Introduction to Nonnegative Matrix Factorisation

Slim ESSID

Telecom ParisTech

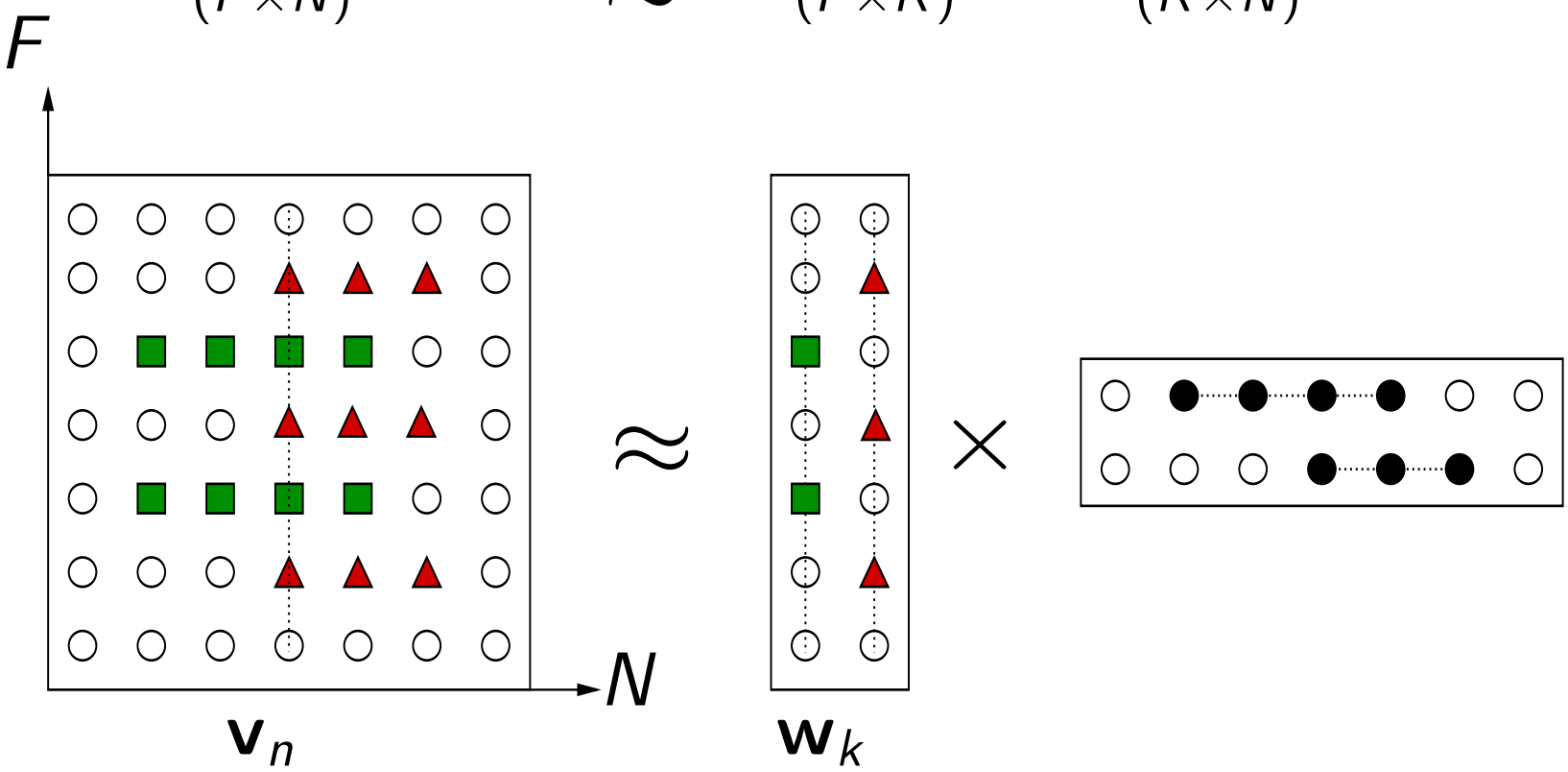
June 2018

- Introduction
- Principal Component Analysis (PCA)
- Introduction to NMF
- NMF models
- Algorithms for solving NMF
- Conclusion

Explaining data by factorisation

General formulation

$$\mathbf{V}_{(F\times N)} \approx \mathbf{W}_{(F\times K)} \times \mathbf{H}_{(K\times N)}$$

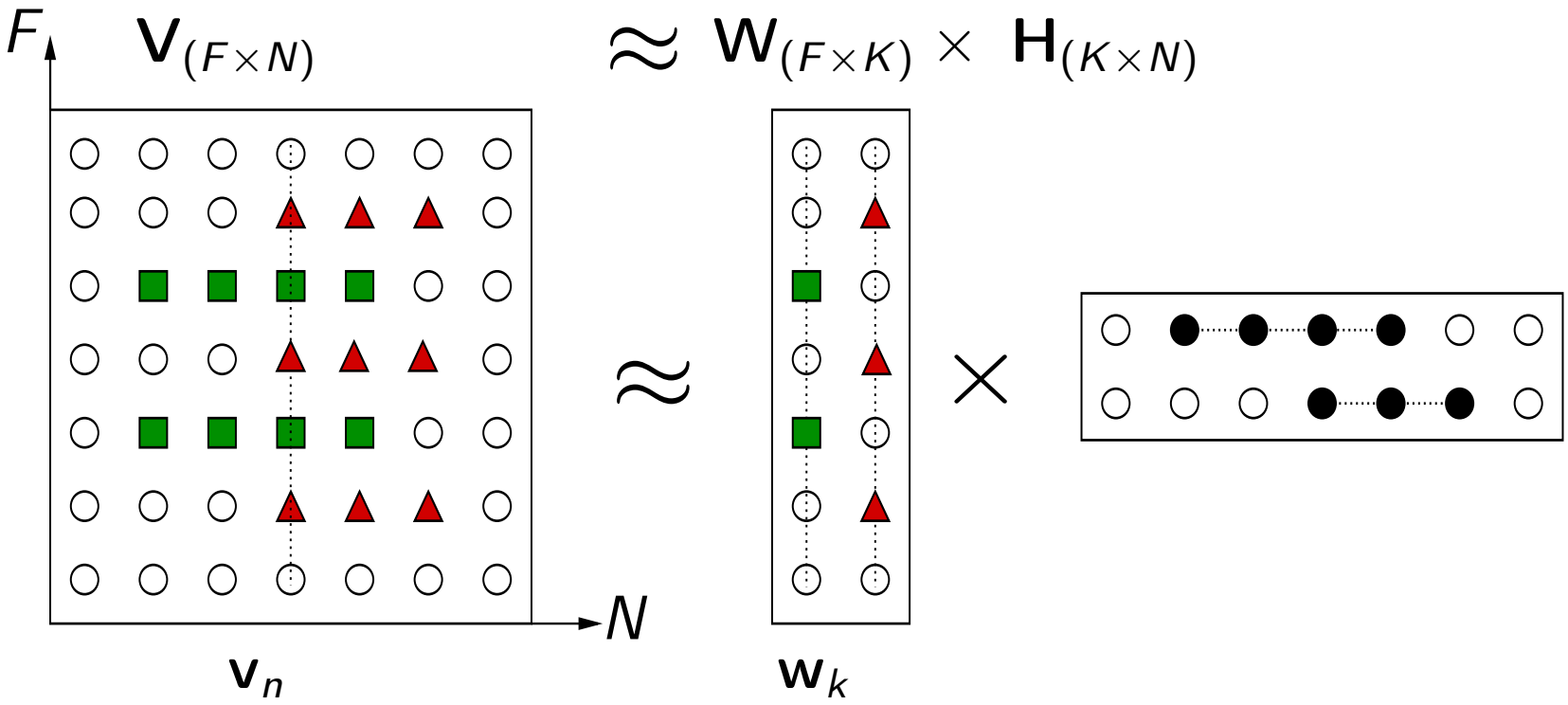


$$\mathbf{v}_n \approx \sum_{k=1}^{K} h_{kn}\mathbf{w}_k$$

Illustration by C. Févotte

Explaining data by factorisation

General formulation



data matrix

"explanatory variables"
"basis", "dictionary",
"patterns", "topics"

"regressors",
"activation coefficients",
"expansion coefficients"

Illustration by C. Févotte

PCA

- The data is assumed real-valued ($\mathbf{v}_n \in \mathbb{R}^F$) and centered ($\mathrm{E}\{\mathbf{v}_n\} = 0$)
- PCA returns the best linear approximation to the data in **least squares** sense:

$$\mathbf{v}_n \approx \hat{\mathbf{v}}_n = \mathbf{W}\mathbf{W}^T\mathbf{v}_n = \sum_{k=1}^{K} < \mathbf{v}_n, \mathbf{w}_k > \mathbf{w}_k$$

where $\mathbf{W} \in \mathbb{R}^{F \times K}$ is such that the least squares error is minimized:

$$\mathbf{W}_{PCA} = \min_{\mathbf{W}} \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{v}_n - \hat{\mathbf{v}}_n\|_2^2 = \frac{1}{N} \|\mathbf{V} - \mathbf{W}\mathbf{W}^T\mathbf{V}\|_F^2$$

PCA

- The solution can be shown to be of the form

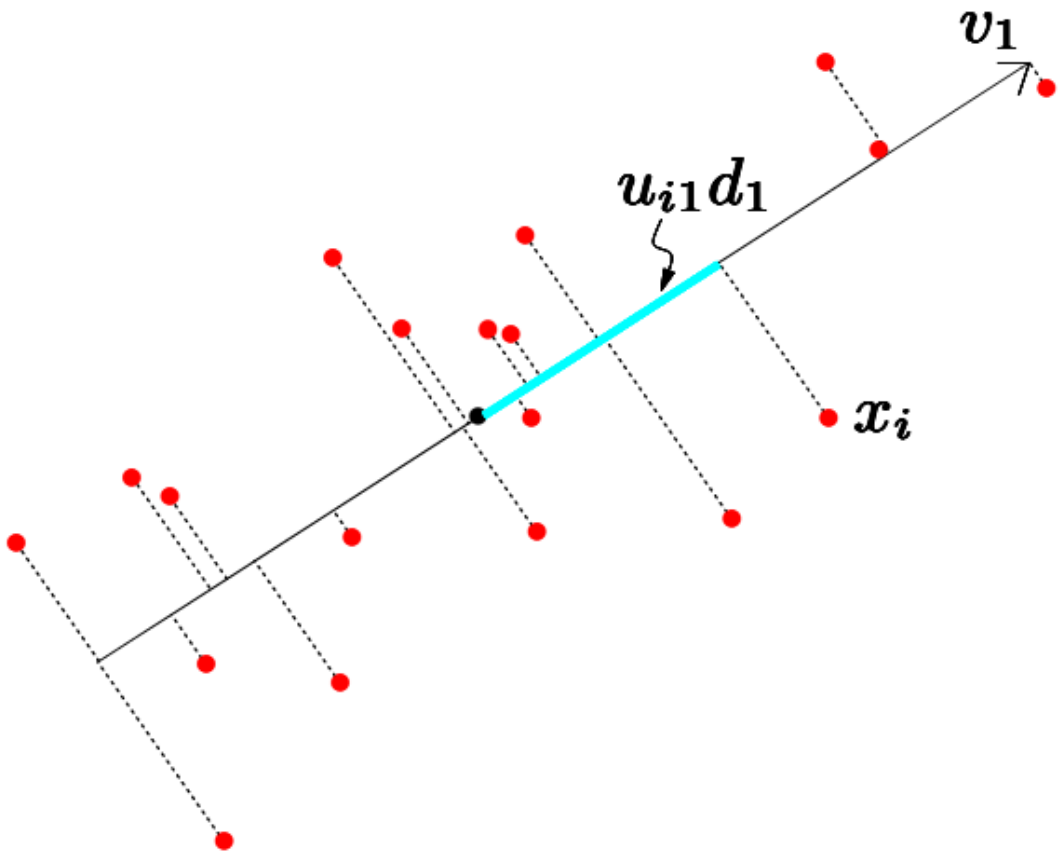
$$\mathbf{W}_{PCA} = \mathbf{E}_{1:K}\mathbf{U}$$

where $\mathbf{E}_{1:K}$ denotes the K dominant eigenvectors of $\mathbf{C_v}$:

$$\mathbf{C_v} = \mathrm{E}\{\mathbf{v}\mathbf{v}^T\} \approx \frac{1}{N}\sum_n \mathbf{v}_n\mathbf{v}_n^T$$

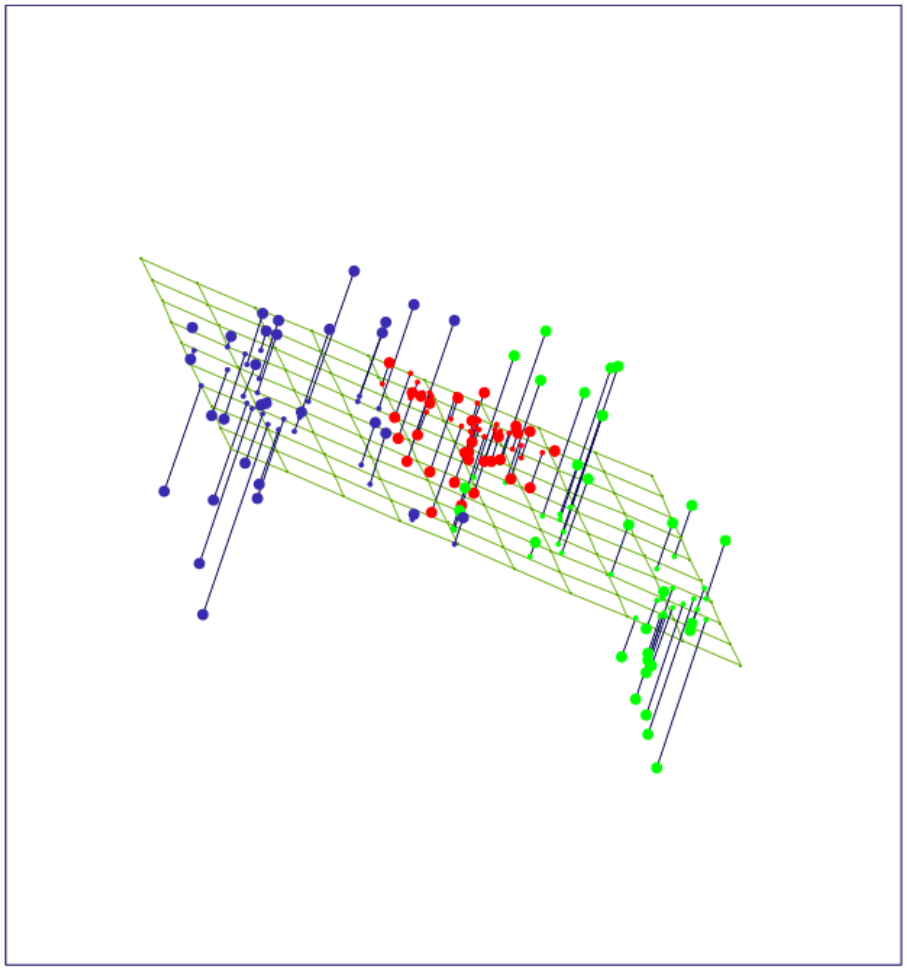
and where $\mathbf{U}$ is any unitary matrix of size $K \times K$.

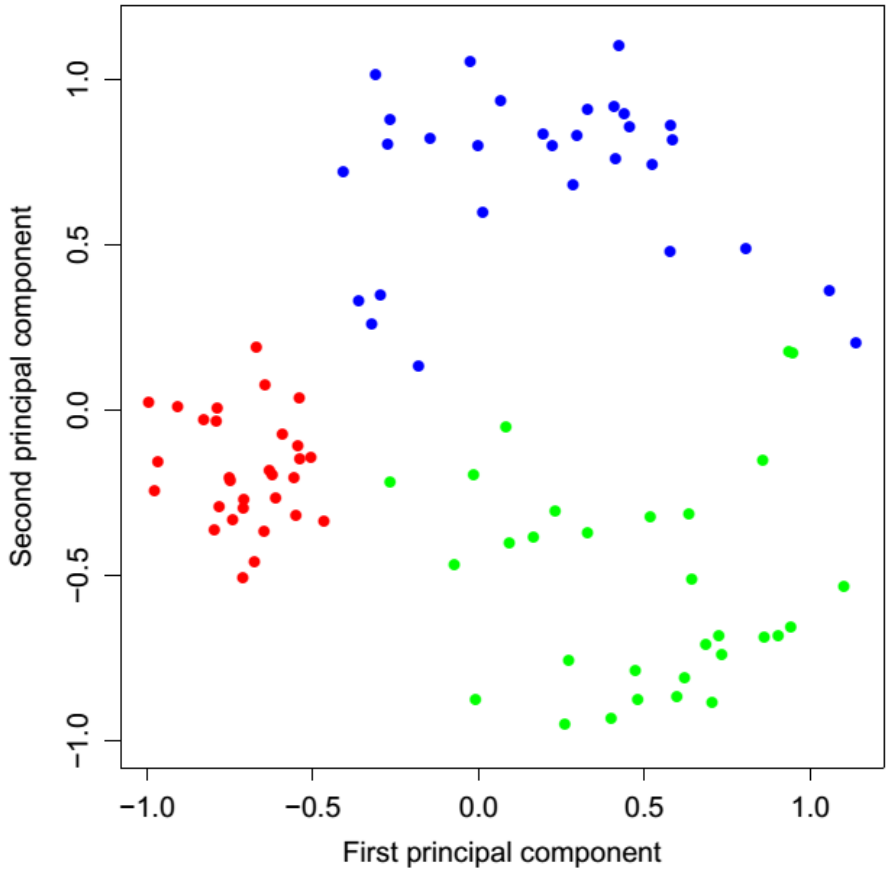
2D data example



After (Hastie *et al.*, 2008)

3D data example





After (Hastie *et al.*, 2008)

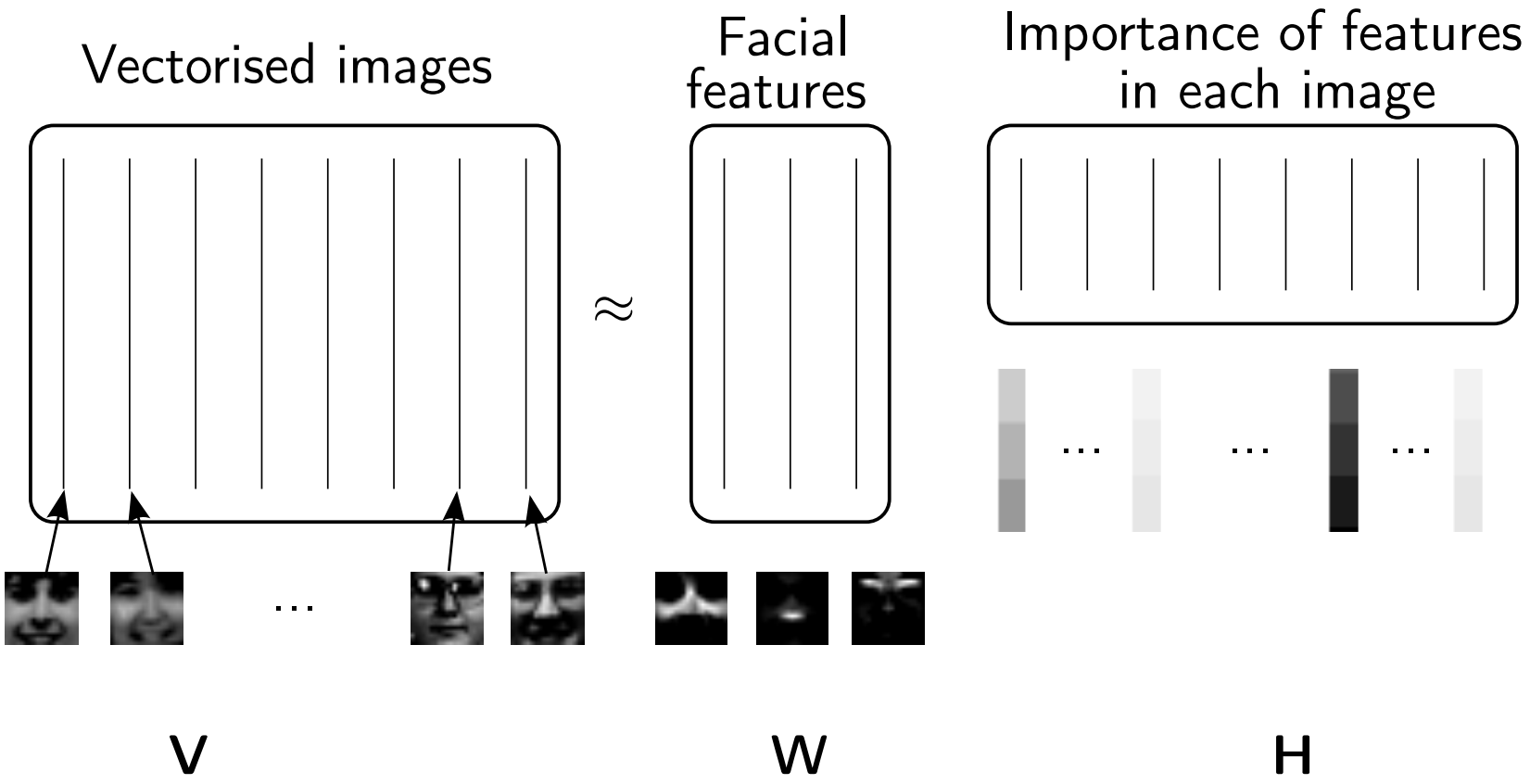
Explaining face images by PCA[1]

Image example: 49 images among 2429 from MIT's CBCL face dataset

[1] slide adapted from (Févotte, 2012).

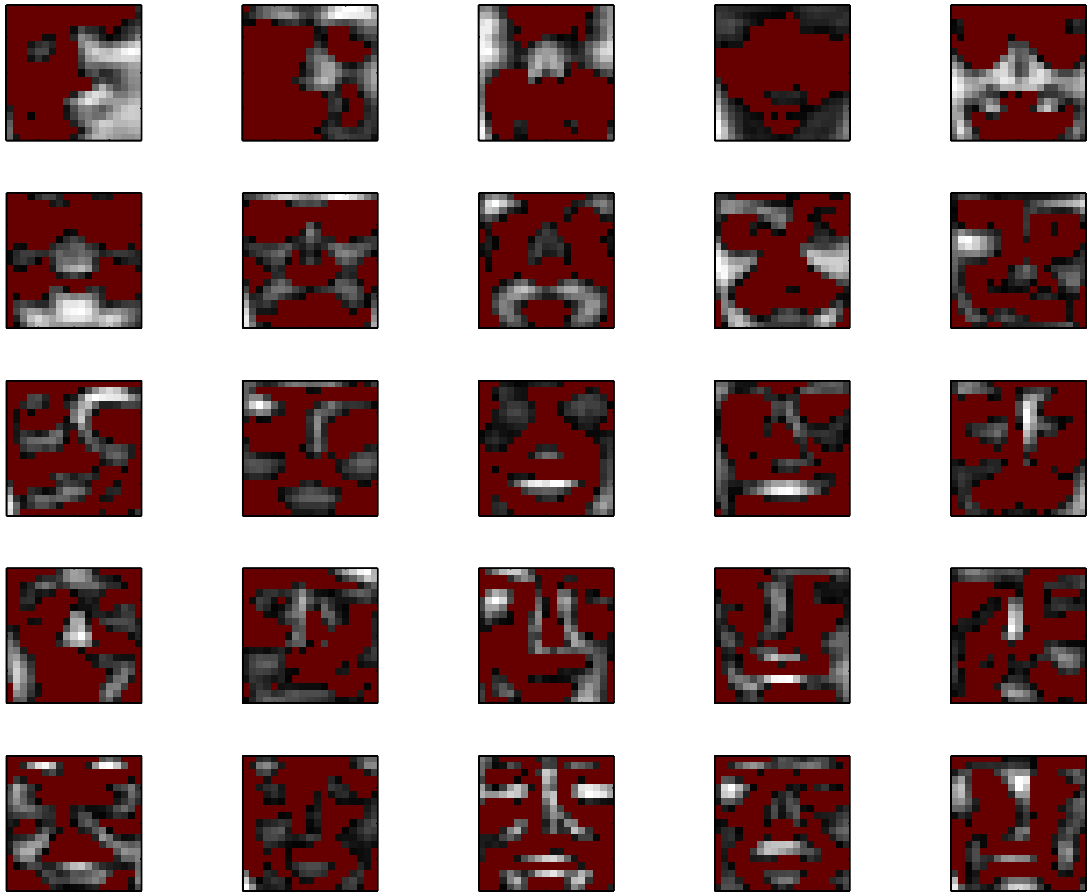
Explaining face images by PCA

Method



Explaining face images by PCA[2]

Eigenfaces



*Red pixels indicate **negative values**! How to interpret this?*

[2] slide adapted from (Févotte, 2012).

Data is often nonnegative by nature[3]

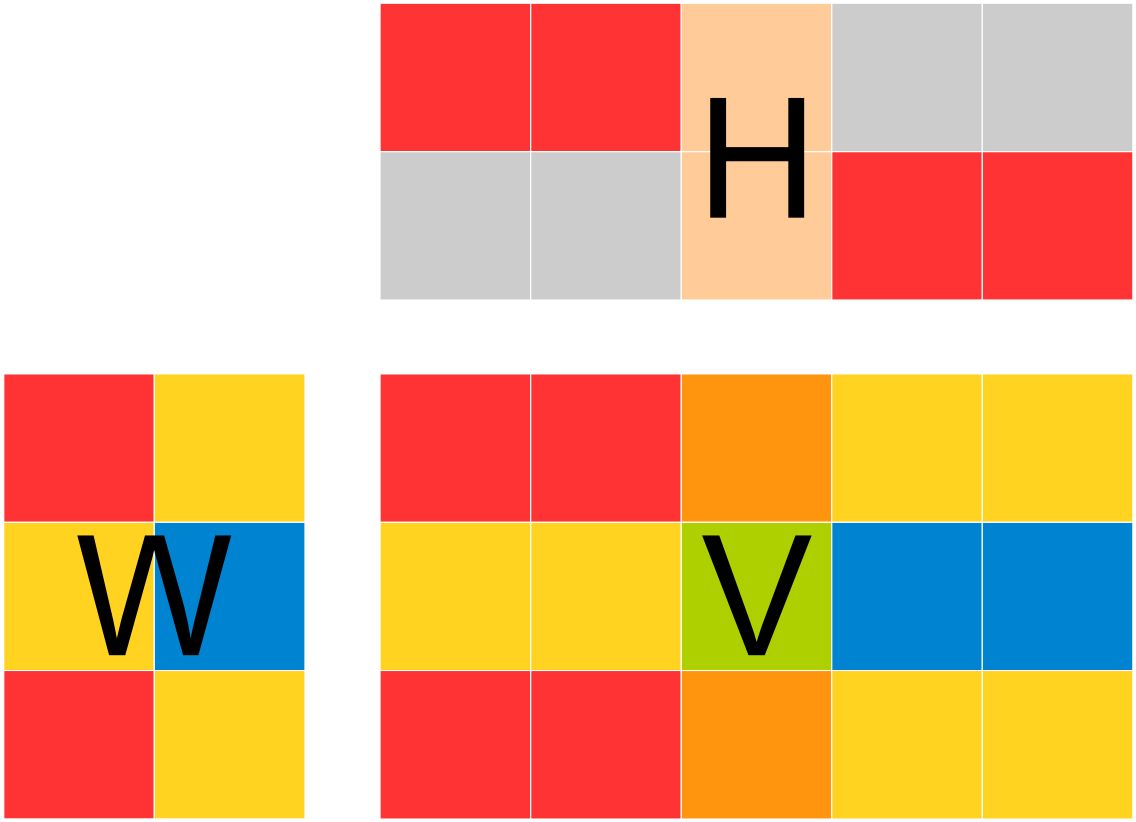
- pixel intensities;
- amplitude spectra;
- occurrence counts;
- food or energy consumption;
- user scores;
- stock market values;
- ...

For the sake of **interpretability** of the results, optimal processing of **nonnegative data** may call for processing under **nonnegativity constraints**.

[3] slide adapted from (Févotte, 2012).

The Nonnegative Matrix Factorisation model

NMF provides an unsupervised linear representation of the data:



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

- $\mathbf{W} = [w_{fk}]$ s.t. $w_{fk} \geq 0$ and
- $\mathbf{H} = [h_{kn}]$ s.t. $h_{kn} \geq 0$.

Illustration by N. Seichepine

Why nonnegative factors?

- Nonnegativity induces **sparsity**.
- Nonnegativity leads to **part-based decompositions**.

"Atoms energy cancellation" is not allowed: once an atom is selected with some energy, it cannot be further concealed by other atoms.

NMF outputs

Image example

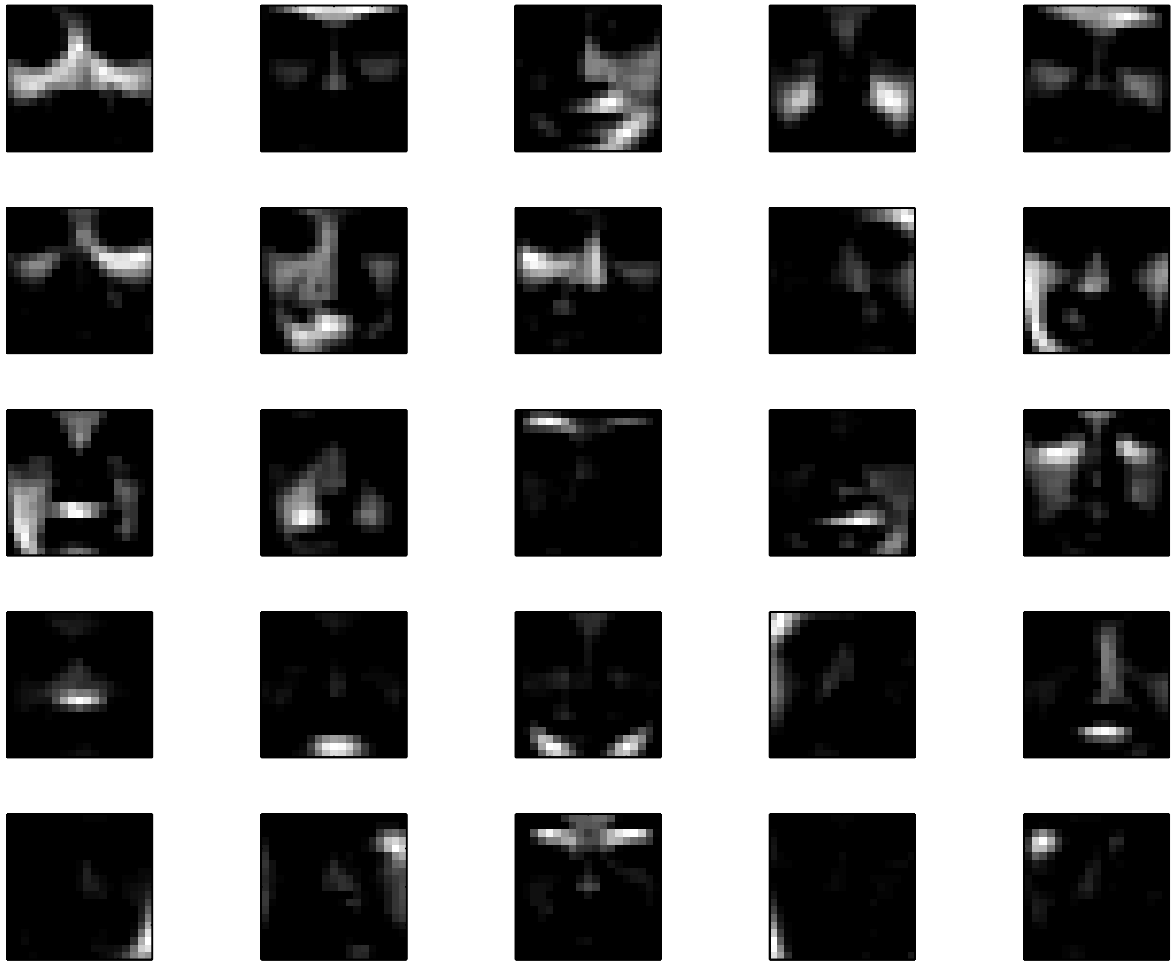


Illustration by C. Févotte

NMF outputs

Audio example

NMF produces **part-based** representations of the data:

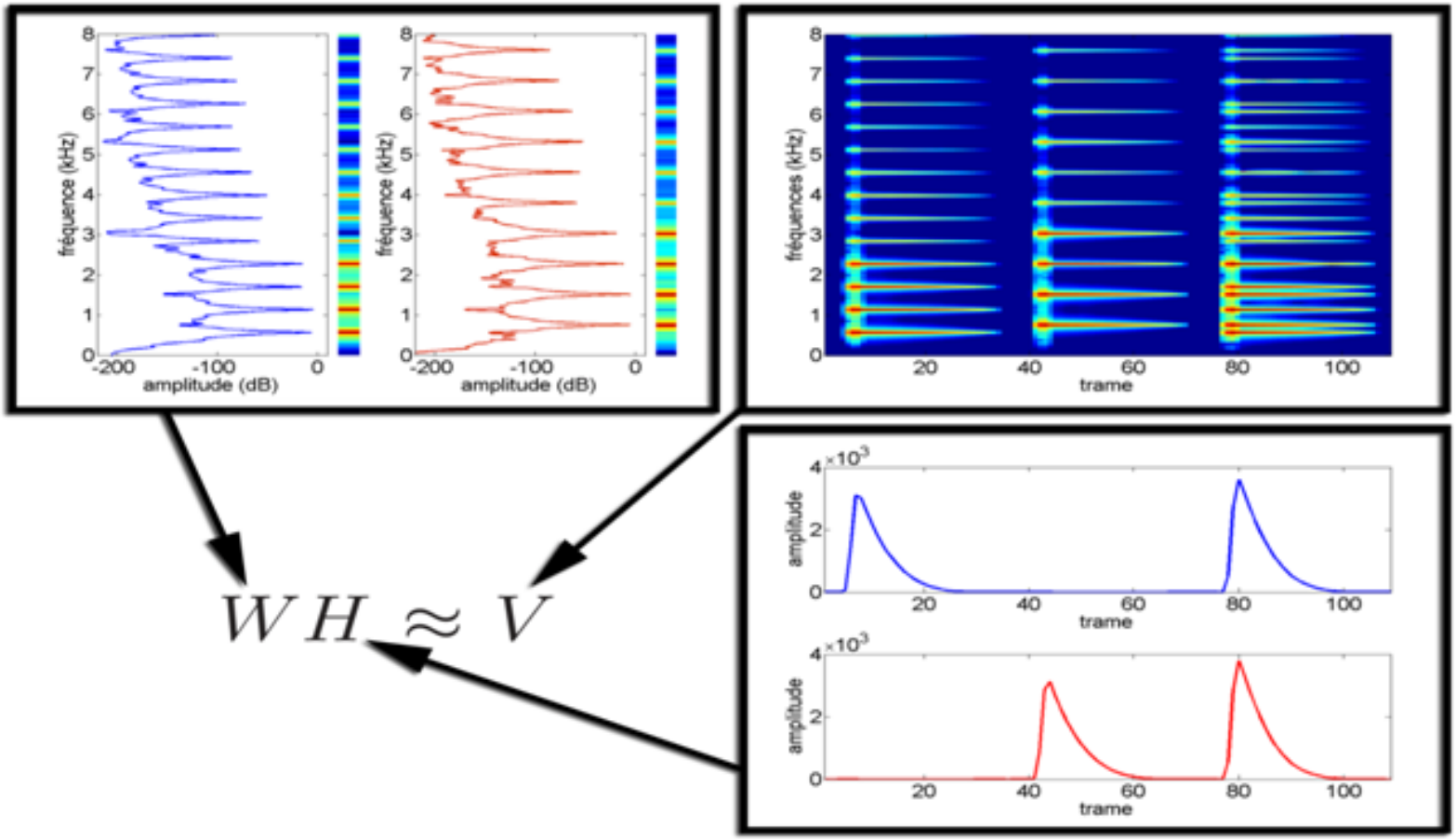


Illustration by R. Hennequin.

History

NMF is more than **30-year** old!

- previous variants referred to as:
  - **nonnegative rank factorisation** (Jeter and Pye, 1981; Chen, 1984);
  - **positive matrix factorisation** (Paatero and Tapper, 1994);
- popularized by Lee and Seung (1999) for "**learning the parts of objects**".

Since then, widely used in various research areas for diverse applications.

Notations I

- $\mathbf{V}$: the $F \times N$ **data matrix**:
  - F features (rows),
  - N observations/examples/feature vectors (columns);
- $\mathbf{v}_n = (v_{1n}, \cdots, v_{Fn})^T$: the n-th **feature vector** observation among a collection of N observations $\mathbf{v}_1, \cdots, \mathbf{v}_N$;
- $\mathbf{v}_n$ is a column vector in $\mathbb{R}_+^F$; $\mathbf{v}_f$ is a row vector;

- $\mathbf{W}$: the $F \times K$ **dictionary matrix**:
  - w_{fk} is one of its coefficients,
  - $\mathbf{w}_k$ a dictionary/basis vector among K elements;

Notations II

- **H** : the $K \times N$ **activation/expansion matrix:**
  - $\mathbf{h}_n$: the **column vector** of activation coefficients for observation $\mathbf{v}_n$:

$$\mathbf{v}_n \approx \sum_{k=1}^{K} h_{kn} \mathbf{w}_k \ ;$$

  - $\mathbf{h}_{k:}$: the **row vector** of activation coefficients relating to basis vector $\mathbf{w}_k$.

General usages of NMF I

What for?

NMF is a non-supervised data decomposition technique, akin to **latent variable analysis**, that can be used for:

- **feature learning**: like Principal Component Analysis (PCA);
  - learn NMF on training dataset $\mathbf{V}_{train} \rightarrow$ dictionary $\mathbf{W}$
  - exploit $\mathbf{W}$ to decompose new test examples $\mathbf{v}_n$: $\mathbf{v}_n \approx \sum_{k=1}^{K} h_{kn}\mathbf{w}_k$; $h_{kn} \geq 0$
  - use $\mathbf{h}_n$ as **feature vector** for example n.

Evaluation for face recognition:

- **Dataset**: Olivetti faces, 40 classes
- **Classifiers**: LDA (Linear Discriminant Analysis)
- **Cross-validated results**:

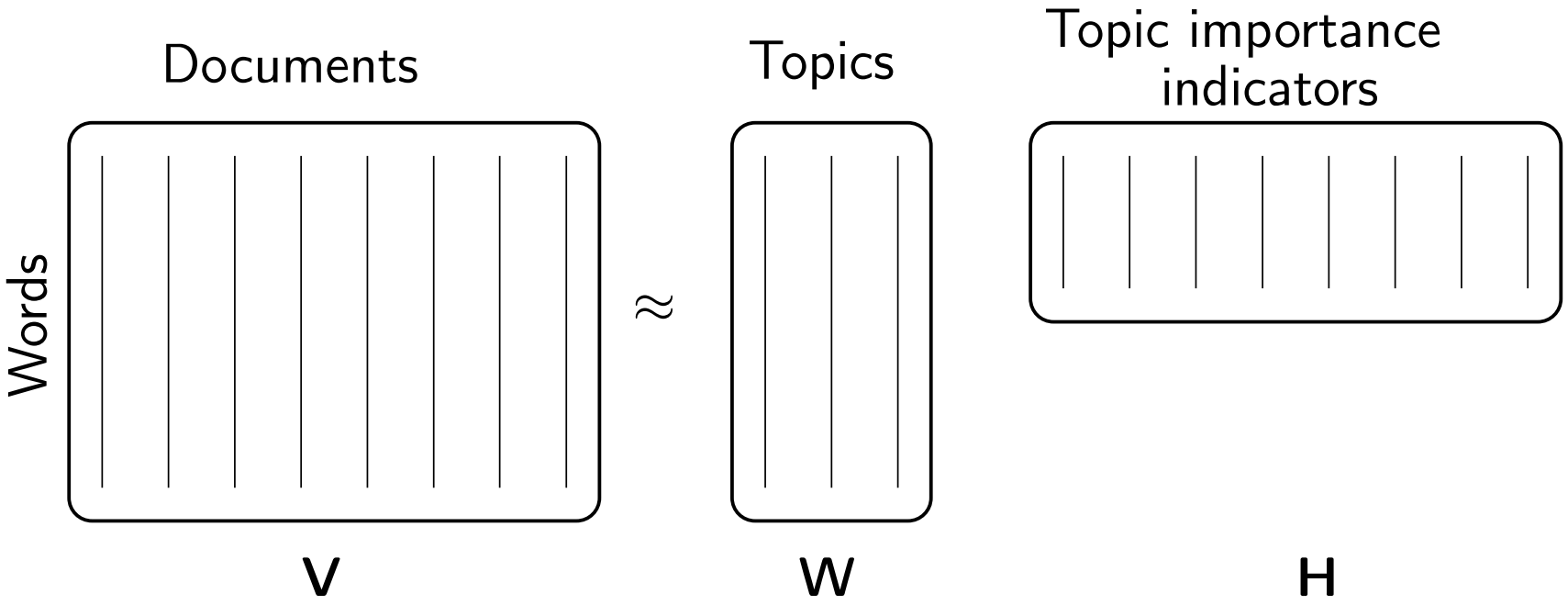
	Accuracy
PCA	93%
ICA	93%
NMF	**96%**

General usages of NMF II

What for?

- **topics recovery**:

  assume $\mathbf{V} = [v_{fn}]$ is a (scaled) **term-document** co-occurrence matrix: v_{fn} is the frequency of occurrences of word m_f in document d_n;



Text document analysis example

After sklearn topics extraction demo (Pedregosa et al., 2011)

Analysing the 20 newsgroups dataset with NMF, the following topics are automatically determined:

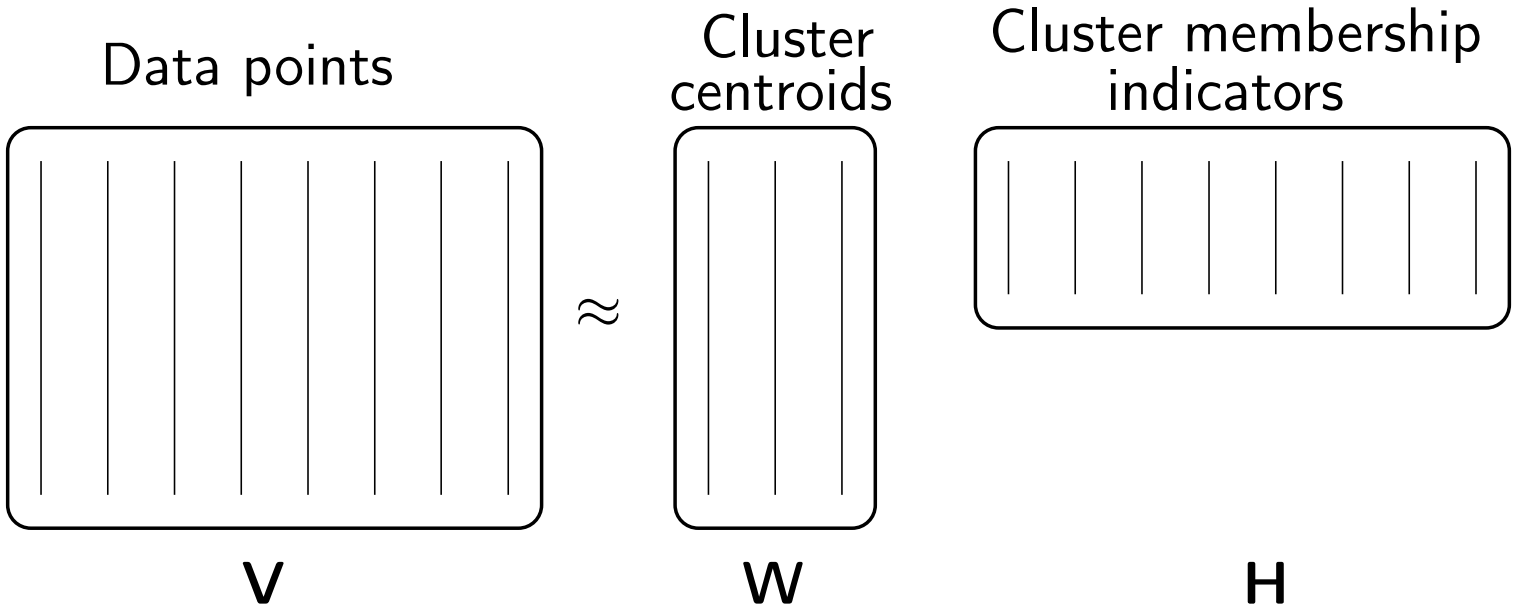
- **Topic #0:** god people bible israel jesus christian true moral think christians believe don say human israeli church life children jewish
- **Topic #1:** drive windows card drivers video scsi software pc thanks vga graphics help disk uni dos file ide controller work
- **Topic #2:** game team nhl games ca hockey players buffalo edu cc year play university teams baseball columbia league player toronto
- **Topic #3:** window manager application mit motif size display widget program xlib windows user color event information use events values
- **Topic #4:** pitt gordon banks cs science pittsburgh univ computer soon disease edu reply pain health david article medical medicine

Topics described by most frequent words in each dictionary element $\mathbf{W}_k$.

General usages of NMF III

What for?

- **clustering**: like K-means (Ding et al., 2005, 2010; Xu et al., 2003):



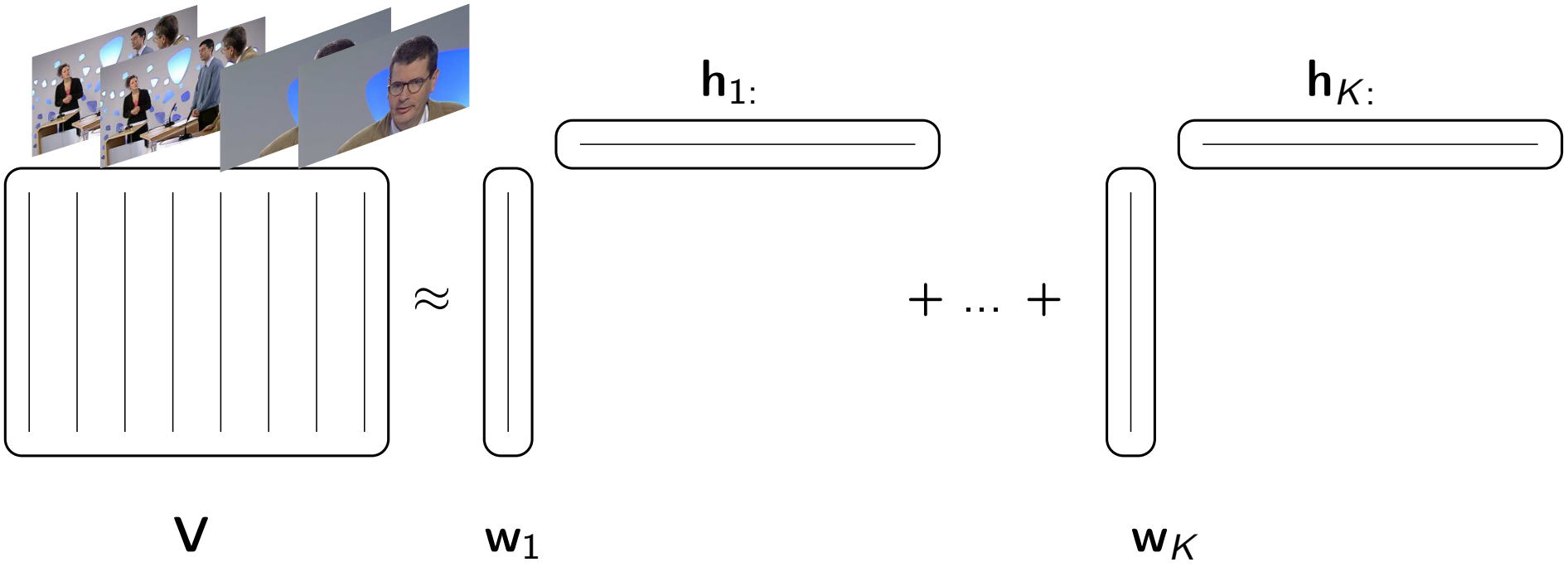
- NMF can handle overlapping clusters and provides *soft* cluster membership indications.

General usages of NMF IV

What for?

- **temporal segmentation**: like Hidden Markov Models (HMM);

  analysing temporal data sequences, *e.g.*, videos:

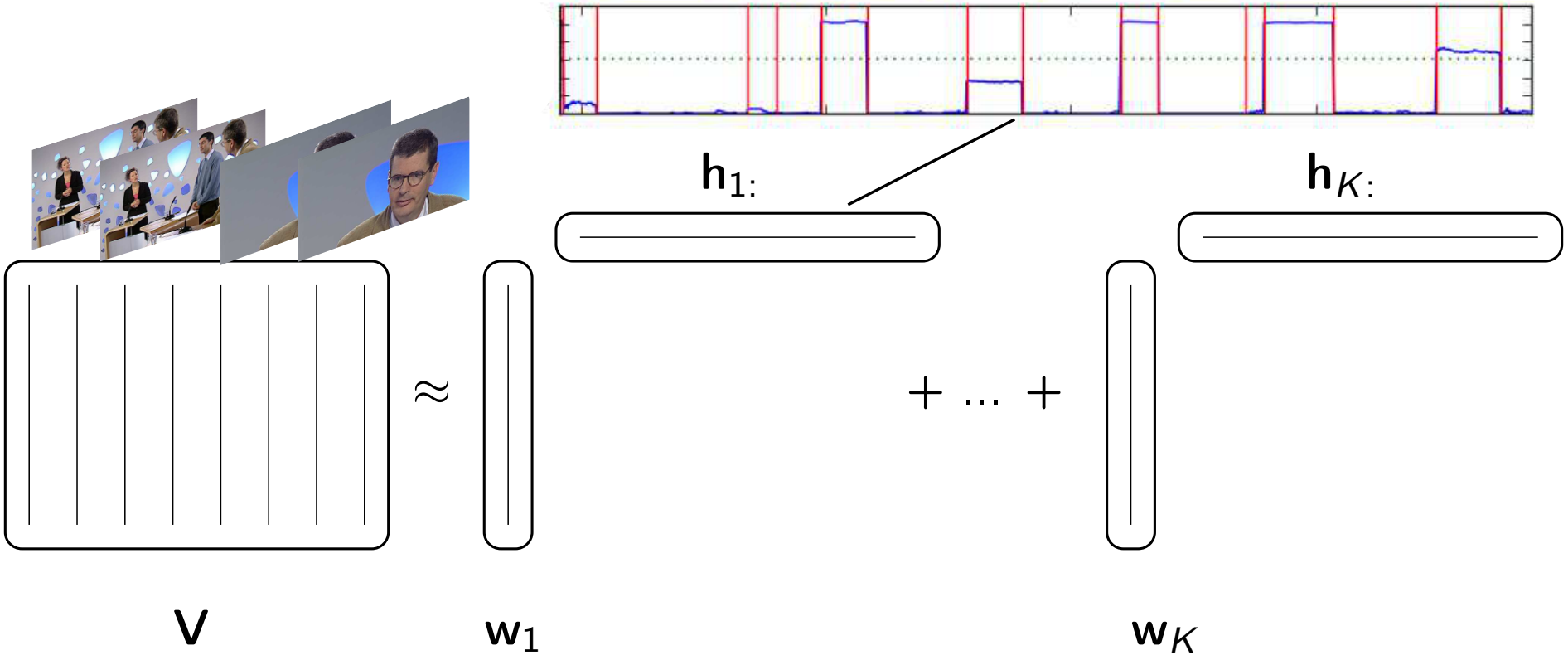


General usages of NMF IV

What for?

- **temporal segmentation**: like Hidden Markov Models (HMM);

  analysing temporal data sequences, *e.g.*, videos:

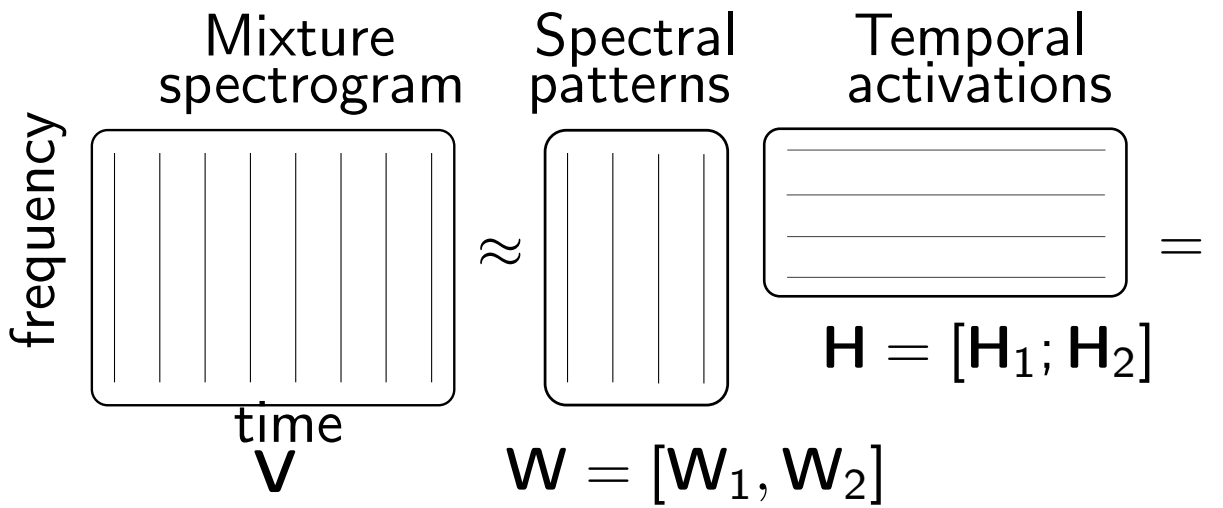


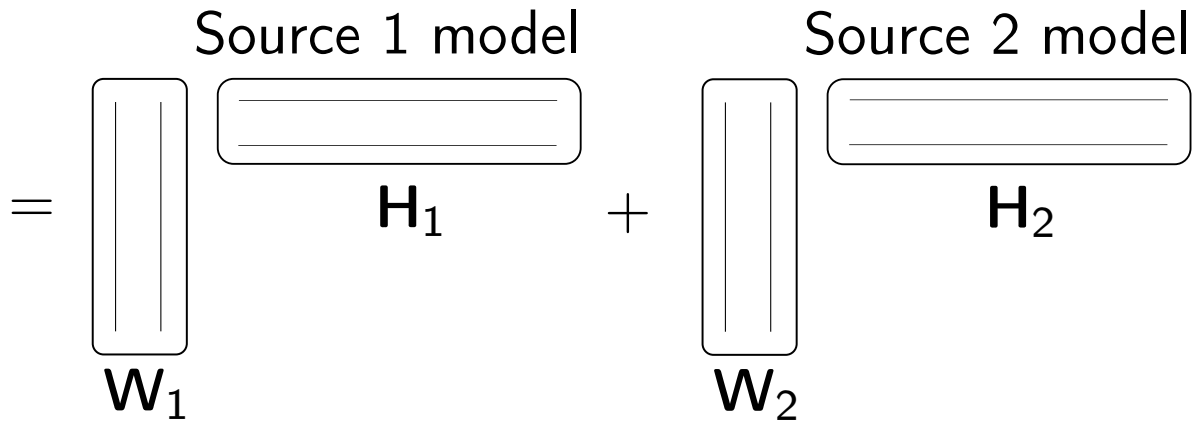
Temporal segmentation can be achieved by thresholding the temporal activations relating to components of interest.

General usages of NMF V

What for?

- **filtering and source separation**: as with Independent Component Analysis (ICA):





In summary...

What for?

NMF is a non-supervised data decomposition technique, akin to **latent variable analysis**, that can be used for:

- **topics recovery**: like Probabilistic Latent Semantic Analysis (PLSA);
- **feature learning**: like Principal Component Analysis (PCA);
- **clustering**: like K-means;
- **temporal segmentation**: like Hidden Markov Models (HMM);
- **filtering and source separation**: as with Independent Component Analysis (ICA);
- **coding** as with vector quantization.

Model order choice

A suitable choice of K is very important

Model order K corresponds to the number of rank-1 matrices within the approximation

The choice of K results in a compromise between

Data fitting

A greater K leads to a better data approximation

Model complexity

A smaller K leads to a less complex model (easier to estimate, less parameters to transmit, etc ...)

A right **model order choice is important** and it depends on the data $\mathbf{V}$ and on the application.

NMF is ill-posed

The solution is not unique

Given $\mathbf{V} = \mathbf{WH}$; $\mathbf{W} \geq 0$, $\mathbf{H} \geq 0$; any matrix $\mathbf{Q}$ such that:

- $\mathbf{WQ} \geq 0$
- $\mathbf{Q}^{-1}\mathbf{H} \geq 0$

provides an alternative factorisation $\mathbf{V} = \tilde{\mathbf{W}}\tilde{\mathbf{H}} = (\mathbf{WQ})(\mathbf{Q}^{-1}\mathbf{H})$.

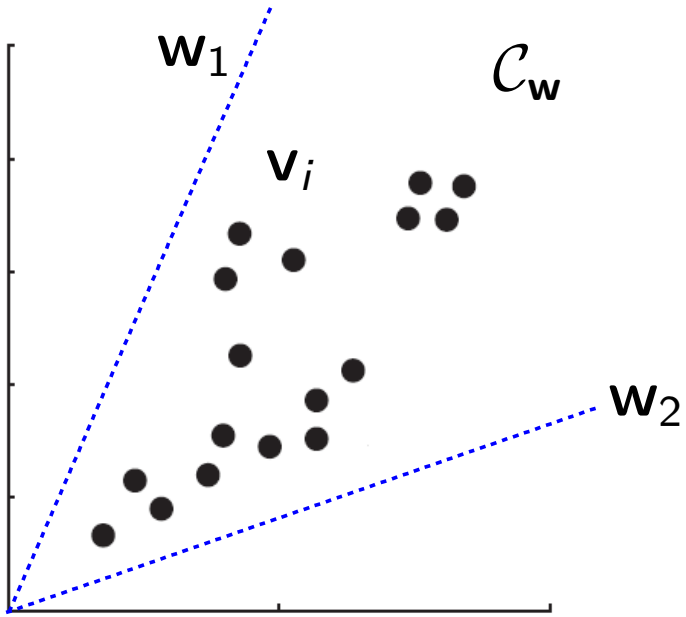
In particular, $\mathbf{Q}$ can be any **nonnegative generalised permutation matrix**; *e.g.*, in $\mathbb{R}^3$:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 3 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

This case is not so problematic: merely accounts for **scaling** and **permutation** of basis vectors $\mathbf{w}_k$.

Geometric interpretation and ill-posedness

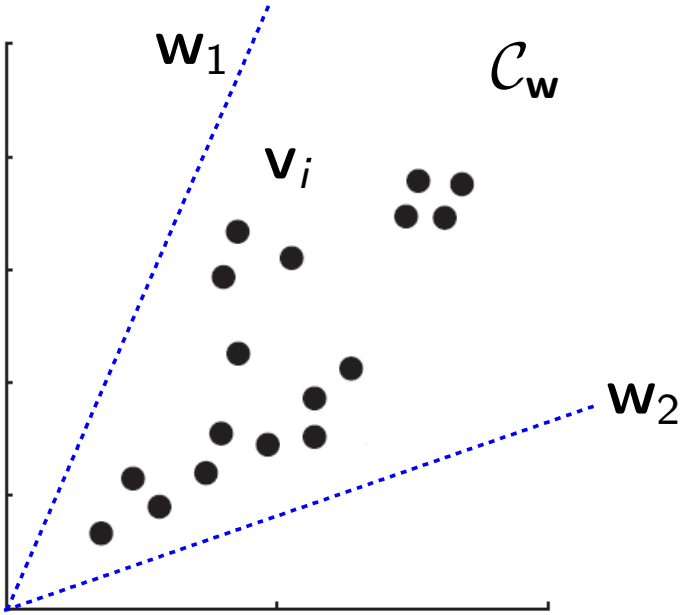
NMF assumes the data is well described by a **simplicial convex cone** $\mathcal{C}_{\mathbf{w}}$ generated by the columns of $\mathbf{W}$:

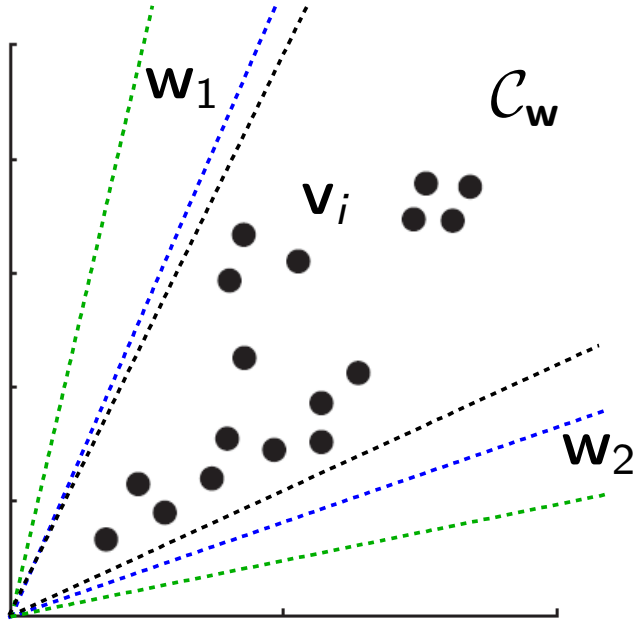


$$\mathcal{C}_{\mathbf{w}} = \left\{ \sum_{k=1}^{K} \lambda_k \mathbf{w}_k;\ \lambda_k \geq 0 \right\}$$

Geometric interpretation and ill-posedness

NMF assumes the data is well described by a **simplicial convex cone** $\mathcal{C}_{\mathbf{w}}$ generated by the columns of $\mathbf{W}$:



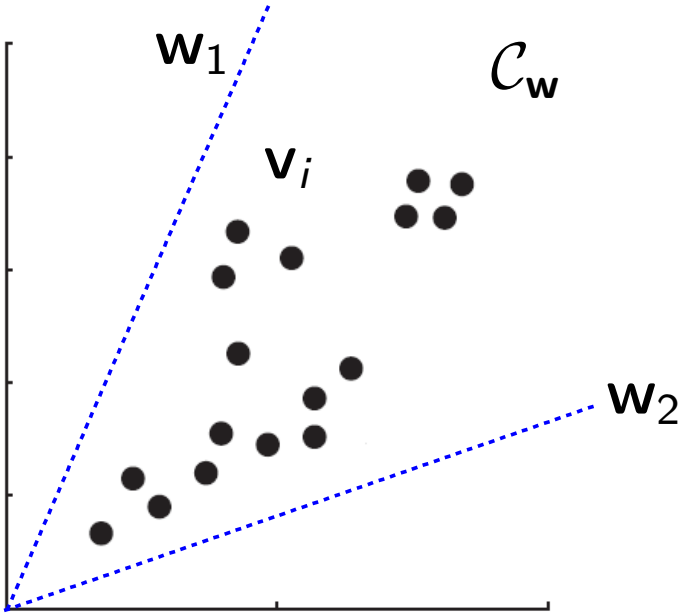


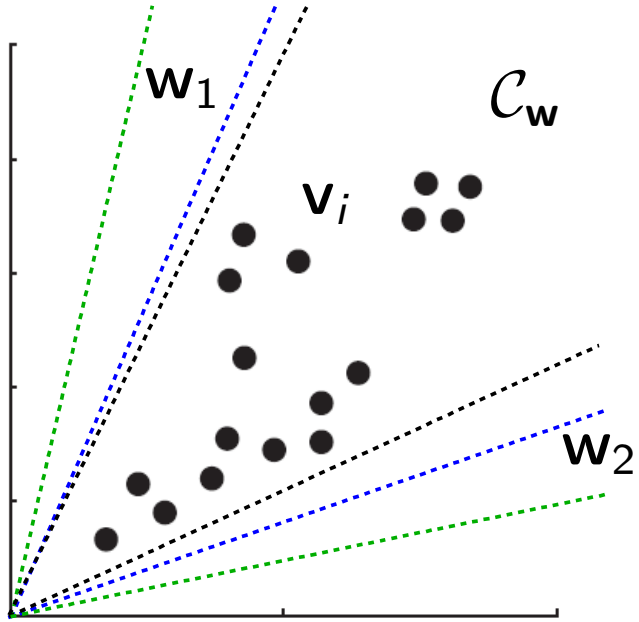
$$\mathcal{C}_{\mathbf{w}} = \left\{ \sum_{k=1}^{K} \lambda_k \mathbf{w}_k;\ \lambda_k \geq 0 \right\}$$

Problem: which $\mathcal{C}_{\mathbf{w}}$?

Geometric interpretation and ill-posedness

NMF assumes the data is well described by a **simplicial convex cone** $\mathcal{C}_{\mathbf{W}}$ generated by the columns of $\mathbf{W}$:





$$\mathcal{C}_{\mathbf{W}} = \left\{ \sum_{k=1}^{K} \lambda_k \mathbf{w}_k ;\ \lambda_k \geq 0 \right\}$$

Problem: which $\mathcal{C}_{\mathbf{W}}$?

→ Need to impose **constraints** on the set of possible solutions to select the most "useful" ones.

Constrained NMF methods

Different types of constraints have been considered in previous works:

- **Sparsity** constraints: either on $\mathbf{W}$ or $\mathbf{H}$ (*e.g.*, Hoyer, 2004; Eggert and Korner, 2004);
- **Shape** constraints on $\mathbf{w}_k$, *e.g.*:
  - **convex NMF**: $\mathbf{w}_k$ are convex combinations of inputs (Ding et al., 2010);
  - **harmonic NMF**: $\mathbf{w}_k$ are mixtures of harmonic spectra (Vincent et al., 2008).
- **Spatial coherence** or **temporal** constraints on $\mathbf{h}_k$: activations are **smooth** (Virtanen, 2007; Jia and Qian, 2009; Essid and Fevotte, 2013);
- **Cross-modal correspondence** constraints: factorisations of related modalities are related, *e.g.*, temporal activations are correlated (Seichepine et al., 2013; Liu et al., 2013; Yilmaz et al., 2011);
- **Geometric** constraints: *e.g.*, select particular cones $\mathcal{C}_{\mathbf{w}}$ (Klingenberg et al., 2009; Essid, 2012).

NMF optimization criteria

NMF approximation $\mathbf{V} \approx \mathbf{WH}$ is usually obtained through:

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

where $D(\mathbf{V}|\widehat{\mathbf{V}})$ is a *separable matrix divergence*:

$$D(\mathbf{V}|\widehat{\mathbf{V}}) = \sum_{f=1}^{F} \sum_{n=1}^{N} d(v_{fn}|\hat{v}_{fn}) \,,$$

and $d(x|y)$ defined for all $x, y \geq 0$ is a *scalar divergence* such that:

- $d(x|y)$ is continuous over x and y;
- $d(x|y) \geq 0$ for all $x, y \geq 0$;
- $d(x|y) = 0$ if and only if $x = y$.

Popular (scalar) divergences

Euclidean (EUC) distance (Lee and Seung, 1999)

$$d_{EUC}(x, y) = (x - y)^2$$

Kullback-Leibler (KL) divergence (Lee and Seung, 1999)

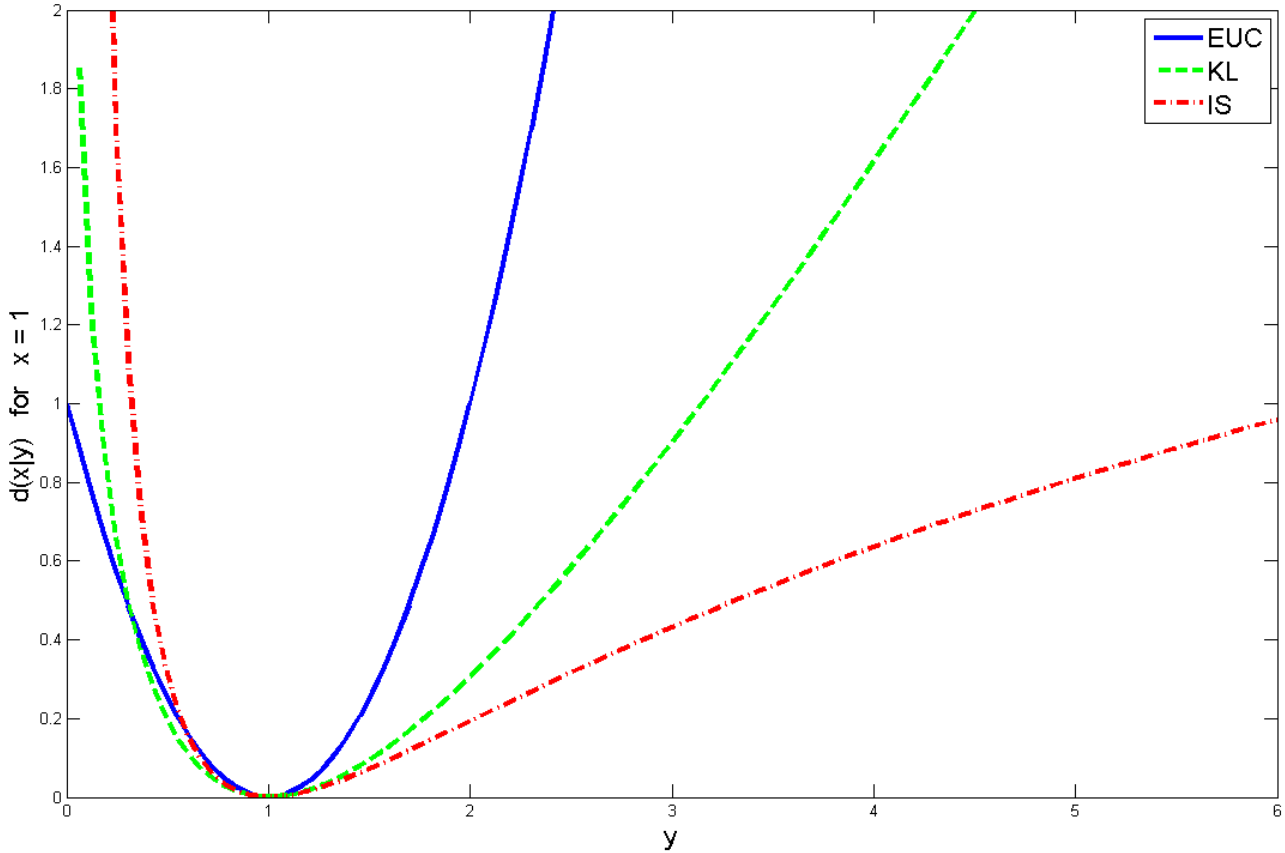
$$d_{KL}(x, y) = x \log \frac{x}{y} - x + y$$

Itakura-Saito (IS) divergence (Févotte et al., 2009)

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

Convexity properties

Divergence $d(x\|y)$	EUC	KL	IS
Convex on x	yes	yes	yes
Convex on y	yes	yes	**no**



Scale invariance properties[4]

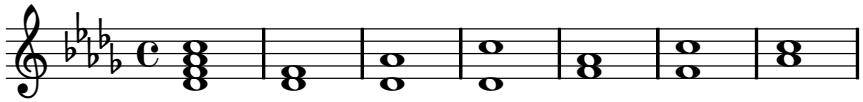
$$
\begin{aligned}
d_{EUC}(\lambda x|\lambda y) &= \lambda^2\, d_{EUC}(x|y) \\
d_{KL}(\lambda x|\lambda y) &= \lambda\, d_{KL}(x|y) \\
d_{IS}(\lambda x|\lambda y) &= d_{IS}(x|y)
\end{aligned}
$$

The IS divergence is **scale-invariant** $\rightarrow$ it provides higher accuracy in the representation of data with large dynamic range, such as audio spectra.

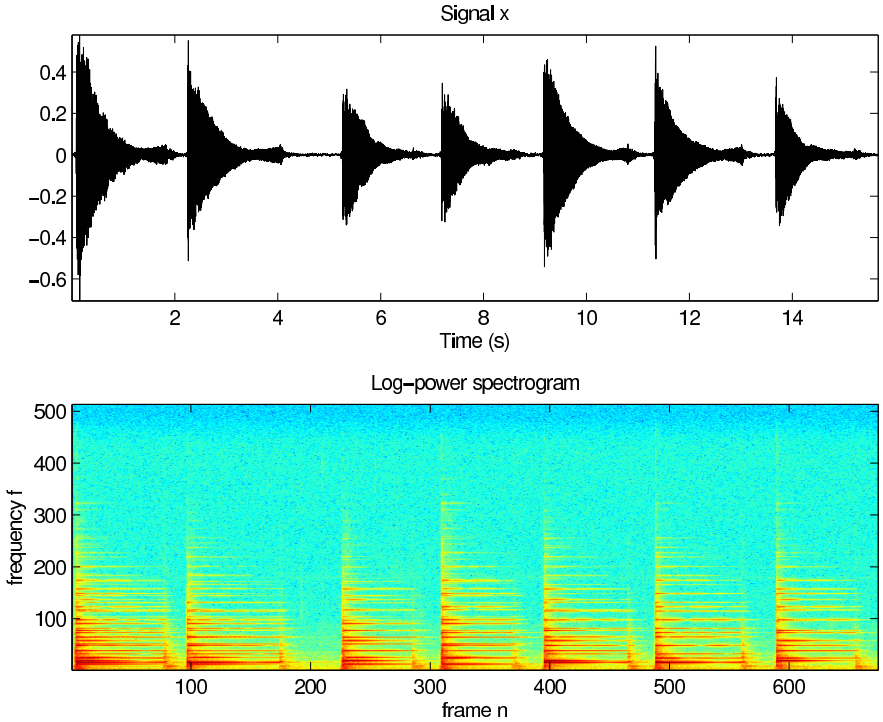
[4] slide adapted from (Févotte, 2012).

Music transcription demo

Demo slide courtesy of C. Févotte (Fevotte et al., 2009)



(MIDI numbers: 61, 65, 68, 72)

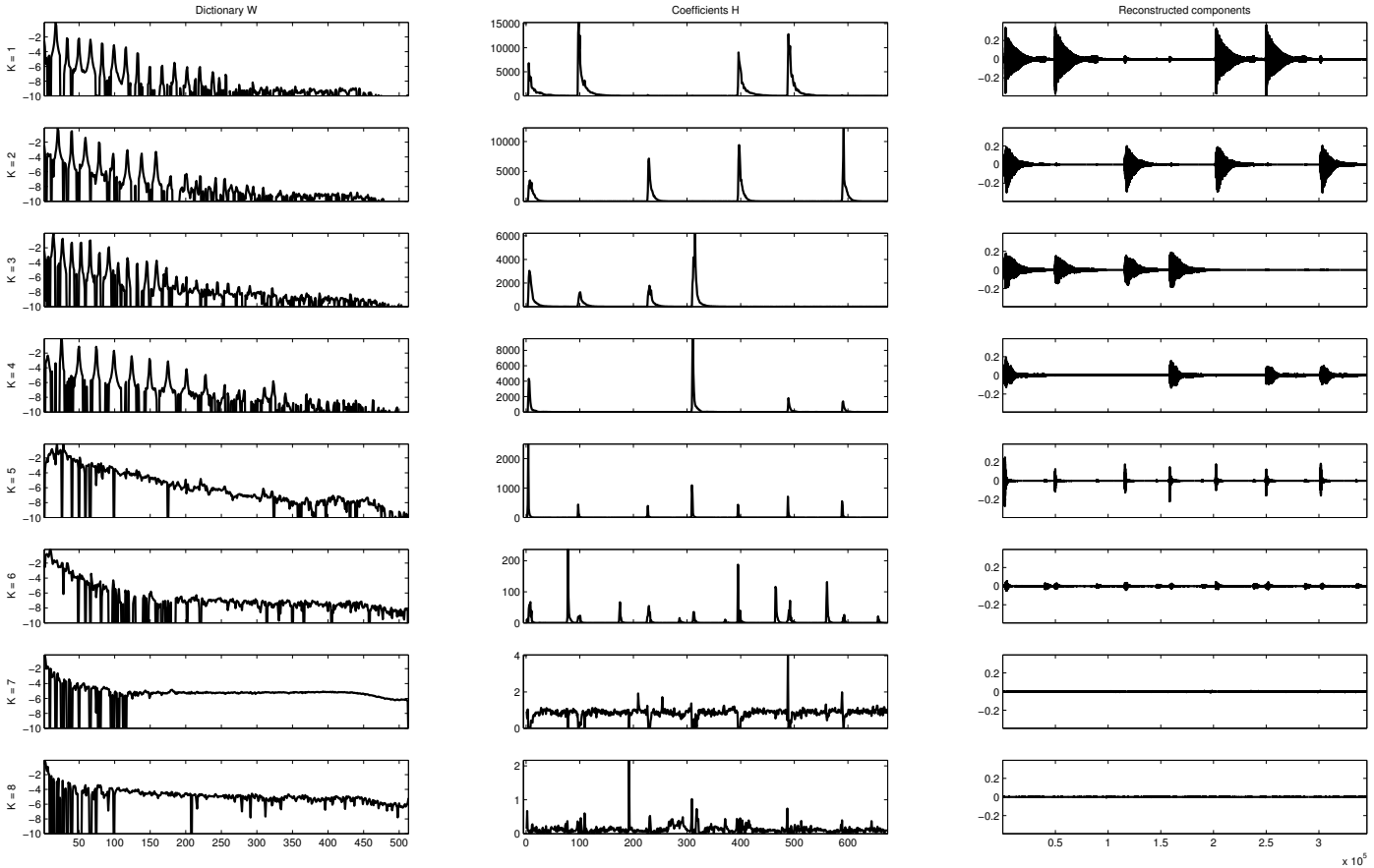


Three representations of the data.

Music transcription demo

Demo slide courtesy of C. Févotte (Fevotte et al., 2009)

NMF decomposition with $K = 8$



Pitch estimates: 65.0 68.0 61.0 72.0 0 0 0 0

(True values: 61, 65, 68, 72)

General parametric families of divergences

β-divergence (Eguchi and Kano., 2001)

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

Generalizes IS ($\beta = 0$), KL ($\beta = 1$) divergences and EUC ($\beta = 2$) distance.

Which divergence to choose?

NMF divergence choice depends on the **data** and on the **application**

One can choose the divergence:

- based on **intuition** or some **prior knowledge on the application goal** (*e.g.*, NMF used for predicting the unseen data while minimizing the mean squared error $\Longrightarrow$ EUC distance) or **invariances** (*e.g.*, scale invariance for audio analysis with IS divergence) ;
- based on some **probabilistic considerations** ;
- so as to **optimize the divergence** (*e.g.* from some parametric family) on some development data within a particular application.

Topics recovery

NMF link to Probabilistic Latent Semantic Analysis (PLSA)

- **Topics recovery** using Probabilistic Latent Semantic Analysis (PLSA):

  assume $\mathbf{V} = [v_{fn}]$ is a (scaled) **term-document** co-occurrence matrix:
  v_{fn} is the frequency of occurrences of word m_f in document d_n;

PLSA model (Hofmann, 1999)

$$P(m_f, d_n) = \sum_{k=1}^{K} P(t_k) P(d_n|t_k) P(m_f|t_k)$$

$\rightarrow$ *the documents can be explained by some underlying topics* t_k.

Topics recovery

NMF link to Probabilistic Latent Semantic Analysis (PLSA)

- Let $w_{fk} = \hat{P}(t_k)\hat{P}(m_f|t_k)$ and $h_{kn} = \hat{P}(d_n|t_k)$;
- the model can be re-written as:

$$[\hat{P}(m_f, d_n)] = [\hat{v}_{fn}] = \mathbf{WH}$$

The $\mathbf{w}_k$ can be interpreted as **topics** explaining the data being analyzed to the extent given by related $\mathbf{h}_{k:}$.

Link between NMF and PLSA (Gaussier and Goutte, 2005)

- Any (local) maximum likelihood solution of PLSA is a solution of NMF with Kullback-Leibler (KL) divergence.
- Any solution of NMF with KL divergence yields a (local) maximum likelihood solution of PLSA.

Weighted NMF

Conventional NMF optimization criterion (separable divergence case):

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \sum_{f=1}^{F}\sum_{n=1}^{N} d(v_{fn}|\hat{v}_{fn})\,.$$

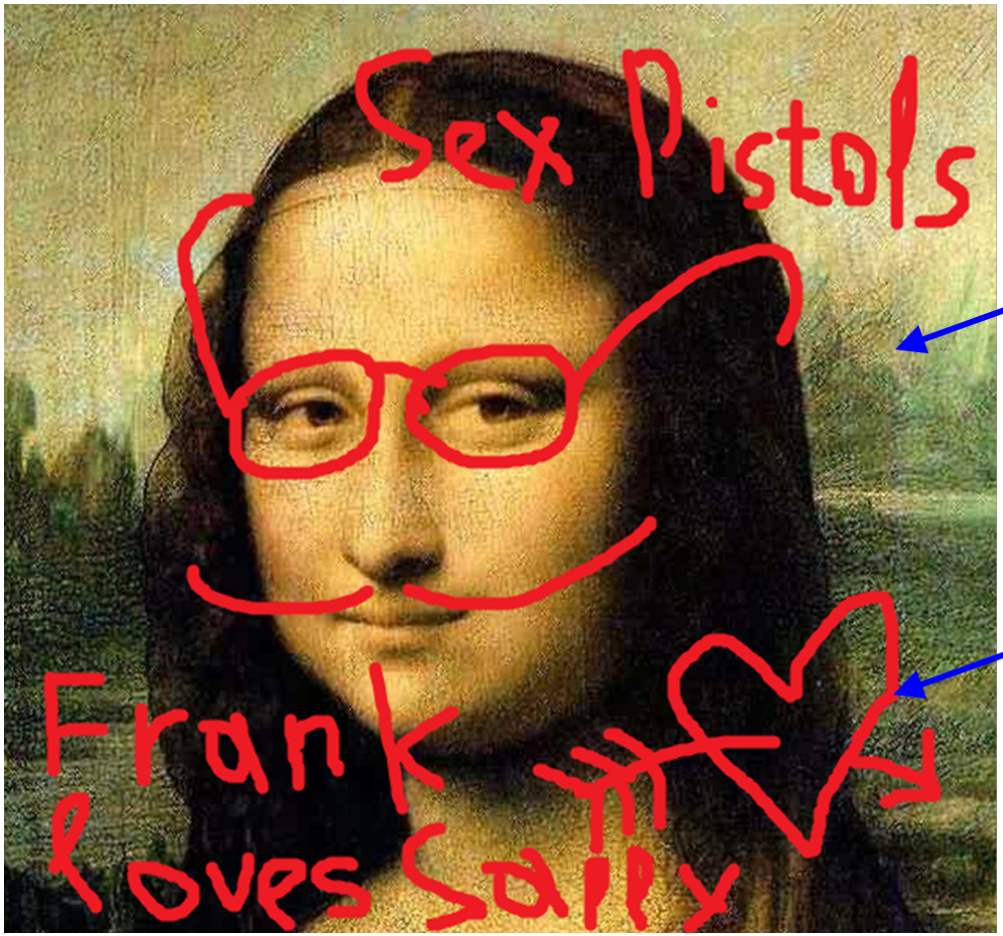
Weighted NMF optimization criterion:

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \sum_{f=1}^{F}\sum_{n=1}^{N} b_{fn}\, d(v_{fn}|\hat{v}_{fn})\,,$$

where b_{fn} $(f = 1, \ldots, F,\ n = 1, \ldots, N)$ are some nonnegative weights representing the contribution of data point v_{fn} into NMF learning.

Weighted NMF application example I

Learning from partial observations (e.g., for **image inpainting** as in (Mairal et al., 2010)):



Observed value

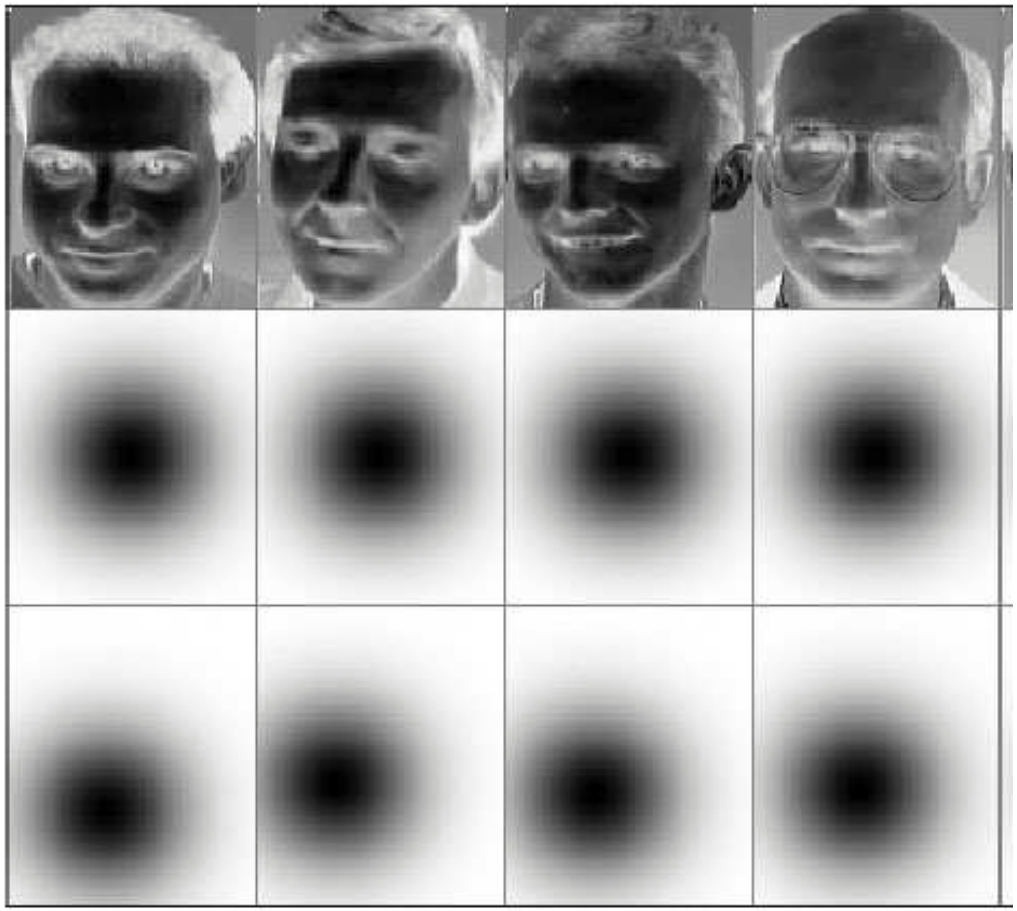
$b_{fn} = 1$

Missing value

$b_{fn} = 0$

Weighted NMF application example II

Face feature extraction (example and figure from (Blondel et al., 2008)):



Data $\mathbf{V}$

Weights $\mathbf{B} = \{b_{fn}\}_{f,n}$

Image-centered weights

Face-centered weights

- Introduction
- Principal Component Analysis (PCA)
- Introduction to NMF
- NMF models
- Algorithms for solving NMF
  - Preliminaries
  - Multiplicative update rules
  - Model order selection, initialization and stopping criteria
- Conclusion

Optimization difficulties

An efficient solution of the NMF optimization problem

$$\min_{\mathbf{W},\mathbf{H}\geq 0} D(\mathbf{V}|\mathbf{WH}) \Leftrightarrow \min_{\boldsymbol{\theta}} C(\boldsymbol{\theta})\,;\;\; C(\boldsymbol{\theta}) \stackrel{\text{def}}{=} D(\mathbf{V}|\mathbf{WH})$$

(where $\boldsymbol{\theta} \stackrel{\text{def}}{=} \{\mathbf{W},\mathbf{H}\}$ denotes the NMF parameters) must cope with the following difficulties:

- the **nonnegativity constraints** must be taken into account;
- **no uniqueness** of the solution is guaranteed in general;
- the optimization problem has usually a **multitude of local and global minima**.

Alternating optimization strategy

The problem is usually easier to optimize over one matrix (say $\mathbf{H}$) given the other matrix (say $\mathbf{W}$) is known and fixed.

Indeed, for several divergences $D(\mathbf{V}|\mathbf{WH})$ is even convex separately w.r.t. $\mathbf{H}$ and w.r.t. $\mathbf{W}$, but not w.r.t. $\{\mathbf{W}, \mathbf{H}\}$.

For this reason many state-of-the-art NMF optimization algorithms rely on the following iterative alternating optimization strategy.

Alternating optimization a.k.a block-coordinate descent (one iteration):

- update $\mathbf{W}$, given $\mathbf{H}$ fixed,
- update $\mathbf{H}$, given $\mathbf{W}$ fixed.

Multiplicative update rules

A heuristic approach introduced by (Lee and Seung, 2001) to solve $\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$

Multiplicative update (MU) rule for $\mathbf{H}$ (similarly for $\mathbf{W}$) is defined as:

$$h_{kn} \leftarrow h_{kn} \left[\nabla_{h_{kn}} C(\boldsymbol{\theta})\right]_- / \left[\nabla_{h_{kn}} C(\boldsymbol{\theta})\right]_+ ,$$

where

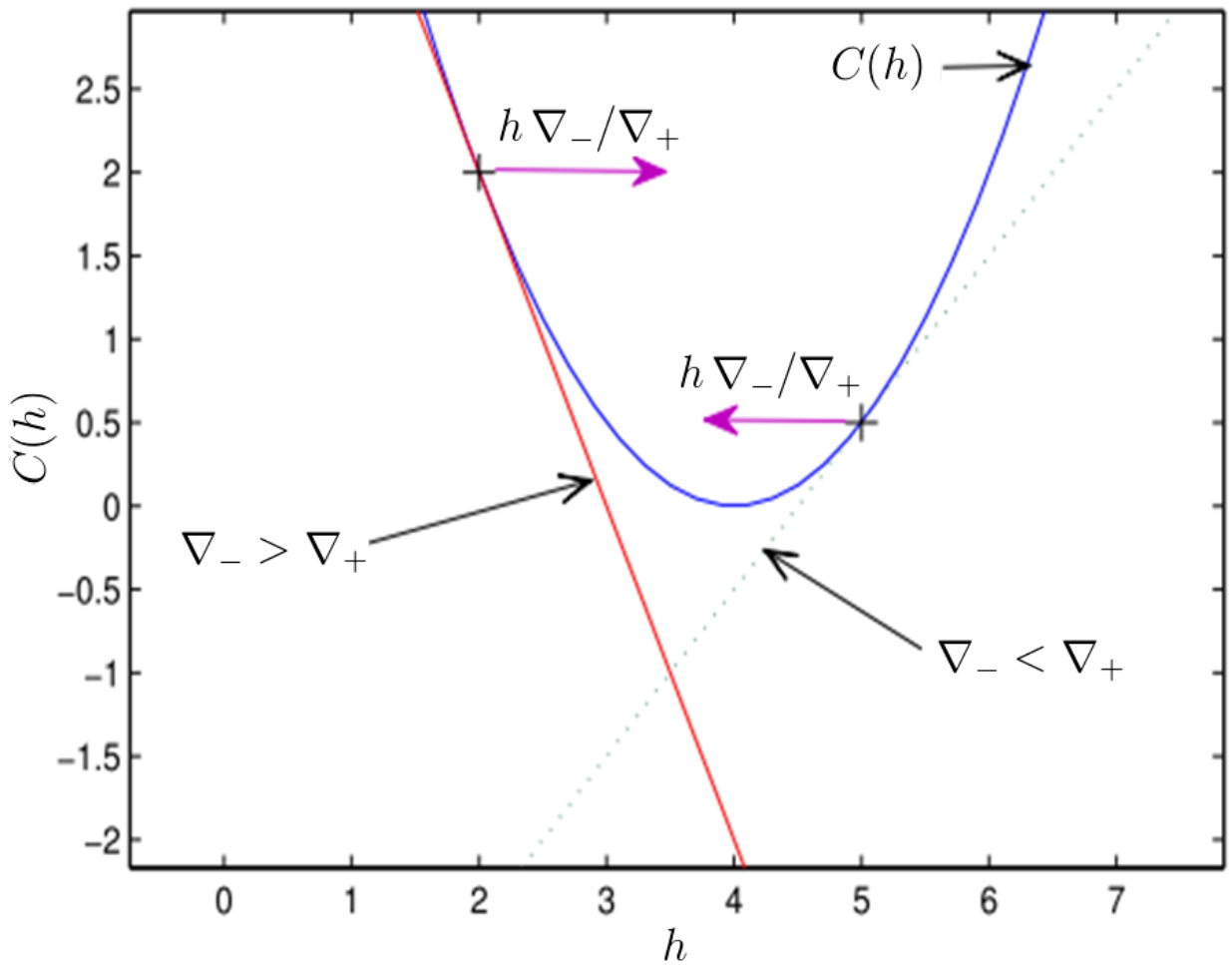
$$\nabla_{h_{kn}} C(\boldsymbol{\theta}) = \left[\nabla_{h_{kn}} C(\boldsymbol{\theta})\right]_+ - \left[\nabla_{h_{kn}} C(\boldsymbol{\theta})\right]_- ,$$

and the summands are both nonnegative.

NOTE: The nonnegativity of $\mathbf{W}$ and $\mathbf{H}$ is guaranteed by construction.

Intuitive explanation

We consider for simplicity $\nabla_h C(h) = \nabla_+ - \nabla_-$



MU rules for the β-divergence

For example, in the case of the β-divergence (generalizing the three popular divergences) the following decomposition:

$$\nabla_y d_\beta(x|y) = \underbrace{y^{\beta-1}}_{[\nabla_y d_\beta(x|y)]_+} - \underbrace{x\, y^{\beta-2}}_{[\nabla_y d_\beta(x|y)]_-}$$

leads to the following MU rules (in matrix form) (Févotte et al., 2009):

MU rules for NMF with the β-divergence (one iteration):

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

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

Re-normalize $\mathbf{W}$ columns and $\mathbf{H}$ rows to address scale-invariance (see Févotte et al. 2009).

Discussion

The only two things guaranteed by this approach:

- the newly updated value lies in the **direction of partial derivative decrease**;
- the newly updated value is **always nonnegative**.

Nothing more can be guaranteed in general, and all other algorithm's properties depend on the **"positive-negative" decomposition chosen:**

$$\nabla_{h_{kn}} C(\boldsymbol{\theta}) = [\nabla_{h_{kn}} C(\boldsymbol{\theta})]_+ - [\nabla_{h_{kn}} C(\boldsymbol{\theta})]_- \ .$$

Gradient descent viewpoint

Each MU rule can be interpreted as a **diagonally rescaled gradient descent** (Lee and Seung, 2001):

$$h_{kn} \leftarrow h_{kn} - \mu_{kn} \nabla_{h_{kn}} C(\boldsymbol{\theta}) \,,$$

where the step-size μ_{kn} is defined as $\mu_{kn} \stackrel{\Delta}{=} h_{kn} / \left[\nabla_{h_{kn}} C(\boldsymbol{\theta})\right]_+$.

Though this re-formulation does not bring any new properties for the algorithm (*e.g.*, the convergence).

Majorisation-minimisation viewpoint

For **many divergences** and **certain "positive-negative" decompositions** each MU rule can be interpreted as a **Majorisation-Minimisation (MM)** procedure (Hunter and Lange, 2004):

To minimise $C(s)$, *e.g.*, $s = w_{fk}$ or $s = h_{kn}$:

- build $G(s|\tilde{s})$ such that $G(s|\tilde{s}) \geq C(s)$ and $G(\tilde{s}|\tilde{s}) = C(\tilde{s})$;
- optimize iteratively $G(s|\tilde{s})$ instead of $C(s)$.

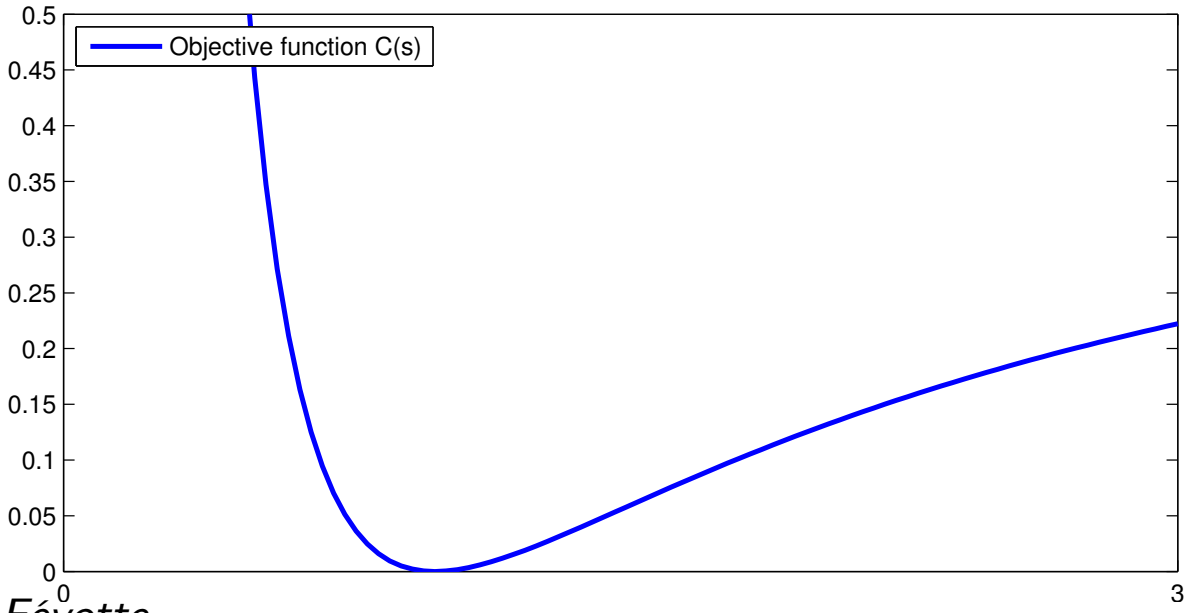


Illustration by C. Févotte

Majorisation-minimisation viewpoint

For **many divergences** and **certain "positive-negative" decompositions** each MU rule can be interpreted as a **Majorisation-Minimisation (MM)** procedure (Hunter and Lange, 2004):

To minimise $C(s)$, *e.g.*, $s = w_{fk}$ or $s = h_{kn}$:

- build $G(s|\tilde{s})$ such that $G(s|\tilde{s}) \geq C(s)$ and $G(\tilde{s}|\tilde{s}) = C(\tilde{s})$;
- optimize iteratively $G(s|\tilde{s})$ instead of $C(s)$.

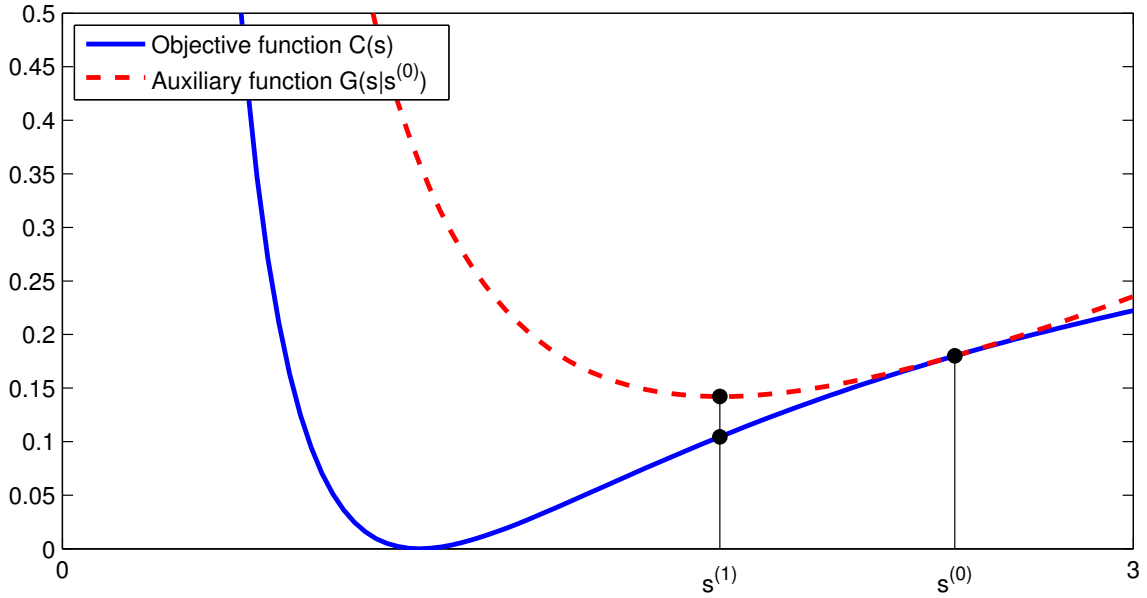


Illustration by C. Févotte

Majorisation-minimisation viewpoint

For **many divergences** and **certain "positive-negative" decompositions** each MU rule can be interpreted as a **Majorisation-Minimisation (MM)** procedure (Hunter and Lange, 2004):

To minimise $C(s)$, *e.g.*, $s = w_{fk}$ or $s = h_{kn}$:

- build $G(s|\tilde{s})$ such that $G(s|\tilde{s}) \geq C(s)$ and $G(\tilde{s}|\tilde{s}) = C(\tilde{s})$;
- optimize iteratively $G(s|\tilde{s})$ instead of $C(s)$.

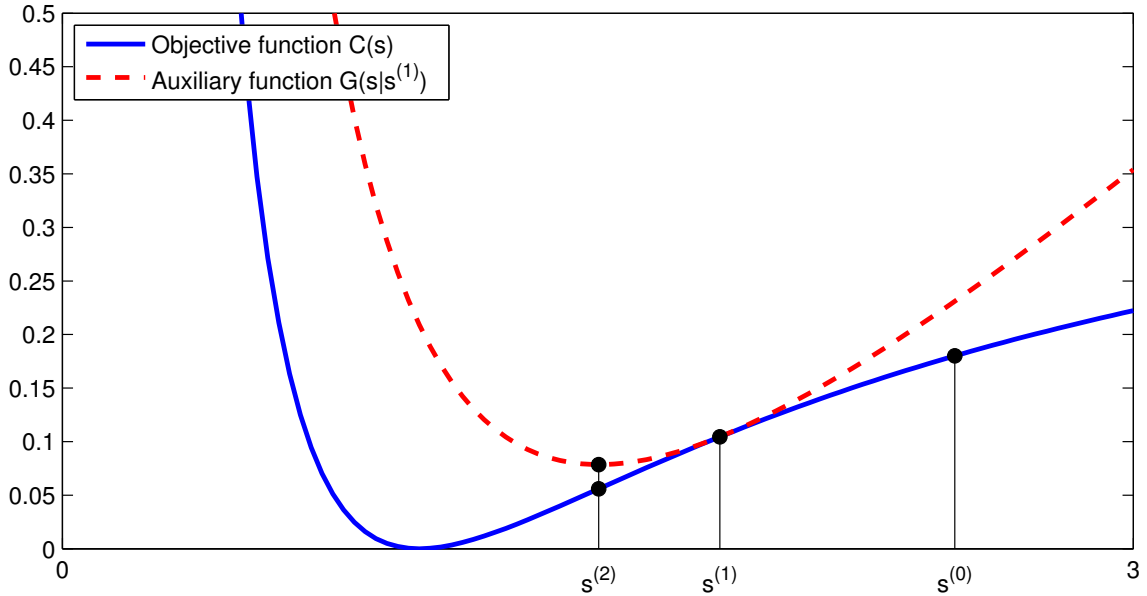


Illustration by C. Févotte

Majorisation-minimisation viewpoint

For **many divergences** and **certain "positive-negative" decompositions** each MU rule can be interpreted as a **Majorisation-Minimisation (MM)** procedure (Hunter and Lange, 2004):

To minimise $C(s)$, *e.g.*, $s = w_{fk}$ or $s = h_{kn}$:

- build $G(s|\tilde{s})$ such that $G(s|\tilde{s}) \geq C(s)$ and $G(\tilde{s}|\tilde{s}) = C(\tilde{s})$;
- optimize iteratively $G(s|\tilde{s})$ instead of $C(s)$.

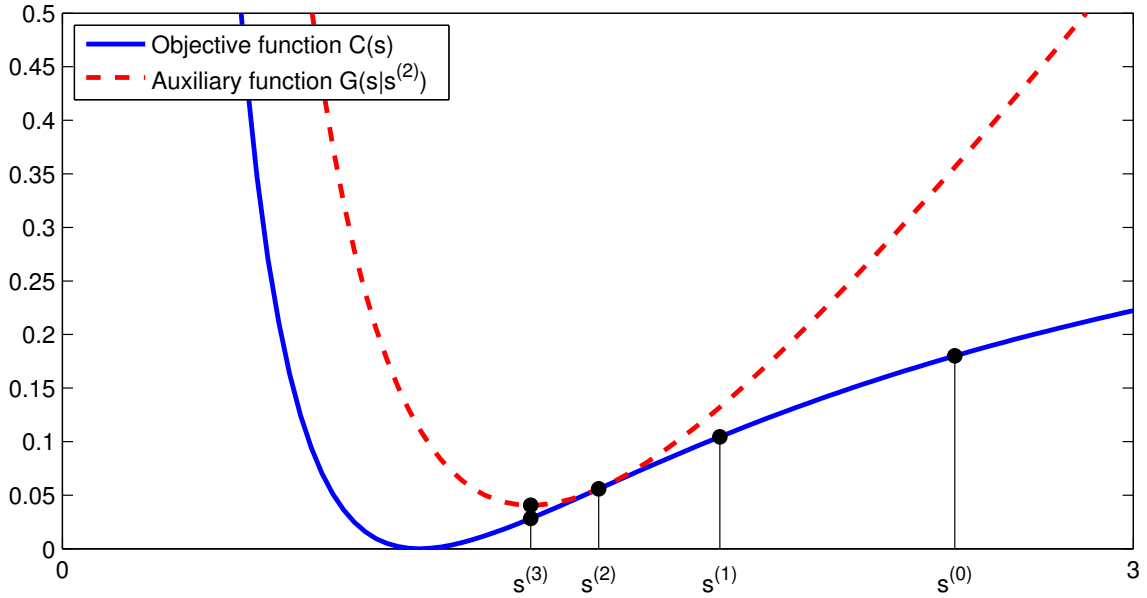


Illustration by C. Févotte

Majorisation-minimisation viewpoint

For **many divergences** and **certain "positive-negative" decompositions** each MU rule can be interpreted as a **Majorisation-Minimisation (MM)** procedure (Hunter and Lange, 2004):

To minimise $C(s)$, *e.g.*, $s = w_{fk}$ or $s = h_{kn}$:

- build $G(s|\tilde{s})$ such that $G(s|\tilde{s}) \geq C(s)$ and $G(\tilde{s}|\tilde{s}) = C(\tilde{s})$;
- optimize iteratively $G(s|\tilde{s})$ instead of $C(s)$.

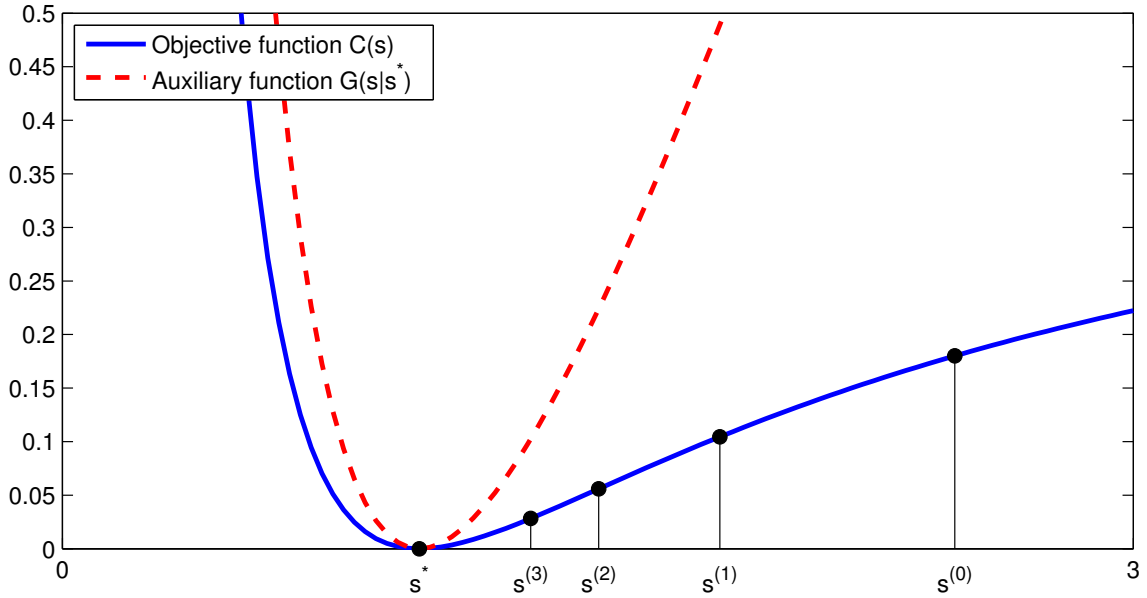


Illustration by C. Févotte

Majorisation-minimisation viewpoint

For **many divergences** and **certain "positive-negative" decompositions** each MU rule can be interpreted as a **Majorisation-Minimisation (MM)** procedure (Hunter and Lange, 2004):

To minimise $C(s)$, *e.g.*, $s = w_{fk}$ or $s = h_{kn}$:

- build $G(s|\tilde{s})$ such that $G(s|\tilde{s}) \geq C(s)$ and $G(\tilde{s}|\tilde{s}) = C(\tilde{s})$;
- optimize iteratively $G(s|\tilde{s})$ instead of $C(s)$.

- **NOTE:** The MM procedure guarantees the cost is non-increasing at each iteration:

$$C(s^{(t+1)}) \leq G(s^{(t+1)}|s^{(t)}) \leq G(s^{(t)}|s^{(t)}) = C(s^{(t)}).$$

Convergence analysis

Monotonicity ("convergence" in terms of **non-increase** of the cost):

- is not guaranteed in general for MU rules;
- is proven (via the majorisation-minimisation formulation) for some divergences (*e.g.*, α and β-divergences) with particular "positive-negative" decompositions (see, *e.g.*, Févotte and Idier 2010; Yang and Oja 2011).

Local convergence in parameters (whether the solution converges to a stationary point?)

- very few positive results for MU rules (see, e.g., Lin 2007a; Badeau et al. 2010);
- the main difficulty is due to non-uniqueness of the NMF.

Summary

Advantages:

- easy to implement;
- non-negativity of **W** and **H** is guaranteed.

Drawbacks:

- monotonicity is not always guaranteed;
- among other algorithms the convergence rate is not the highest one.

Other alternating optimization algorithms

Gradient-like algorithms (Lin, 2007b)

- Advantages: may "**converge**" **faster** than MU rules
- Drawbacks: nonnegativity **constraints must be explicitly handled.**

Newton-like algorithms (Zdunek and Cichocki, 2006)

- Advantages: "**converge**" **faster** than Gradient-like algos and MU rules
- Drawbacks: nonnegativity **constraints must be explicitly handled**; limited to **convex divergences**

Expectation-maximization (EM) algorithms (Févotte et al., 2009; Cemgil, 2009)

- Advantages: nonnegativity **constraints are implicitly handled**; possibility of **introducing other constraints** via probabilistic priors
- Drawbacks: may "**converge**" **slower** than MU rules; limited to NMF with **probabilistic formulation**

Online algorithms

Online algorithms to handle **continuous data streams** (Bucak and Gunsel, 2009; Simon and Vincent, 2012)

Online algorithms to handle **big data** (stochastic gradient-like) (Mairal et al., 2010)

How to choose model order?

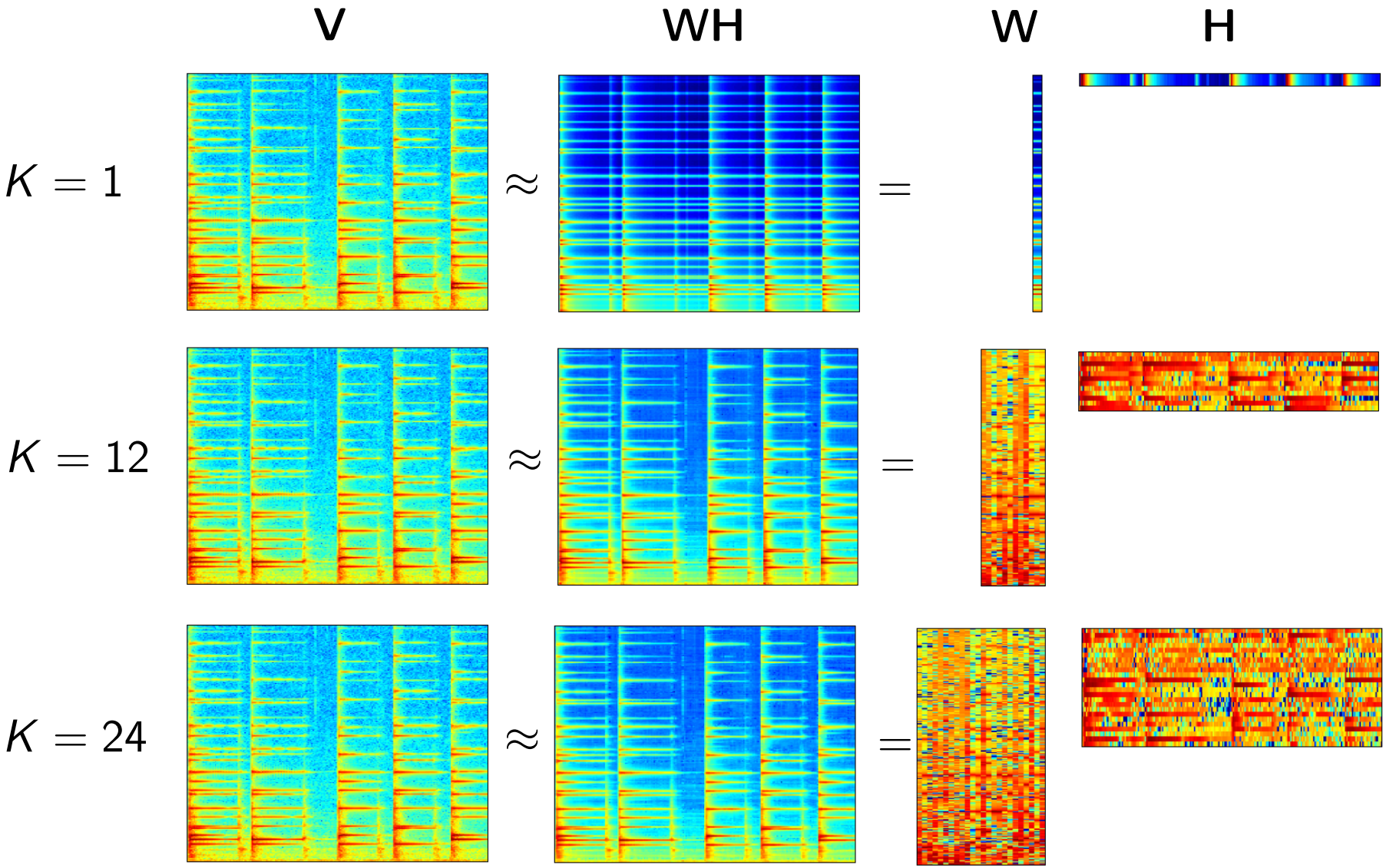
A right **model order choice is important** and it depends on the data $\mathbf{V}$ and on the application.

The following strategies are usually used to set up an appropriate model order:

- **Choose model order** K by trial and error / cross-validation
- **Estimate it automatically** within the NMF decomposition (Tan and Févotte, 2013; Schmidt and Morup, 2010).

Model order choice

Illustration on audio data



Initialization

A good **initialization** of parameters (**W** and **H**) **is important for any local optimization** approach (including MU rules) due to the existence of many local minima.

Random initializations:

- initialize (nonnegative) parameters **randomly several times**;
- keep the solution with the lowest final cost.

Strcutural data-driven initializations:

- initialize **W** by **clustering** of data points **V** (Kim and Choi, 2007);
- initialize **W** by **singular value decomposition (SVD)** of data points **V** (Boutsidis and Gallopoulos, 2008);
- ...

Stopping criteria

How many iterations?

For any iterative optimization strategy (including MU rules) **the total number of iterations is important** and results in a tradeoff between:

- the computational load from one side, and
- the data fitting (approximation error) and model quality from the other side.

Stopping criteria (Albright et al., 2006):

- after a **fixed number of iterations**;
- once the **approximation error** (the cost) is below a pre-defined **threshold**;
- once the **approximation error relative decrease** is below a pre-defined **threshold**;
- etc.

Take-home messages I

- NMF is a **versatile** data decomposition technique that has proven effective for **diverse applications** across **numerous disciplines**,
  - it tends to provide "meaningful" and "natural" **part-based** data representations,
  - it can be used both for feature learning, topic extraction, clustering, segmentation, source separation, coding...
- For NMF to be successful, it has to be estimated using **appropriate cost-functions** reflecting prior knowledge about the data.
- Being non-unique, NMF should **incorporate constraints** relating to the data, either though:
  - **regularized cost-functions** accounting for sparsity, shape, smoothness, cross-modal dependency constraints..., or
  - alternative formulations, *e.g*, **geometric** approaches having the potential to estimate **exact NMF** models.

Take-home messages II

- Many algorithms are available to estimate NMF, mostly alternating updates of **W** and **H**; variants include:
  - **multiplicative updates:** heuristic, simple and easy to implement, but slow and instable,
  - **majorisation-minimisation:** well-founded for a variety of cost functions, stable, still slow,
  - **gradient-descent** and **Newton**: fast but unstable.
- NMF is a state-of-the-art technique for a number of audio-processing tasks (transcription, source separation...),
- it has a great potential for video (and RBG+depth) analysis tasks, especially temporal structure analysis.

Ongoing and future research

- How to properly estimate the **model-order** K?
- How to achieve **better** and **faster** "convergence"?
- How to perform **non-linear** data decompositions?
- How to handle **big data**?

A selection of NMF software

Software	Language	Main features
beta_ntf	Python	Weighted tensor decomposition, all β-divergences, MM
sklearn.decomposition.NMF	Python	ℓ_2-norm, gradient-descent, sparsity
IMM DTU NMF toolbox	Matlab	ℓ_2-norm, MM, gradient-descent, ALS
Févotte's matlab scripts	Matlab	ℓ_2-norm, KL and IS-div, MM, probabilistic
Seichepine's matlab scripts	Matlab	Soft **co-factorisation**, ℓ_2-norm, KL and IS-div, ℓ_1/ℓ_2-norm **temporal smoothing**, MM
svmnmf	Matlab	Geometric SVM-based NMF, **kernel**-based non-linear decompositions, fast
libNMF	C	ℓ_2-norm, MM, gradient-descent, ALS, multi-core, fast

Bibliography I

R. Albright, J. Cox, D. Duling, A. Langville, and C. Meyer. Algorithms, initializations, and convergence for the nonnegative matrix factorization. Technical Report Math 81706, NCSU, 2006.

R. Badeau, N. Bertin, and E. Vincent. Stability analysis of multiplicative update algorithms and application to non-negative matrix factorization. *IEEE Transactions on Neural Networks*, 21(12):1869–1881, Dec. 2010.

V. D. Blondel, N.-D. Ho, and P. V. Dooren. Weighted non-negative matrix factorization and face feature extraction. In *Image and Vision Computing*, 2008.

C. Boutsidis and E. Gallopoulos. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41:1350–1362, 2008.

S. Bucak and B. Gunsel. Incremental subspace learning via non-negative matrix factorization. *Pattern Recognition*, 42(5):788–797, May 2009.

A. T. Cemgil. Bayesian Inference for Nonnegative Matrix Factorisation Models, Feb. 2009. URL http://www.hindawi.com/journals/cin/2009/785152.abs.html.

J.-C. Chen. The nonnegative rank factorizations of nonnegative matrices. 62:207–217, Nov 1984. ISSN 00243795. doi: 10.1016/0024-3795(84)90096-X. URL http://www.sciencedirect.com/science/article/pii/002437958490096X.

C. Ding, X. He, and H. Simon. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In *SIAM Data Mining Conference*, number 4, 2005. URL http://epubs.siam.org/doi/abs/10.1137/1.9781611972757.70.

C. H. Ding, T. Li, and M. I. Jordan. Convex and Semi-Nonnegative Matrix Factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):45–55, 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.277. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=5339303&arnumber=4685898&count=16&index=4.

Bibliography II

J. Eggert and E. Korner. Sparse coding and NMF. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 4, pages 2529–2533. IEEE, 2004. ISBN 0-7803-8359-1. doi: 10.1109/IJCNN.2004.1381036. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1381036.

S. Eguchi and Y. Kano. Robustifying maximum likelihood estimation. Technical report, Institute of Statistical Mathematics, June 2001. Research Memo. 802.

S. Essid. A single-class SVM based algorithm for computing an identifiable NMF. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 2012.

S. Essid and C. Fevotte. Smooth Nonnegative Matrix Factorization for Unsupervised Audiovisual Document Structuring. *IEEE Transactions on Multimedia*, 15(2):415–425, 2013. ISSN 1520-9210. doi: 10.1109/TMM.2012.2228474.

C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the beta-divergence. Oct. 2010. URL http://arxiv.org/abs/1010.1763.

C. Fevotte, N. Bertin, and J.-L. Durrieu. Nonnegative Matrix Factorization with the Itakura-Saito Divergence. With Application to Music Analysis. *Neural Computation*, 21(3), Mar. 2009.

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis. *Neural Computation*, 21(3):793–830, 2009.

E. Gaussier and C. Goutte. Relation between PLSA and NMF and implications. In *Proc. 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'05)*, pages 601–602, New York, NY, USA, 2005. ACM. ISBN 1595930345. URL http://dl.acm.org/citation.cfm?id=1076148.

T. Hofmann. Probabilistic latent semantic analysis. *Proceedings of the Fifteenth conference on Uncertainty . . .*, 1999. URL http://dl.acm.org/citation.cfm?id=2073829.

P. O. Hoyer. Non-negative Matrix Factorization with Sparseness Constraints. *The Journal of Machine Learning Research*, 5:1457–1469, Dec. 2004. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1005332.1044709.

D. R. Hunter and K. Lange. A tutorial on MM algorithms. *Amer. Stat.*, 58(1):30–37, Feb. 2004.

Bibliography III

M. Jeter and W. Pye. A note on nonnegative rank factorizations. *Linear Algebra and its Applications*, 38:171–173, Jun 1981. ISSN 00243795. doi: 10.1016/0024-3795(81)90018-5. URL http://www.sciencedirect.com/science/article/pii/0024379581900185.

S. Jia and Y. Qian. Constrained Nonnegative Matrix Factorization for Hyperspectral Unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 47(1):161–173, Jan. 2009. ISSN 0196-2892. doi: 10.1109/TGRS.2008.2002882. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4694061.

Y.-D. Kim and S. Choi. A Method of Initialization for Nonnegative Matrix Factorization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07)*, volume 2, pages 537–540, Honolulu, Hawaii, 2007.

B. Klingenberg, J. Curry, and A. Dougherty. Non-negative matrix factorization: Ill-posedness and a geometric algorithm. *Pattern Recognition*, 42(5):918–928, May 2009. ISSN 0031-3203. doi: 10.1016/j.patcog.2008.08.026. URL http://linkinghub.elsevier.com/retrieve/pii/S0031320308003403http://www.sciencedirect.com/science/article/B6V14-4TCR1KR-1/2/8bedc245dc0fd9ba7487561f8df431cahttp://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V14-4TCR1KR-1&_user=771355&_coverDate=05/31/2009&_rdoc=1&_fmt=high&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1272450458&_rerunOrigin=google&_acct=C000028498&_version=1&_urlVersion=0&_userid=771355&md5=35ba172c08afbfa2b0676a0f2dca1897.

D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401: 788–791, 1999.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural and Information Processing Systems 13*, pages 556–562, 2001.

C.-J. Lin. On the Convergence of Multiplicative Update Algorithms for Nonnegative Matrix Factorization. *IEEE Transactions on Neural Networks*, 18:1589–1596, 2007a.

C.-J. Lin. Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Computation*, 19:2756–2779, 2007b.

Bibliography IV

J. Liu, C. Wang, J. Gao, and J. Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proc. of SDM*, 2013. URL http://epubs.siam.org/doi/abs/10.1137/1.9781611972832.28.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11(10-60), 2010.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, Jun 1994. ISSN 11804009. doi: 10.1002/env.3170050203. URL http://doi.wiley.com/10.1002/env.3170050203.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

M. N. Schmidt and M. Morup. Infinite non-negative matrix factorizations. In *Proc. European Signal Processing Conference (EUSIPCO)*, 2010.

N. Seichepine, S. Essid, C. Fevotte, and O. Cappe. Soft nonnegative matrix co-factorization with application to multimodal speaker diarization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, 2013.

L. S. R. Simon and E. Vincent. A general framework for online audio source separation. In *International conference on Latent Variable Analysis and Signal Separation*, Tel-Aviv, Israel, Mar. 2012.

Y. F. Tan and C. Févotte. Automatic relevance determination in nonnegative matrix factorization with the β-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1592–1605, 2013.

E. Vincent, N. Bertin, and R. Badeau. Harmonic and inharmonic Nonnegative Matrix Factorization for Polyphonic Pitch transcription. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 109–112. IEEE, Mar. 2008. ISBN 978-1-4244-1483-3. doi: 10.1109/ICASSP.2008.4517558. URL http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4517558.

T. Virtanen. Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1066–1074, 2007.

Bibliography V

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03*, page 267, New York, New York, USA, July 2003. ACM Press. ISBN 1581136463. doi: 10.1145/860435.860485. URL http://dl.acm.org/citation.cfm?id=860435.860485.

Z. Yang and E. Oja. Unified Development of Multiplicative Algorithms for Linear and Quadratic Nonnegative Matrix Factorization. *IEEE Trans. Neural Networks*, 22(12):1878–1891, 2011.

K. Y. Yilmaz, A. T. Cemgil, and U. Simsekli. Generalized coupled tensor factorization. In *NIPS*, 2011.

R. Zdunek and A. Cichocki. Non-negative matrix factorization with quasi-Newton optimization. In *Eighth International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, pages 870–879, 2006.