Nonnegative Matrix Factorization for Audio Source Separation

The rise, fall, and resurgence

Paul Magron, Researcher - INRIA Centre at Université de Lorraine

Workshop on Low-Rank Models and Applications (LRMA), Mons - September 11th, 2025

Inria MULTISPEECH

Audio source separation

Audio signals are composed of several constitutive sounds.

- multiple speakers, background noise, domestic sounds, musical instruments . . .

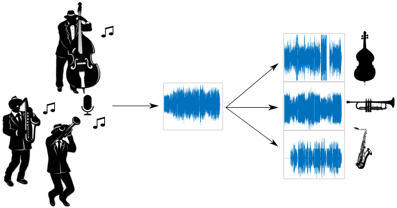
Audio source separation

Audio signals are composed of several constitutive sounds.

- multiple speakers, background noise, domestic sounds, musical instruments . . .

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

- An important preprocessing for many downstream tasks.
  - Automatic speech recognition.
  - Music transcription / information retrieval.
  - Acoustic scene analysis / sound event detection.
- A goal in itself for synthesis purposes.
  - Augmented mixing, e.g., from mono to stereo.
  - Backing track generation / karaoke.



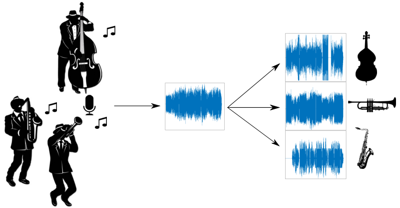
Audio source separation

Audio signals are composed of several constitutive sounds.

- ▷ multiple speakers, background noise, domestic sounds, musical instruments ...

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

- ▷ An important preprocessing for many downstream tasks.
  - ▷ Automatic speech recognition.
  - ▷ Music transcription / information retrieval.
  - ▷ Acoustic scene analysis / sound event detection.
- ▷ A goal in itself for synthesis purposes.
  - ▷ Augmented mixing, e.g., from mono to stereo.
  - ▷ Backing track generation / karaoke.



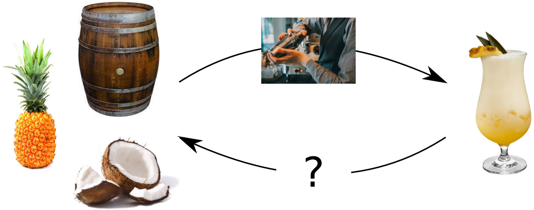
Beyond audio: Biomedical signals, astronomy imaging, fluorescence spectroscopy, etc.

A difficult task?

Mixing is easy . . .

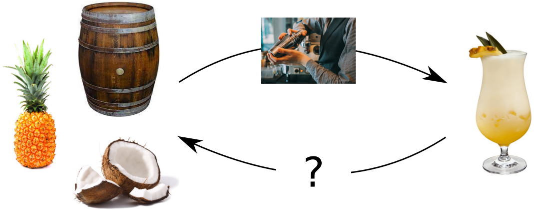
A difficult task?

Mixing is easy . . . but demixing is not.



A difficult task?

Mixing is easy ... but demixing is not.

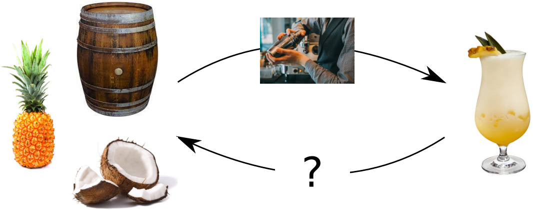


Finding $\{\mathbf{s}_j \in \mathbb{R}^N\}_{j=1}^J$ such that $\mathbf{x} = \sum_{j=1}^{J} \mathbf{s}_j$ is an under-determined problem.

+ domain-specific challenges: correlated music sources, speaker variability, reverberation/noise...

A difficult task?

Mixing is easy . . . but demixing is not.



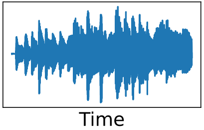
Finding $\{\mathbf{s}_j \in \mathbb{R}^N\}_{j=1}^J$ such that $\mathbf{x} = \sum_{j=1}^{J} \mathbf{s}_j$ is an under-determined problem.

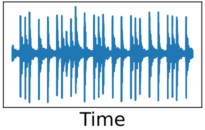
$+$ domain-specific challenges: correlated music sources, speaker variability, reverberation/noise. . .

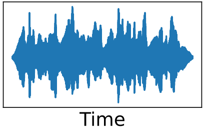
- ▷ Need to incorporate additional information / constraints / structure.
- ▷ Either via expert knowledge or by leveraging data.

Setting the stage

The raw material are audio signals $\mathbf{s}_j$

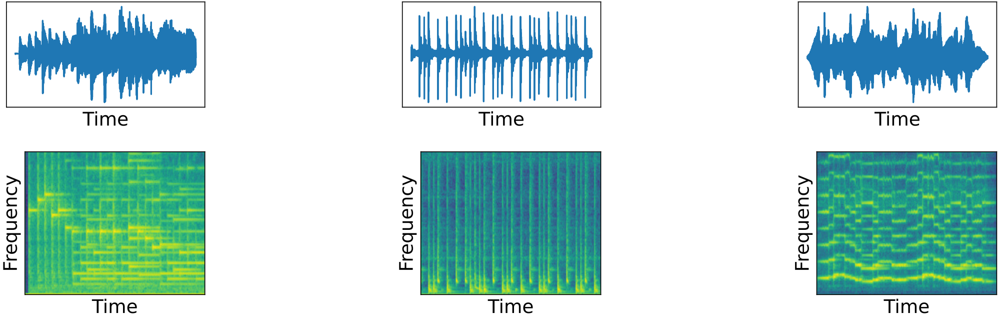






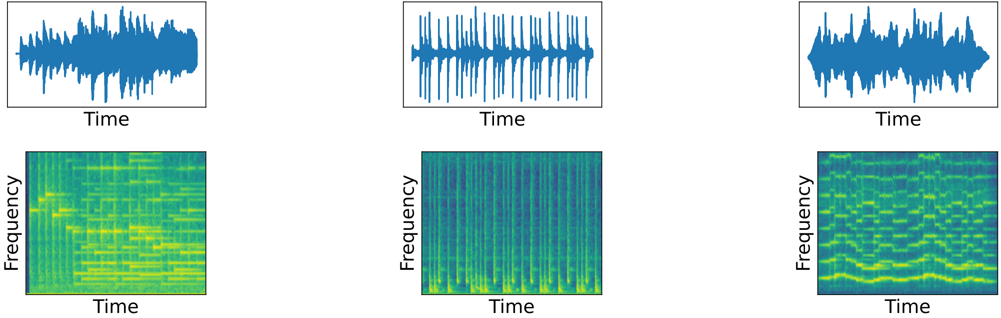
Setting the stage

The raw material are audio signals $\mathbf{s}_j$, but we rather consider a time-frequency representation.

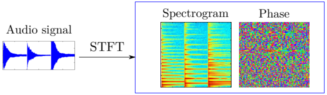


Setting the stage

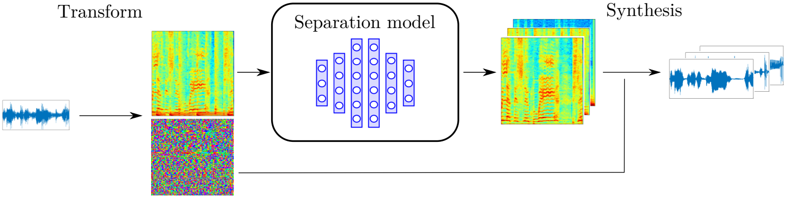
The raw material are audio signals $\mathbf{s}_j$, but we rather consider a time-frequency representation.



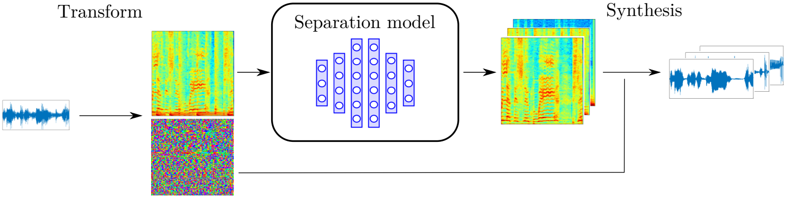
- A popular choice: the short-time Fourier transform (STFT).
- The mixture model becomes $\mathbf{X} = \sum_{j=1}^{J} \mathbf{S}_j \in \mathbb{C}^{F \times T}$.



The separation pipeline



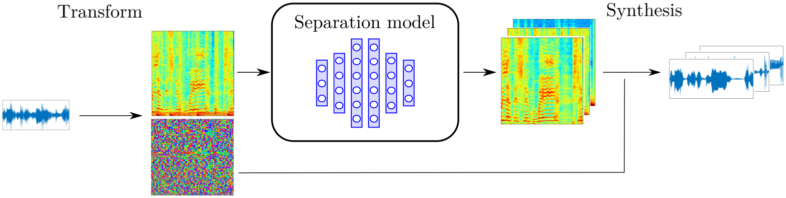
The separation pipeline



The separator consists of a spectral model:

- A (linear) low-rank / matrix factorization approximation, with some constraints, e.g., statistical independance, sparsity, nonnegativity.
- A nonlinear model based on deep neural networks (DNNs).

The separation pipeline



The separator consists of a spectral model:

- ▷ A (linear) low-rank / matrix factorization approximation, with some constraints, e.g., statistical independance, sparsity, nonnegativity.
- ▷ A nonlinear model based on deep neural networks (DNNs).

Synthesis is performed by inverse STFT on top of:

- ▷ Spectral and/or spatial filtering (e.g., Wiener filtering).
- ▷ A phase recovery / spectrogram inversion stage.

The NMF trend

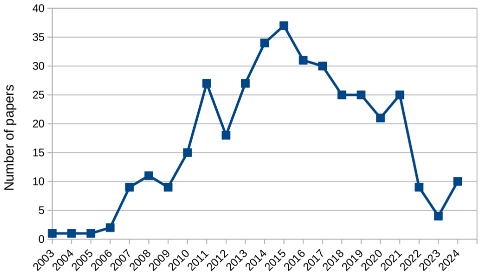
Let's have some fun on IEEE Xplore!

- Search for papers whose title contains "nonnegative matrix factorization", up to variations ("non-negative" instead of "nonnegative", "NMF", etc.).
- Filter publication topics containing "source separation" and variants ("blind source separation", "music separation", etc.).

The NMF trend

Let's have some fun on IEEE Xplore!

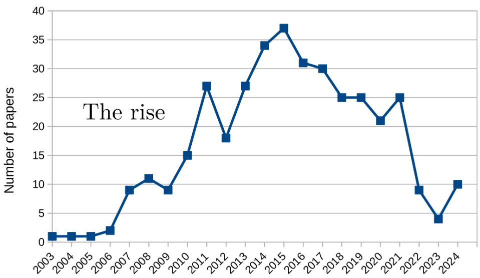
- Search for papers whose title contains "nonnegative matrix factorization", up to variations ("non-negative" instead of "nonnegative", "NMF", etc.).
- Filter publication topics containing "source separation" and variants ("blind source separation", "music separation", etc.).



The NMF trend

Let's have some fun on IEEE Xplore!

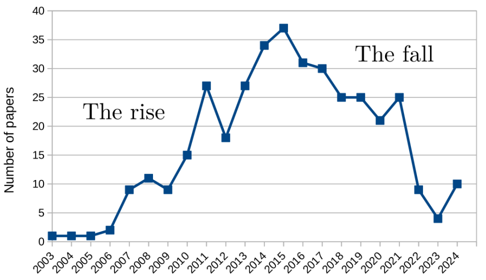
- Search for papers whose title contains "nonnegative matrix factorization", up to variations ("non-negative" instead of "nonnegative", "NMF", etc.).
- Filter publication topics containing "source separation" and variants ("blind source separation", "music separation", etc.).



The NMF trend

Let's have some fun on IEEE Xplore!

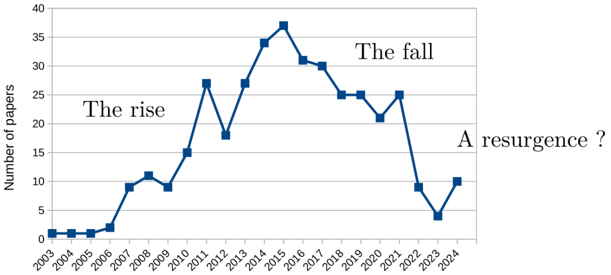
- ▷ Search for papers whose title contains "nonnegative matrix factorization", up to variations ("non-negative" instead of "nonnegative", "NMF", etc.).
- ▷ Filter publication topics containing "source separation" and variants ("blind source separation", "music separation", etc.).



The NMF trend

Let's have some fun on IEEE Xplore!

- Search for papers whose title contains "nonnegative matrix factorization", up to variations ("non-negative" instead of "nonnegative", "NMF", etc.).
- Filter publication topics containing "source separation" and variants ("blind source separation", "music separation", etc.).



The rise

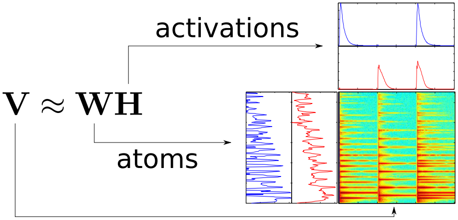
Nonnegative matrix factorization (NMF)

Given a (nonnegative) data matrix $\mathbf{V} \in \mathbb{R}^{F \times T}$, find a factorization $\mathbf{WH}$ such that the factors $\mathbf{W} \in \mathbb{R}^{F \times K}$ and $\mathbf{H} \in \mathbb{R}^{K \times T}$ are low-rank $(K \ll \min(F, T))$ and nonnegative.

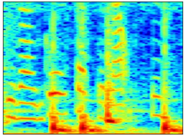
- ▷ $\mathbf{V}$ is usually a magnitude $|\mathbf{X}|$ or power $|\mathbf{X}|^2$ spectrogram.
- ▷ $\mathbf{W}$ is a dictionary of spectral atoms.
- ▷ $\mathbf{H}$ is a matrix of temporal activation.

Nonnegativity favors:

- ▷ interpretability of the factors.
- ▷ a part-based decomposition of the data.

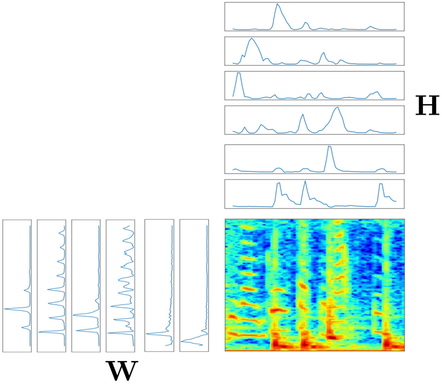


NMF for (blind) source separation



Exploit additivity for getting each source spectrogram.

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



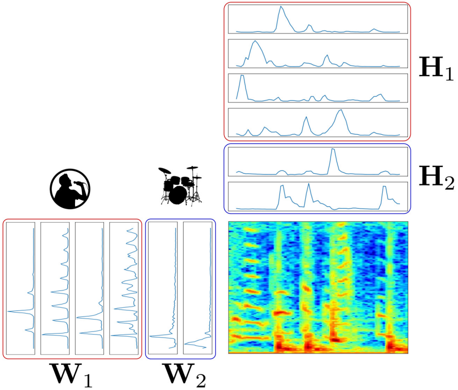
Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).

NMF for (blind) source separation



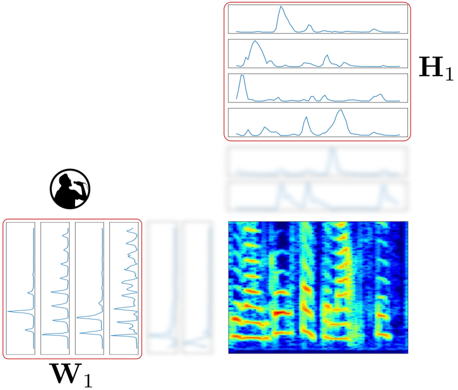
Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).
2. Cluster atoms $\mathbf{w}_k$ that belong to the same source to build source-specific matrices: $\mathbf{W}_j = \{\mathbf{w}_k\}_{k \in \mathcal{K}_j}$

NMF for (blind) source separation



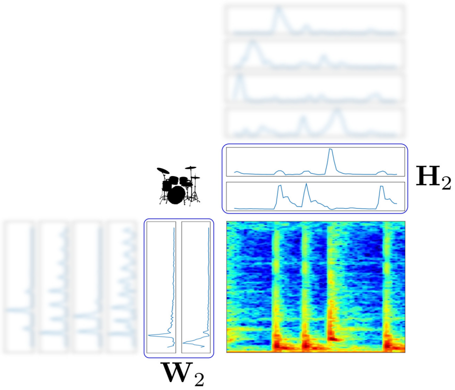
Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).
2. Cluster atoms $\mathbf{w}_k$ that belong to the same source to build source-specific matrices: $\mathbf{W}_j = \{\mathbf{w}_k\}_{k \in \mathcal{K}_j}$
3. Multiply each dictionary with the corresponding activations to retrieve each source spectrogram.

NMF for (blind) source separation



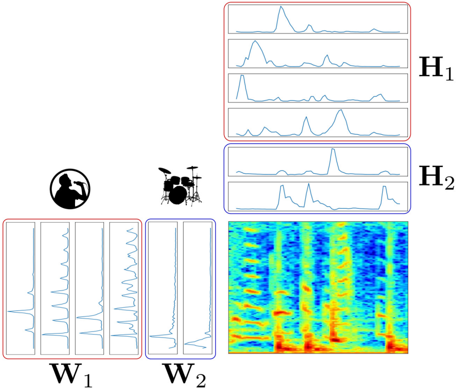
Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).
2. Cluster atoms $\mathbf{w}_k$ that belong to the same source to build source-specific matrices: $\mathbf{W}_j = \{\mathbf{w}_k\}_{k \in \mathcal{K}_j}$
3. Multiply each dictionary with the corresponding activations to retrieve each source spectrogram.

NMF for (blind) source separation



Exploit additivity for getting each source spectrogram.

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

Procedure

1. Factorize the mixture's spectrogram (i.e., find $\mathbf{W}$ and $\mathbf{H}$ by solving the optimization problem).
2. Cluster atoms $\mathbf{w}_k$ that belong to the same source to build source-specific matrices: $\mathbf{W}_j = \{\mathbf{w}_k\}_{k \in \mathcal{K}_j}$
3. Multiply each dictionary with the corresponding activations to retrieve each source spectrogram.

Introducing supervision

Assume a set of isolated source signals is available (= a training dataset).

- ▷ Pretrain each source dictionary from the corresponding isolated spectrogram $\mathbf{V}_j^{\text{pretrain}}$:

$$\mathbf{W}_j^{\text{pretrain}} = \arg\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}_j^{\text{pretrain}}, \mathbf{WH})$$

Introducing supervision

Assume a set of isolated source signals is available (= a training dataset).

- ▷ Pretrain each source dictionary from the corresponding isolated spectrogram $\mathbf{V}_j^{\text{pretrain}}$:

$$\mathbf{W}_j^{\text{pretrain}} = \arg\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}_j^{\text{pretrain}}, \mathbf{W}\mathbf{H})$$

- ▷ On the mixture, fix the dictionaries and only estimate the activation:

$$[\mathbf{H}_1, \ldots, \mathbf{H}_J] = \arg\min_{\mathbf{H}} D(\mathbf{V}, [\mathbf{W}_1^{\text{pretrain}}, \ldots, \mathbf{W}_J^{\text{pretrain}}]\mathbf{H})$$

Introducing supervision

Assume a set of isolated source signals is available (= a training dataset).

- ▷ Pretrain each source dictionary from the corresponding isolated spectrogram $\mathbf{V}_j^{\text{pretrain}}$:

$$\mathbf{W}_j^{\text{pretrain}} = \arg\min_{\mathbf{W},\mathbf{H}} D(\mathbf{V}_j^{\text{pretrain}}, \mathbf{WH})$$

- ▷ On the mixture, fix the dictionaries and only estimate the activation:

$$[\mathbf{H}_1, \ldots, \mathbf{H}_J] = \arg\min_{\mathbf{H}} D(\mathbf{V}, [\mathbf{W}_1^{\text{pretrain}}, \ldots, \mathbf{W}_J^{\text{pretrain}}]\mathbf{H})$$

- ▷ Retrieve each source's spectrogram via $\mathbf{V}_j = \mathbf{W}_j^{\text{pretrain}}\mathbf{H}_j$.

Estimation - problem setting

Optimization-based model estimation:

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} D(\mathbf{V}, \mathbf{WH}) \quad + \text{regularizations}$$

The literature is (very) abundant (Gillis 2020).

Estimation - problem setting

Optimization-based model estimation:

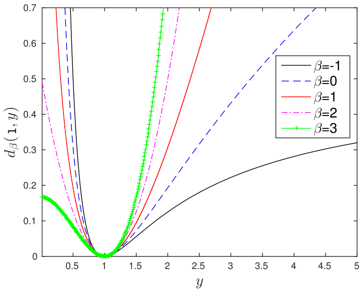
$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} D(\mathbf{V}, \mathbf{WH}) \quad + \text{regularizations}$$

The literature is (very) abundant (Gillis 2020).

NMF with the beta-divergences (Févotte et al. 2009)

$$D(\mathbf{V}, \mathbf{WH}) = \sum_{f,t} d_\beta(v_{f,t}, [\mathbf{WH}]_{f,t})$$

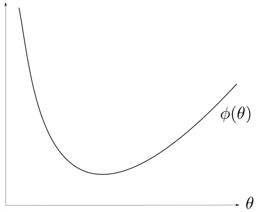
- Interesting in audio: (quasi)-scale invariance, better fits human perception.
- Popular special cases:
  - Euclidean distance ($\beta = 2$).
  - Kullback-Leibler (KL) divergence ($\beta = 1$).
  - Itakura-Saito (IS) divergence ($\beta = 0$).



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

Estimation via majorization-minimization (MM)

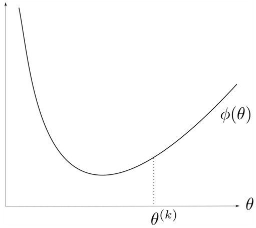
Procedure to minimize ϕ:



Estimation via majorization-minimization (MM)

Procedure to minimize ϕ:

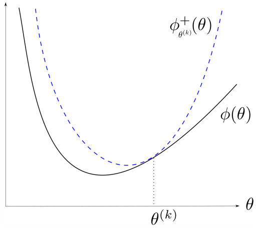
- ▷ Given a current estimate $\theta^{(k)}$



Estimation via majorization-minimization (MM)

Procedure to minimize ϕ:

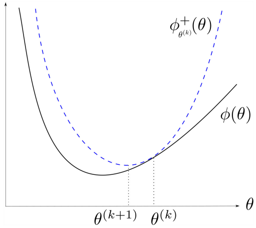
- Given a current estimate $\theta^{(k)}$, construct a majorizing function ϕ^+ of ϕ that is tight at $\theta^{(k)}$.



Estimation via majorization-minimization (MM)

Procedure to minimize ϕ:

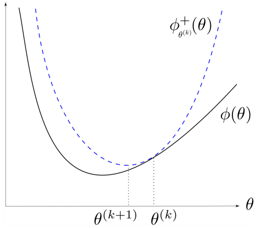
- Given a current estimate $\theta^{(k)}$, construct a majorizing function ϕ^+ of ϕ that is tight at $\theta^{(k)}$.
- Minimize ϕ^+ to get an updated estimate $\theta^{(k+1)}$.



Estimation via majorization-minimization (MM)

Procedure to minimize ϕ:

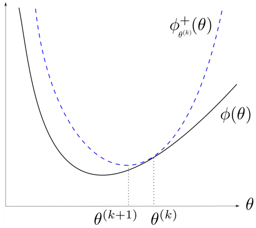
- Given a current estimate $\theta^{(k)}$, construct a majorizing function ϕ^+ of ϕ that is tight at $\theta^{(k)}$.
- Minimize ϕ^+ to get an updated estimate $\theta^{(k+1)}$.
- Then $\phi(\theta^{(k+1)}) \leq \phi(\theta^{(k)})$.



Estimation via majorization-minimization (MM)

Procedure to minimize ϕ:

- Given a current estimate $\theta^{(k)}$, construct a majorizing function ϕ^+ of ϕ that is tight at $\theta^{(k)}$.
- Minimize ϕ^+ to get an updated estimate $\theta^{(k+1)}$.
- Then $\phi(\theta^{(k+1)}) \leq \phi(\theta^{(k)})$.



For NMF (Févotte et al. 2011)

- The divergence is split into a convex and a concave part.
- Majorization using convexity and tangent inequalities.
- Solving yields multiplicative updates.
- Convergence-guaranteed, no hyperparameter to tune.

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \frac{(\mathbf{V} \cdot [\mathbf{WH}]^{\beta-2})\mathbf{H}^T}{[\mathbf{WH}]^{\beta-1}\mathbf{H}^T}$$

$$\mathbf{H} \leftarrow \mathbf{H} \cdot \frac{\mathbf{W}^T(\mathbf{V} \cdot [\mathbf{WH}]^{\beta-2})}{\mathbf{W}^T[\mathbf{WH}]^{\beta-1}}$$

Refining the model

Regularizations injected in the optimization problem as soft penalties.

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} D(\mathbf{V}, \mathbf{WH}) + \lambda \mathcal{R}$$

- Sparsity of the activations (Le Roux et al. 2015b): $\mathcal{R}(\mathbf{H}) = \sum_{k,t} h_{k,t}$
- Temporal smoothness (Virtanen 2007): $\mathcal{R}(\mathbf{H}) = \sum_{k,t} (h_{k,t} - h_{k,t-1})^2$

Refining the model

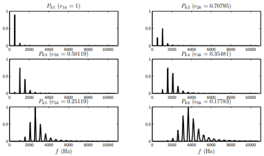
Regularizations injected in the optimization problem as soft penalties.

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} D(\mathbf{V}, \mathbf{WH}) + \lambda \mathcal{R}$$

- Sparsity of the activations (Le Roux et al. 2015b): $\mathcal{R}(\mathbf{H}) = \sum_{k,t} h_{k,t}$
- Temporal smoothness (Virtanen 2007): $\mathcal{R}(\mathbf{H}) = \sum_{k,t} (h_{k,t} - h_{k,t-1})^2$

or model variants where the constraint is hard-coded.

- Harmonic spectra (Bertin et al. 2010): $w_{f,k} = \sum_m e_{m,k} P_{k,m,f}$ where patterns $P_{k,m}$ contain equally-spaced partials.



Source: (Bertin et al. 2010)

Refining the model

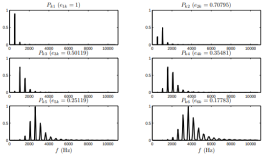
Regularizations injected in the optimization problem as soft penalties.

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} D(\mathbf{V}, \mathbf{WH}) + \lambda \mathcal{R}$$

- Sparsity of the activations (Le Roux et al. 2015b): $\mathcal{R}(\mathbf{H}) = \sum_{k,t} h_{k,t}$
- Temporal smoothness (Virtanen 2007): $\mathcal{R}(\mathbf{H}) = \sum_{k,t} (h_{k,t} - h_{k,t-1})^2$

or model variants where the constraint is hard-coded.

- Harmonic spectra (Bertin et al. 2010): $w_{f,k} = \sum_m e_{m,k} P_{k,m,f}$ where patterns $P_{k,m}$ contain equally-spaced partials.
- Orthogonality (Choi 2008) of the activations $\mathbf{H}\mathbf{H}^T = \mathbf{I}$ or the spectra $\mathbf{W}^T\mathbf{W} = \mathbf{I}$.



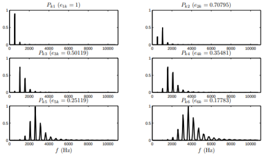
Source: (Bertin et al. 2010)

Refining the model

Regularizations injected in the optimization problem as soft penalties.

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} D(\mathbf{V},\mathbf{WH}) + \lambda\mathcal{R}$$

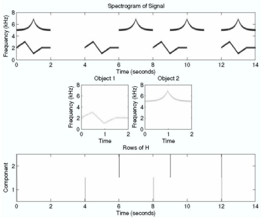
- Sparsity of the activations (Le Roux et al. 2015b): $\mathcal{R}(\mathbf{H}) = \sum_{k,t} h_{k,t}$
- Temporal smoothness (Virtanen 2007): $\mathcal{R}(\mathbf{H}) = \sum_{k,t}(h_{k,t} - h_{k,t-1})^2$



Source: (Bertin et al. 2010)

or model variants where the constraint is hard-coded.

- Harmonic spectra (Bertin et al. 2010): $w_{f,k} = \sum_m e_{m,k} P_{k,m,f}$ where patterns $P_{k,m}$ contain equally-spaced partials.
- Orthogonality (Choi 2008) of the activations $\mathbf{HH}^T = \mathbf{I}$ or the spectra $\mathbf{W}^T\mathbf{W} = \mathbf{I}$.
- Convolutive NMF (O'Grady et al. 2006): $\mathbf{V} \approx \mathbf{W} \circledast \mathbf{H}$, where $\mathbf{W} \in \mathbb{R}^{F\times K\times L}$ contains time-varying templates.

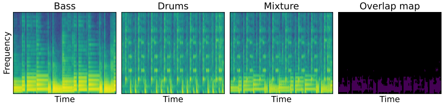


Source: (O'Grady et al. 2006)

An example: complex NMF

NMF-based spectrogram decomposition

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

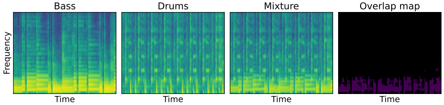


- Additivity of the sources' magnitudes / phase is ignored.
- Limiting assumption when sources *overlap*.

An example: complex NMF

NMF-based spectrogram decomposition

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



- ▷ Additivity of the sources' magnitudes / phase is ignored.
- ▷ Limiting assumption when sources *overlap*.

Complex NMF (Kameoka et al. 2009)

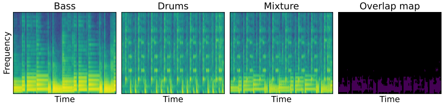
✓ Assumes additivity of the sources' STFTs, and factorizes each source's magnitude.

$$\mathbf{X} \approx \sum_{j=1}^{J} \mathbf{W}_j \mathbf{H}_j e^{\mathrm{i}\boldsymbol{\mu}_j}$$

An example: complex NMF

NMF-based spectrogram decomposition

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



- ▷ Additivity of the sources' magnitudes / phase is ignored.
- ▷ Limiting assumption when sources *overlap*.

Complex NMF (Kameoka et al. 2009)

✓ Assumes additivity of the sources' STFTs, and factorizes each source's magnitude.

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

- ▷ Model-based phase regularizations (Le Roux et al. 2009; Bronson et al. 2014; Magron et al. 2016).

Extending complex NMF to beta-divergences

- NMF can be estimated using a variety of loss functions (e.g., beta-divergences).
- Complex NMF is estimated using the Euclidean distance.
- Most beta-divergences are defined for nonnegative quantities only.

Extending complex NMF to beta-divergences

- NMF can be estimated using a variety of loss functions (e.g., beta-divergences).
- Complex NMF is estimated using the Euclidean distance.
- Most beta-divergences are defined for nonnegative quantities only.

Problem: How to extend complex NMF to non-Euclidean losses?

Extending complex NMF to beta-divergences

- ▷ NMF can be estimated using a variety of loss functions (e.g., beta-divergences).
- ▷ Complex NMF is estimated using the Euclidean distance.
- ▷ Most beta-divergences are defined for nonnegative quantities only.

Problem: How to extend complex NMF to non-Euclidean losses?

A probabilistic view on NMF (Simsekli et al. 2013)

- ▷ NMF can be used in a probabilistic model to structure some distribution's parameter.
- ▷ Maximum likelihood estimation yields a loss function that depends on the statistical model.

Extending complex NMF to beta-divergences

- NMF can be estimated using a variety of loss functions (e.g., beta-divergences).
- Complex NMF is estimated using the Euclidean distance.
- Most beta-divergences are defined for nonnegative quantities only.

Problem: How to extend complex NMF to non-Euclidean losses?

A probabilistic view on NMF (Simsekli et al. 2013)

- NMF can be used in a probabilistic model to structure some distribution's parameter.
- Maximum likelihood estimation yields a loss function that depends on the statistical model.

	Variable	Distribution	Model	Parametrization	Loss
Euc. NMF	Magnitude	(Real) Gaussian	$\mathcal{N}(m, \sigma^2)$	$m = wh$	Euclidean

Extending complex NMF to beta-divergences

- NMF can be estimated using a variety of loss functions (e.g., beta-divergences).
- Complex NMF is estimated using the Euclidean distance.
- Most beta-divergences are defined for nonnegative quantities only.

Problem: How to extend complex NMF to non-Euclidean losses?

A probabilistic view on NMF (Simsekli et al. 2013)

- NMF can be used in a probabilistic model to structure some distribution's parameter.
- Maximum likelihood estimation yields a loss function that depends on the statistical model.

	Variable	Distribution	Model	Parametrization	Loss
Euc. NMF	Magnitude	(Real) Gaussian	$\mathcal{N}(m, \sigma^2)$	$m = wh$	Euclidean
KLNMF	Magnitude	Poisson	$\mathcal{P}(v)$	$v = wh$	Kullback-Leibler

Extending complex NMF to beta-divergences

- NMF can be estimated using a variety of loss functions (e.g., beta-divergences).
- Complex NMF is estimated using the Euclidean distance.
- Most beta-divergences are defined for nonnegative quantities only.

Problem: How to extend complex NMF to non-Euclidean losses?

A probabilistic view on NMF (Simsekli et al. 2013)

- NMF can be used in a probabilistic model to structure some distribution's parameter.
- Maximum likelihood estimation yields a loss function that depends on the statistical model.

	Variable	Distribution	Model	Parametrization	Loss
Euc. NMF	Magnitude	(Real) Gaussian	$\mathcal{N}(m, \sigma^2)$	$m = wh$	Euclidean
KLNMF	Magnitude	Poisson	$\mathcal{P}(v)$	$v = wh$	Kullback-Leibler
ISNMF	STFT	Isotropic Gaussian	$\mathcal{N}_{\mathbb{C}}(0, v^2 I)$	$v^2 = wh$	Itakura-Saito

Extending complex NMF to beta-divergences

- NMF can be estimated using a variety of loss functions (e.g., beta-divergences).
- Complex NMF is estimated using the Euclidean distance.
- Most beta-divergences are defined for nonnegative quantities only.

Problem: How to extend complex NMF to non-Euclidean losses?

A probabilistic view on NMF (Simsekli et al. 2013)

- NMF can be used in a probabilistic model to structure some distribution's parameter.
- Maximum likelihood estimation yields a loss function that depends on the statistical model.

	Variable	Distribution	Model	Parametrization	Loss
Euc. NMF	Magnitude	(Real) Gaussian	$\mathcal{N}(m, \sigma^2)$	$m = wh$	Euclidean
KLNMF	Magnitude	Poisson	$\mathcal{P}(v)$	$v = wh$	Kullback-Leibler
ISNMF	STFT	Isotropic Gaussian	$\mathcal{N}_{\mathbb{C}}(0, v^2 I)$	$v^2 = wh$	Itakura-Saito
Complex NMF	STFT	Isotropic Gaussian	$\mathcal{N}_{\mathbb{C}}(m, \sigma^2 I)$	$m = whe^{i\mu}$	Euclidean

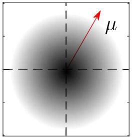
Complex ISNMF (Magron et al. 2019; Magron et al. 2018)

From isotropic sources (ISNMF)

$$\Gamma_j = \begin{pmatrix} w_j h_j & 0 \\ 0 & w_j h_j \end{pmatrix}$$

$$s_j \sim \mathcal{N}_{\mathbb{C}}(0, \Gamma_j)$$

Isotropic

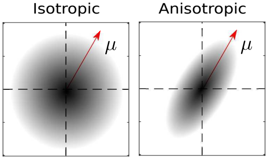


Complex ISNMF (Magron et al. 2019; Magron et al. 2018)

From isotropic sources (ISNMF) to anisotropic sources (Complex ISNMF).

$$\Gamma_j = \begin{pmatrix} \lambda w_j h_j & \rho w_j h_j e^{2i\mu_j} \\ \rho w_j h_j e^{-2i\mu_j} & \lambda w_j h_j \end{pmatrix}$$

$$s_j \sim \mathcal{N}_{\mathbb{C}}(0, \Gamma_j)$$



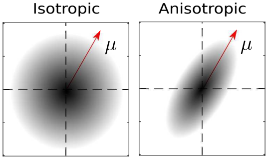
- Non-zero *relation parameter*: the phase is no longer uniform.
- λ / ρ adjust the importance of the phase.
- Prior on the phase parameters μ_j (e.g., Markov chain).

Complex ISNMF (Magron et al. 2019; Magron et al. 2018)

From isotropic sources (ISNMF) to anisotropic sources (Complex ISNMF).

$$\Gamma_j = \begin{pmatrix} \lambda w_j h_j & \rho w_j h_j e^{2i\mu_j} \\ \rho w_j h_j e^{-2i\mu_j} & \lambda w_j h_j \end{pmatrix}$$

$$s_j \sim \mathcal{N}_{\mathbb{C}}(0, \Gamma_j)$$



- ▷ Non-zero *relation parameter*: the phase is no longer uniform.
- ▷ λ / ρ adjust the importance of the phase.
- ▷ Prior on the phase parameters μ_j (e.g., Markov chain).

Estimation via expectation-maximization:

- ▷ E-step: compute the posterior moments.
- ▷ M-step: an IS divergence minimization problem.
  - ✓ Outperforms complex NMF and ISNMF.

The fall

What happened?

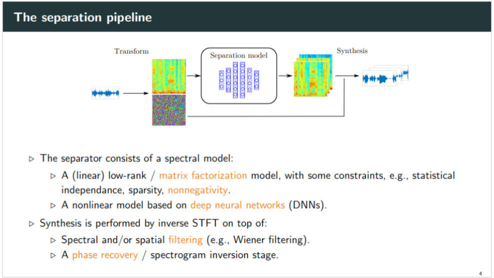
Some limitations of NMF:

- Spectrograms are perhaps *not low-rank*.
- Interactions between spectral templates and activations are perhaps *not linear*.

What happened?

Some limitations of NMF:

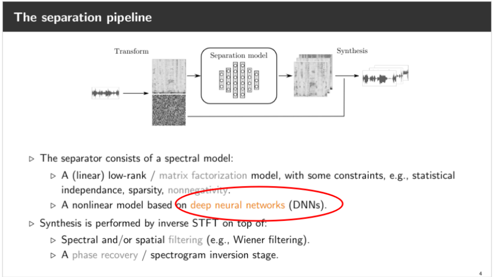
- ▷ Spectrograms are perhaps *not low-rank*.
- ▷ Interactions between spectral templates and activations are perhaps *not linear*.



What happened?

Some limitations of NMF:

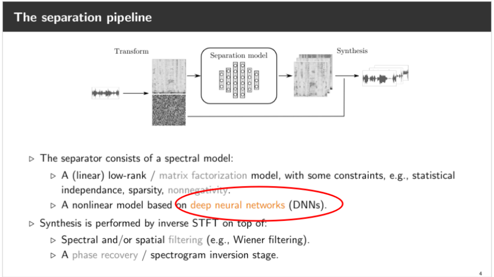
- ▷ Spectrograms are perhaps *not low-rank*.
- ▷ Interactions between spectral templates and activations are perhaps *not linear*.



What happened?

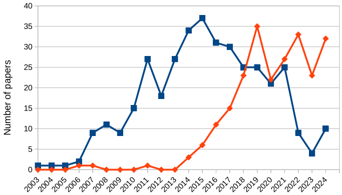
Some limitations of NMF:

- ▷ Spectrograms are perhaps *not low-rank*.
- ▷ Interactions between spectral templates and activations are perhaps *not linear*.



Enter the deep learning era.

- ▷ Abundance of large-scale datasets.
- ▷ Computing capabilities (GPUs) have exploded.
- ▷ Efficient training algorithms (backpropagation).
- ▷ User-friendly frameworks (Pytorch, Keras)



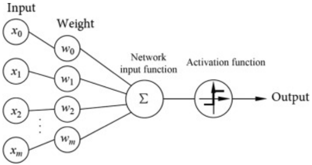
Deep neural networks (DNNs)

Model: a mapping function g with parameters θ: $\mathbf{y} \approx g_\theta(\mathbf{x})$

- Inputs $\mathbf{x}$ / outputs $\mathbf{y}$ are high-dimensional audio data, e.g., spectrograms for source separation.
- g_θ is built by assembling elementary *neurons*, e.g.:

$$\mathbf{x}^{(l+1)} = \sigma(\mathbf{W}^{(l)}\mathbf{x}^{(l)} + \mathbf{b}^{(l)})$$

- For source separation $|\theta| \sim 10^7 - 10^8$.
- Many possible neural architectures: MLP, CNN, RNN, etc.



Source: ScienceDirect

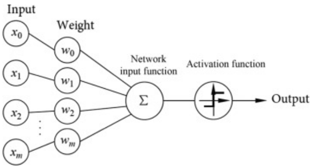
Deep neural networks (DNNs)

Model: a mapping function g with parameters θ: $\mathbf{y} \approx g_\theta(\mathbf{x})$

- Inputs $\mathbf{x}$ / outputs $\mathbf{y}$ are high-dimensional audio data, e.g., spectrograms for source separation.
- g_θ is built by assembling elementary *neurons*, e.g.:

$$\mathbf{x}^{(l+1)} = \sigma(\mathbf{W}^{(l)}\mathbf{x}^{(l)} + \mathbf{b}^{(l)})$$

- For source separation $|\theta| \sim 10^7 - 10^8$.
- Many possible neural architectures: MLP, CNN, RNN, etc.



Source: ScienceDirect

Supervised learning

- A training dataset = a collection of input/output pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{I}$.
- The parameters of the network are learned via: $\min_\theta \sum_{i=1}^{I} \mathcal{L}(\mathbf{y}_i, g_\theta(\mathbf{x}_i))$.
- Solved with a stochastic gradient descent algorithm (e.g., ADAM).

A paradigm shift

From expert knowledge research

- How do I refine this model to overcome its limitation (e.g., convolutive NMF)?
- Which regularization would fit this intrument (sparsity, (in)harmonicity)?
- How do I model it mathematically (trade-off between complexity and generalizability)?
- Which loss would be more perceptually-relevant?
- How do I (efficiently) solve the new optimization problem?

A paradigm shift

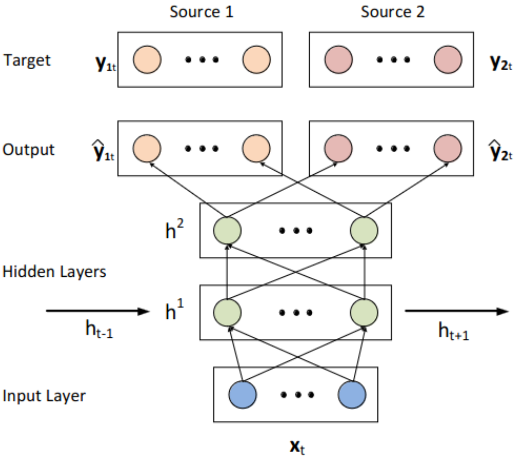
From expert knowledge research

- How do I refine this model to overcome its limitation (e.g., convolutive NMF)?
- Which regularization would fit this intrument (sparsity, (in)harmonicity)?
- How do I model it mathematically (trade-off between complexity and generalizability)?
- Which loss would be more perceptually-relevant?
- How do I (efficiently) solve the new optimization problem?

to data-driven model engineering.

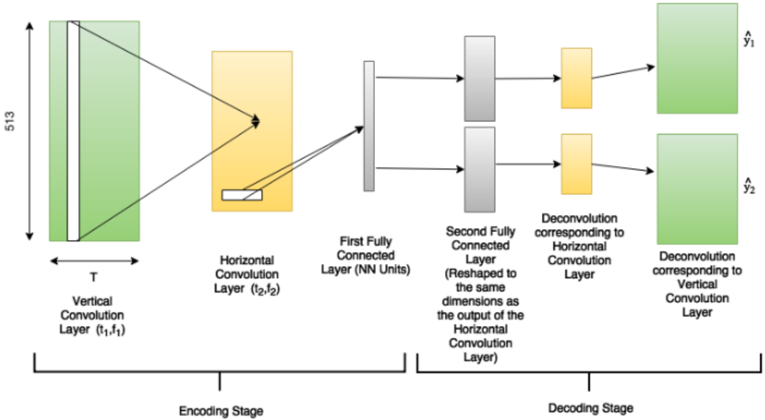
- Which architecture would be more powerful?
- Should I re-test every hyperparameter value upon a minor additional change?
- How can I parallelize / reduce training time / optimally use my hardware?
- How can I use more data / better exploit my available data / cope with data scarcity?

A few architectures



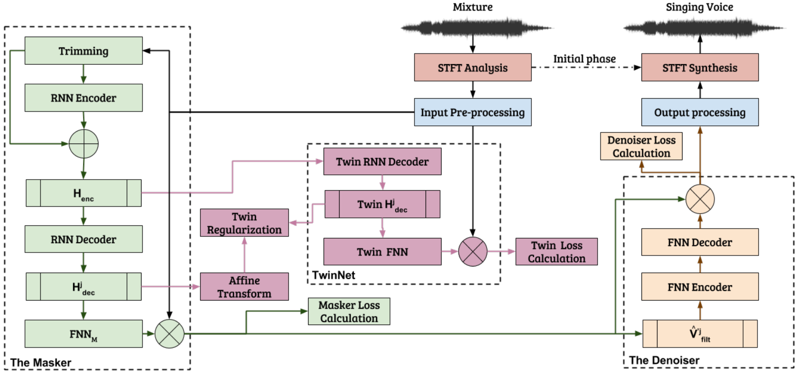
Source: (Huang et al. 2014)

A few architectures



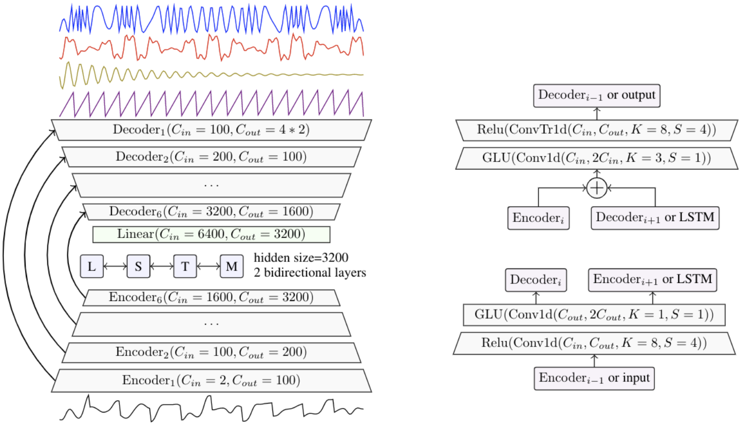
Source: (Chandna et al. 2017)

A few architectures



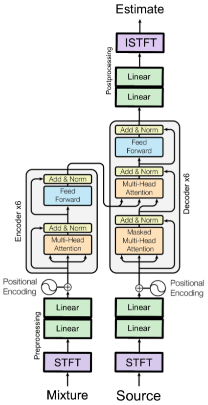
Source: (Drossos et al. 2018)

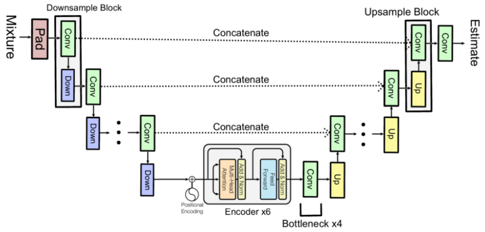
A few architectures



Source: (Défossez 2021)

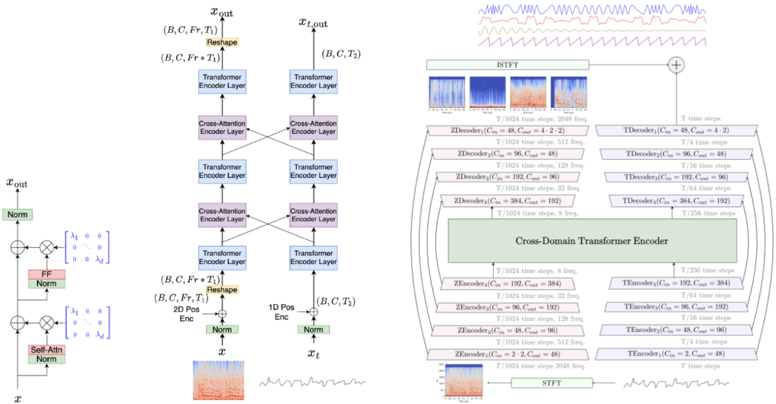
A few architectures





Source: (Yang et al. 2023)

A few architectures



(a) Transformer Encoder Layer

(b) Cross-domain Transformer Encoder of depth 5

(c) Hybrid Transformer Demucs

Source: (Rouard et al. 2023)

Results

Impressive performance

	Vocals	Bass	Drums	Guitar
UMX (2018)				
BSRNN (2023)				

Results

Impressive performance

	Vocals	Bass	Drums	Guitar
UMX (2018)				
BSRNN (2023)				

A few drawbacks . . .

- ▷ Black boxes / lack of interpretability.
- ▷ Difficult to adapt to new tasks.
- ▷ Energy / environmental costs.
- ▷ Exacerbates the reproductibility crisis.

We trained three separation models respectively for vocals, bass, and drums using In-House and the Musdb18HQ training set. For the "other" stem, we subtracted the vocals, bass, and drums signals from the input mixture in the time domain. For each model, the training process lasted for 4 weeks using 16 Nvidia A100-80GB GPUs with a total batch size of 128 (i.e., 8 for each GPU). The model checkpoint with the best validation result was selected.

Source: (Lu et al. 2024)

The Costs of Reproducibility in Music Separation Research: a Replication of Band-Split RNN

Paul Magron, Romain Serizel, Constance Douwes

lator [36], which approximates energy consumption base on hardware specifications (we consider a 3 W power pe 8 GB of memory).[3] This amounts to 19,030 kWh, whic is more than 44 times the energy consumption of trainin the best model, or 150 times that of the base model.

In all fairness, part of this cost is due to our own im plementation errors, which resulted in, e.g., interrupted o redundant training runs. However, we believe that mos

Source: Coming soon...

What to do then?

The rise and fall

- A paradigm shift from expert knowledge research to data-driven model engineering.
- Clear pros and cons for both approaches.

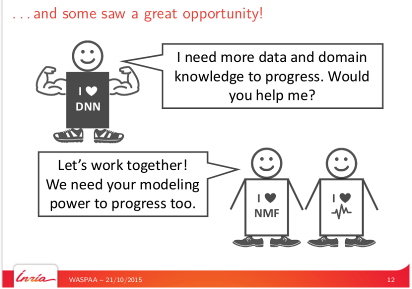
What to do then?

The rise and fall

- A paradigm shift from expert knowledge research to data-driven model engineering.
- Clear pros and cons for both approaches.

The obvious solution: combine them.

- Not exactly breaking news.
- But still relevant!
- (Beyond source separation,) many recent works combine DNNs and factorization / low-rank models.



Source: Vincent, "Is audio signal processing still useful in the era of machine learning?", 2015.

A resurgence?

Low-rank weights

Main idea: enforce the weights of a neural network to be low-rank.

- Either at training, inference, or for fine-tuning.
- Allows to achieve significant size reduction.

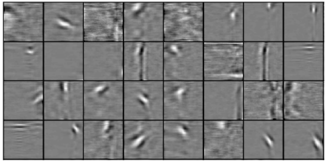
Low-rank weights

Main idea: enforce the weights of a neural network to be low-rank.

- ▷ Either at training, inference, or for fine-tuning.
- ▷ Allows to achieve significant size reduction.

Early examples (in audio / speech!), replacing one or more layers' weights with two low-rank independent factors. Yields a 30-75 % size reduction.

- ▷ The first layer's weights correspond to low-level filters that have a simple structure (Nakkiran et al. 2015).
- ▷ Output dimension is very large (for speech recognition), so the last layer tends to be overparametrized (Sainath et al. 2013).
- ▷ Or just apply SVD everywhere (Xue et al. 2013).



Source: (Nakkiran et al. 2015)

Low-rank weights

Large networks compression has motivated more recent approaches:

- Exploiting orthogonality constraints (Povey et al. 2018).
- Sparse SVD (Swaminathan et al. 2020) and other SVD variants (Cai et al. 2023).
- Adapting the rank to each layer (Idelbayev et al. 2020).

Low-rank weights

Large networks compression has motivated more recent approaches:

- Exploiting orthogonality constraints (Povey et al. 2018).
- Sparse SVD (Swaminathan et al. 2020) and other SVD variants (Cai et al. 2023).
- Adapting the rank to each layer (Idelbayev et al. 2020).

Low-rank regularization implicitly, e.g. via a nuclear norm penalty (Scarvelis et al. 2024).

Low-rank weights

Large networks compression has motivated more recent approaches:

- ▷ Exploiting orthogonality constraints (Povey et al. 2018).
- ▷ Sparse SVD (Swaminathan et al. 2020) and other SVD variants (Cai et al. 2023).
- ▷ Adapting the rank to each layer (Idelbayev et al. 2020).

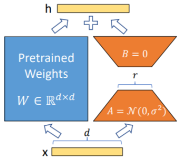
Low-rank regularization implicitly, e.g. via a nuclear norm penalty (Scarvelis et al. 2024).

LLM fine-tuning in particular poses computational challenges. The LoRA method (Hu et al. 2021):

- ▷ Does not approximate all weights, but only an additional term that corresponds to fine-tuning.

$$\mathbf{W}_{\text{finetuned}} = \mathbf{W}_{\text{pretrained}} + \mathbf{\Delta} \quad \text{with} \quad \mathbf{\Delta} = \mathbf{AB}$$

- ▷ Reduces the amount of trainable parameters (in the fine-tuning stage) by a large factor (10000).

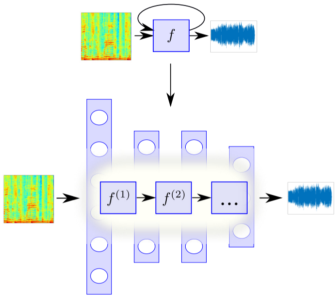


Source: (Hu et al. 2021)

Deep NMF

Deep unfolding (or unrolling)

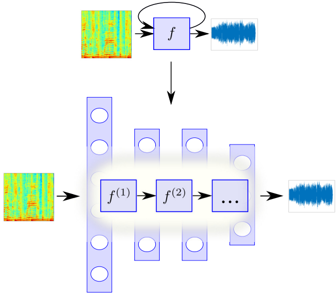
- Each algorithm's iteration = one layer of a neural network.
- Train via backpropagation through the unfolded algorithm.
- Lighter and more interpretable networks.



Deep NMF

Deep unfolding (or unrolling)

- ▷ Each algorithm's iteration = one layer of a neural network.
- ▷ Train via backpropagation through the unfolded algorithm.
- ▷ Lighter and more interpretable networks.



Deep NMF (Le Roux et al. 2015a)

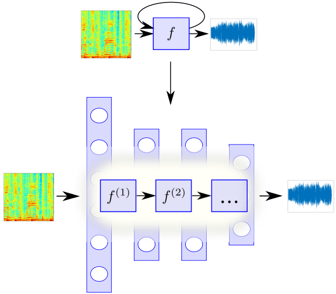
$$\mathbf{H}^{(l+1)} = \mathbf{H}^{(l)} \cdot \frac{(\mathbf{W}^{(l)})^T(\mathbf{V} \cdot [\mathbf{W}^{(l)}\mathbf{H}^{(l)}]^{\beta-2})}{(\mathbf{W}^{(l)})^T[\mathbf{W}^{(l)}\mathbf{H}^{(l)}]^{\beta-1}}$$

- ▷ $\mathbf{H}^{(l)}$ is the output of the l-th layer.
- ▷ $\mathbf{W}^{(l)}$ are the learnable weights of the l-th layer.

Deep NMF

Deep unfolding (or unrolling)

- Each algorithm's iteration = one layer of a neural network.
- Train via backpropagation through the unfolded algorithm.
- Lighter and more interpretable networks.



Deep NMF (Le Roux et al. 2015a)

$$\mathbf{H}^{(l+1)} = \mathbf{H}^{(l)} \cdot \frac{(\mathbf{W}^{(l)})^T (\mathbf{V} \cdot [\mathbf{W}^{(l)}\mathbf{H}^{(l)}]^{\beta-2})}{(\mathbf{W}^{(l)})^T [\mathbf{W}^{(l)}\mathbf{H}^{(l)}]^{\beta-1}}$$

- $\mathbf{H}^{(l)}$ is the output of the l-th layer.
- $\mathbf{W}^{(l)}$ are the learnable weights of the l-th layer.

A very active research topic!

- Unfolding other update schemes: ISTA (Wisdom et al. 2017), ALS (Xiong et al. 2022).
- Unfold both factor updates and add other learnable parameters (Kervazo et al. 2024).
- Adapt the loss / formulate alternative optimization problems (Leplat et al. 2024).

Factorized latent space

Main idea: learn a transformation of the data such that the low-rank asumption better holds.

Factorized latent space

Main idea: learn a transformation of the data such that the low-rank asumption better holds.

- Early approach, purely optimization-based (Fagot et al. 2018).

$$\min_{\mathbf{\Phi}\text{ is orthogonal},\mathbf{W}\geq 0,\mathbf{H}\geq 0} D(|\mathbf{\Phi}(\mathbf{x})|^2, \mathbf{WH})$$

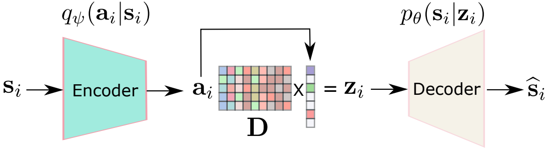
Factorized latent space

Main idea: learn a transformation of the data such that the low-rank asumption better holds.

- ▷ Early approach, purely optimization-based (Fagot et al. 2018).

$$\min_{\boldsymbol{\Phi} \text{ is orthogonal}, \mathbf{W}\geq 0, \mathbf{H}\geq 0} D(|\boldsymbol{\Phi}(\mathbf{x})|^2, \mathbf{WH})$$

- ▷ More recently: a variational auto-encoder and a (fixed) latent dictionary (Sadeghi et al. 2022).



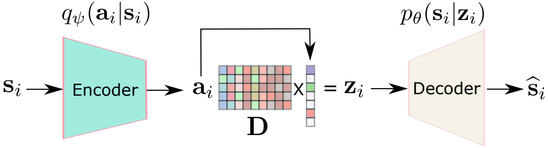
Factorized latent space

Main idea: learn a transformation of the data such that the low-rank asumption better holds.

- ▷ Early approach, purely optimization-based (Fagot et al. 2018).

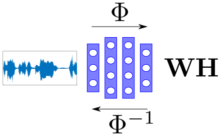
$$\min_{\mathbf{\Phi} \text{ is orthogonal}, \mathbf{W}\geq 0, \mathbf{H}\geq 0} D(|\mathbf{\Phi}(\mathbf{x})|^2, \mathbf{WH})$$

- ▷ More recently: a variational auto-encoder and a (fixed) latent dictionary (Sadeghi et al. 2022).



Perspective: learning both factors and the DNN jointly.

$$\min_{\theta, \mathbf{W}\geq 0, \mathbf{H}\geq 0} D(\mathbf{\Phi}_\theta(\mathbf{x}), \mathbf{WH})$$



- ▷ A connexion with disentangled latent spaces (Luo et al. 2024).

NMF + deep models

Additional approaches combine (nonnegative) matrix factorization and DNNs for flexible modeling.

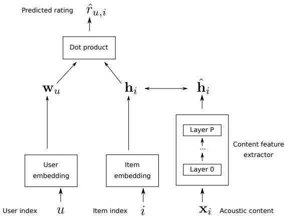
NMF + deep models

Additional approaches combine (nonnegative) matrix factorization and DNNs for flexible modeling.

Deep prior: Regularizing a factorization model with a DNN.

- For image denoising (Lin et al. 2020), restoration (Chen et al. 2022).
- Recommender systems (Magron et al. 2022), where deep acoustic features regularize an item embedding:

$$\min ||\mathbf{R} - \mathbf{W}\mathbf{H}||^2 + \lambda \sum_i ||\mathbf{h}_i - \text{DNN}(\mathbf{x}_i)||^2$$



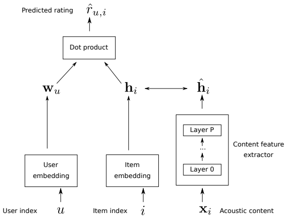
NMF + deep models

Additional approaches combine (nonnegative) matrix factorization and DNNs for flexible modeling.

Deep prior: Regularizing a factorization model with a DNN.

- For image denoising (Lin et al. 2020), restoration (Chen et al. 2022).
- Recommender systems (Magron et al. 2022), where deep acoustic features regularize an item embedding:

$$\min ||\mathbf{R} - \mathbf{WH}||^2 + \lambda \sum_i ||\mathbf{h}_i - \text{DNN}(\mathbf{x}_i)||^2$$



Hybrid models optimally leverage DNNs and NMF, e.g. for speech enhancement (Leglaive et al. 2018).

$$\mathbf{X} = \mathbf{S} + \mathbf{N} \quad \text{with} \quad \underbrace{\mathbf{S} = \text{DNN}}_{\text{speech}} \quad \text{and} \quad \underbrace{\mathbf{N} = \mathbf{WH}}_{\text{noise}}$$

Conclusion

- ▷ NMF has been particularly successful for source separation until the mid 2010s.
- ▷ Then, it declined as deep learning has shown powerful for solving signal processing problems.
- ▷ But this comes with some drawbacks: black boxes, energy costs, reproductibility crisis, etc.

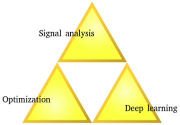
Conclusion

- ▷ NMF has been particularly successful for source separation until the mid 2010s.
- ▷ Then, it declined as deep learning has shown powerful for solving signal processing problems.
- ▷ But this comes with some drawbacks: black boxes, energy costs, reproductibility crisis, etc.

Key message

Combining expert knowledge and data-driven models:
a promising approach for machine leaning / source separation research.

- ▷ Enable networks to exploit prior information.
- ▷ Improve their robustness and reduce their size.
- ▷ More interpretable and principled networks.



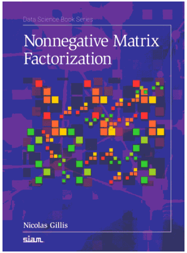
Useful ressources

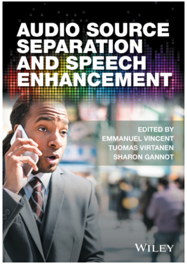
home
CV
publications
people
demos
talks

Cédric Févotte

Selected talks

- Non-negative matrix factorizations with the beta-divergence, Tutorial at Peyresq signal & image summer school, 2024.
- Recent advances in nonnegative matrix factorization, Tutorial at ICASSP, Singapore, 2022.
- Robust nonnegative matrix factorisation with the beta-divergence and applications in imaging, Workshop Imaging & Machine Learning, Institut Henri Poincaré, Paris, 2019.
- Temporal models with low-rank spectrogram, Keynote at IEEE MLSP, Aalborg, 2018.
- Nonnegative matrix factorisation & friends for audio signal separation, Tutorial at SPARS summer school, Lisbon, 2017.





Bertin, N. et al. (2010). **"Enforcing Harmonicity and Smoothness in Bayesian Non-Negative Matrix Factorization Applied to Polyphonic Music Transcription".** In: *IEEE Transactions on Audio, Speech, and Language Processing*.

Bronson, J. and P. Depalle (2014). **"Phase constrained complex NMF: Separating overlapping partials in mixtures of harmonic musical sources".** In: *Proc. ICASSP*.

Cai, G. et al. (2023). **"Learning and Compressing: Low-Rank Matrix Factorization for Deep Neural Network Compression".** In: *Applied Sciences*.

Chandna, P. et al. (2017). **"Monoaural Audio Source Separation Using Deep Convolutional Neural Networks".** In: *Latent Variable Analysis and Signal Separation*. Ed. by P. Tichavský et al.

Chen, L. et al. (2022). **"Reweighted Low-Rank Factorization With Deep Prior for Image Restoration".** In: *IEEE Transactions on Signal Processing*.

Choi, S. (2008). **"Algorithms for orthogonal nonnegative matrix factorization".** In: *Proc. IJCNN*.

Défossez, A. (2021). **"Hybrid Spectrogram and Waveform Source Separation".** In: *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*.

Drossos, K. et al. (2018). **"MaD TwinNet: Masker-Denoiser Architecture with Twin Networks for Monaural Sound Source Separation".** In: *Proc. IJCNN*.

Fagot, D. et al. (2018). **"Nonnegative Matrix Factorization with Transform Learning".** In: *Proc. ICASSP*.

References ii

Févotte, C. and J. Idier (2011). **"Algorithms for Nonnegative Matrix Factorization with the beta-Divergence".** In: *Neural Computation* 23.9.

Févotte, C. et al. (2009). **"Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis".** In: *Neural computation*.

Gillis, N. (2020). **Nonnegative Matrix Factorization.** Society for Industrial and Applied Mathematics.

Hu, J. E. et al. (2021). **"LoRA: Low-Rank Adaptation of Large Language Models".** In: *Proc. ICLR*.

Huang, P.-S. et al. (2014). **"Deep learning for monaural speech separation".** In: *Proc. ICASSP*.

Idelbayev, Y. and M. A. Carreira-Perpiñán (2020). **"Low-Rank Compression of Neural Nets: Learning the Rank of Each Layer".** In: *Proc. CVPR*.

Kameoka, H. et al. (2009). **"Complex NMF: A new sparse representation for acoustic signals".** In: *Proc. ICASSP*.

Kervazo, C. et al. (2024). **"Deep Unrolling of the Multiplicative Updates Algorithm for Blind Source Separation, with Application to Hyperspectral Unmixing".** In: *Proc. EUSIPCO*.

Le Roux, J. et al. (2009). **"Complex NMF under spectrogram consistency constraints".** In: *Proc. of Acoustical Society of Japan Autumn Meeting*.

Le Roux, J. et al. (2015a). **"Deep NMF for speech separation".** In: *Proc. ICASSP*.

Le Roux, J. et al. (2015b). **"Sparse NMF – half-baked or well done?"** In: *Technical report, Mitsubishi Electric Research Labs (MERL)*.

Leglaive, S. et al. (2018). **“A variance modeling framework based on variational autoencoders for speech enhancement”.** In: *Proc. MLSP*.

Leplat, V. et al. (2024). **“Deep Nonnegative Matrix Factorization With Beta Divergences”.** In: *Neural Computation*.

Lin, B. et al. (2020). **“Hyperspectral Image Denoising via Matrix Factorization and Deep Prior Regularization”.** In: *IEEE Transactions on Image Processing*.

Lu, W.-T. et al. (2024). **“Music Source Separation With Band-Split Rope Transformer”.** In: *Proc. ICASSP*.

Luo, Y.-J. et al. (2024). **“Disentangling Multi-instrument Music Audio for Source-level Pitch and Timbre Manipulation”.** In: *Proc. NeurIPS 2024*.

Magron, P. et al. (2016). **“Complex NMF under phase constraints based on signal modeling: application to audio source separation”.** In: *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Magron, P. and C. Févotte (2022). **“Neural content-aware collaborative filtering for cold-start music recommendation”.** In: *Data Min. Knowl. Discov.*

Magron, P. and T. Virtanen (2018). **“Towards complex nonnegative matrix factorization with the beta-divergence”.** In: *Proc. iWAENC*.

— (2019). **“Complex ISNMF: A Phase-Aware Model for Monaural Audio Source Separation”.** In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Nakkiran, P. et al. (2015). **“Compressing deep neural networks using a rank-constrained topology”.** In: *Proc. Interspeech*.

O’Grady, P. D. and B. A. Pearlmutter (2006). **“Convolutive Non-Negative Matrix Factorisation with a Sparseness Constraint”.** In: *Proc. MLSP*.

Povey, D. et al. (2018). **“Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks”.** In: *Proc. Interspeech*.

Rouard, S. et al. (2023). **“Hybrid Transformers for Music Source Separation”.** In: *Proc. ICASSP*.

Sadeghi, M. and P. Magron (2022). **“A Sparsity-promoting Dictionary Model for Variational Autoencoders”.** In: *Proc. Interspeech*.

Sainath, T. N. et al. (2013). **“Low-rank matrix factorization for Deep Neural Network training with high-dimensional output targets”.** In: *Proc. ICASSP*.

Scarvelis, C. and J. Solomon (2024). **“Nuclear Norm Regularization for Deep Learning”.** In: *Proc. NeurIPS*.

Simsekli, U. et al. (2013). **“Learning the beta-Divergence in Tweedie Compound Poisson Matrix Factorization Models”.** In: *Proc. ICML*.

Swaminathan, S. et al. (2020). **“Sparse low rank factorization for deep neural network compression”.** In: *Neurocomputing*.

Virtanen, T. (2007). **“Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria”.** In: *IEEE Transactions on Audio, Speech, and Language Processing*.

References v

Wisdom, S. et al. (2017). **“Deep recurrent NMF for speech separation by unfolding iterative thresholding”.** In: *Proc. WASPAA*.

Xiong, F. et al. (2022). **“SNMF-Net: Learning a Deep Alternating Neural Network for Hyperspectral Unmixing”.** In: *IEEE Transactions on Geoscience and Remote Sensing*.

Xue, J. et al. (2013). **“Restructuring of deep neural network acoustic models with singular value decomposition”.** In: *Proc. Interspeech*.

Yang, M. et al. (2023). **“A Transformer-Based Approach to Music Separation”.** In: *Technical report, Univesity of Washington*.