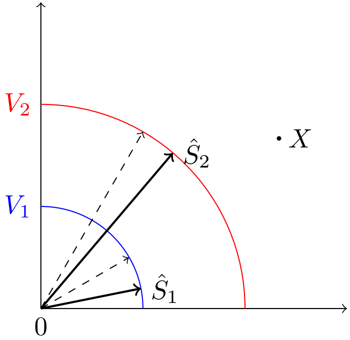
An iterative source separation algorithm

Problem Given target magnitude spectrograms $\mathbf{V}_j$, solve:

$$\min_{\{\hat{\mathbf{S}}_j\}} ||\mathbf{X} - \sum_{j=1}^{J} \hat{\mathbf{S}}_j||^2 \quad \text{s.t.} \quad |\hat{\mathbf{S}}_j| = \mathbf{V}_j$$

Strategy

- ▷ Obtain an iterative procedure using some optimization framework (e.g., majorization-minimization).
- ▷ Initialize the procedure using the sinusoidal phase model.



✓ Leveraging the sinusoidal phase model reduces interference between source estimates.

Perspective: towards deep phase recovery

Recently: Some attempts at predicting the phase using DNNs.

- ✗ Generic architectures which do not account for the particular phase structure.

Perspective: towards deep phase recovery

Recently: Some attempts at predicting the phase using DNNs.

- ✗ Generic architectures which do not account for the particular phase structure.

Proposal: Generalize phase models from signal analysis with deep learning.

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + l\boldsymbol{\nu}_t \quad \rightarrow \quad \boldsymbol{\mu}_t = \underbrace{\mathcal{R}(\boldsymbol{\nu}_t, \boldsymbol{\mu}_{t-1}, \ldots, \boldsymbol{\mu}_{t-\tau})}_{\text{temporal dynamics}} \quad \text{with} \quad \boldsymbol{\nu}_t = \underbrace{\mathcal{C}(|\mathbf{x}|_t)}_{\text{frequency extraction}}$$

Perspective: towards deep phase recovery

Recently: Some attempts at predicting the phase using DNNs.

✗ Generic architectures which do not account for the particular phase structure.

Proposal: Generalize phase models from signal analysis with deep learning.

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t-1} + l\boldsymbol{\nu}_t \quad \rightarrow \quad \boldsymbol{\mu}_t = \underbrace{\mathcal{R}(\boldsymbol{\nu}_t, \boldsymbol{\mu}_{t-1}, \ldots, \boldsymbol{\mu}_{t-\tau})}_{\text{temporal dynamics}} \quad \text{with} \quad \boldsymbol{\nu}_t = \underbrace{\mathcal{C}(|\mathbf{x}|_t)}_{\text{frequency extraction}}$$

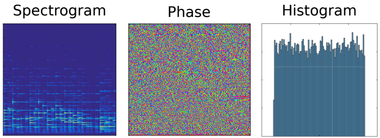
- Architectural choices (non-linearities, loss functions) adapted to the phase (periodicity).
- Identify and exploit perceptual phase invariants.

Probabilistic phase modeling

A statistical view

A simple example

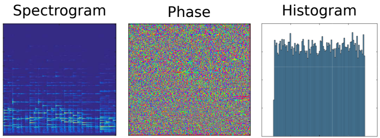
- ▷ The phase appears uniformly-distributed.



A statistical view

A simple example

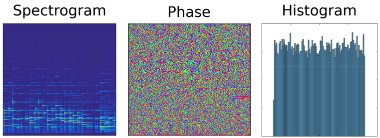
- The phase appears uniformly-distributed.
- But is that consistent with, e.g., the sinusoidal model?



A statistical view

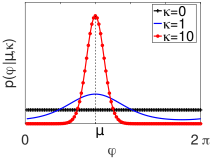
A simple example

- ▷ The phase appears uniformly-distributed.
- ▷ But is that consistent with, e.g., the sinusoidal model?



Von Mises phase $\phi_{f,t} \sim \mathcal{VM}(\mu_{f,t}, \kappa)$

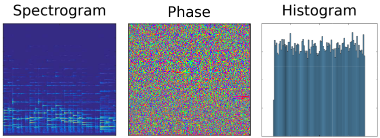
- ▷ Assume some structure (e.g., sinusoidal) for the location parameter $\mu_{f,t}$.



A statistical view

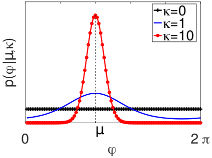
A simple example

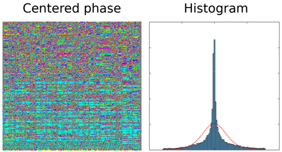
- ▷ The phase appears uniformly-distributed.
- ▷ But is that consistent with, e.g., the sinusoidal model?



Von Mises phase $\phi_{f,t} \sim \mathcal{VM}(\mu_{f,t}, \kappa)$

- ▷ Assume some structure (e.g., sinusoidal) for the location parameter $\mu_{f,t}$.





✓ Both models are statistically relevant, but convey a different information about the phase.

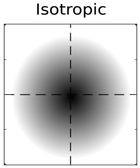
- ▷ Uniform → describes the *global* behavior.
- ▷ Von Mises → accounts for the *local* structure.

Modeling complex-valued coefficients

Isotropic Gaussian model

$$s \sim \mathcal{N}_{\mathbb{C}}(m, \Gamma) \text{ with } \Gamma = \begin{pmatrix} \gamma & 0 \\ 0 & \gamma \end{pmatrix}$$

✗ Equivalent to assuming a uniform phase: $\angle s_j = \phi_j \sim \mathcal{U}_{[0,2\pi[}$



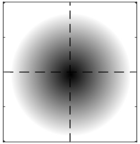
Modeling complex-valued coefficients

Isotropic Gaussian model

$$s \sim \mathcal{N}_{\mathbb{C}}(m, \Gamma) \text{ with } \Gamma = \begin{pmatrix} \gamma & 0 \\ 0 & \gamma \end{pmatrix}$$

✗ Equivalent to assuming a uniform phase: $\angle s_j = \phi_j \sim \mathcal{U}_{[0,2\pi[}$

Isotropic



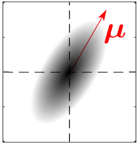
Anisotropic Gaussian model

$$s \sim \mathcal{N}_{\mathbb{C}}(m, \Gamma) \text{ with } \Gamma = \begin{pmatrix} \gamma & c \\ \bar{c} & \gamma \end{pmatrix}$$

c is the *relation* term, defined as a function of the phase parameter μ.

✓ Allows to incorporate phase priors.

Anisotropic



Application to demixing

Mixture model In each time-frequency bin: $x = \sum_j s_j$ with $s_j \sim \mathcal{N}_{\mathbb{C}}(m_j, \Gamma_j)$.

- ▷ Choose an appropriate parametrization for m_j and Γ_j (a bit technical).
- ▷ Estimate the models' parameters (e.g., maximum likelihood estimation).

Application to demixing

Mixture model In each time-frequency bin: $x = \sum_j s_j$ with $s_j \sim \mathcal{N}_{\mathbb{C}}(m_j, \Gamma_j)$.

- ▷ Choose an appropriate parametrization for m_j and Γ_j (a bit technical).
- ▷ Estimate the models' parameters (e.g., maximum likelihood estimation).

Anisotropic Wiener filter

- ▷ Posterior mean of the sources: $\hat{\mathbf{S}}_j = \mathbb{E}(\mathbf{S}_j | \mathbf{X})$.
- ▷ Optimal in the MMSE sense, conservative set of estimates.
- ▷ A generalization of the (phase-unaware) Wiener filter.

Application to demixing

Mixture model In each time-frequency bin: $x = \sum_j s_j$ with $s_j \sim \mathcal{N}_{\mathbb{C}}(m_j, \Gamma_j)$.

- Choose an appropriate parametrization for m_j and Γ_j (a bit technical).
- Estimate the models' parameters (e.g., maximum likelihood estimation).

Anisotropic Wiener filter

- Posterior mean of the sources: $\hat{\mathbf{S}}_j = \mathbb{E}(\mathbf{S}_j | \mathbf{X})$.
- Optimal in the MMSE sense, conservative set of estimates.
- A generalization of the (phase-unaware) Wiener filter.

Performance

✓ Including a phase prior in the filter improves the separation quality.

Perspective: anisotropic deep learning

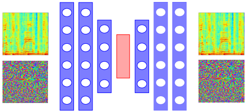
✗ Bayesian deep learning / variational autoencoders (VAE) are limited to isotropic distributions.

Perspective: anisotropic deep learning

✗ Bayesian deep learning / variational autoencoders (VAE) are limited to isotropic distributions.

Proposal: Combine deep learning and anisotropic modeling, e.g., via anisotropic VAEs.

$$\underbrace{\mathbf{z}|\mathbf{x} \sim \mathcal{N}_{\mathbb{C}}(\psi_{\text{enc}}(\mathbf{x}), \mathbf{\Gamma}_{\text{enc}})}_{\text{encoder}} \quad \text{and} \quad \underbrace{\mathbf{s}|\mathbf{z} \sim \mathcal{N}_{\mathbb{C}}(\psi_{\text{dec}}(\mathbf{z}), \mathbf{\Gamma}_{\text{dec}})}_{\text{decoder}}$$

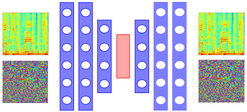


Perspective: anisotropic deep learning

✗ Bayesian deep learning / variational autoencoders (VAE) are limited to isotropic distributions.

Proposal: Combine deep learning and anisotropic modeling, e.g., via anisotropic VAEs.

$$\underbrace{\mathbf{z}|\mathbf{x} \sim \mathcal{N}_{\mathbb{C}}(\psi_{\text{enc}}(\mathbf{x}), \mathbf{\Gamma}_{\text{enc}})}_{\text{encoder}} \quad \text{and} \quad \underbrace{\mathbf{s}|\mathbf{z} \sim \mathcal{N}_{\mathbb{C}}(\psi_{\text{dec}}(\mathbf{z}), \mathbf{\Gamma}_{\text{dec}})}_{\text{decoder}}$$



▷ A strong effort in modeling and optimization is needed for deriving appropriate estimation techniques.

Factorization methods

A leap in the past: nonnegative matrix factorization (NMF)

Given a (nonnegative) spectrogram $\mathbf{V}$, find a factorization $\mathbf{WH}$ such that the factors $\mathbf{W}$ and $\mathbf{H}$ are:

- ▷ low rank.
- ▷ nonnegative.

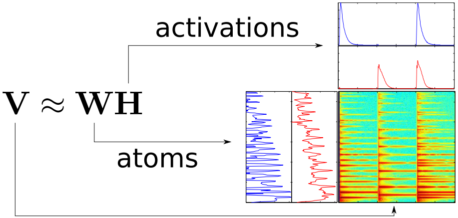
A leap in the past: nonnegative matrix factorization (NMF)

Given a (nonnegative) spectrogram $\mathbf{V}$, find a factorization $\mathbf{WH}$ such that the factors $\mathbf{W}$ and $\mathbf{H}$ are:

- ▷ low rank.
- ▷ nonnegative.

Nonnegativity favors interpretability.

- ▷ $\mathbf{W}$ is a dictionary of spectral atoms.
- ▷ $\mathbf{H}$ is a matrix of temporal activation.



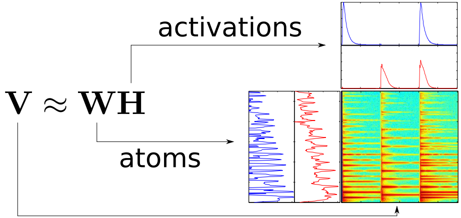
A leap in the past: nonnegative matrix factorization (NMF)

Given a (nonnegative) spectrogram $\mathbf{V}$, find a factorization $\mathbf{WH}$ such that the factors $\mathbf{W}$ and $\mathbf{H}$ are:

- ▷ low rank.
- ▷ nonnegative.

Nonnegativity favors interpretability.

- ▷ $\mathbf{W}$ is a dictionary of spectral atoms.
- ▷ $\mathbf{H}$ is a matrix of temporal activation.

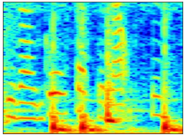


Estimation via an optimization problem:

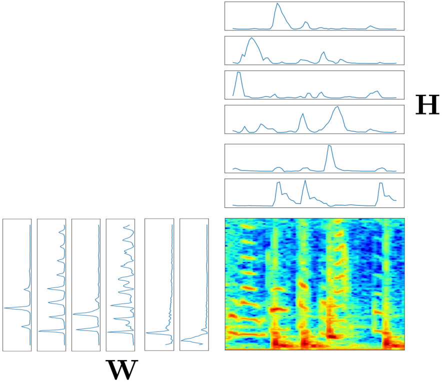
$$\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}, \mathbf{WH})$$

NMF for audio demixing

$$|\mathbf{X}| \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$



NMF for audio demixing

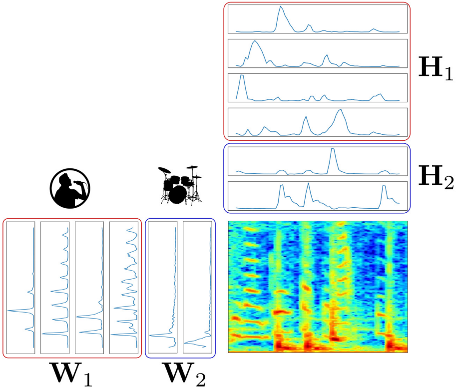


$$|\mathbf{X}| \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram.

NMF for audio demixing

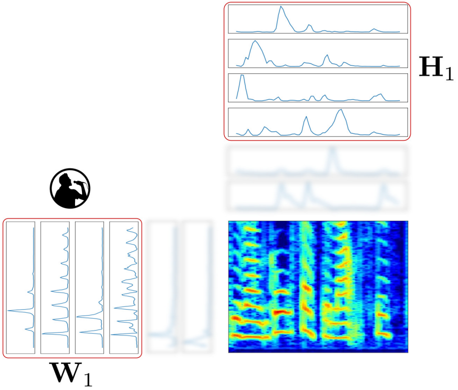


$$|\mathbf{X}| \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.

NMF for audio demixing

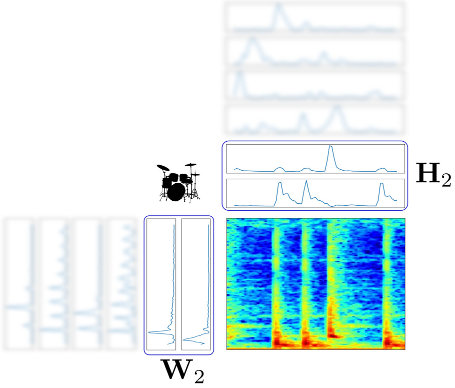


$$|\mathbf{X}| \approx \mathbf{WH} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.
2. Multiply each dictionary with the corresponding activations.

NMF for audio demixing

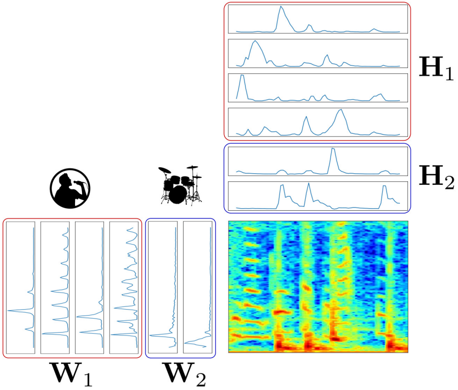


$$|\mathbf{X}| \approx \mathbf{WH} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.
2. Multiply each dictionary with the corresponding activations.

NMF for audio demixing



$$|\mathbf{X}| \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

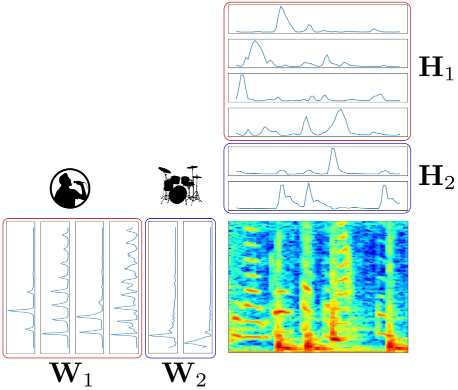
Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.
2. Multiply each dictionary with the corresponding activations.

Supervised demixing

Pretrain $\mathbf{W}_1$ and $\mathbf{W}_2$ on subsets of isolated tracks.

NMF for audio demixing



$$|\mathbf{X}| \approx \mathbf{W}\mathbf{H} = \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j = \sum_{j=1}^{J} \mathbf{V}_j$$

Procedure

1. Factorize the mixture's spectrogram.
2. Cluster atoms that belong to the same source.
2. Multiply each dictionary with the corresponding activations.

Supervised demixing

Pretrain $\mathbf{W}_1$ and $\mathbf{W}_2$ on subsets of isolated tracks.

- ✗ Ignores the phase / assumes the magnitudes are additive.
- ✗ The low-rank assumption is not verified in practice.

Complex NMF

Instead of assuming additive magnitudes $|\mathbf{X}| = |\mathbf{S}_1| + |\mathbf{S}_2| + \ldots + |\mathbf{S}_J|$

Complex NMF

Instead of assuming additive magnitudes $|\mathbf{X}| = |\mathbf{S}_1| + |\mathbf{S}_2| + \ldots + |\mathbf{S}_J|$, consider additive complex-valued sources:

$$\begin{aligned}\mathbf{X} &= \mathbf{S}_1 + \mathbf{S}_2 + \ldots + \mathbf{S}_J \\ &= \mathbf{V}_1 e^{\mathrm{i}\boldsymbol{\mu}_1} + \mathbf{V}_2 e^{\mathrm{i}\boldsymbol{\mu}_2} + \ldots + \mathbf{V}_J e^{\mathrm{i}\boldsymbol{\mu}_J}\end{aligned}$$

Complex NMF

Instead of assuming additive magnitudes $|\mathbf{X}| = |\mathbf{S}_1| + |\mathbf{S}_2| + \ldots + |\mathbf{S}_J|$, consider additive complex-valued sources:

$$\begin{aligned}\mathbf{X} &= \mathbf{S}_1 + \mathbf{S}_2 + \ldots + \mathbf{S}_J \\ &= \mathbf{V}_1 e^{\mathrm{i}\boldsymbol{\mu}_1} + \mathbf{V}_2 e^{\mathrm{i}\boldsymbol{\mu}_2} + \ldots + \mathbf{V}_J e^{\mathrm{i}\boldsymbol{\mu}_J}\end{aligned}$$

And factorize each spectrogram with NMF: $\mathbf{V}_j = \mathbf{W}_j \mathbf{H}_j$

Complex NMF

Instead of assuming additive magnitudes $|\mathbf{X}| = |\mathbf{S}_1| + |\mathbf{S}_2| + \ldots + |\mathbf{S}_J|$, consider additive complex-valued sources:

$$\begin{aligned}\mathbf{X} &= \mathbf{S}_1 + \mathbf{S}_2 + \ldots + \mathbf{S}_J \\ &= \mathbf{V}_1 e^{\mathrm{i}\boldsymbol{\mu}_1} + \mathbf{V}_2 e^{\mathrm{i}\boldsymbol{\mu}_2} + \ldots + \mathbf{V}_J e^{\mathrm{i}\boldsymbol{\mu}_J}\end{aligned}$$

And factorize each spectrogram with NMF: $\mathbf{V}_j = \mathbf{W}_j \mathbf{H}_j$

Estimation

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\mu}} ||\mathbf{X} - \sum_{j=1}^{J} [\mathbf{W}_j \mathbf{H}_j] e^{\mathrm{i}\boldsymbol{\mu}_j}||^2$$

Complex NMF

Instead of assuming additive magnitudes $|\mathbf{X}| = |\mathbf{S}_1| + |\mathbf{S}_2| + \ldots + |\mathbf{S}_J|$, consider additive complex-valued sources:

$$\begin{aligned} \mathbf{X} &= \mathbf{S}_1 + \mathbf{S}_2 + \ldots + \mathbf{S}_J \\ &= \mathbf{V}_1 e^{\mathrm{i}\boldsymbol{\mu}_1} + \mathbf{V}_2 e^{\mathrm{i}\boldsymbol{\mu}_2} + \ldots + \mathbf{V}_J e^{\mathrm{i}\boldsymbol{\mu}_J} \end{aligned}$$

And factorize each spectrogram with NMF: $\mathbf{V}_j = \mathbf{W}_j \mathbf{H}_j$

Estimation

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\mu}} ||\mathbf{X} - \sum_{j=1}^{J} [\mathbf{W}_j \mathbf{H}_j] e^{\mathrm{i}\boldsymbol{\mu}_j}||^2 + \mathcal{C}(\boldsymbol{\mu})$$

▷ Add some model-based phase regularization (e.g., sinusoidal).

Complex NMF

Instead of assuming additive magnitudes $|\mathbf{X}| = |\mathbf{S}_1| + |\mathbf{S}_2| + \ldots + |\mathbf{S}_J|$, consider additive complex-valued sources:

$$\begin{aligned} \mathbf{X} &= \mathbf{S}_1 + \mathbf{S}_2 + \ldots + \mathbf{S}_J \\ &= \mathbf{V}_1 e^{\mathrm{i}\boldsymbol{\mu}_1} + \mathbf{V}_2 e^{\mathrm{i}\boldsymbol{\mu}_2} + \ldots + \mathbf{V}_J e^{\mathrm{i}\boldsymbol{\mu}_J} \end{aligned}$$

And factorize each spectrogram with NMF: $\mathbf{V}_j = \mathbf{W}_j \mathbf{H}_j$

Estimation

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\mu}} ||\mathbf{X} - \sum_{j=1}^{J} [\mathbf{W}_j \mathbf{H}_j] e^{\mathrm{i}\boldsymbol{\mu}_j}||^2 + \mathcal{C}(\boldsymbol{\mu})$$

▷ Add some model-based phase regularization (e.g., sinusoidal).

Performance

▷ Complex NMF > NMF: the advantage of accounting for the phase.

Complex NMF

Instead of assuming additive magnitudes $|\mathbf{X}| = |\mathbf{S}_1| + |\mathbf{S}_2| + \ldots + |\mathbf{S}_J|$, consider additive complex-valued sources:

$$\begin{aligned}\mathbf{X} &= \mathbf{S}_1 + \mathbf{S}_2 + \ldots + \mathbf{S}_J \\ &= \mathbf{V}_1 e^{\mathrm{i}\boldsymbol{\mu}_1} + \mathbf{V}_2 e^{\mathrm{i}\boldsymbol{\mu}_2} + \ldots + \mathbf{V}_J e^{\mathrm{i}\boldsymbol{\mu}_J}\end{aligned}$$

And factorize each spectrogram with NMF: $\mathbf{V}_j = \mathbf{W}_j \mathbf{H}_j$

Estimation

$$\min_{\mathbf{W},\mathbf{H},\boldsymbol{\mu}} ||\mathbf{X} - \sum_{j=1}^{J} [\mathbf{W}_j \mathbf{H}_j] e^{\mathrm{i}\boldsymbol{\mu}_j}||^2 + \mathcal{C}(\boldsymbol{\mu})$$

- ▷ Add some model-based phase regularization (e.g., sinusoidal).

Performance

- ▷ Complex NMF $>$ NMF: the advantage of accounting for the phase.
- ▷ Complex NMF $>$ NMF $+$ phase recovery: the advantage of a joint training approach.

Perspective: learning to factorize

- ✗ The low-rank assumption does not hold in practice for spectrograms.

Perspective: learning to factorize

✗ The low-rank assumption does not hold in practice for spectrograms.

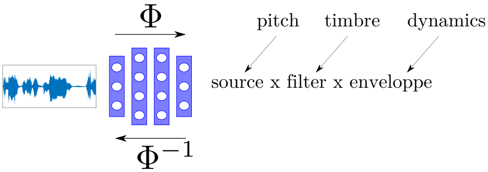
Proposal: leverage DNNs to transform the data and make it "factorizable".

Perspective: learning to factorize

✗ The low-rank assumption does not hold in practice for spectrograms.

Proposal: leverage DNNs to transform the data and make it "factorizable".

- ✓ High expressive power of DNNs.
- ✓ Interpretability of the factorization.



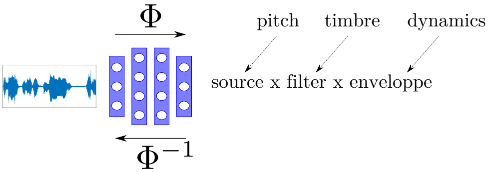
Perspective: learning to factorize

✗ The low-rank assumption does not hold in practice for spectrograms.

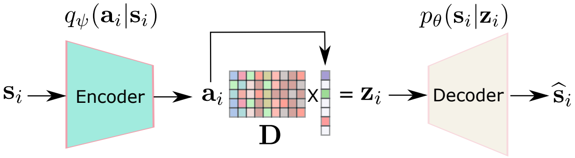
Proposal: leverage DNNs to transform the data and make it “factorizable”.

✓ High expressive power of DNNs.

✓ Interpretability of the factorization.



A first attempt: VAE with a sparse dictionary model.



✓ Nice performance in terms of sparsity and speech modeling / reconstruction.

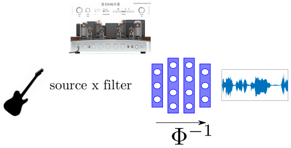
✗ Fixed dictionary and no nonnegativity: non-interpretable factors.

A (distant) dream: joint synthesis / separation

- Assume we have a high quality parametric generative model for a given source.

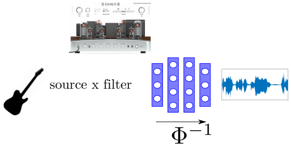
A (distant) dream: joint synthesis / separation

- Assume we have a high quality parametric generative model for a given source.
- Reconsider it as an synthesis model from some factorized latent space.

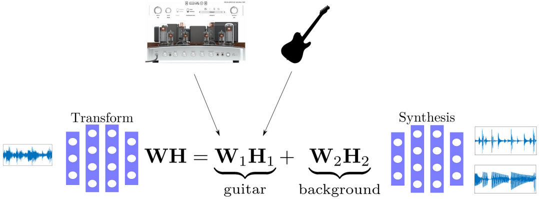


A (distant) dream: joint synthesis / separation

- Assume we have a high quality parametric generative model for a given source.
- Reconsider it as an synthesis model from some factorized latent space.

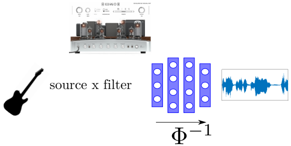


Proposal: incorporate the generative model into the demixing pipeline.

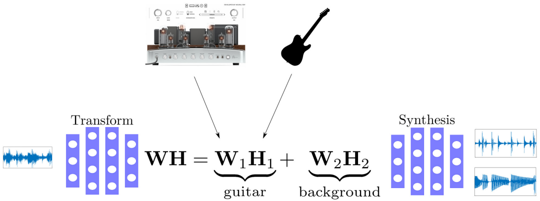


A (distant) dream: joint synthesis / separation

- Assume we have a high quality parametric generative model for a given source.
- Reconsider it as an synthesis model from some factorized latent space.

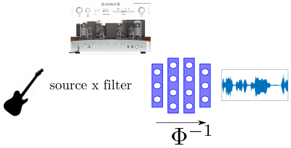


Proposal: incorporate the generative model into the demixing pipeline.



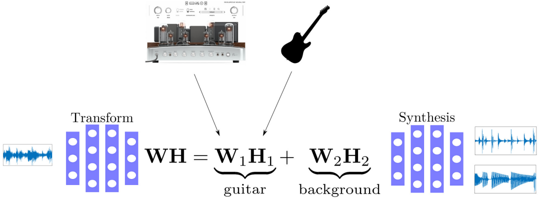
✓ High quality backing track generation.

A (distant) dream: joint synthesis / separation



- ▷ Assume we have a high quality parametric generative model for a given source.
- ▷ Reconsider it as an synthesis model from some factorized latent space.

Proposal: incorporate the generative model into the demixing pipeline.



✓ High quality backing track generation.

✓ Optimal generative model parameters = a preset!

Conclusion

Where are we now?

Main messages

- ▷ The room for improvement of phase recovery: more potential gain than with magnitudes.
- ▷ A promising approach: leveraging model-based phase properties.

Where are we now?

Main messages

- The room for improvement of phase recovery: more potential gain than with magnitudes.
- A promising approach: leveraging model-based phase properties.

The current trend: from nonnegative to time-domain deep learning.

Where are we now?

Main messages

- ▷ The room for improvement of phase recovery: more potential gain than with magnitudes.
- ▷ A promising approach: leveraging model-based phase properties.

The current trend: from nonnegative to time-domain deep learning.

✓ Performance in controlled conditions.

✓ No more phase problem.

Where are we now?

Main messages

- ▷ The room for improvement of phase recovery: more potential gain than with magnitudes.
- ▷ A promising approach: leveraging model-based phase properties.

The current trend: from nonnegative to time-domain deep learning.

✓ Performance in controlled conditions.

✓ No more phase problem.

✗ Greediness in (annotated) training data.

Where are we now?

Main messages

- ▷ The room for improvement of phase recovery: more potential gain than with magnitudes.
- ▷ A promising approach: leveraging model-based phase properties.

The current trend: from nonnegative to time-domain deep learning.

✓ Performance in controlled conditions.

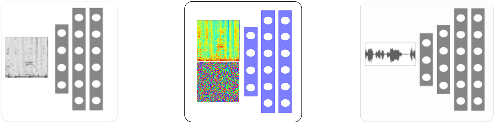
✓ No more phase problem.

✗ Greediness in (annotated) training data.

✗ Lacks interpretability and flexibility.

The proposed alternative

Complex-domain / phase-aware deep learning

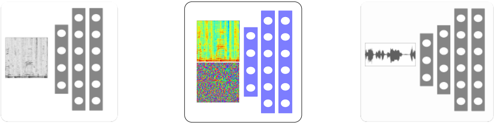


✓ Robustness/flexibility of time-frequency processing.

✓ Performance of processing all the data exhaustively.

The proposed alternative

Complex-domain / phase-aware deep learning



✓ Robustness/flexibility of time-frequency processing.

✓ Performance of processing all the data exhaustively.

Open questions

▷ How to handle phase in deep learning?

▷ How to exploit anisotropic probabilistic modeling?

▷ How to efficiently learn to factorize?

References

Magron et al., “Phase reconstruction of spectrograms with linear unwrapping: application to audio signal restoration”, *Proc. EUSIPCO*, August 2015.

Magron et al., “Model-based STFT phase recovery for audio source separation”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, June 2018.

Magron and Virtanen, “On modeling the STFT phase of audio signals with the von Mises distribution”, *Proc. IWAENC*, September 2018.

Magron et al., “Phase-dependent anisotropic Gaussian model for audio source separation”, *Proc. IEEE ICASSP*, March 2017.

Magron et al., “Complex NMF under phase constraints based on signal modeling: application to audio source separation”, *Proc. IEEE ICASSP*, March 2016.

Sadeghi and Magron, “A sparsity-promoting dictionary model for variational autoencoders”, *Proc. of Interspeech*, September 2022.

Thanks!

https://magronp.github.io/

https://github.com/magronp/