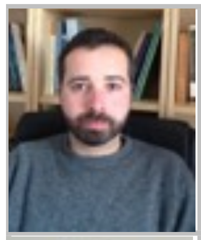
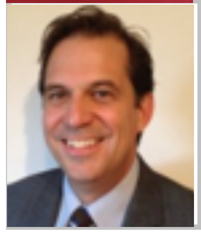
Audio Source Separation

Elias Kokkinis, CTO
elias@accusonus.com

accusonus

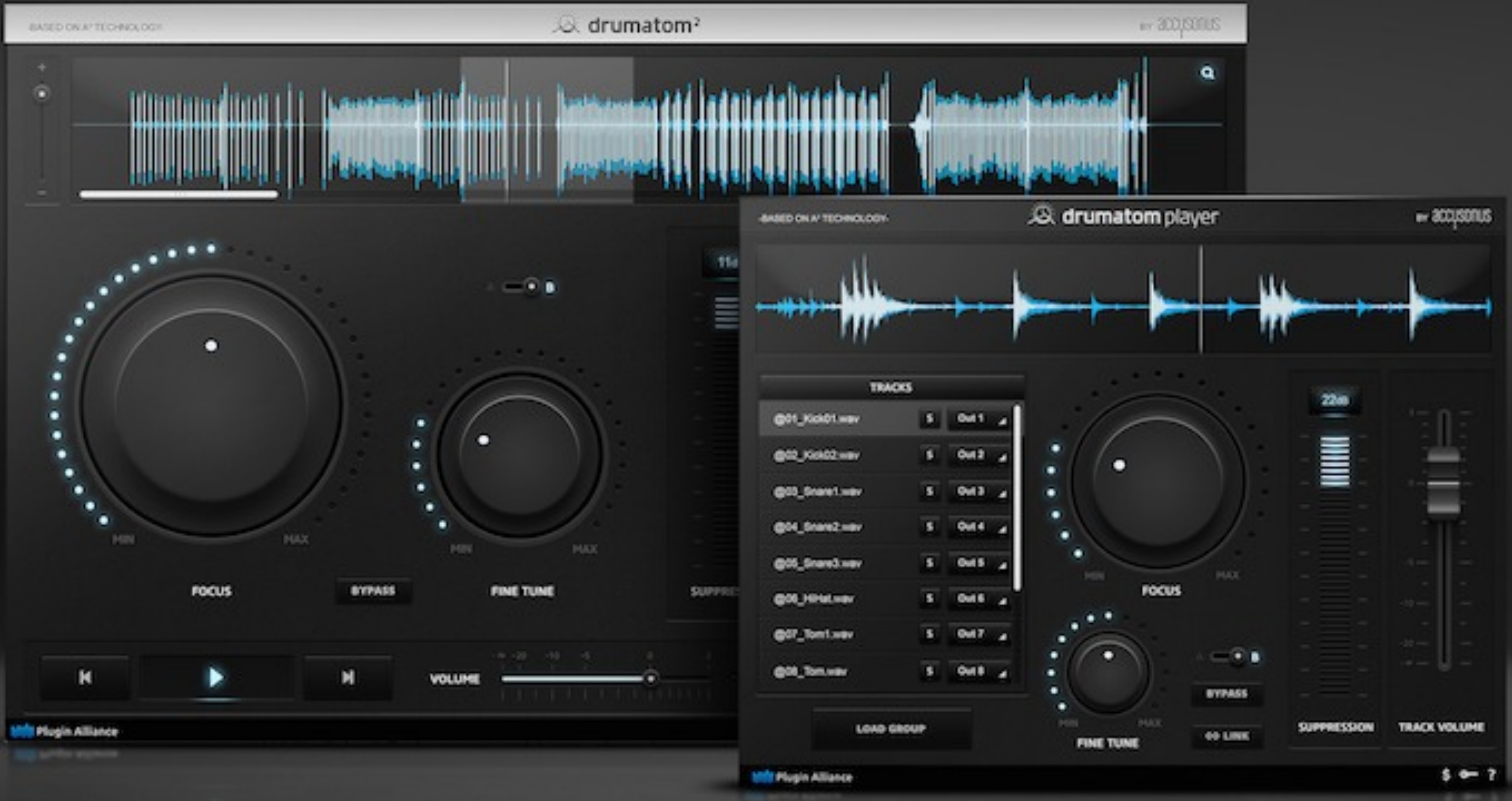
Accusonus is a Greek start-up, focusing on innovative digital audio technologies. The company's mission is to offer advanced solutions and unique business services for the music and speech technology sectors.

Board of Directors

	Alexandros Tsilfidis, PhD Founder & CEO	·Electrical and Computer Engineer ·MPhil and PhD in digital audio
	Elias Kokkinis, PhD Founder & CTO	·Electrical and Computer Engineer ·PhD in digital audio
	Michael Tzannes, PhD Board member and advisor	·Founder and ex-CEO of Aware Inc (Nasdaq: AWRE) ·Founder and Executive Manager at Tzannes Patent management LLC

+ a great team!

accusonus



drumatom²
drumatom player
TRACKS
@01_Kick01.wav
@02_Kick02.wav
@03_Snare1.wav
@04_Snare2.wav
@05_Snare3.wav
@06_HiHat.wav
@07_Tom1.wav
@08_Tom.wav
LOAD GROUP
FOCUS
FINE TUNE
BYPASS
LINK
SUPPRESSION
TRACK VOLUME
VOLUME
Plugin Alliance

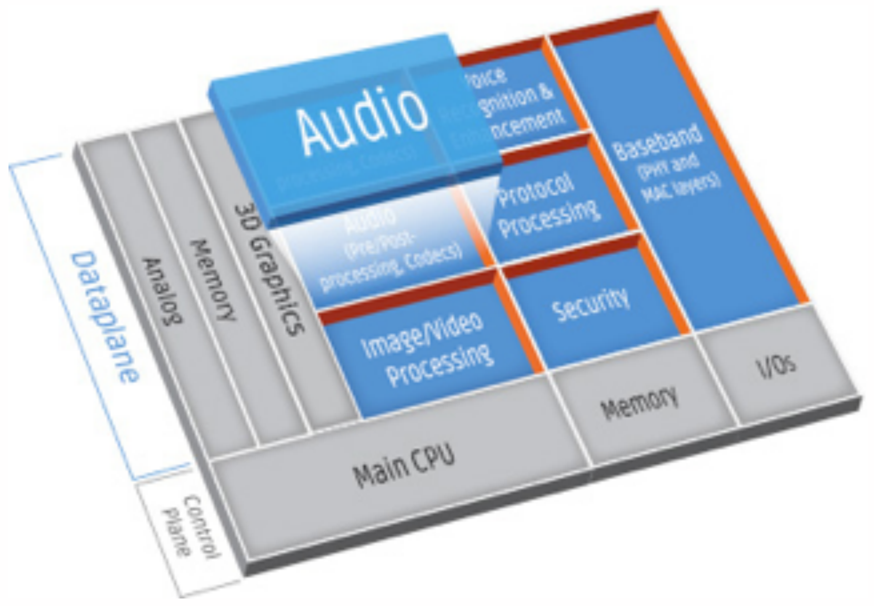
Focus Voice Engine

Focus-BDR/Focus-MDR

- Patent-pending
- The first dereverberation product in the market
- The first dual channel dereverberation in the market
- Combines state-of-the-art dereverberation algorithm with room acoustics

Focus-DNR

- State-of-the art denoise algorithm reengineered for tight integration with dereverberation



cādence™

https://github.com/EliasKokkinis/audio-source-separation

Audio Source Separation

Part I - Why do we need it?

Speech

- Can you understand what is being said?

Music

- Can you focus on the piano?
- Can you focus on the guitar?

Man vs. Machine

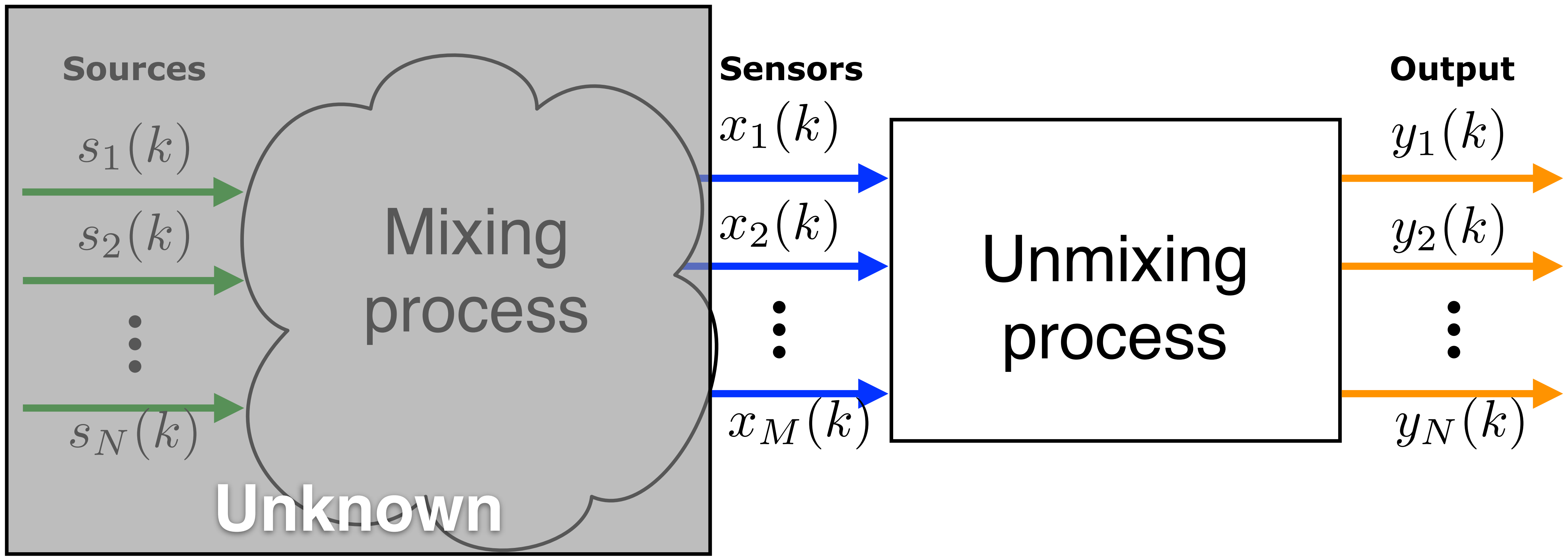
- Would a computer be able to do it?
- Probably **no!**
- We need audio source separation to *help computers understand* auditory events
- Applications:
  - speech enhancement and ASR performance improvement
  - music information retrieval applications (MIR)
  - better audio in studio and live (**that's what we do!**)
  - and many more...

Audio Source Separation

Part II - Blind source separation

Problem formulation

- Consider **N** sources and **M** sensors (microphones)



Problem formulation

- Consider **N** sources and **M** sensors (microphones)
- For each time instant

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$$

- **A** is the MxN **mixing matrix**
  - instantaneous mixing
  - three distinct cases:
    - overdetermined (M > N)
    - determined (M = N)
    - underdetermined (M < N). when M = 1 things get really tough...

Problem formulation

- Consider **N** sources and **M** sensors (microphones)
- For each time instant

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$$

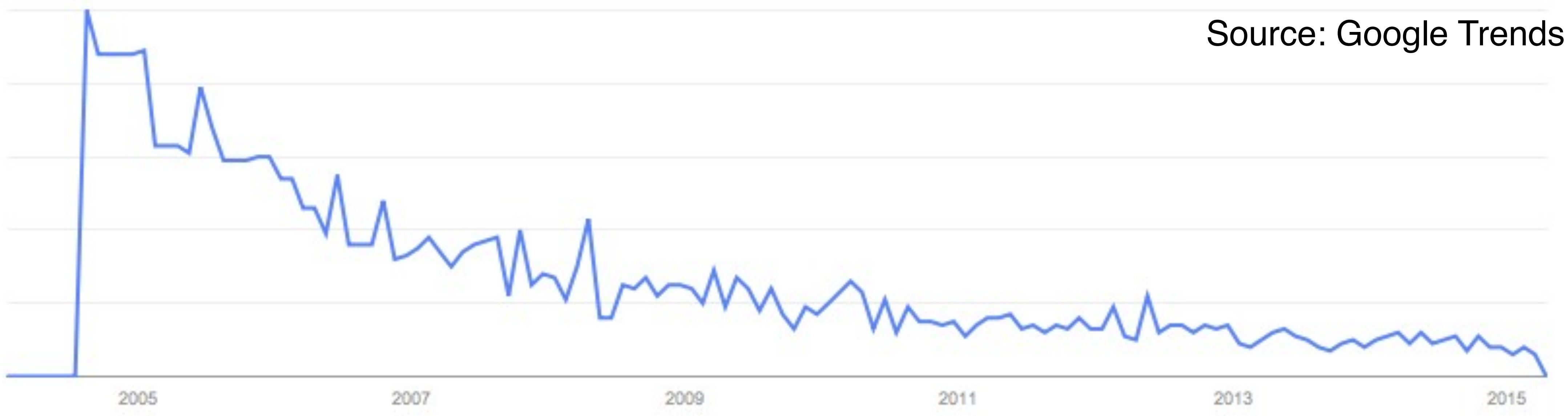
- Blind source separation
  - estimate an **unmixing matrix** **W** such that

$$\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t)$$

  - with **only** **x**(t)
  - we **don't** know anything about **A** or **s**(t)

Problem formulation

- The problem **cannot** be solved completely blind
- We have to make some **assumptions** about the source signals
- BSS was first introduced in the 1980s
- A very active research field for more than 20 years



Source: Google Trends

ICA

- Main assumption: **statistical independence (SI)**
  - the most widely used assumption
  - leads to Independent Component Analysis (ICA)
- Intuition:
  - knowing something about one signal does not give you any information about another signal
  - this is violated when signals are mixed

ICA

- ICA methods estimate an unmixing matrix so that the estimated signals are as SI as possible
- *How can we measure statistical independence?*
  - non-Gaussianity
  - mixed SI signals tend towards Gaussian (via CLT)
- *How can we measure non-Gaussianity?*
  - kurtosis: how “spiky” is a pdf
    - 4th order statistic, sensitive to outliers
  - negentropy: difference between the entropy of Gaussian and a given distribution
    - difficult to calculate

FastICA

- We want to maximize a measure of non-Gaussianity

$$\mathcal{J}_G = \sum_{i=1}^{N} \mathrm{E}\{G(\mathbf{w}_i^T \mathbf{x})\}$$

- G is a non-linear function
  - this function defines the measure that is maximized
- FastICA: an iterative method to maximize this cost function

Hyvärinen, A.; Oja, E. (2000). "Independent component analysis: Algorithms and applications". *Neural Networks* **13** (4–5): 411–430

FastICA

- The update for FastICA is

$$\mathbf{w}_i = \mathrm{E}\{\mathbf{x}g(\mathbf{w}_i^T\mathbf{x})\} - \mathrm{E}\{g'(\mathbf{w}_i^T\mathbf{x})\}\mathbf{w}_i$$

  - g() is the derivative of G()
  - g'() is the derivative of g()

- For kurtosis maximization

$$g(y) = y^3 \qquad g'(y) = 3y^2$$

- For negentropy maximization

$$g(y) = \tanh(y) \qquad g'(y) = 1 - (\tanh(y))^2$$

FastICA

- Implementation issues
  - Orthonormalization of W after each iteration

$$\bar{\mathbf{W}} = \mathbf{W}\left(\mathbf{W}^T\mathbf{W}\right)^{-\frac{1}{2}}$$

  - Preprocessing
    - Center the data (remove the mean)
    - Whiten the data: uncorrelated data with unity variance (use eigenvalues)

Ambiguities

- Ideally

$$\mathbf{W}\mathbf{A} = \mathbf{I}$$

- BSS suffers from a set of indeterminancies
- Formally this is stated as

$$\mathbf{W}\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{I}$$

  - permutation matrix
  - scaling matrix (diagonal)
  - identity matrix

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

- Permuted and scaled signals are still SI!

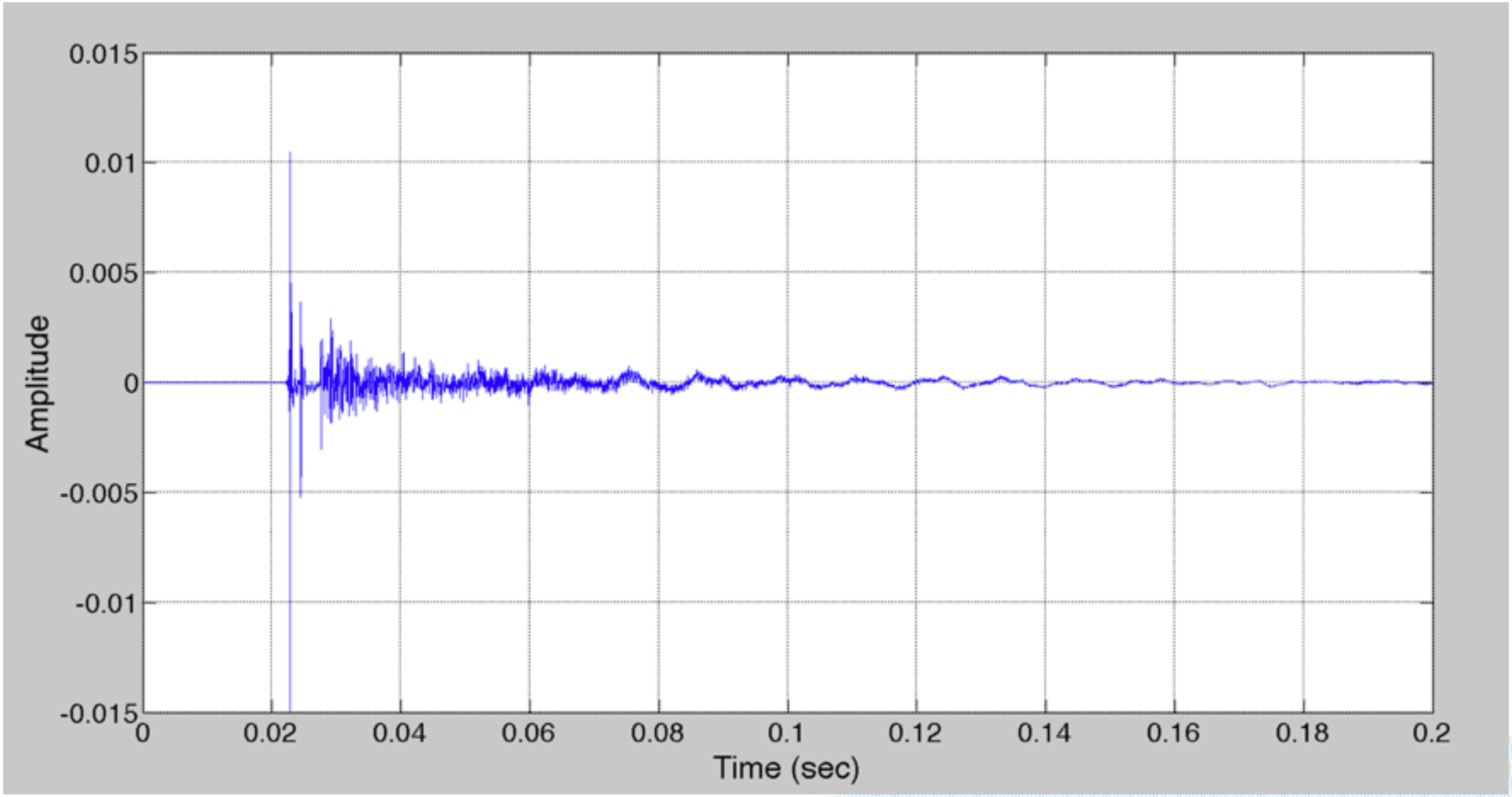
Implementation

Convolutive mixing

- Instantaneous mixing does not actually occur!
  - but it is very useful of biomedical applications
- Audio sources are typically inside rooms!
  - The mixing matrix becomes a set of filters
  - The number of parameters to estimate increases drastically!
- Convolutive BSS: a much harder problem!

Convolutive mixing

- The room as a filter

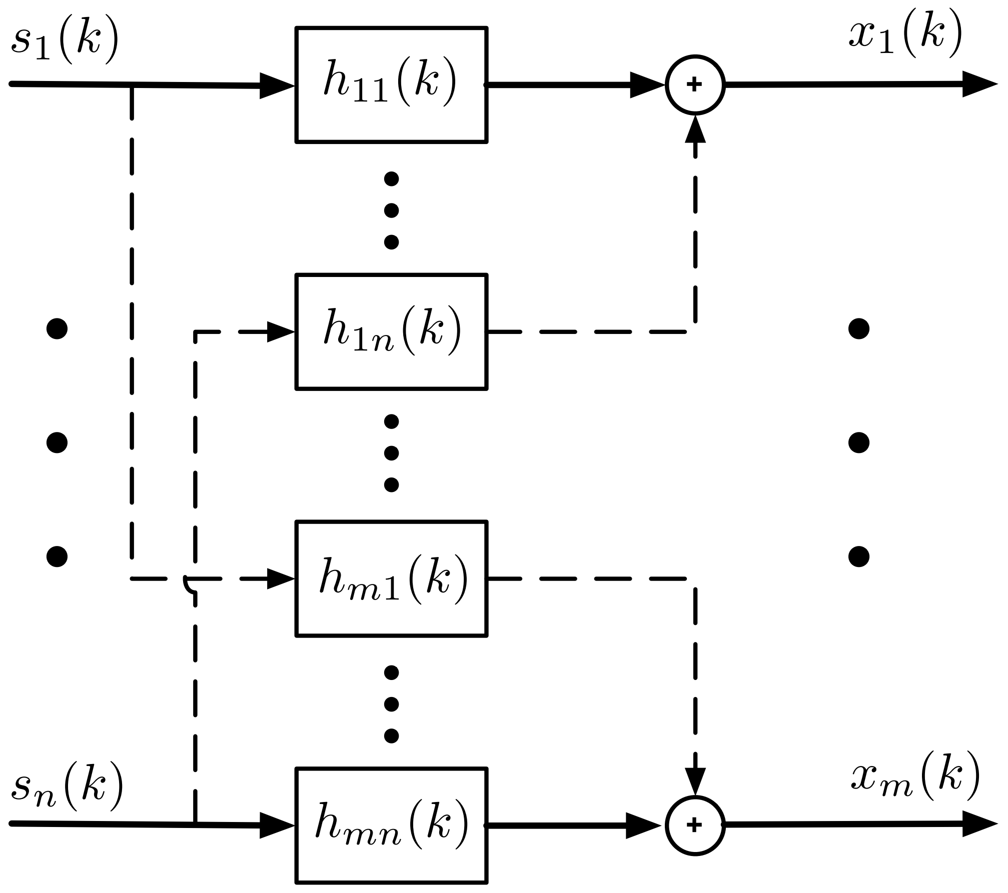


Convolutive mixing

- The room as a filter
  - Modeled as an FIR filter
  - The effect on a signal: convolution
  - The number of coefficients is related to fs and RT60
  - The room impulse response (RIR) characterizes a room
    - for the specific source-microphone position!
  - RIRs are non-minimum phase
    - their inversion is not straightforward!

Convolutive mixing

- Convolutive mixing
  - A block diagram



  - or in equation form

$$x_i(t) = \sum_j \sum_k a_{ij}(t) s_j(t-k)$$

Freq. domain ICA

- An elegant solution: transform to frequency domain
  - convolution becomes a multiplication

$$\mathbf{X}(\omega) = \mathbf{A}(\omega)\mathbf{S}(\omega)$$

- Solve an instantaneous ICA problem for each bin

Freq. domain ICA

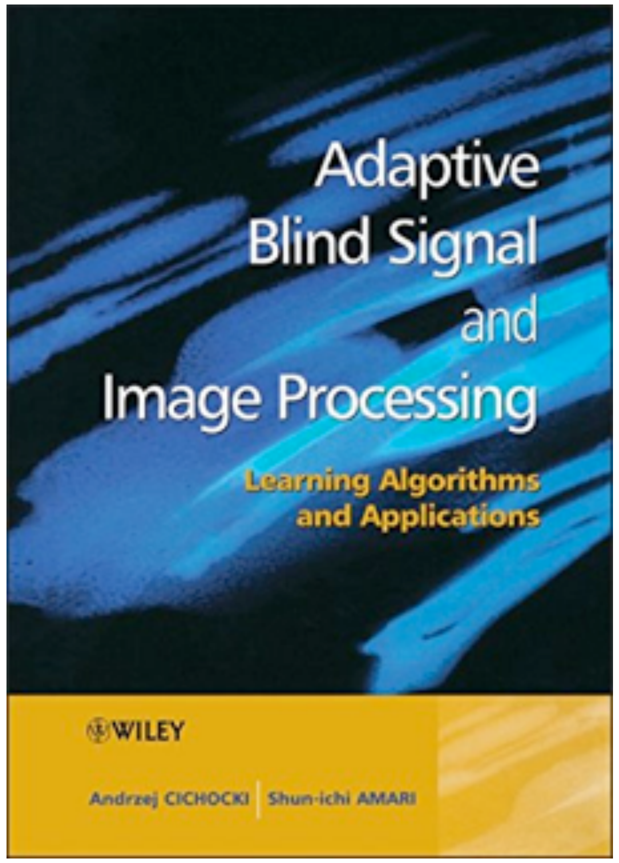
- But: ambiguities need to be resolved!
  - the **permutation** problem is quite complex but can be solved
  - **scaling** ambiguities introduce spectral distortions
- Further issues:
  - the length of the FFT has to be greater than the RIR length for the multiplication assumption to hold (due to **circular convolution**)
  - however a long FFT increases the number of parameters to adjust and **slows convergence**
  - also for longer windows the SI assumption **collapses**

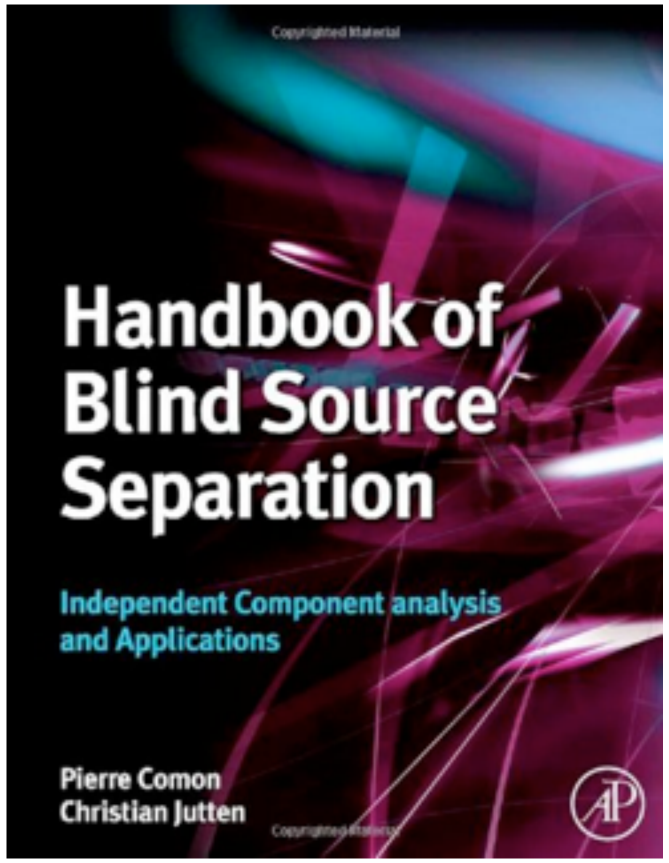
Convolutive BSS

- Convolutive BSS methods are quite involved
  - beyond the scope of this workshop
- They suffer from significant limitations in realistic audio applications

References

- Two very good books


Adaptive Blind Signal and Image Processing
Learning Algorithms and Applications
WILEY
Andrzej CICHOCKI | Shun-ichi AMARI


Handbook of Blind Source Separation
Independent Component analysis and Applications
Pierre Comon
Christian Jutten

- Review chapter *"Convolutive Blind Source Separation Methods"* by M. Pedersen et al. in "Handbook of Speech Processing" (Springer)

Audio Source Separation

Part III - Non-negative matrix factorization

Definition

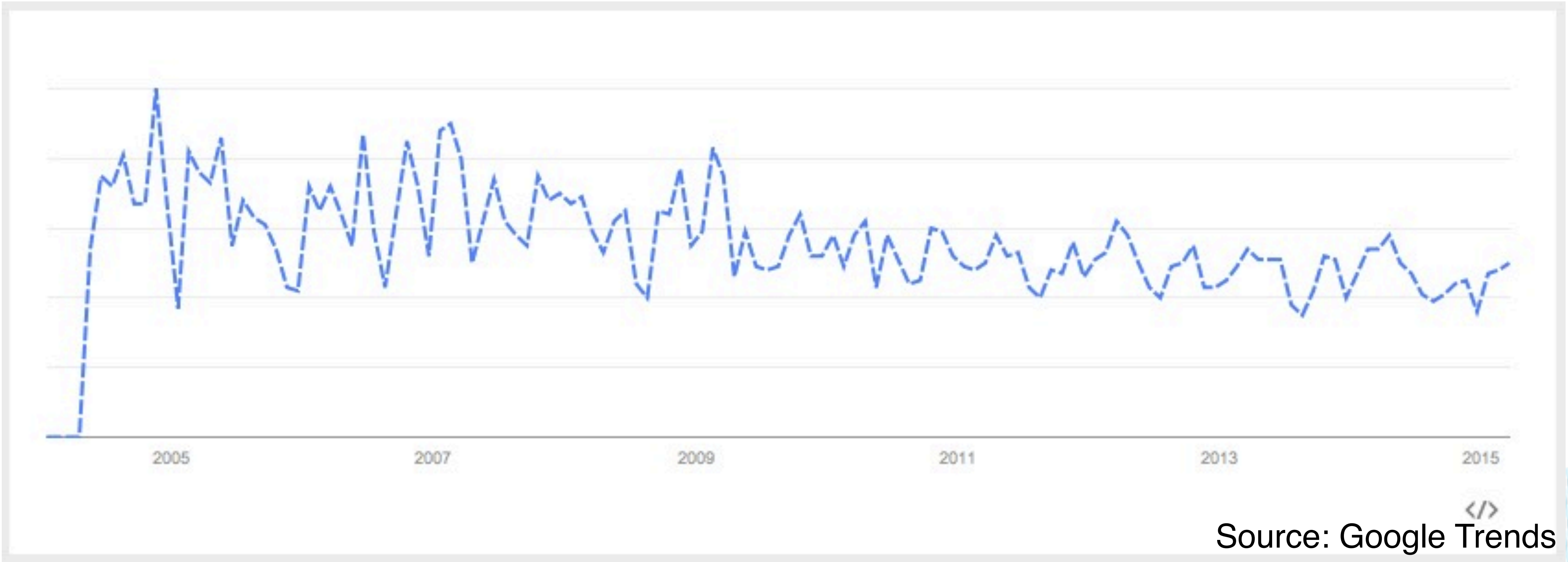
- Non-negative matrix factorization (NMF)
  - Given a non-negative matrix **V**,
  - find two matrices **W** and **H** such that

$$\mathbf{V} \approx \mathbf{W}\mathbf{H}$$

$$\mathbb{R}_+ : \{x \in \mathbb{R}, x \geq 0\}$$
$$\mathbf{V} \in \mathbb{R}_+^{F \times N}, \mathbf{W} \in \mathbb{R}_+^{F \times K}, \mathbf{H} \in \mathbb{R}_+^{K \times N}$$

A bit of history

- First introduced by Paatero and Tapper (1994)
- Became widely known with Lee and Seung (1999)
- Google Scholar reports more than 132.000 results
- A very strong research field even after 15 years.



Source: Google Trends

The matrices (1)

- **V** of size FxN is a collection of column vectors $\mathbf{v}_n$
  - each vector is a **data point**
  - each vector is F-dimensional (or it has F features)
  - there are N data points in total

$$\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \cdots \mathbf{v}_N]$$

$$\mathbf{v}_n = \begin{bmatrix} v_{1n} \\ v_{2n} \\ \vdots \\ v_{Fn} \end{bmatrix}$$

The matrices (2)

- **W** is the basis matrix of size FxK
- **W** has K columns of size F
- The columns of **W** span a linear space

The matrices (3)

- **H** is the weight matrix of KxN
- **H** has N columns of size K
  - The n-th column of **H** represents the coordinates of the n-th data point in the space defined by **W**

$$\mathbf{v}_n \approx \mathbf{W}\mathbf{h}_n$$

Dimensions

- K is the **rank** of the factorization
  - Typically K is chosen such that K ≪ FN
  - This leads to a dimensionality reduction in the data
  - For K > F, matrix **W** becomes overcomplete.
  - Overcomplete basis matrices are related to sparse signal processing.
  - K is the number of “building blocks”

Terminology & Notation

- A note on terminology:
  - **W** is called basis matrix
    - The columns of W are called basis or *atoms*
    - **W** is also called a dictionary (in spase signal processing)
  - **H** is called the weight or *gain* matrix
  - K is the *rank* of the factorization (also called number of components)

NMF discovers parts!

- **Non-negativity** leads to **parts-based** decomposition
  - The “building blocks” of **W** must be combined **constructively**
  - The algorithm is forced to discover the parts that make up **V**

Applications

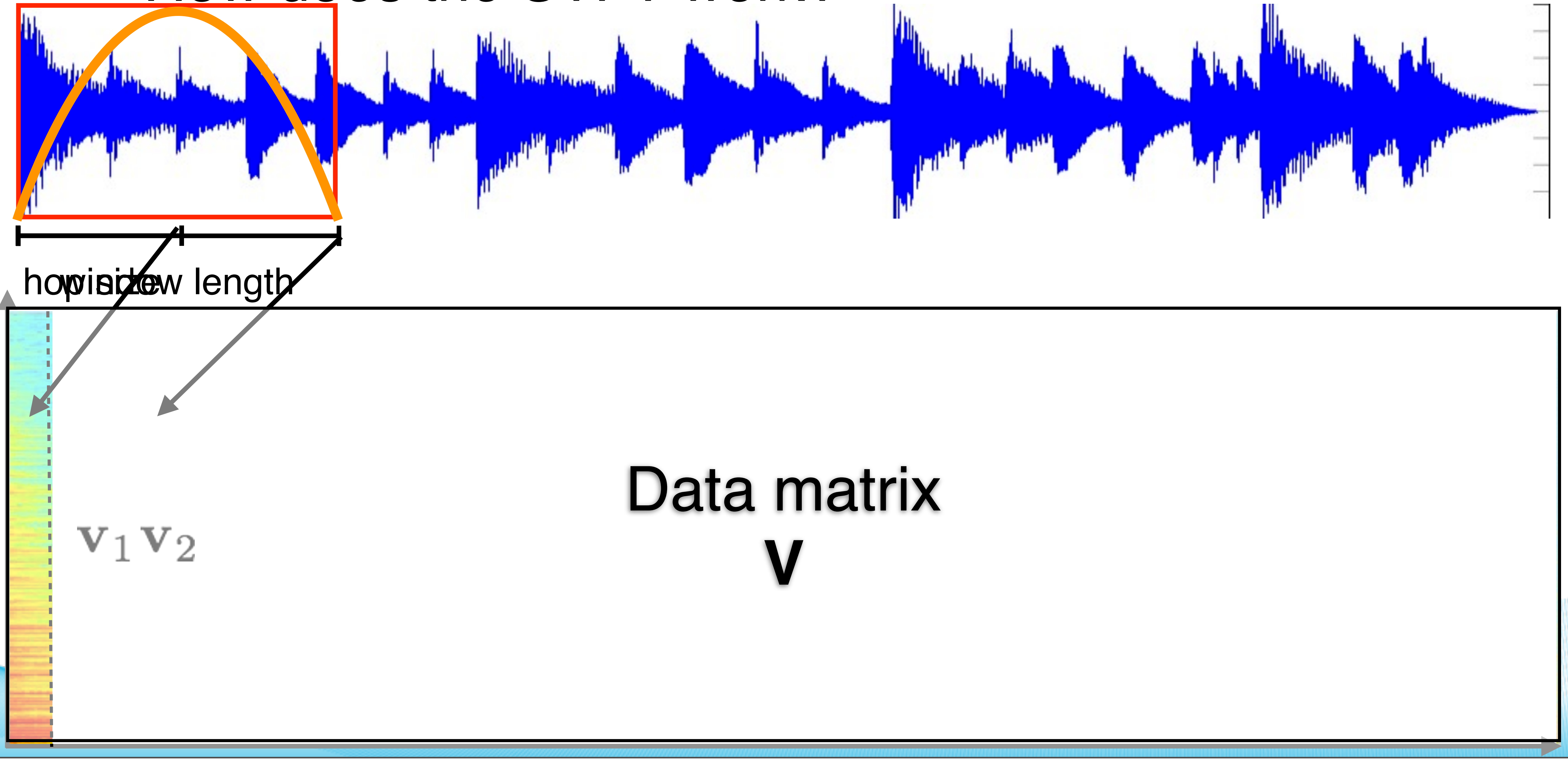
- Some applications of NMF
  - image processing
  - text mining
- We are concerned with audio applications.
- *How can we interpret **V, W** and **H** in audio?*

NMF in audio context

- **V** represents the audio spectrogram
  - F is the number of frequency bins
  - N is the number of audio frames
- **W** consists of *spectral profiles*
  - i.e. the magnitude spectra of elementary sources in **V**
- **H** represents the *gains or activation functions* of the spectral profiles
  - i.e. in a given frame which elementary sources are active

The spectrogram

- The spectrogram is the magnitude of the STFT
- How does the STFT work?

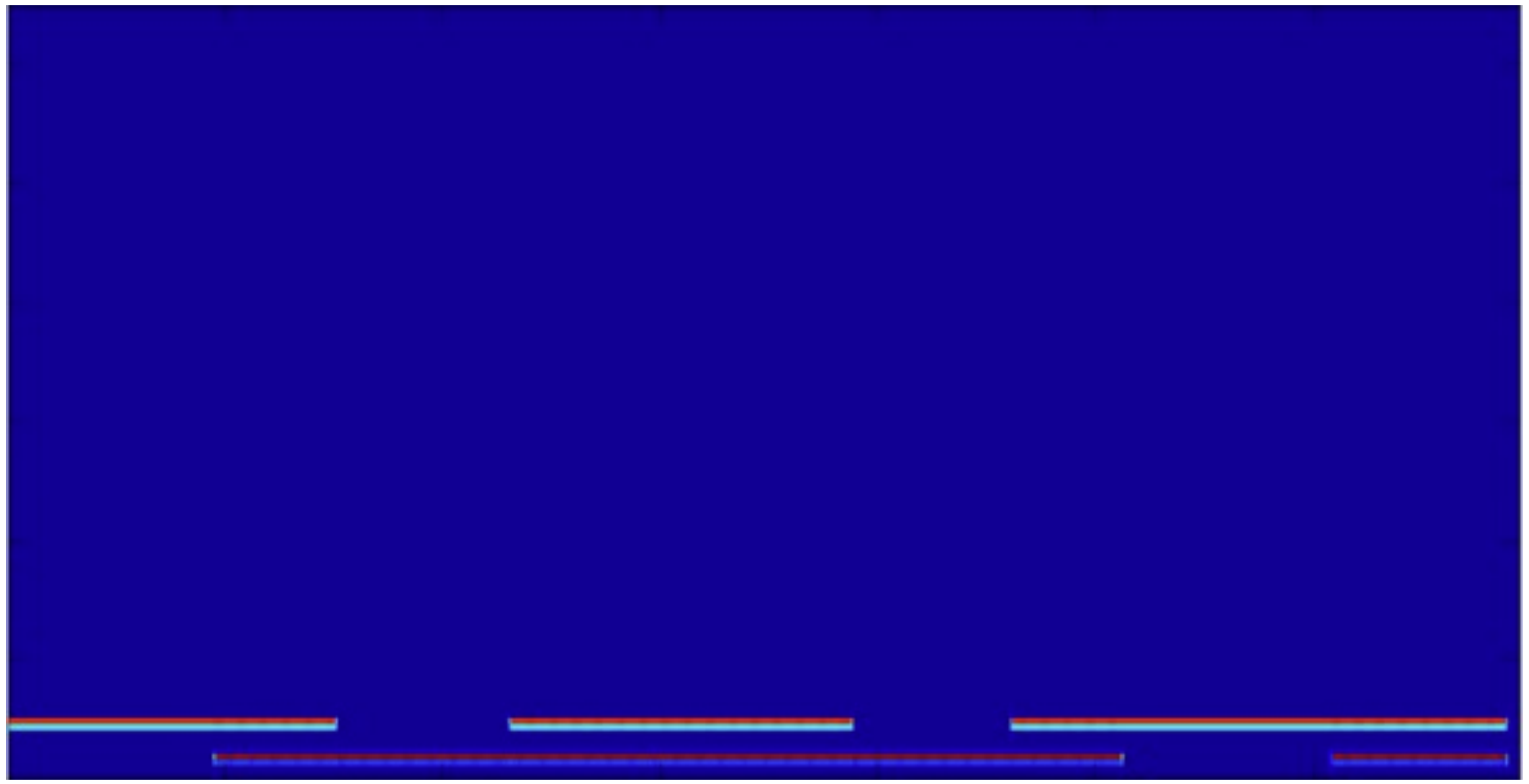


NMF in audio context

- **V** represents the audio spectrogram
  - F is the number of frequency bins
  - N is the number of audio frames
- **W** consists of *spectral profiles*
  - i.e. the magnitude spectra of elementary sources in **V**
- **H** represents the *gains or activation functions* of the spectral profiles
  - i.e. in a given frame which elementary sources are active

Toy example

- Two sine waves that come and go...



accusonus

NMF algorithms

- NMF can be expressed as an optimization problem

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} \mathcal{D}\left(\mathbf{V}, \mathbf{WH}\right)$$

- The distance between the matrices is calculated element-wise

$$\mathcal{D}\left(\mathbf{V}, \mathbf{WH}\right) = \sum_f \sum_n \underbrace{\boxed{d(v_{fn}, w_{fk}h_{kn})}}_{\text{scalar distance}}$$

- Only **V** is known.
  - NMF performs **unsupervised** learning

NMF algorithms

- Which distance to choose?
  - The most straightforward: Euclidean **distance**

$$d_{EUC}(x|y) = \frac{1}{2}(x-y)^2$$

  - (Generalized) Kullback-Leibler **divergence**

$$d_{KL}(x|y) = x \log \frac{x}{y} - x + y$$

  - Itakura-Saito **divergence**

$$\boxed{d_{IS}(x|y) = \frac{x}{y} - \log \frac{x}{y} - 1}$$

Itakura-Saito distance

- Proposed by Itakura and Saito in 1968
- Measures the difference between two **spectra**
- Reflects **perceptual similarity** between two spectra
  - used as a speech enhancement performance metric
- It is **scale invariant**
  - low and high energy data are treated the same

$$d_{IS}(x|y) = d_{IS}(\alpha x|\alpha y)$$

NMF algorithms

- A statistical insight behind the choice of distance
  - Euclidean distance: ML estimation of W, H in AGN

$$\mathbf{V} = \mathbf{W}\mathbf{H} + \mathbf{E}$$

  - KL divergence: ML estimation of W, H in Poisson noise
  - IS divergence: ML estimation of W, H in Gamma multiplicative noise

$$\mathbf{V} = \mathbf{W}\mathbf{H} \odot \mathbf{E}$$

NMF algorithms

- All the previous measures belong to the family of **beta divergences**:

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

NMF algorithms

- How can we **solve** the optimization problem?

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} \mathcal{D}\left(\mathbf{V}, \mathbf{WH}\right)$$

- It cannot be solved for **both** **W** and **H.**
    - Keep **W** **fixed** and optimize **H** and vice versa
- Common minimization approach: **gradient descent**

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla \mathcal{J}(\boldsymbol{\theta})$$

- batch approach - all data are considered
- calculate the gradient of the distance w.r.t. W and H
- rearrange terms in order to remove negative terms
- choose the appropriate step size

NMF algorithms

- Let's do this for **W**
  - The derivative of the beta divergence

$$d_\beta(x|y) = y^{\beta-2}(y-x)$$

  - The gradient with respect to **W**

$$\nabla_W D_\beta(\mathbf{V}|\mathbf{WH}) = \left(\mathbf{WH}^{\beta-2} \odot (\mathbf{WH}-\mathbf{V})\right)\mathbf{H}^T$$

  - Rearrange

$$\nabla_W D_\beta(\mathbf{V}|\mathbf{WH}) = \mathbf{WH}^{\beta-1}\mathbf{H}^T - \left(\mathbf{WH}^{\beta-2} \odot \mathbf{V}\right)\mathbf{H}^T$$

$$\nabla^+_W D_\beta(\mathbf{V}|\mathbf{WH}) \qquad \nabla^-_W D_\beta(\mathbf{V}|\mathbf{WH})$$

NMF algorithms

- Let’s do this for **W**
  - The gradient descent method

$$\mathbf{W}^{i+1} \leftarrow \mathbf{W}^{i} - \eta \left(\nabla_W D_\beta(\mathbf{V}|\mathbf{WH}) \right)$$

$$\mathbf{W}^{i+1} \leftarrow \mathbf{W}^{i} - \eta \left(\nabla^+_W D_\beta(\mathbf{V}|\mathbf{WH}) - \nabla^-_W D_\beta(\mathbf{V}|\mathbf{WH}) \right)$$

  - Choose the step size as

$$\eta = \frac{\mathbf{W}^i}{\nabla^+_W D_\beta(\mathbf{V}|\mathbf{WH})}$$

  - Substitute

$$\mathbf{W}^{i+1} \leftarrow \mathbf{W}^{i} \frac{\nabla^-_W D_\beta(\mathbf{V}|\mathbf{WH})}{\nabla^+_W D_\beta(\mathbf{V}|\mathbf{WH})}$$

NMF algorithms

- NMF multiplicative update rules for beta divergence

$$\mathbf{W} \leftarrow \mathbf{W}\frac{\left((\mathbf{WH})^{\beta-2}\odot\mathbf{V}\right)\mathbf{H}^T}{(\mathbf{WH})^{\beta-1}\mathbf{H}^T}$$

$$\mathbf{H} \leftarrow \mathbf{H}\frac{\mathbf{W}^T\left((\mathbf{WH})^{\beta-2}\odot\mathbf{V}\right)}{\mathbf{W}^T(\mathbf{WH})^{\beta-1}}$$

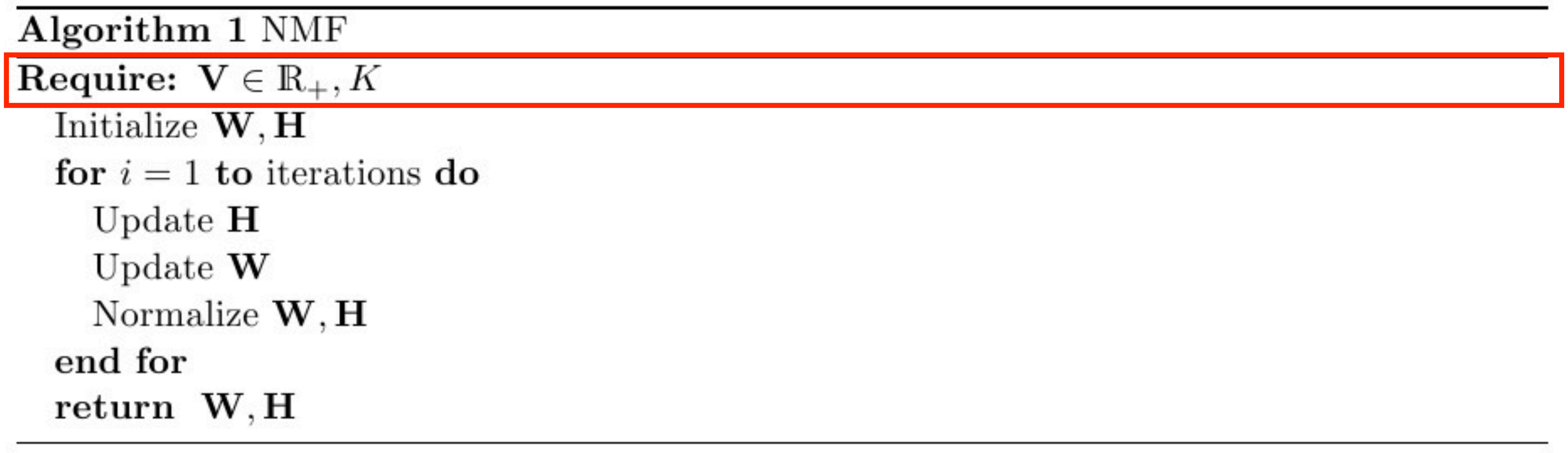
- Why multiplicative?
  - non-negativity is preserved
  - easy to implement

NMF algorithms

- The MU updates are proved to **converge**
    - Lee & Seung’s classic paper
- The NMF problem is **not convex for both** **W** and **H**
    - it is convex for W or H
    - It is **not guaranteed** that a global minimum will be found

NMF algorithms

- The basic NMF algorithm



```
Algorithm 1 NMF
Require: $\mathbf{V} \in \mathbb{R}_+, K$
  Initialize $\mathbf{W}, \mathbf{H}$
  for $i = 1$ to iterations do
    Update $\mathbf{H}$
    Update $\mathbf{W}$
    Normalize $\mathbf{W}, \mathbf{H}$
  end for
  return $\mathbf{W}, \mathbf{H}$
```

How to start?

- Which are the initial values of **W** and **H**?
  - the most common approach: **random non-negative values!**
    - the specific random value distribution may affect results
  - NMF is generally **sensitive** to initial values
  - more involved initialization strategies have been proposed
  - initialization **depends** on application
    - for audio applications, random is enough

When to stop?

- NMF is calculated iteratively.
  - How **many** iterations do we need?
- Typically you check that the cost function decreases more than ε (tolerance) in each iteration
- You can also set a **maximum** number of iterations.
- For audio applications: trial and error
  - a smaller value of the cost function does not mean better perceptual quality

Other issues

- How do we choose K?
  - model order selection is a difficult problem
  - most approaches are trial and error
  - some approaches based on the eigenvalues of **V**
  - the number of components K is not necessarily equal to the number of sources
- Normalization
  - avoid getting stuck by small values due to multiplicative updates
  - normalize columns of **W**
  - the cost function changes!

NMF algorithms

- Adding constraints: the beauty of NMF
  - incorporate **prior knowledge** about the data
  - easy to extend the core method for specific applications
- We now want to solve

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} \mathcal{J}\left(\mathbf{W},\mathbf{H}\right)$$

- where the cost function is

$$\mathcal{J}\left(\mathbf{W},\mathbf{H}\right) = \mathcal{D}_\beta\left(\mathbf{V},\mathbf{WH}\right) + c_1\Phi(\mathbf{W}) + c_2\Psi(\mathbf{H})$$

NMF algorithms

- The multiplicative updates have the same form:

$$\mathbf{W} \leftarrow \mathbf{W} \frac{\nabla^-_W \mathcal{J}(\mathbf{W},\mathbf{H})}{\nabla^+_W \mathcal{J}(\mathbf{W},\mathbf{H})}$$

- where each gradient is rearranged to form

$$\boxed{\nabla^+_W \mathcal{J}(\mathbf{W},\mathbf{H}) = \nabla^+_W \mathcal{D}_\beta(\mathbf{V},\mathbf{WH}) + c_1 \nabla^+_W \Phi(\mathbf{W}) + \cancel{c_2 \nabla^+_W \Psi(\mathbf{H})}}$$

$$\boxed{\nabla^-_W \mathcal{J}(\mathbf{W},\mathbf{H}) = \nabla^-_W \mathcal{D}_\beta(\mathbf{V},\mathbf{WH}) + c_1 \nabla^-_W \Phi(\mathbf{W}) + \cancel{c_2 \nabla^-_W \Psi(\mathbf{H})}}$$

- Of course the same holds for **H**

NMF algorithms

- The most common constraint: **sparsity!**
  - each data point is a combination of **some** sources
- How is sparsity measured?
  - There are several ways which can be summarized as

$$\Psi(\mathbf{H}) = \sum_{k,n} f(h_{kn})$$

  - A very simple and efficient way is to choose $f(x) = x$
  - Since all elements are non-negative it is equivalent to the entrywise l_1 norm

$$\Psi(\mathbf{H}) = ||\mathbf{H}||_1$$

NMF algorithms

- So the cost function now is

$$\mathcal{J}(\mathbf{W},\mathbf{H}) = \mathcal{D}_\beta(\mathbf{V},\mathbf{W}\mathbf{H}) + \lambda\Psi(\mathbf{H})$$

- The gradients of the constraint are

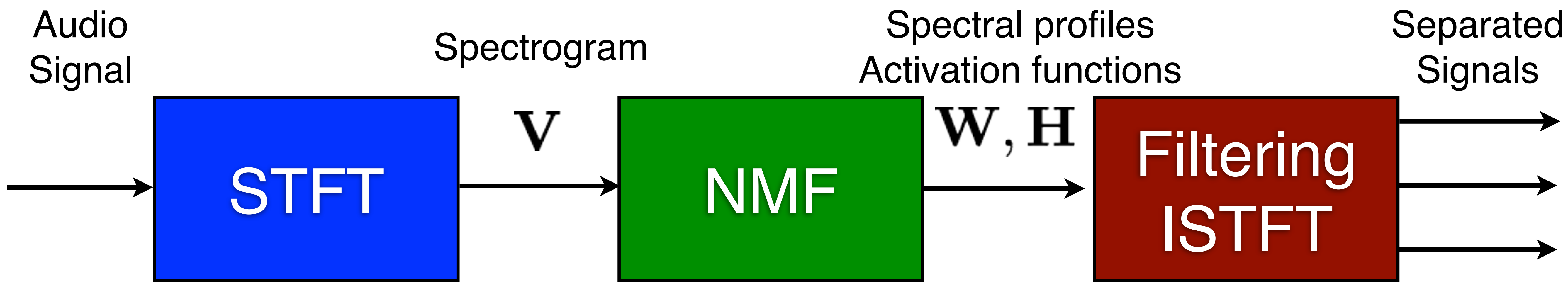
$$\nabla_H^+\Psi(\mathbf{H}) = 1 \qquad \nabla_H^-\Psi(\mathbf{H}) = 0$$

- And the new multiplicative update rules are

$$\mathbf{W} \leftarrow \mathbf{W}\frac{\left((\mathbf{W}\mathbf{H})^{\beta-2}\odot\mathbf{V}\right)\mathbf{H}^T}{(\mathbf{W}\mathbf{H})^{\beta-1}\mathbf{H}^T} \qquad \mathbf{H} \leftarrow \mathbf{H}\frac{\mathbf{W}^T\left(\mathbf{V}\odot(\mathbf{W}\mathbf{H})^{\beta-2}\right)}{\mathbf{W}^T(\mathbf{W}\mathbf{H})^{\beta-1}+\lambda}$$

How to separate using NMF?

- Building blocks of an NMF based separation system



STFT details

- STFT: from time to time-frequency domain
- Why? Audio signals are sparse in this domain
- Time-frequency resolution:
  - trade-off: choose good time OR frequency resolution
  - workaround: decrease hop size
    - this increases the number of data points tremendously
- Other things to consider:
  - analysis window type
  - spectrogram domain (magnitude or power)

Separating with masks

- We know how to get **V**, **W** and **H**.
- How do we get the separated signals?
- Time-frequency masks
  - Recall the Wiener filter

$$H(\omega) = \frac{P_{ss}(\omega)}{P_{ss}(\omega) + P_{nn}(\omega)}$$

  - Pseudo-Wiener masks

$$\mathbf{M}_k = \frac{\mathbf{w}_k \mathbf{h}^k}{\mathbf{W}\mathbf{H}}$$

  - Masks are real. Signal phase is left untouched.

Separating with masks

- Masks are applied on the complex spectrogram **X**
- Each mask produces a new spectrogram
- We perform ISTFT to obtain the separated signal

Implementation

Beyond NMF

- NMF can be trained!
  - Semi-supervised case
  - Supervised case
- Sort components: Make sense of the output data!
- Multichannel extensions: tensor factorizations
- Bayesian formulations
- Probabilistic Latent Component Analysis (PLCA)

References

- D.D. Lee and H.B. Seung, “Learning the parts of objects by non-negative matrix factorization”, *Nature* **401**, 788-791
- D.D. Lee and H.B. Seung, “Algorithms for non-negative matrix factorization”, NIPS 2000
- N. Gillis, *"The Why and How of Nonnegative Matrix Factorization"*, In: "Regularization, Optimization, Kernels, and Support Vector Machines", J.A.K. Suykens, M. Signoretto and A. Argyriou (eds), Chapman & Hall/CRC, Machine Learning and Pattern Recognition Series, pp. 257-291, 2014.
- Y.-X. Wang and Y.-J. Zhang, “Nonnegative Matrix Factorization: A Comprehensive Review,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353.
- T. Virtanen, J. F. Gemmeke, B. Raj, and P. Smaragdis, “Compositional Models for Audio Processing: Uncovering the structure of sound mixtures,” *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 125–144, Feb. 2015.
- T. Virtanen, “Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 3, pp. 1066–1074.
- A. Cichocki, R. Zdunek, A.H. Phan, “Non-negative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation”

Thank you!

accusonus