

Tutorial 5, ISMIR
Milan, November 5, 2023

Learning with Music Signals: Technology Meets Education

Audio Decomposition

Meinard Müller

International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de

Source Separation

- Decomposition of audio stream into different sound sources
- Central task in digital signal processing
- "Cocktail party effect"

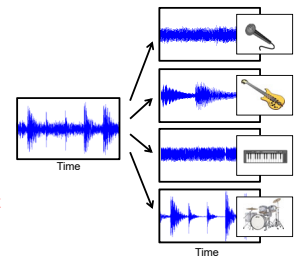


Source Separation

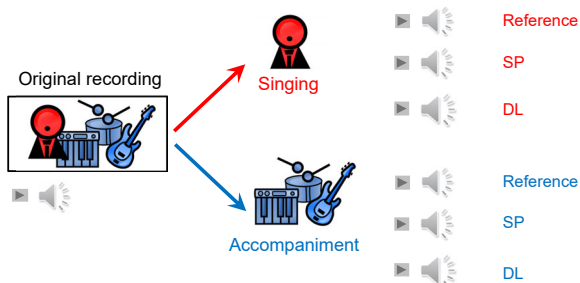
- Decomposition of audio stream into different sound sources
- Central task in digital signal processing
- "Cocktail party effect"
- Several input signals
- Sources are assumed to be statistically independent

Source Separation (Music)

- Main melody, accompaniment, drum track
- Instrumental voices
- Individual note events
- Only mono or stereo
- Sources are often highly dependent



Source Separation (Singing Voice)



DL-Based Source Separation
Stöter, Uhlisch Luitkus, Mitsufuji: Open-
Unmix – A Reference Implementation for
Music Source Separation. JOSS, 2019.

- Reference: Best possible result
- SP: Traditional signal processing
- DL: Deep Learning

Score-Informed Source Separation

Exploit musical score to support
decomposition process

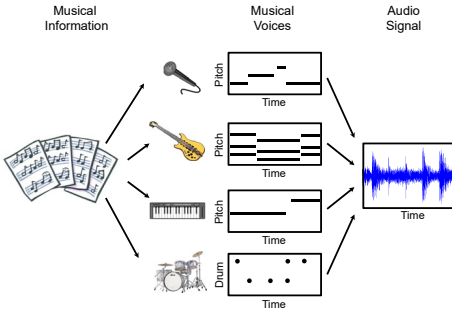
Prior Knowledge
Ewert, Pardo, Müller, Plumbley:
Score-Informed Source Separation
for Musical Audio Recordings.
IEEE SPM 31(3), 2014.



Score-Informed Source Separation

Exploit musical score to support decomposition process

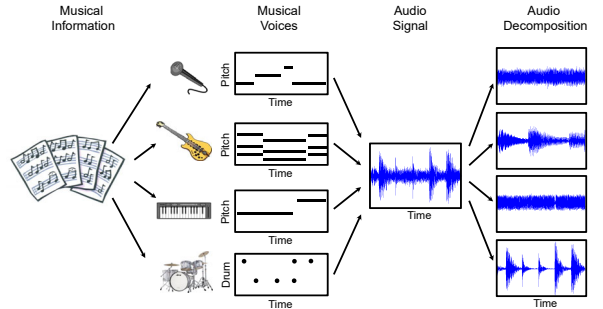
Prior Knowledge
Ewert, Pardo, Müller, Plumbley:
Score-Informed Source Separation
for Musical Audio Recordings.
IEEE SPM 31(3), 2014.



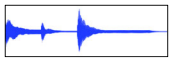
Score-Informed Source Separation

Exploit musical score to support decomposition process

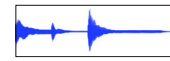
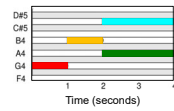
Prior Knowledge
Ewert, Pardo, Müller, Plumbley:
Score-Informed Source Separation
for Musical Audio Recordings.
IEEE SPM 31(3), 2014.



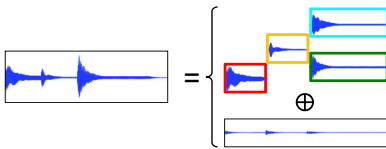
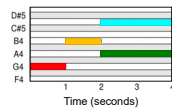
Score-Informed Audio Decomposition



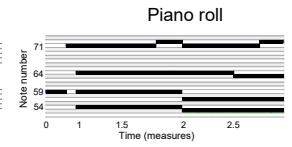
Score-Informed Audio Decomposition



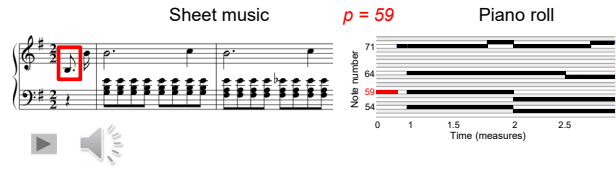
Score-Informed Audio Decomposition



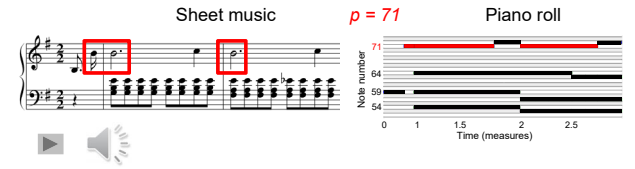
Score-Informed Audio Decomposition



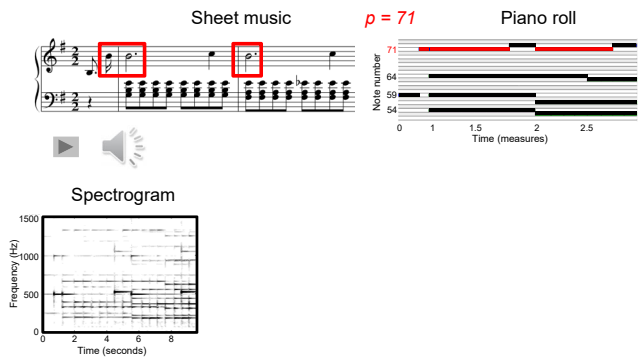
Score-Informed Audio Decomposition



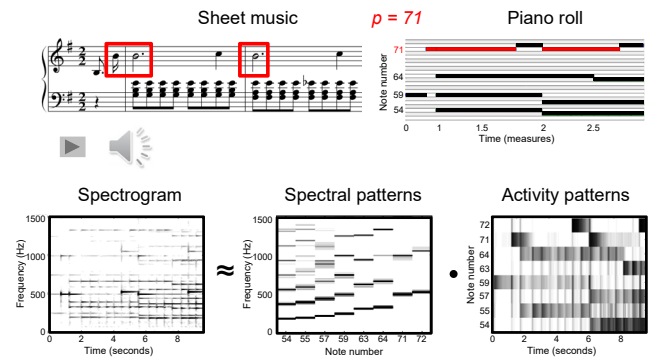
Score-Informed Audio Decomposition



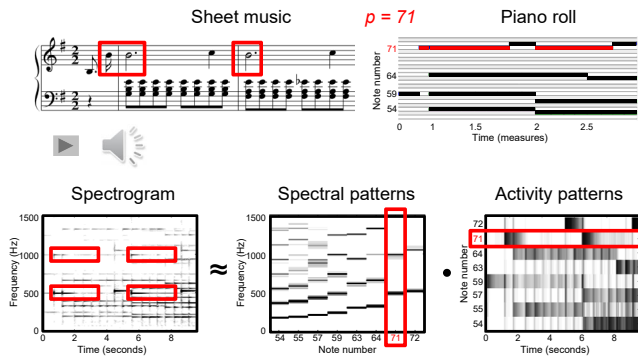
Score-Informed Audio Decomposition



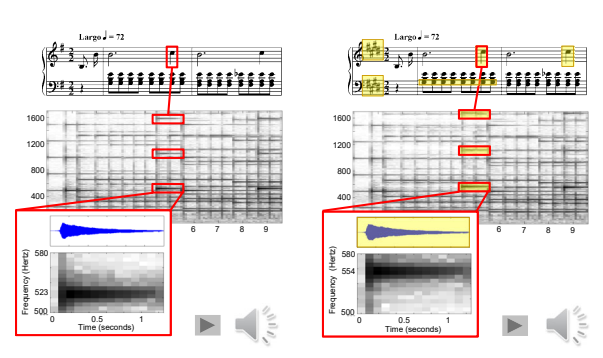
Score-Informed Audio Decomposition



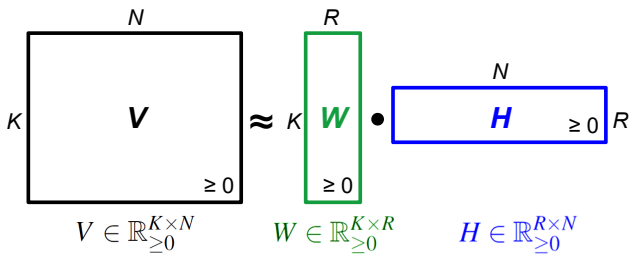
Score-Informed Audio Decomposition



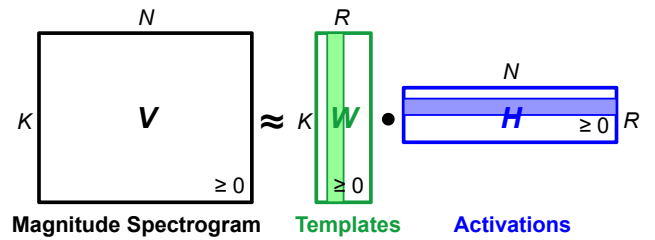
Score-Informed Audio Decomposition



Nonnegative Matrix Factorization (NMF)



Nonnegative Matrix Factorization (NMF)



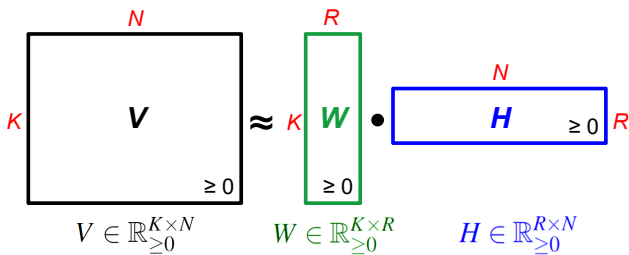
Templates: Pitch + Timbre

“How does it sound”

Activations: Onset time + Duration

“When does it sound”

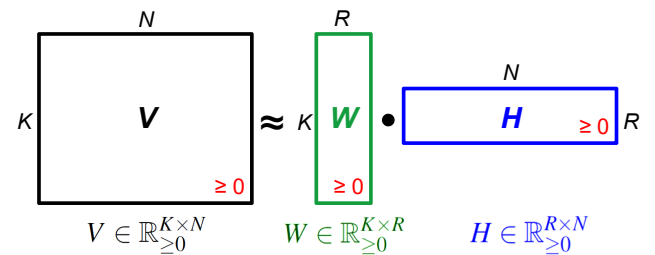
Nonnegative Matrix Factorization (NMF)



Dimensionality reduction

- K, N typically much larger than R (maximal rank)
- Example: $N = 1000, K = 500, R = 20$
 $K \times N = 500,000, K \times R = 10,000, R \times N = 20,000$

Nonnegative Matrix Factorization (NMF)



Nonnegativity:

- Prevents mutual cancellation of template vectors
- Encourages semantically meaningful decomposition

NMF Optimization

Optimization problem:

Given $V \in \mathbb{R}_{\geq 0}^{K \times N}$ and rank parameter R minimize

$$\|V - WH\|^2$$

with respect to $W \in \mathbb{R}_{\geq 0}^{K \times R}$ and $H \in \mathbb{R}_{\geq 0}^{R \times N}$.

Optimization not easy:

- Nonnegativity constraints
- Nonconvexity when jointly optimizing W and H

Strategy: Iteratively optimize W and H via gradient descent

NMF Optimization

Computation of gradient with respect to H (fixed W)

$$D := RN$$

$$\varphi^W : \mathbb{R}^D \rightarrow \mathbb{R}$$

$$\varphi^W(H) := \|V - WH\|^2$$

Variables

$$H \in \mathbb{R}^{R \times N}$$

$$H_{\rho v}$$

$$\rho \in [1 : R]$$

$$v \in [1 : N]$$

NMF Optimization

Computation of gradient with respect to H (fixed W)

$$D := RN$$

$$\varphi^W : \mathbb{R}^D \rightarrow \mathbb{R}$$

$$\varphi^W(H) := \|V - WH\|^2$$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^K \sum_{n=1}^N (V_{kn} - \sum_{r=1}^R W_{kr} H_{rn})^2 \right)}{\partial H_{\rho\nu}}$$

Variables

$$H \in \mathbb{R}^{R \times N}$$

$$H_{\rho\nu}$$

$$\rho \in [1 : R]$$

$$\nu \in [1 : N]$$

NMF Optimization

Computation of gradient with respect to H (fixed W)

$$D := RN$$

$$\varphi^W : \mathbb{R}^D \rightarrow \mathbb{R}$$

$$\varphi^W(H) := \|V - WH\|^2$$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^K \sum_{n=1}^N (V_{kn} - \sum_{r=1}^R W_{kr} H_{rn})^2 \right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^K (V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu})^2 \right)}{\partial H_{\rho\nu}}$$

Variables

$$H \in \mathbb{R}^{R \times N}$$

$$H_{\rho\nu}$$

$$\rho \in [1 : R]$$

$$\nu \in [1 : N]$$

Summand that does not depend on $H_{\rho\nu}$ must be zero

NMF Optimization

Computation of gradient with respect to H (fixed W)

$$D := RN$$

$$\varphi^W : \mathbb{R}^D \rightarrow \mathbb{R}$$

$$\varphi^W(H) := \|V - WH\|^2$$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^K \sum_{n=1}^N (V_{kn} - \sum_{r=1}^R W_{kr} H_{rn})^2 \right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^K (V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu})^2 \right)}{\partial H_{\rho\nu}}$$

Variables

$$H \in \mathbb{R}^{R \times N}$$

$$H_{\rho\nu}$$

$$\rho \in [1 : R]$$

$$\nu \in [1 : N]$$

Apply chain rule from calculus

NMF Optimization

Computation of gradient with respect to H (fixed W)

$$D := RN$$

$$\varphi^W : \mathbb{R}^D \rightarrow \mathbb{R}$$

$$\varphi^W(H) := \|V - WH\|^2$$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^K \sum_{n=1}^N (V_{kn} - \sum_{r=1}^R W_{kr} H_{rn})^2 \right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^K (V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu})^2 \right)}{\partial H_{\rho\nu}}$$

$$= \sum_{k=1}^K 2 \left(V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu} \right) \cdot (-W_{k\rho})$$

$$= 2 \left(\sum_{r=1}^R \sum_{k=1}^K W_{k\rho} W_{kr} H_{r\nu} - \sum_{k=1}^K W_{k\rho} V_{k\nu} \right)$$

Variables

$$H \in \mathbb{R}^{R \times N}$$

$$H_{\rho\nu}$$

$$\rho \in [1 : R]$$

$$\nu \in [1 : N]$$

Rearrange summands

NMF Optimization

Computation of gradient with respect to H (fixed W)

$$D := RN$$

$$\varphi^W : \mathbb{R}^D \rightarrow \mathbb{R}$$

$$\varphi^W(H) := \|V - WH\|^2$$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^K \sum_{n=1}^N (V_{kn} - \sum_{r=1}^R W_{kr} H_{rn})^2 \right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^K (V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu})^2 \right)}{\partial H_{\rho\nu}}$$

$$= \sum_{k=1}^K 2 \left(V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu} \right) \cdot (-W_{k\rho})$$

$$= 2 \left(\sum_{r=1}^R \sum_{k=1}^K W_{k\rho} W_{kr} H_{r\nu} - \sum_{k=1}^K W_{k\rho} V_{k\nu} \right)$$

$$= 2 \left(\sum_{r=1}^R \left(\sum_{k=1}^K W_{\rho k}^\top W_{kr} \right) H_{r\nu} - \sum_{k=1}^K W_{\rho k}^\top V_{k\nu} \right)$$

Variables

$$H \in \mathbb{R}^{R \times N}$$

$$H_{\rho\nu}$$

$$\rho \in [1 : R]$$

$$\nu \in [1 : N]$$

Introduce transposed W^\top

NMF Optimization

Computation of gradient with respect to H (fixed W)

$$D := RN$$

$$\varphi^W : \mathbb{R}^D \rightarrow \mathbb{R}$$

$$\varphi^W(H) := \|V - WH\|^2$$

$$\frac{\partial \varphi^W}{\partial H_{\rho\nu}} = \frac{\partial \left(\sum_{k=1}^K \sum_{n=1}^N (V_{kn} - \sum_{r=1}^R W_{kr} H_{rn})^2 \right)}{\partial H_{\rho\nu}}$$

$$= \frac{\partial \left(\sum_{k=1}^K (V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu})^2 \right)}{\partial H_{\rho\nu}}$$

$$= \sum_{k=1}^K 2 \left(V_{k\nu} - \sum_{r=1}^R W_{kr} H_{r\nu} \right) \cdot (-W_{k\rho})$$

$$= 2 \left(\sum_{r=1}^R \sum_{k=1}^K W_{k\rho} W_{kr} H_{r\nu} - \sum_{k=1}^K W_{k\rho} V_{k\nu} \right)$$

$$= 2 \left(\sum_{r=1}^R \left(\sum_{k=1}^K W_{\rho k}^\top W_{kr} \right) H_{r\nu} - \sum_{k=1}^K W_{\rho k}^\top V_{k\nu} \right)$$

$$= 2 \left((W^\top WH)_{\rho\nu} - (W^\top V)_{\rho\nu} \right)$$

Variables

$$H \in \mathbb{R}^{R \times N}$$

$$H_{\rho\nu}$$

$$\rho \in [1 : R]$$

$$\nu \in [1 : N]$$

NMF Optimization

Gradient descent

Initialization $H^{(0)} \in \mathbb{R}^{R \times N}$

Iteration for $\ell = 0, 1, 2, \dots$

$$H_{rn}^{(\ell+1)} = H_{rn}^{(\ell)} - \gamma_{rn}^{(\ell)} \cdot \left((W^T W H^{(\ell)})_{rn} - (W^T V)_{rn} \right)$$

with suitable learning rate $\gamma_{rn}^{(\ell)} \geq 0$

NMF Optimization

Gradient descent

Initialization $H^{(0)} \in \mathbb{R}^{R \times N}$

Iteration for $\ell = 0, 1, 2, \dots$

$$H_{rn}^{(\ell+1)} = H_{rn}^{(\ell)} - \gamma_{rn}^{(\ell)} \cdot \left((W^T W H^{(\ell)})_{rn} - (W^T V)_{rn} \right)$$

with suitable learning rate $\gamma_{rn}^{(\ell)} \geq 0$

Issues:

- How to do the initialization?
- How to choose the learning rate?
- How to ensure nonnegativity?

NMF Optimization

Gradient descent

Initialization $H^{(0)} \in \mathbb{R}^{R \times N}$

Iteration for $\ell = 0, 1, 2, \dots$

$$H_{rn}^{(\ell+1)} = H_{rn}^{(\ell)} - \gamma_{rn}^{(\ell)} \cdot \left((W^T W H^{(\ell)})_{rn} - (W^T V)_{rn} \right)$$

$$= H_{rn}^{(\ell)} \cdot \frac{(W^T V)_{rn}}{(W^T W H^{(\ell)})_{rn}}$$

Issues:

- How to do the initialization?
- How to choose the learning rate?
- How to ensure nonnegativity?

Choose adaptive learning rate:

$$\gamma_{rn}^{(\ell)} := \frac{H_{rn}^{(\ell)}}{(W^T W H^{(\ell)})_{rn}}$$

NMF Optimization

Gradient descent

Initialization $H^{(0)} \in \mathbb{R}^{R \times N}$

Iteration for $\ell = 0, 1, 2, \dots$

$$H_{rn}^{(\ell+1)} = H_{rn}^{(\ell)} - \gamma_{rn}^{(\ell)} \cdot \left((W^T W H^{(\ell)})_{rn} - (W^T V)_{rn} \right)$$

$$= H_{rn}^{(\ell)} \cdot \frac{(W^T V)_{rn}}{(W^T W H^{(\ell)})_{rn}}$$

Issues:

- How to do the initialization?
- How to choose the learning rate?
- How to ensure nonnegativity?

Choose adaptive learning rate:

$$\gamma_{rn}^{(\ell)} := \frac{H_{rn}^{(\ell)}}{(W^T W H^{(\ell)})_{rn}}$$

- Update rule become multiplicative
- Nonnegative values stay nonnegative

NMF Optimization

NMF Algorithm

Lee, Seung: Algorithms for Non-Negative Matrix Factorization, Proc. NIPS, 2000.

Algorithm: NMF ($V \approx WH$)

Input: Nonnegative matrix V of size $K \times N$
Rank parameter $R \in \mathbb{N}$
Threshold ϵ used as stop criterion

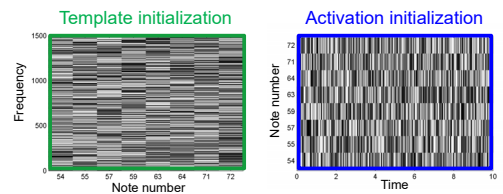
Output: Nonnegative template matrix W of size $K \times R$
Nonnegative activation matrix H of size $R \times N$

Procedure: Define nonnegative matrices $W^{(0)}$ and $H^{(0)}$ by some random or informed initialization. Furthermore set $\ell = 0$. Apply the following update rules (written in matrix notation):

- $H^{(\ell+1)} = H^{(\ell)} \odot \left(\frac{(W^{(\ell)})^T V}{(W^{(\ell)})^T W^{(\ell)} H^{(\ell)}} \right)$
- $W^{(\ell+1)} = W^{(\ell)} \odot \left(\frac{V (H^{(\ell+1)})^T}{W^{(\ell)} H^{(\ell+1)} (H^{(\ell+1)})^T} \right)$
- Increase ℓ by one.

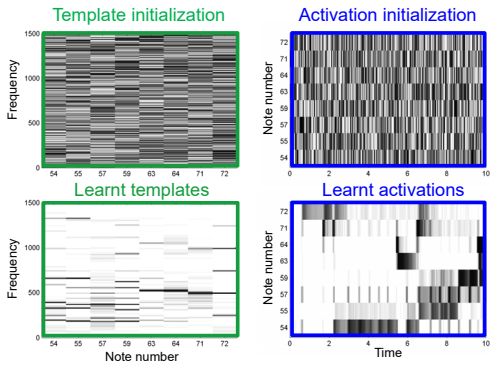
Repeat the steps (1) to (3) until $\|H^{(\ell)} - H^{(\ell-1)}\| \leq \epsilon$ and $\|W^{(\ell)} - W^{(\ell-1)}\| \leq \epsilon$ (or until some other stop criterion is fulfilled). Finally, set $H = H^{(\ell)}$ and $W = W^{(\ell)}$.

NMF-based Spectrogram Decomposition



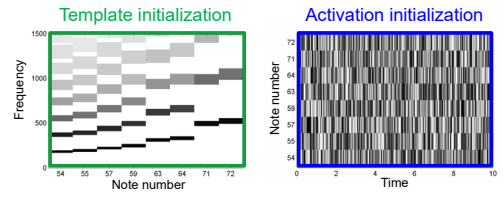
Random initialization

NMF-based Spectrogram Decomposition



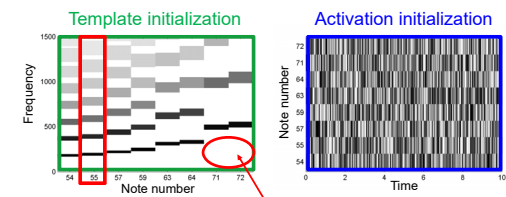
Random initialization → No semantic meaning

Constrained NMF: Templates



Enforce harmonic structure with zero-valued entries

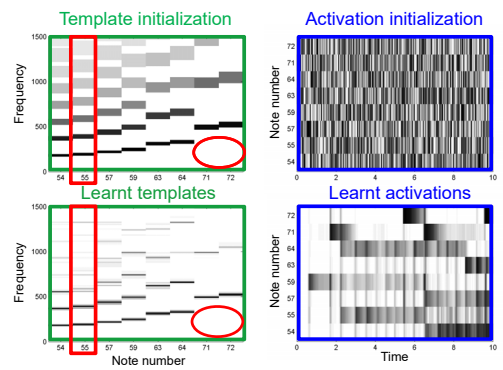
Constrained NMF: Templates



Template constraint for $p=55$

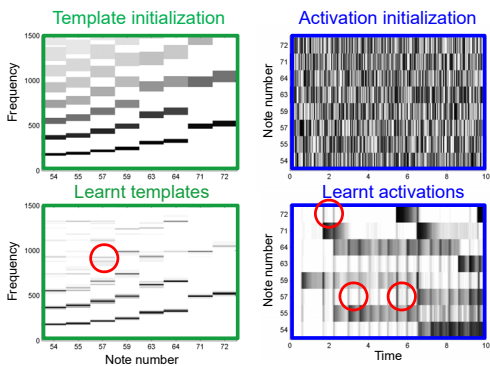
Enforce harmonic structure with zero-valued entries

Constrained NMF: Templates



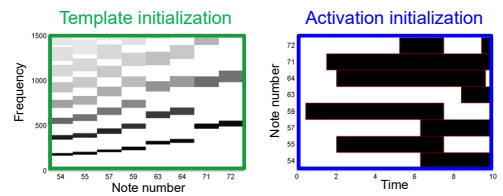
Zero-valued entries remain zero-valued entries!

Constrained NMF: Templates

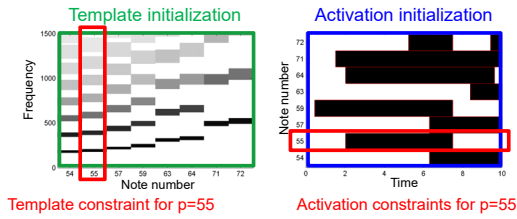


Pitch templates misused to represent onsets

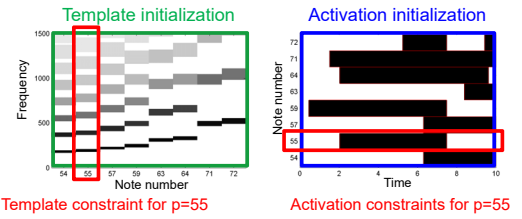
Constrained NMF: Double Constraints



Constrained NMF: Double Constraints



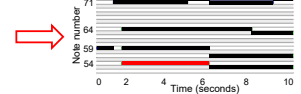
Constrained NMF: Double Constraints



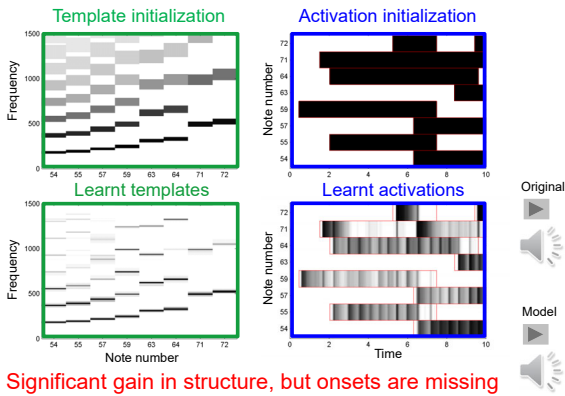
Sheet music



Such information may come from a synchronized score

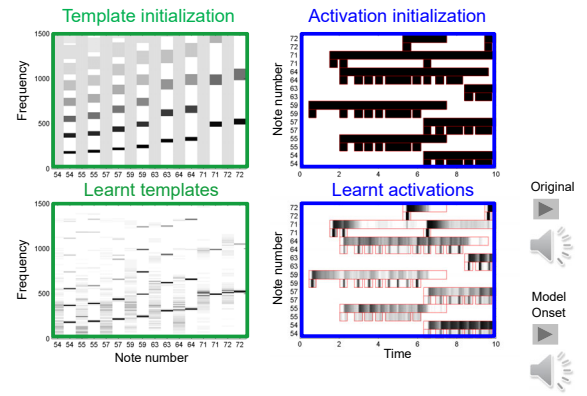


Constrained NMF: Double Constraints



Significant gain in structure, but onsets are missing

Constrained NMF: Onset Templates

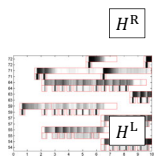


Score-Informed Audio Decomposition

Application: Separating left and right hands for piano



1. Split activation matrix

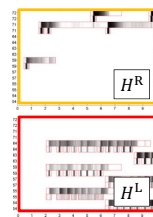


Score-Informed Audio Decomposition

Application: Separating left and right hands for piano



1. Split activation matrix

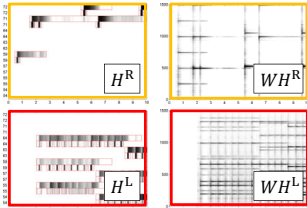


Score-Informed Audio Decomposition

Application: Separating left and right hands for piano



1. Split activation matrix
2. Model spectrogram for left/right

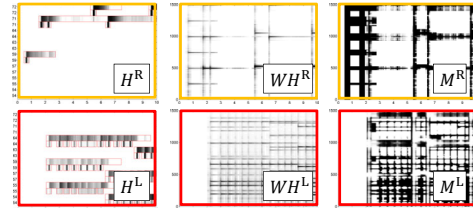


Score-Informed Audio Decomposition

Application: Separating left and right hands for piano



1. Split activation matrix
2. Model spectrogram for left/right
3. Separation masks for left/right

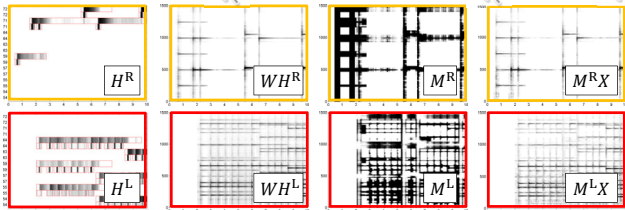


Score-Informed Audio Decomposition

Application: Separating left and right hands for piano



1. Split activation matrix
2. Model spectrogram for left/right
3. Separation masks for left/right
4. Estimated spectrograms for left/right



Score-Informed Audio Decomposition

Application: Separating left and right hands for piano

Chopin, Waltz Op. 64, No. 1



Original



Score-Informed Constraints

Ewert, Müller: Using Score-Informed Constraints for NMF-based Source Separation. Proc. ICASSP, 2012.

Further results available at <http://www.mpi-inf.mpg.de/resources/MIR/ICASSP2012-ScoreInformedNMF/>

Score-Informed Audio Decomposition

Application: Separating left and right hands for piano

Chopin, Waltz Op. 64, No. 1



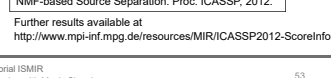
Original



Left/right hand



Right hand



Left hand



Score-Informed Constraints

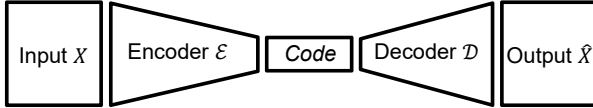
Ewert, Müller: Using Score-Informed Constraints for NMF-based Source Separation. Proc. ICASSP, 2012.

Further results available at <http://www.mpi-inf.mpg.de/resources/MIR/ICASSP2012-ScoreInformedNMF/>

Conclusions (NMF)

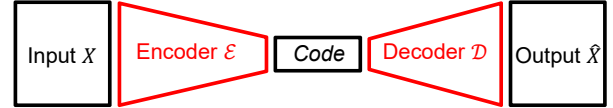
- NMF used for spectrogram decomposition
- Multiplicative update rules make it easy to constrain NMF model via zero initialization
- Exploiting score information to guide separation process (requires score-audio synchronization)
- Application: Separation of arbitrary note groups from given audio recording

Autoencoder



- Specific type of neural network
- Encoder: Compress input X into a low-dimensional code
- Decoder: Reconstruct output \hat{X} from code

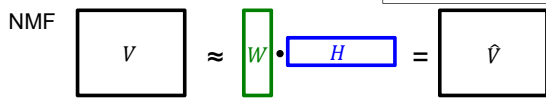
Autoencoder



- Specific type of neural network
- Encoder: Compress input X into a low-dimensional code
- Decoder: Reconstruct output \hat{X} from code
- Goal: Learn **parameters** for **encoder** and **decoder** such that output is close to input with respect to some loss function:

$$\mathcal{L}(X, \hat{X}) \approx 0$$

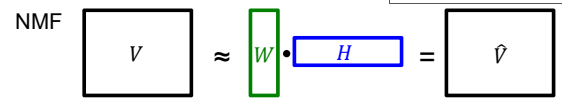
NMF and Autoencoder (AE)



$V \approx WH$ implies $W^+V \approx H$ with pseudoinverse W^+

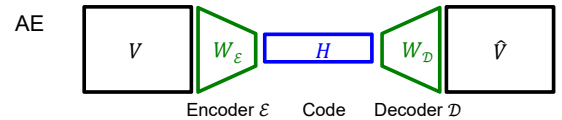
Nonnegative Autoencoder
Smaragdīs, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models, Proc. ICASSP 2017.

NMF and Autoencoder (AE)



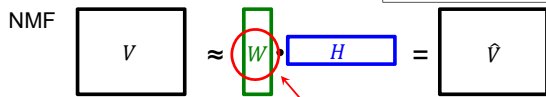
$V \approx WH$ implies $W^+V \approx H$ with pseudoinverse W^+

Nonnegative Autoencoder
Smaragdīs, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models, Proc. ICASSP 2017.



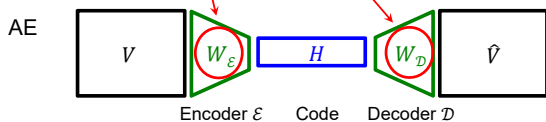
1. Layer: $H = W_\epsilon V$
2. Layer: $\hat{V} = W_D H$

NMF and Autoencoder (AE)



$V \approx WH$ implies $W^+V \approx H$ with pseudoinverse W^+

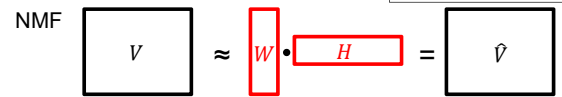
Nonnegative Autoencoder
Smaragdīs, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models, Proc. ICASSP 2017.



1. Layer: $H = W_\epsilon V$
2. Layer: $\hat{V} = W_D H$

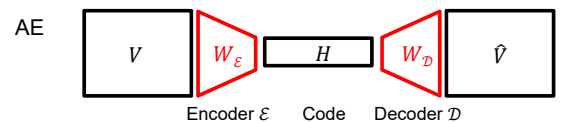
Fully connected network

NMF and Autoencoder (AE)



$V \approx WH$ implies $W^+V \approx H$ with pseudoinverse W^+

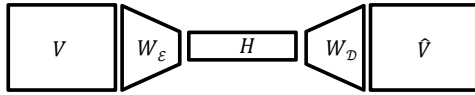
Nonnegative Autoencoder
Smaragdīs, Venkataramani: A Neural Network Alternative to Non-Negative Audio Models, Proc. ICASSP 2017.



1. Layer: $H = W_\epsilon V$
2. Layer: $\hat{V} = W_D H$

NMF: Learn H and W
AE: Learn W_ϵ and W_D

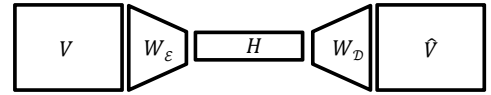
Nonnegative Autoencoder (NAE)



1. Layer: $H = W_\epsilon V$
2. Layer: $\hat{V} = W_D H$

- How can one adjust the AE to simulate NMF?
- How can one achieve nonnegativity?
- How can one incorporate musical knowledge?
- ...

Nonnegative Autoencoder (NAE)

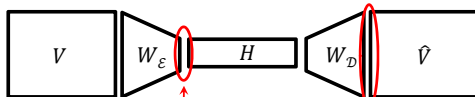


1. Layer: $H = W_\epsilon V$
2. Layer: $\hat{V} = W_D H$

$$\mathcal{L}(V, \hat{V}) = \|V - \hat{V}\|^2$$

- **Loss function:** same as in NMF

Nonnegative Autoencoder (NAE)

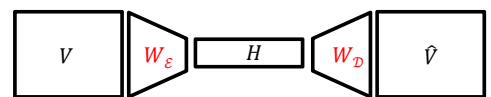


1. Layer: $H = \max(W_\epsilon V, 0)$
2. Layer: $\hat{V} = \max(W_D H, 0)$

$$\mathcal{L}(V, \hat{V}) = \|V - \hat{V}\|^2$$

- Loss function: same as in NMF
- Activation function (**ReLU**) makes H and \hat{V} nonnegative

Nonnegative Autoencoder (NAE)



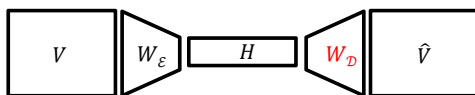
1. Layer: $H = \max(W_\epsilon V, 0)$
2. Layer: $\hat{V} = \max(W_D H, 0)$

$$\mathcal{L}(V, \hat{V}) = \|V - \hat{V}\|^2$$

$$W_D \leftarrow \max\left(W_D - \gamma \frac{\partial \mathcal{L}}{\partial W_D}, 0\right)$$

- Loss function: same as in NMF
- Activation function (**ReLU**) makes H and \hat{V} nonnegative
- **Projected gradient descent** can be used to keep W_D (and W_ϵ) nonnegative

Musical Constraints



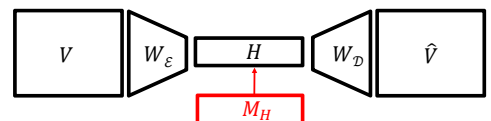
$$H = \max(W_\epsilon V, 0)$$

$$\hat{V} = \max(W_D H, 0)$$

- **Template constraints:** Project certain entries in W_D to zero values (using projected gradient decent)

Musical Constraints

Ewert, Sandler: Structured Dropout for Weak Label and Multi-Instance Learning and Its Application to Score-Informed Source Separation. Proc. ICASSP, 2017.



$$H' = H \odot M_H$$

$$\hat{V} = \max(W_D H', 0)$$

- **Template constraints:** Project certain entries in W_D to zero values (using projected gradient decent)
- **Activation constraints:** Use structured dropout by applying pointwise multiplication with binary mask M_H

NAE with Multiplicative Update Rules

- Multiplicative update rules in NMF:
 - Preserve nonnegativity
 - Lead to fast convergence
- Question: Can one introduce multiplicative update rules to train network weights for NAE?
- Use in additive gradient descent

$$W^{(\ell+1)} = W^{(\ell)} - \gamma \cdot \frac{\partial \mathcal{L}}{\partial W}$$

a suitable (adaptive) learning rate γ .

NAE with Multiplicative Update Rules

- Encoder:

$$H = W_{\mathcal{E}}V$$

- Structured Dropout:

$$H' = H \odot M_H$$

- Decoder:

$$\hat{V} = W_{\mathcal{D}}H'$$

NMF vs. NAE

Ozer, Hansen, Zunner, Müller: Investigating Nonnegative Autoencoders for Efficient Audio Decomposition. Proc. EUSIPCO, 2022.

NAE with Multiplicative Update Rules

- Encoder:

$$H = W_{\mathcal{E}}V$$

$$W_{\mathcal{E},rk}^{(\ell+1)} = W_{\mathcal{E},rk}^{(\ell)} \cdot \frac{\left((W_{\mathcal{D}}^{\top} V) \odot M_H \right)_{rk} V^{\top}}{\left((W_{\mathcal{D}}^{\top} W_{\mathcal{D}} H^{(\ell)}) \odot M_H \right)_{rk} V^{\top}}$$

- Structured Dropout:

$$H' = H \odot M_H$$

- Decoder:

$$\hat{V} = W_{\mathcal{D}}H'$$

$$W_{\mathcal{D},kr}^{(\ell+1)} = W_{\mathcal{D},kr}^{(\ell)} \cdot \frac{(V H'^{\top})_{kr}}{(W_{\mathcal{D}}^{(\ell)} H' H'^{\top})_{kr}}$$

Similar idea and computation as for NMF.

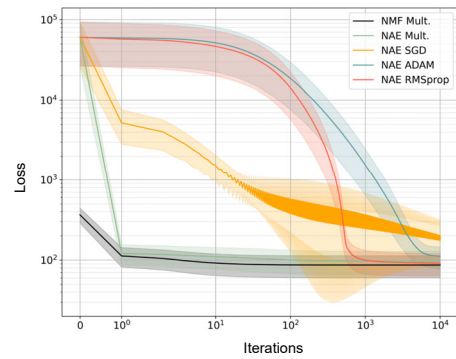
NMF vs. NAE

Ozer, Hansen, Zunner, Müller: Investigating Nonnegative Autoencoders for Efficient Audio Decomposition. Proc. EUSIPCO, 2022.

Approximation Loss

NMF vs. NAE

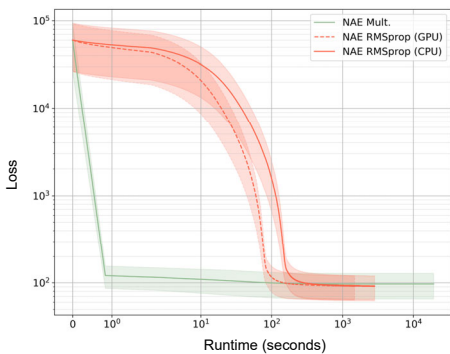
Ozer, Hansen, Zunner, Müller: Investigating Nonnegative Autoencoders for Efficient Audio Decomposition. Proc. EUSIPCO, 2022.



Approximation Loss

NMF vs. NAE

Ozer, Hansen, Zunner, Müller: Investigating Nonnegative Autoencoders for Efficient Audio Decomposition. Proc. EUSIPCO, 2022.



Conclusions (NAE)

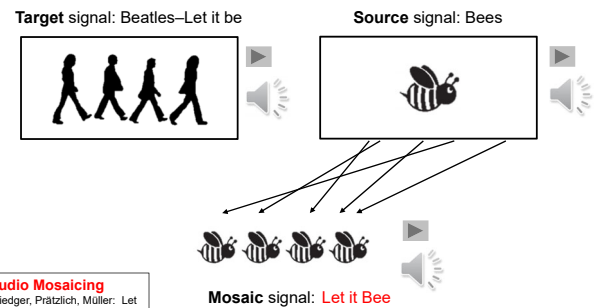
- Simulation of NMF:
 - Decoder corresponds to NMF templates
 - Encoder learns a kind of pseudo-inverse
 - Code corresponds to NMF activations
- Nonnegativity can be achieved via
 - activation function (ReLU)
 - projected gradient descent
 - multiplicative update rules
- Musical knowledge can be integrated via
 - removing network weights (template constraints)
 - structured dropout (activation constraints)

Outlook

- More complex networks
 - Deeper networks (more layers)
 - Different layer types (CNN, RNN, ...) and activation functions
 - Modification of loss function and regularization terms
- Understanding encoder – decoder relationship
 - Nonnegativity
 - Pseudo-inverse
- Update rules
 - Constraints and convergence issues
 - Adaptive learning rates and projected gradient descent

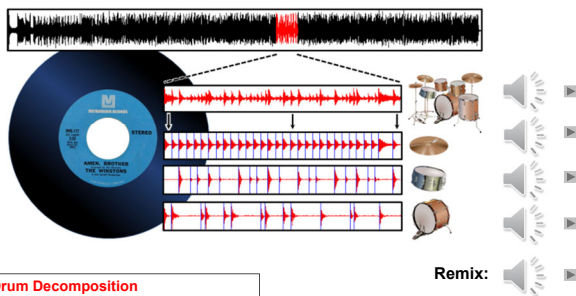
Score-Informed Audio Decomposition

Audio mosaicing (style transfer)



Score-Informed Audio Decomposition

Informed Drum-Sound Decomposition



Drum Decomposition
Dittmar, Müller: Reverse Engineering the Amen Break – Score-Informed Separation and Restoration Applied to Drum Recordings. IEEE/ACM TASLP 24(9), 2016.

Score-Informed Audio Decomposition

Major challenge: Reconstructed sound events often have artifacts

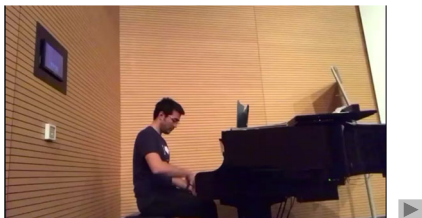
Approaches:

- Resynthesize certain sound components
- Differentiable Digital Signal Processing (DDSP) combines classical DSP and deep learning
- Generative adversarial networks may help to reduce the artifacts

DDSP
Engel et al.: DDSP: Differentiable Digital Signal Processing. ICLR, 2020.

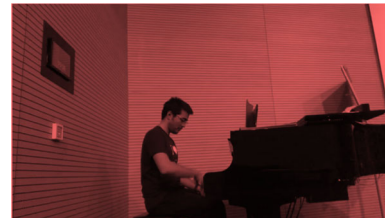
Source Separation (Piano Concerto)

- Yigitcan Özer
- PhD student in engineering
- Pianist



Source Separation (Piano Concerto)

- Yigitcan Özer
- PhD student in engineering
- Pianist



Only Piano!



Where is the orchestra?



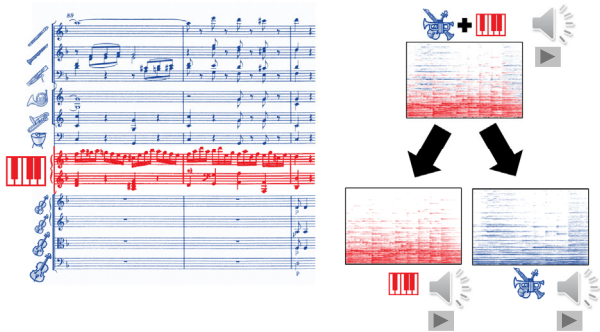
Source Separation (Piano Concerto)



Source Separation (Piano Concerto)



Source Separation (Piano Concerto)



Source Separation (Piano Concerto)



References (NMF, NAE)

- Daniel Lee and Sebastian Seung: Algorithms for Non-Negative Matrix Factorization. Proc. NIPS, 2000.
- Sebastian Ewert and Meinard Müller: Using Score-Informed Constraints for NMF-Based Source Separation. Proc. ICASSP, 2012.
- Paris Smaragdis and Shrikant Venkataramani: A Neural Network Alternative to Non-Negative Audio Models. Proc. ICASSP, 2017.
- Sebastian Ewert and Mark B. Sandler: Structured Dropout for Weak Label and Multi-Instance Learning and Its Application to Score-Informed Source Separation. Proc. ICASSP, 2017.
- Yigitcan Özer, Jonathan Hansen, Tim Zunner, and Meinard Müller: Investigating Nonnegative Autoencoders for Efficient Audio Decomposition. Proc. EUSIPCO, 2022.