Unsupervised data decompositions

Slim ESSID & Alexey OZEROV

Telecom ParisTech

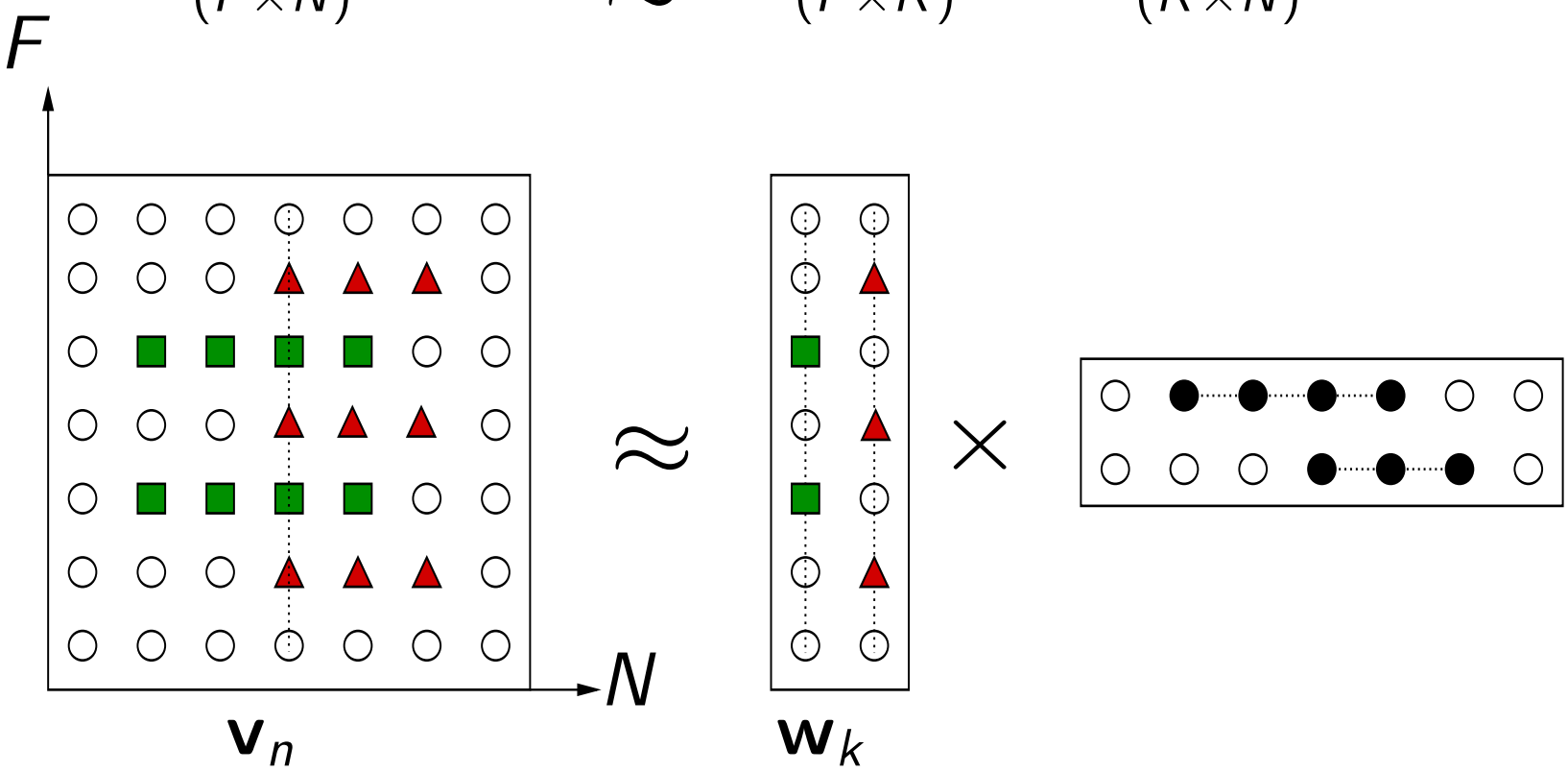
April 2015

- Introduction
- Principal Component Analysis (PCA)
- NMF models
- Algorithms for solving NMF
- Constrained NMF schemes
- Multi-stream and cross-modal NMF schemes
- Applications
- Conclusion

Explaining data by factorisation

General formulation

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

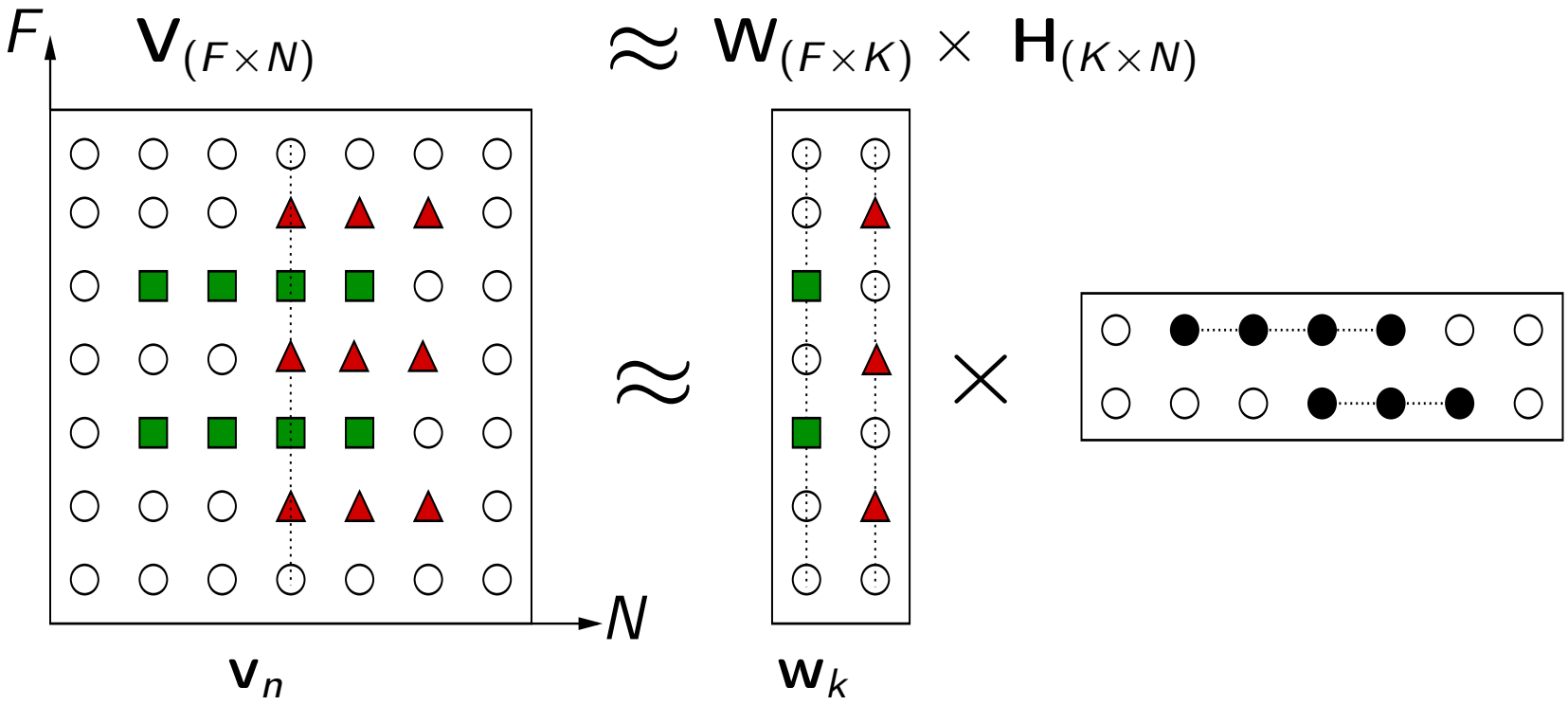


$$\mathbf{v}_n \approx \sum_{k=1}^{K} h_{kn}\mathbf{w}_k$$

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 $K \times K$.

Compression

The residual least square error of the decomposition can be shown to be

$$\frac{1}{N}\sum_{n}\|\mathbf{v}_n - \hat{\mathbf{v}}_n\|_2^2 = \sum_{i=K+1}^{F} d_i$$

where $\{d_i\}_i$ are the eigenvalues of $\mathbf{C_v}$, sorted in order of decreasing value. PCA can be used for **compression** and **dimensionality reduction**: the

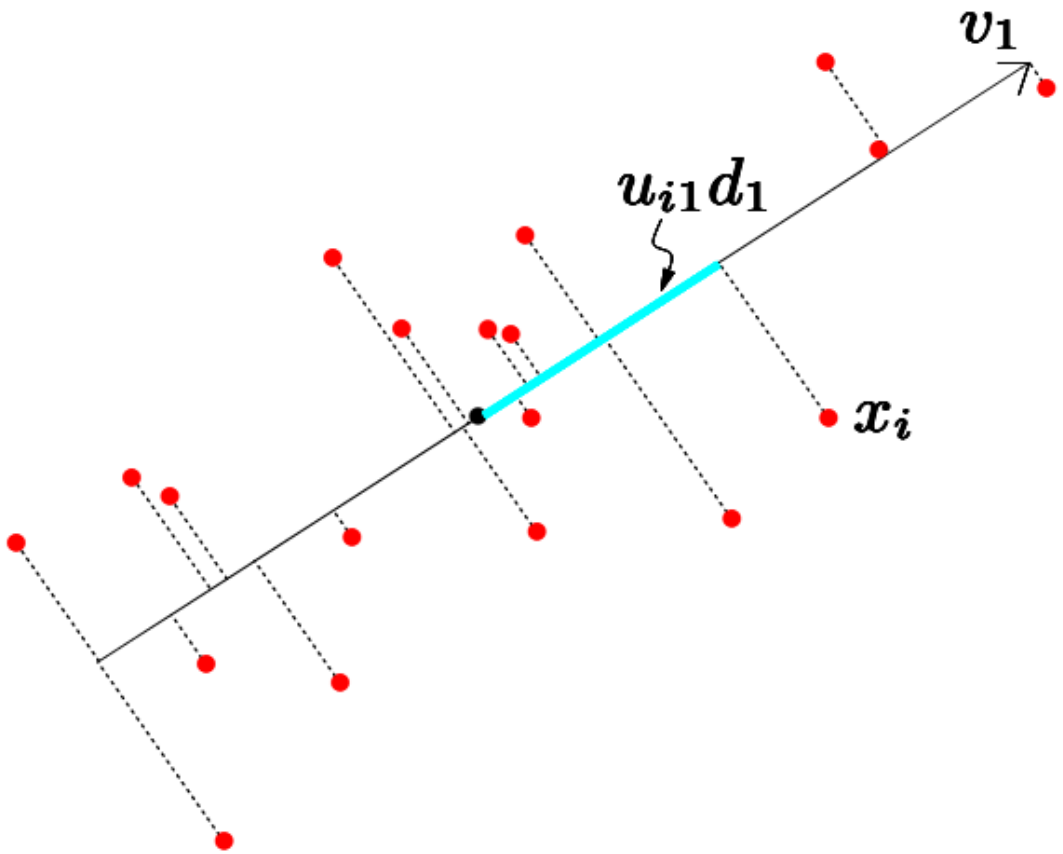
original $F \times N$ data matrix $\mathbf{V}$ can be approximated by $FK + KN$ coefficients of the matrices $\mathbf{W}_{PCA}$ and $\mathbf{H}_{PCA} = \mathbf{W}_{PCA}^T\mathbf{V}$.

Uncorrelatedness

When $\mathbf{U} = \mathbf{I}$, the expansion coefficients in the PCA model are uncorrelated; indeed, we have

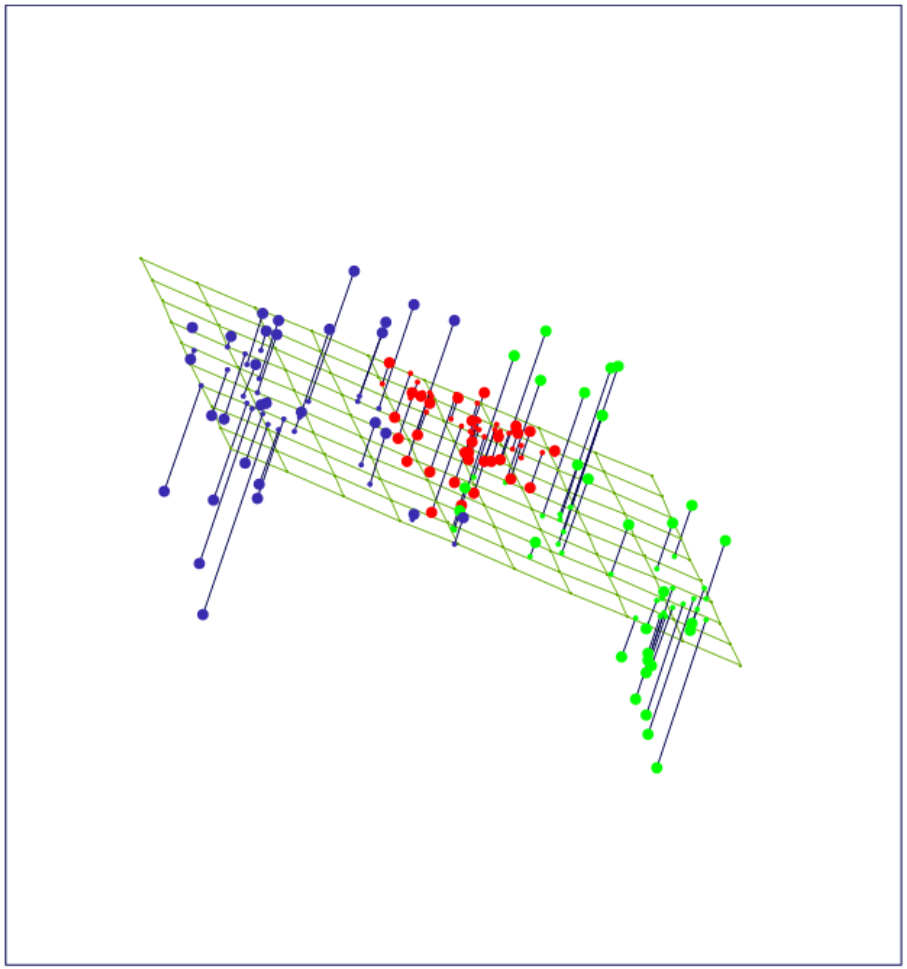
$$\begin{aligned} \mathbf{C_h} &= \mathrm{E}\{(\mathbf{W}^T\mathbf{v})(\mathbf{W}^T\mathbf{v})^T\} \\ &= \mathbf{W}^T\mathbf{C_v}\mathbf{W} \\ &= \mathrm{diag}\left([d_1, \ldots, d_K]\right) \\ &\stackrel{\text{def}}{=} \mathbf{D}_K \end{aligned}$$

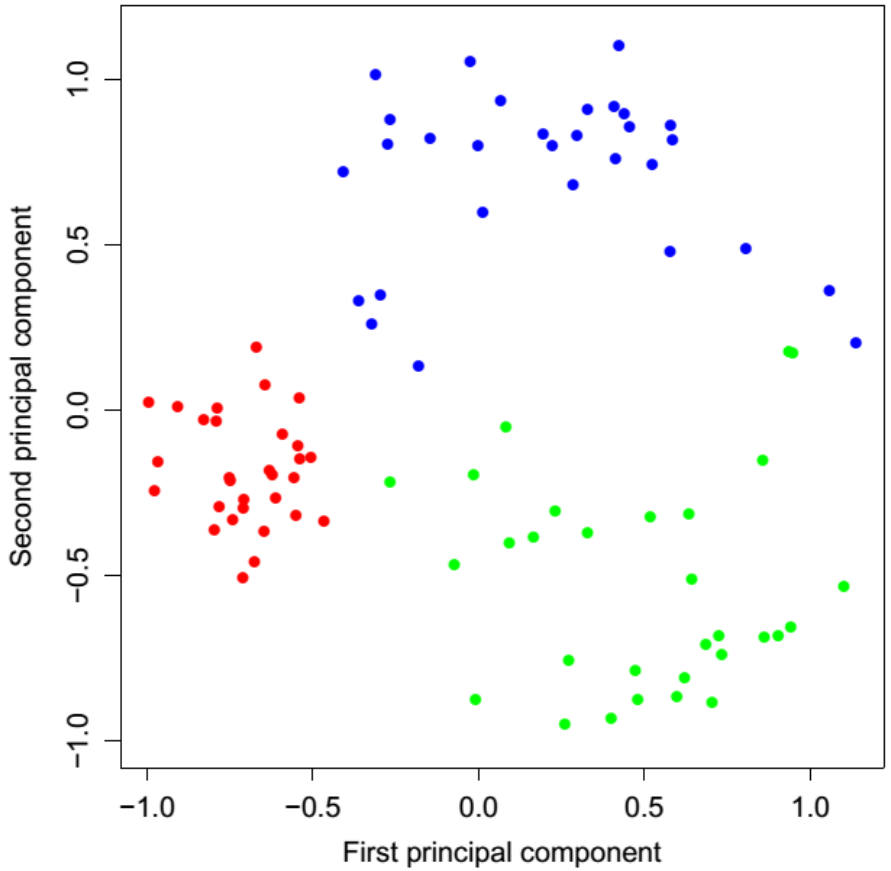
2D data example



After (Hastie *et al.*, 2008)

3D data example





After (Hastie *et al.*, 2008)

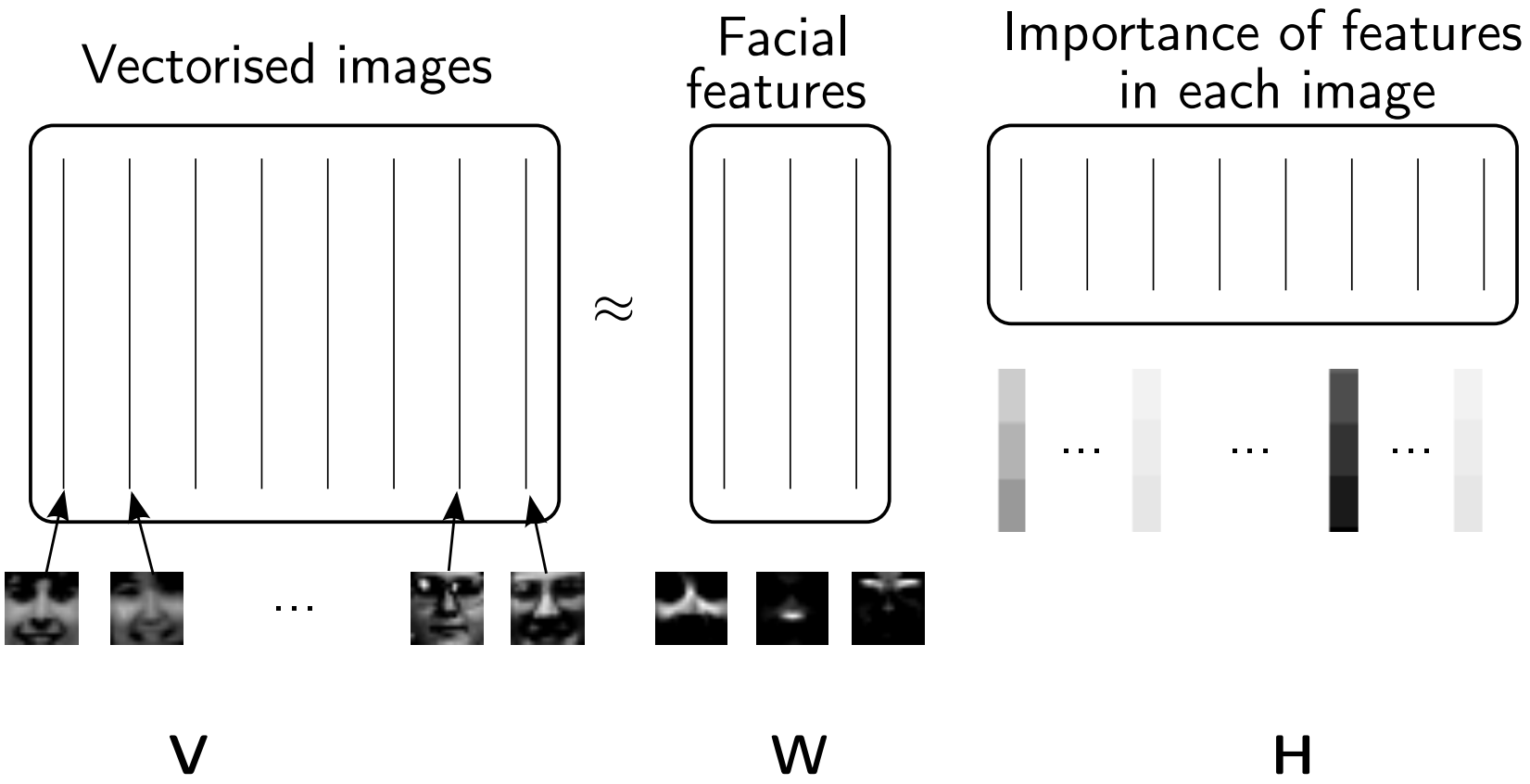
Explaining face images by PCA[1]

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

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

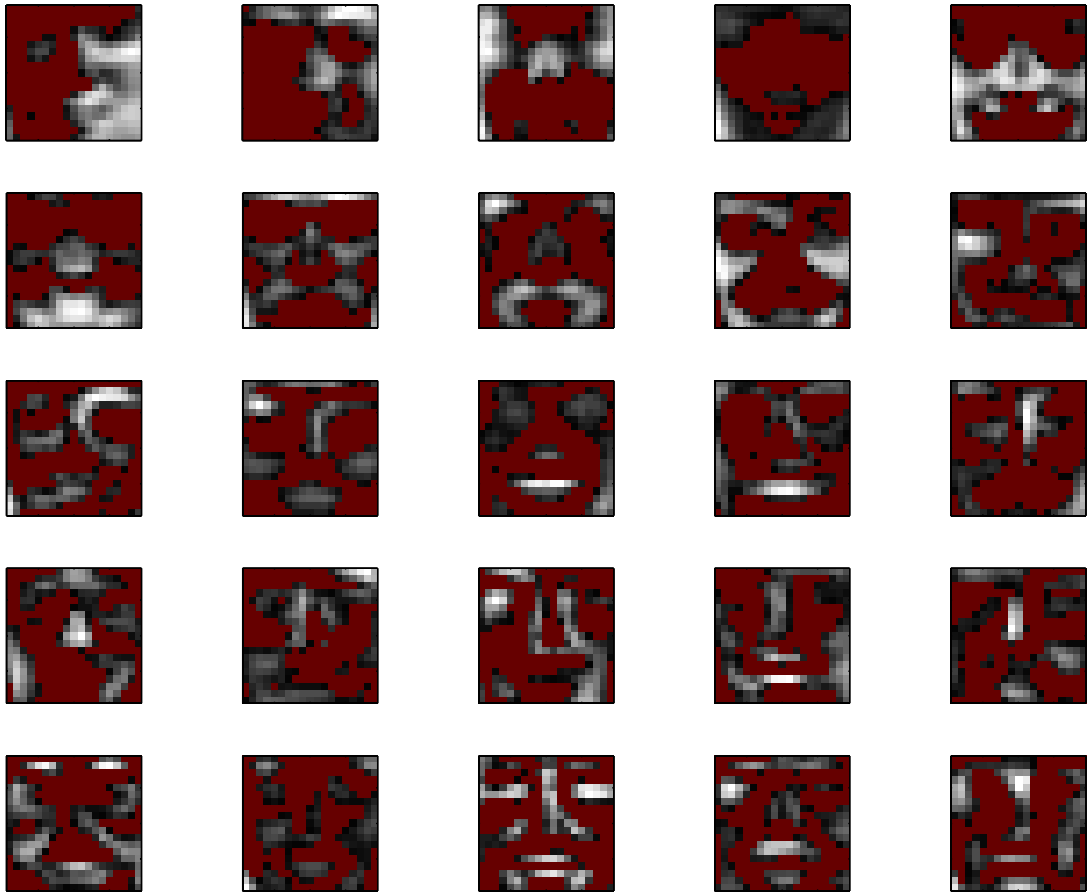
Explaining face images by PCA

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]

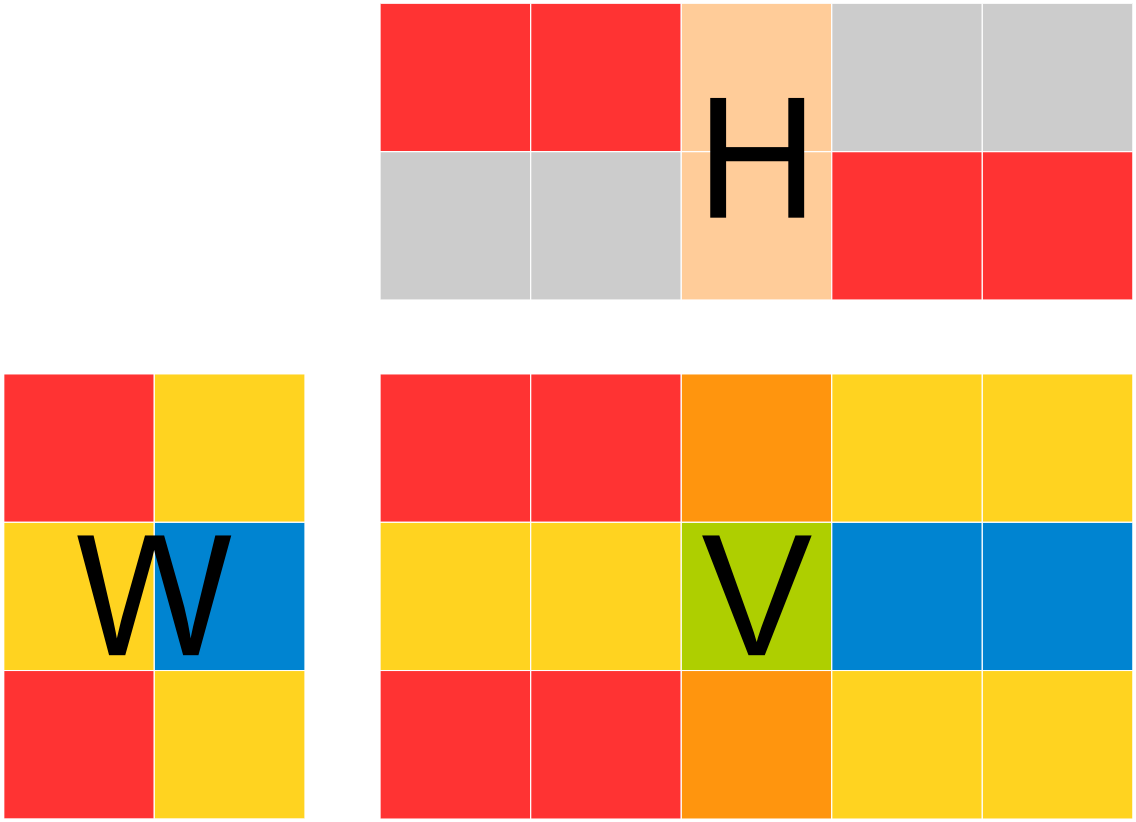
- pixel intensities;
- amplitude spectra;
- occurrence counts;
- food or energy consumption;
- user scores;
- stock market values;
- ...

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

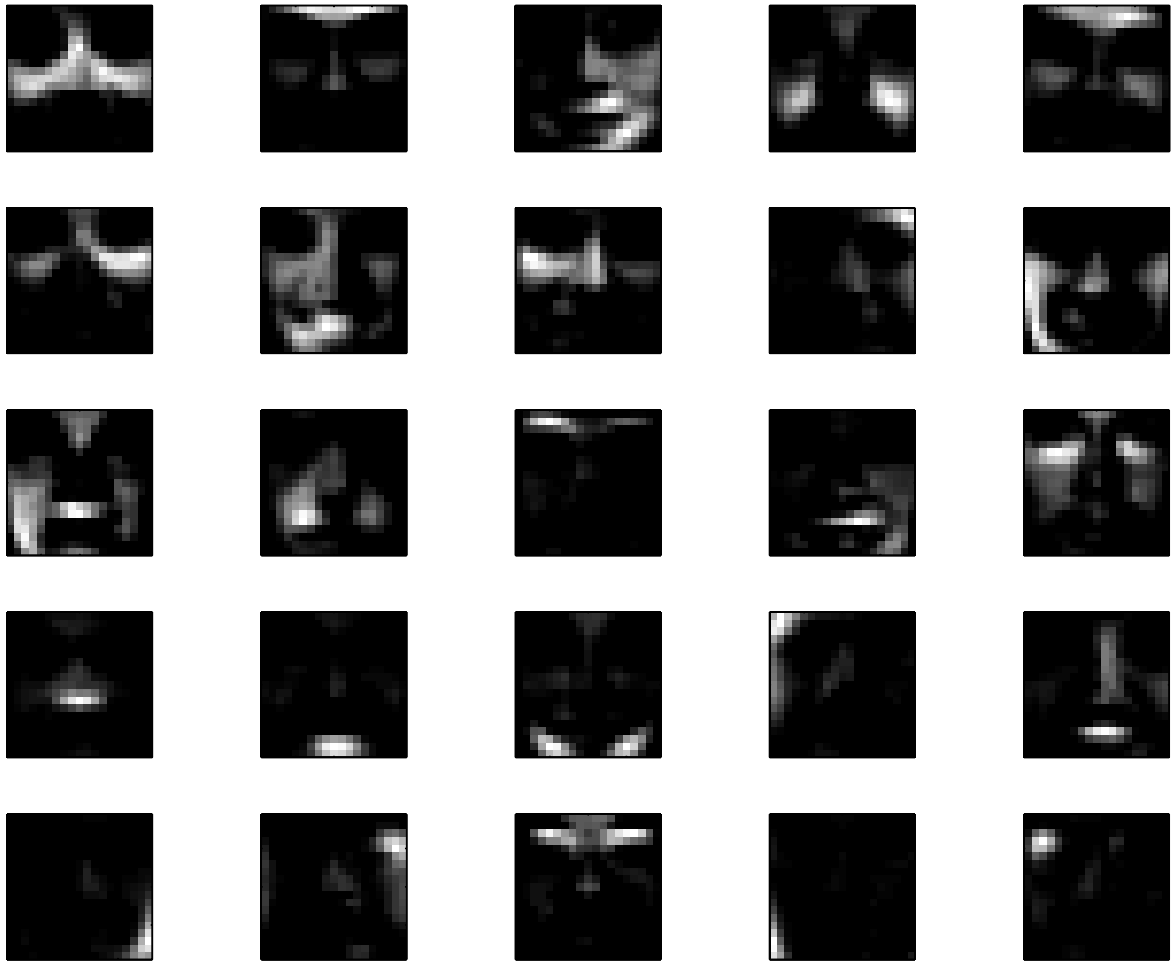


Illustration by C. Févotte

NMF outputs

Audio example

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

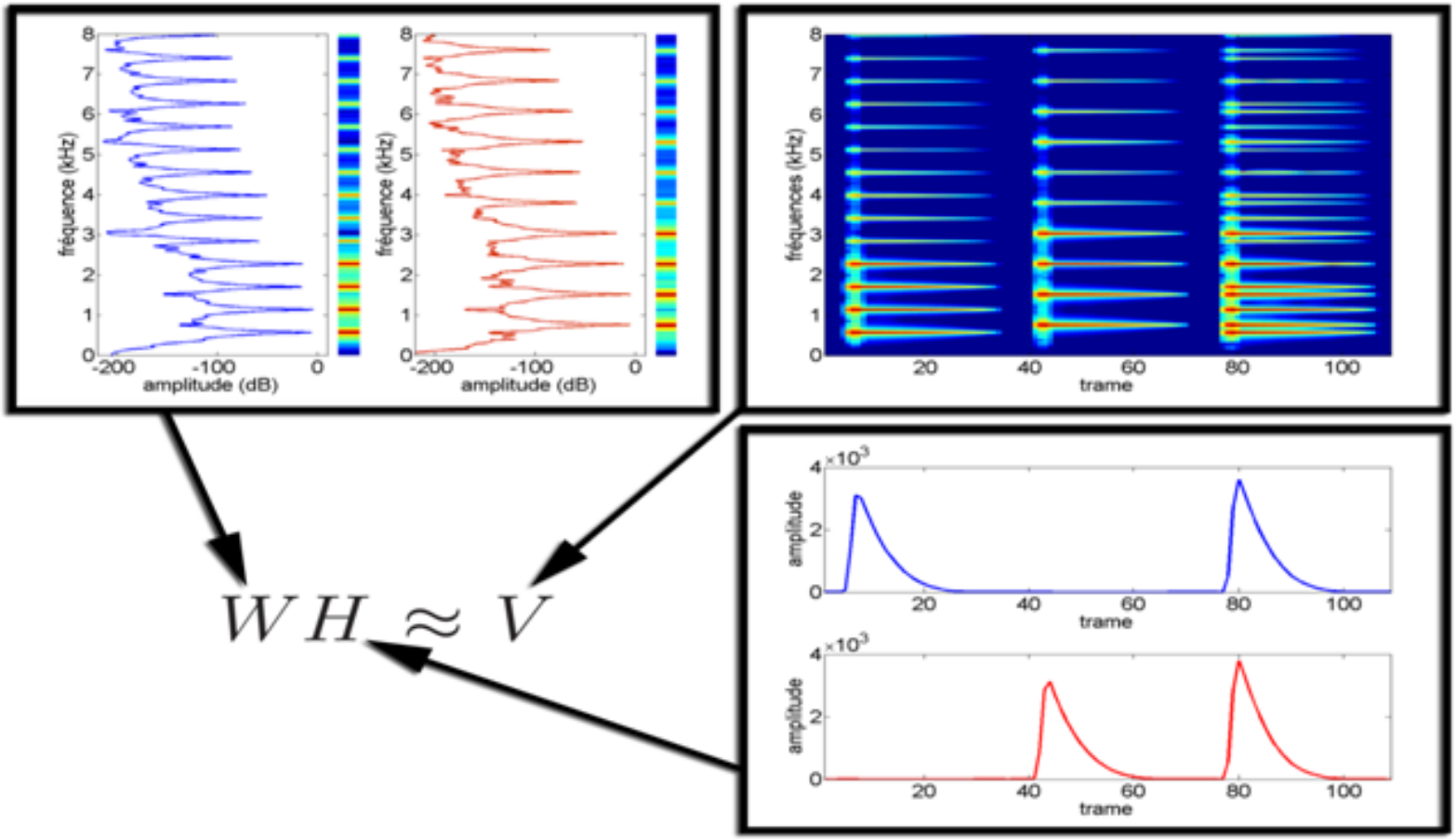


Illustration by R. Hennequin.

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**:

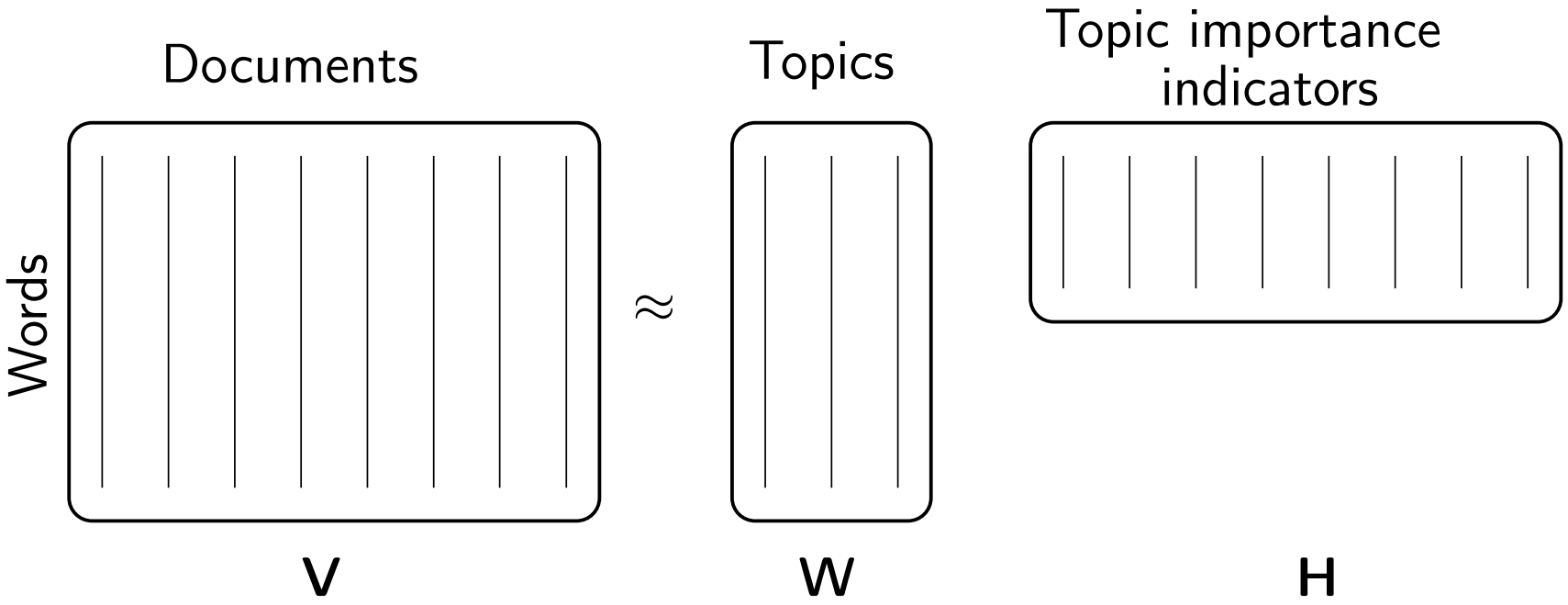
	Accuracy
PCA	93%
ICA	93%
NMF	**96%**

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:

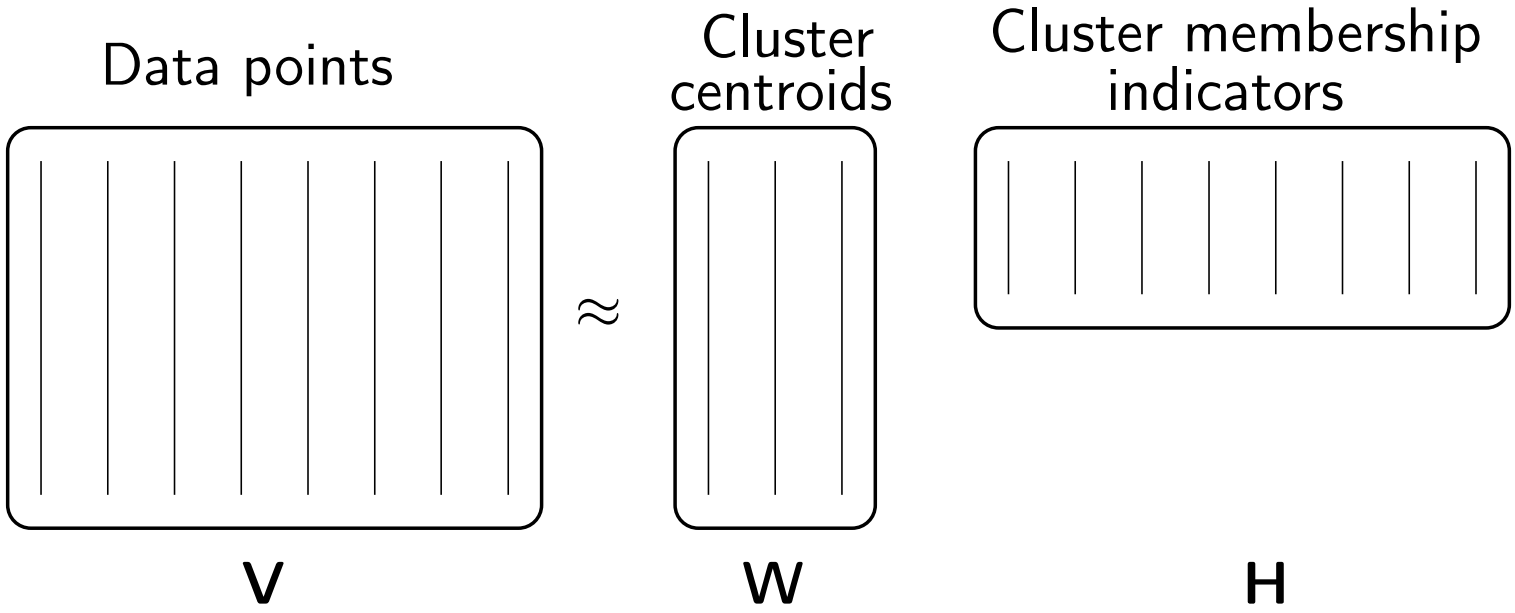
- **Topic #0:** god people bible israel jesus christian true moral think christians believe don say human israeli church life children jewish
- **Topic #1:** drive windows card drivers video scsi software pc thanks vga graphics help disk uni dos file ide controller work
- **Topic #2:** game team nhl 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):



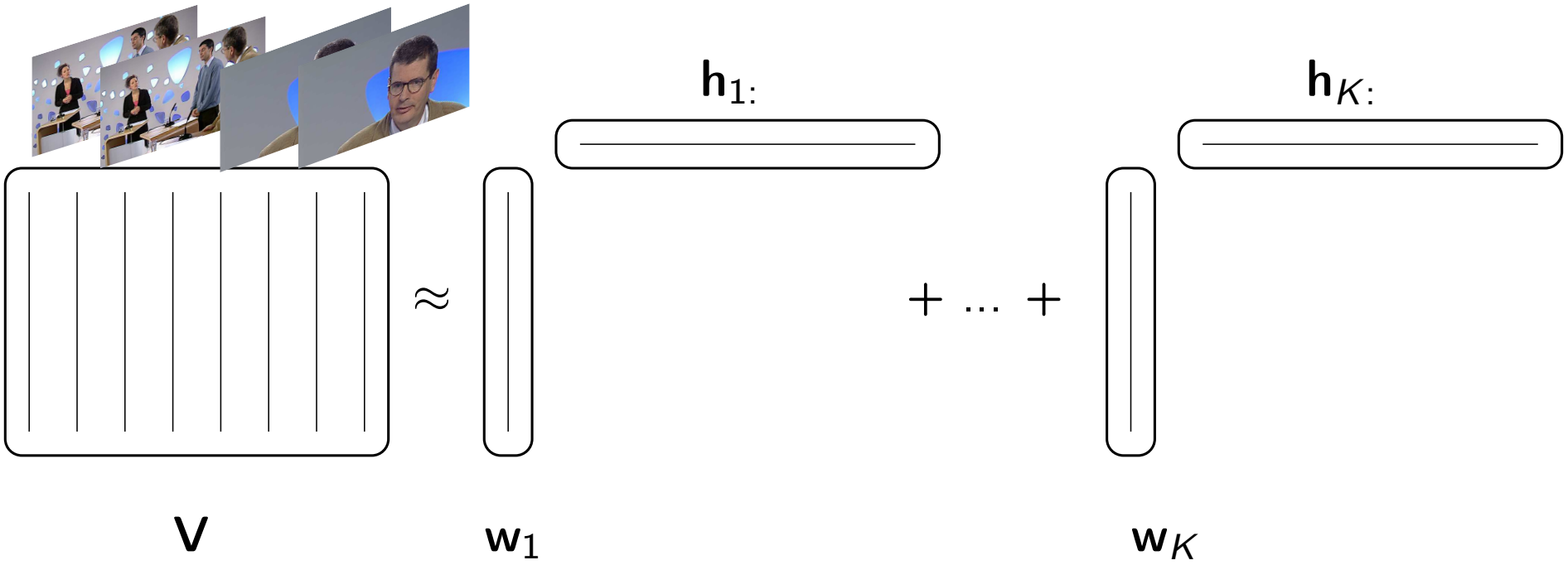
- NMF can handle overlapping clusters and provides *soft* cluster membership indications.

General usages of NMF IV

What for?

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

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

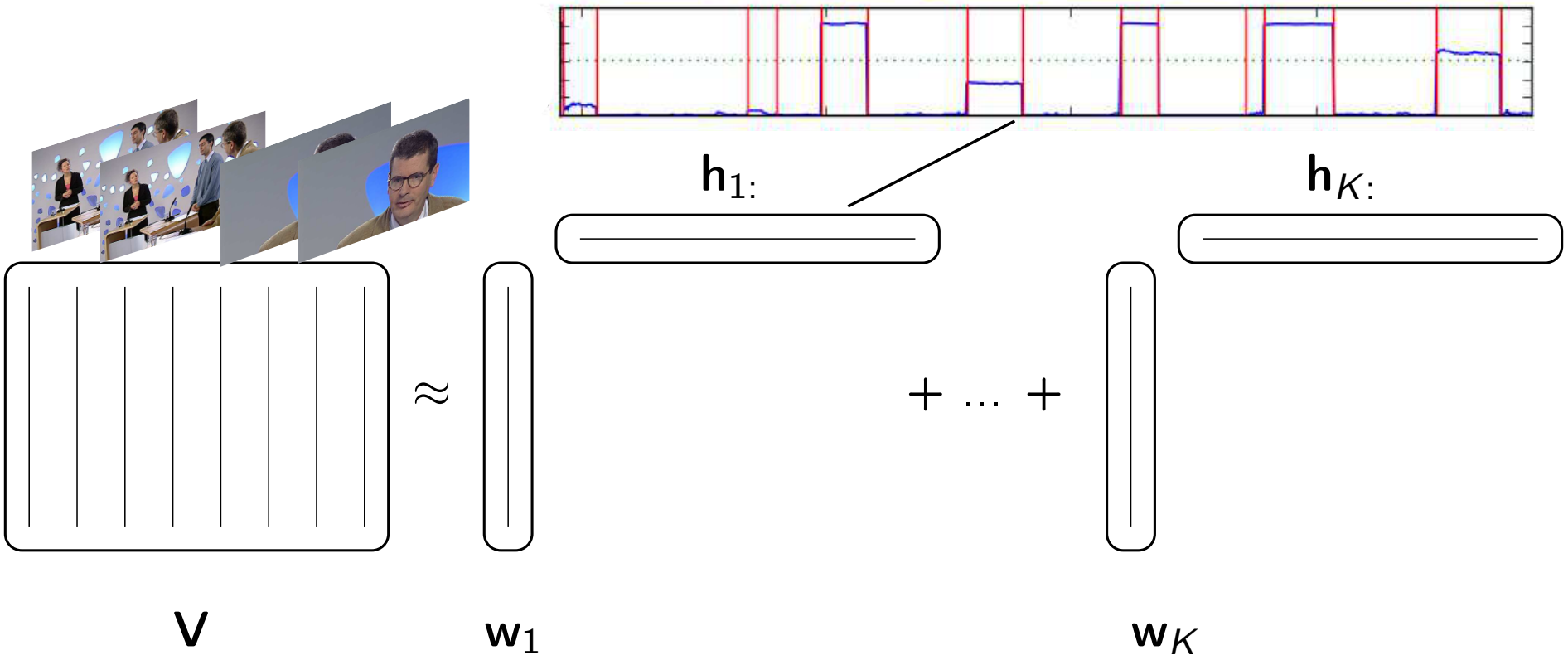


General usages of NMF IV

What for?

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

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

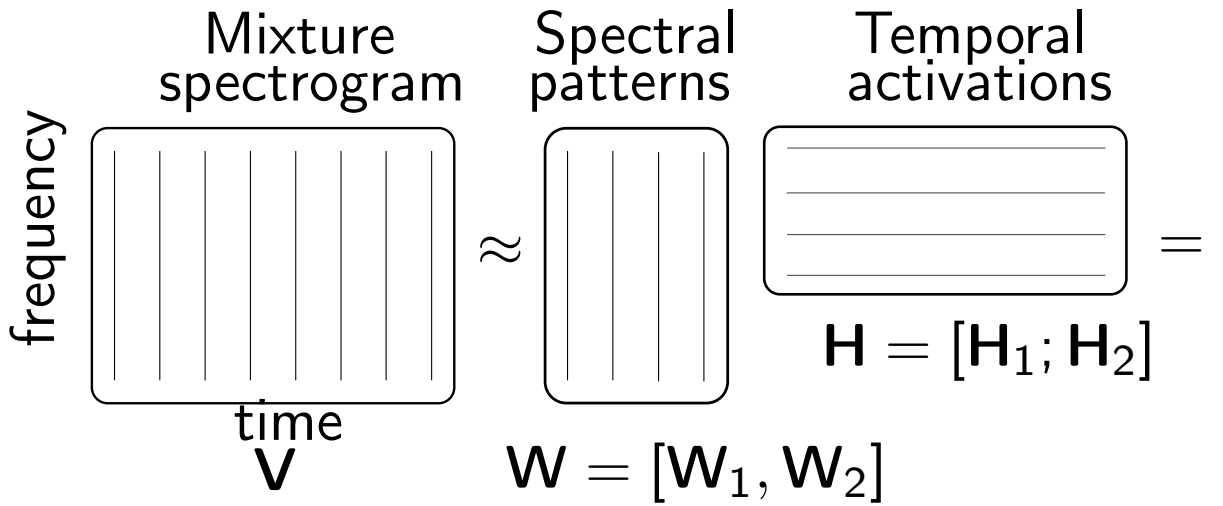


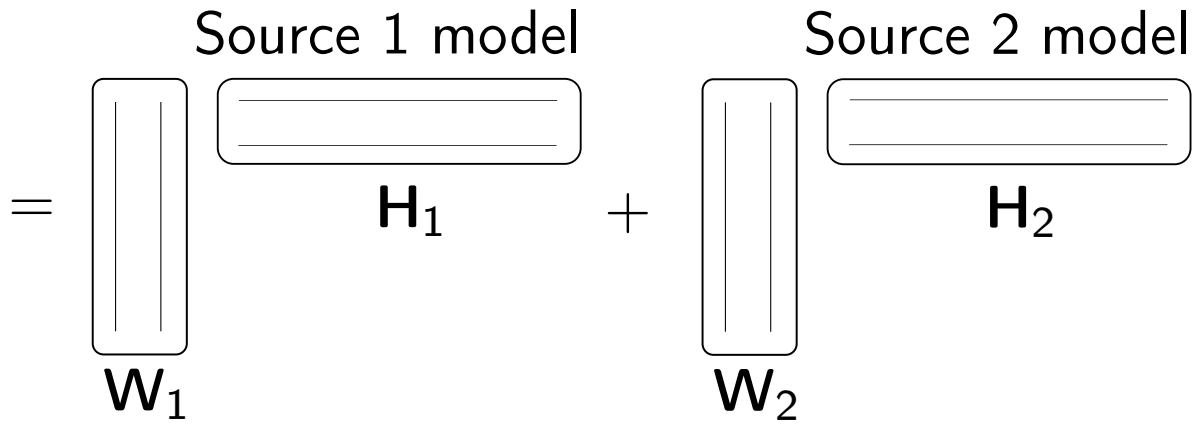
Temporal segmentation can be achieved by thresholding the temporal activations relating to components of interest.

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.

Overview of NMF application domains I

A variety of successful applications:

- **Text mining**: (Xu et al., 2003; Berry and Browne, 2006; Kim and Park, 2008)
- **Images**:
  - unsupervised object discovery (Sivic et al., 2005)
  - object and face recognition (Soukup and Bajla, 2008)
  - tagging (Kalayeh et al., 2014)
  - denoising and inpainting (Mairal et al., 2010)
  - texture classification (Sandler and Lindenbaum, 2011)
  - spectral data (Berry et al.)
  - hashing (Monga and Mihcak, 2007)
  - watermarking (Lu et al., 2009)
- **Electroencephalography (EEG) data**:
  - feature extraction (Cichocki and Rutkowski, 2006; Lee et al., 2009)
  - artifact rejection (Damon et al., 2013a,b)

Overview of NMF application domains II

- **Bioinformatics**:
  - gene expression analysis (Brunet et al., 2004; Gao and Church, 2005)
  - protein interaction clustering (Greene et al., 2008)
- **Other**:
  - collaborative filtering (Melville and Sindhwai, 2010)
  - community discovery (Wang et al., 2010)
  - portfolio diversification (Drakakis et al., 2007)
  - food consumption analysis (Zetlaoui et al., 2010)
  - industrial source apportionment (Limem et al., 2013)
- **Audio and music**
- **Videos**

Audio and music processing

- **Source separation** (NMF is state-of-the art):
  - **speech**: separating voices in speech mixtures or voice from background (Virtanen, 2007; Virtanen and Cemgil, 2009; Mohammadiha et al., 2013)
  - **music**: separating singing voice/melody from accompaniment or musical instruments in polyphonic mixtures (Durrieu et al., 2009; Ozerov and Fevotte, 2010; Hennequin et al., 2011; Ozerov et al., 2013; Rafii et al., 2013)
- **Signal enhancement/denoising**:

  (Wilson et al., 2008; Schmidt et al., 2007; Sun and Mazumder, 2013)
- **Audio inpainting**

  (Roux et al., 2011; Yilmaz et al., 2011)

Audio and music processing

- **Compression**

  (Ozerov et al., 2011b; Nikunen et al., 2011)

- **Music transcription:** recognizing musical notes played by Piano, Drums or multiple instruments

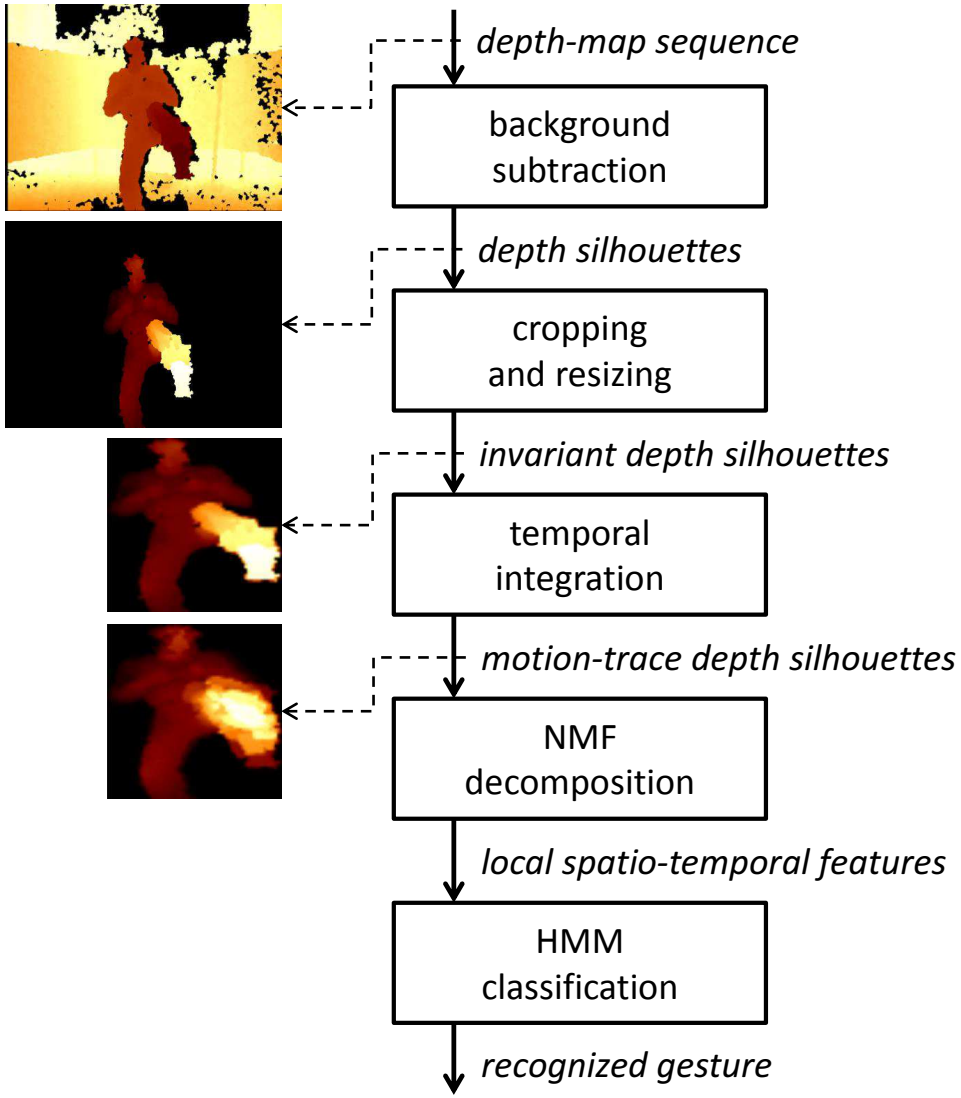
  (Smaragdis and Brown, 2003; Abdallah and Plumbley, 2004; Vincent et al., 2007; E. Vincent et al., 2008; Févotte et al., 2009; Bertin et al., 2010; Vincent et al., 2010)

Video processing

- NMF use for video processing remains quite limited, despite its potential.
- Known works:
  - Video summarization (Cooper and Foote, 2002)
  - Dynamic video content representation and scene change detection (Bucak and Gunsel, 2007)
  - Onscreen person spotting and shot-type classification (Essid and Fevotte, 2012, 2013)
  - Fingerprinting (Cirakman et al., 2010)
  - Action recognition (Krausz and Bauckhage, 2010; Masurelle et al., 2014)
  - Compression (Türkan and Guillemot, 2011)

Action recognition using depth silhouettes

Using NMF for feature learning (Masurelle et al., 2014)



Skeleton features	PCA	NMF
78%	89%	**91%**

Recognition accuracies

- considering Kinect recordings of 8 actions;
- using Huawei/3DLife grand challenge dataset for action recognition.

Video Structuring

Using NMF for temporal segmentation and soft-clustering (Essid and Fevotte, 2013)

Discovering the video editing structure (Essid and Fevotte, 2012)

Performing **speaker diarization** (Seichepine et al., 2013)

"Who spoke when?"

illustration by N. Seichepine

Using the **Canal9 political debates** database (Vinciarelli et al., 2009).

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

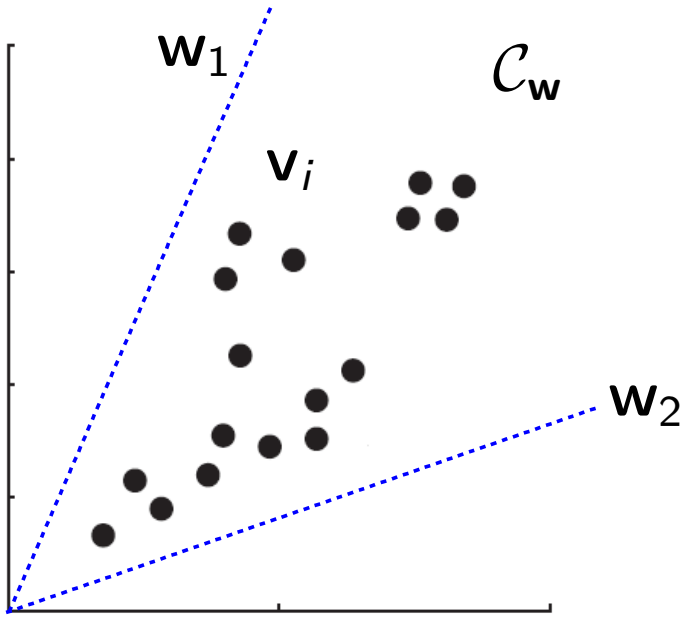
In particular, $\mathbf{Q}$ can be any **nonnegative generalised permutation matrix**; *e.g.*, in $\mathbb{R}^3$:

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

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

Geometric interpretation and ill-posedness

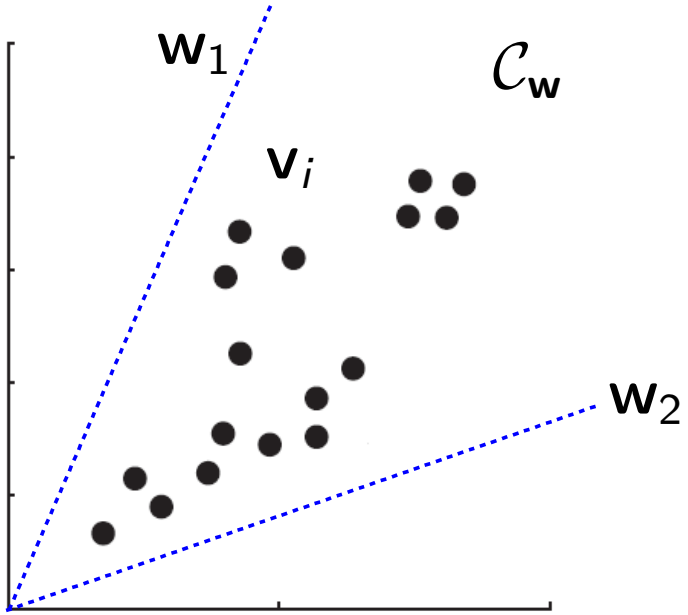
NMF assumes the data is well described by a **simplicial convex cone** $\mathcal{C}_{\mathbf{w}}$ generated by the columns of $\mathbf{W}$:

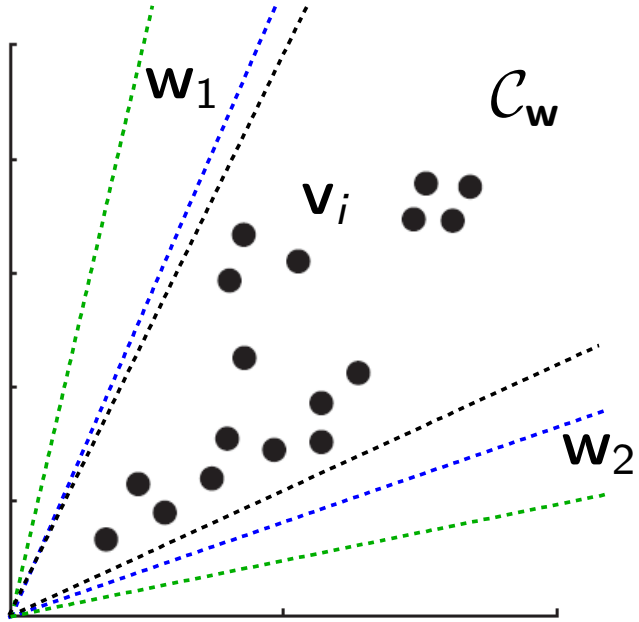


$$\mathcal{C}_{\mathbf{w}} = \left\{ \sum_{k=1}^{K} \lambda_k \mathbf{w}_k;\ \lambda_k \geq 0 \right\}$$

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 $\mathbf{W}$:



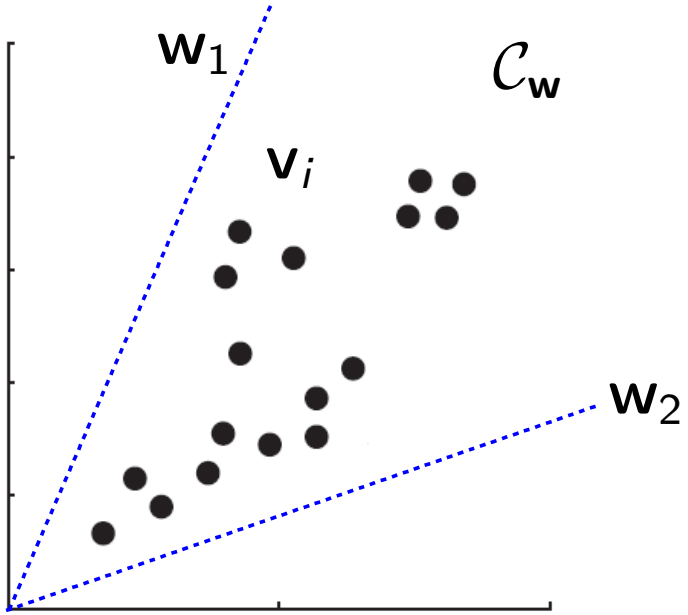


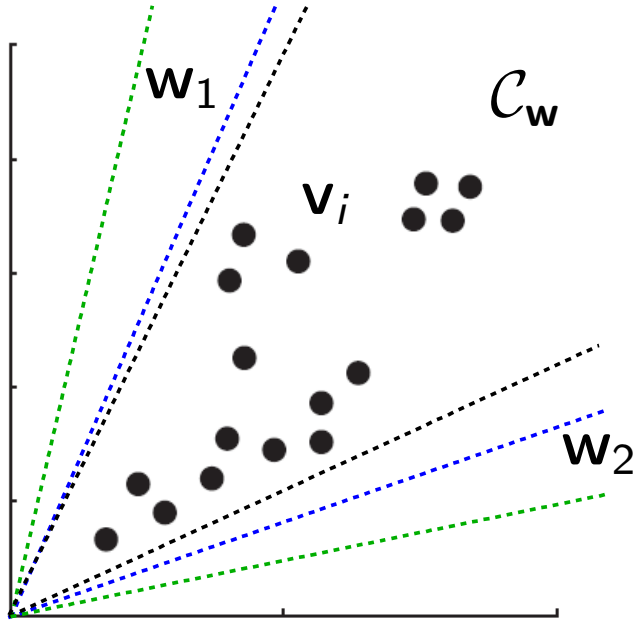
$$\mathcal{C}_{\mathbf{w}} = \left\{ \sum_{k=1}^{K} \lambda_k \mathbf{w}_k;\ \lambda_k \geq 0 \right\}$$

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)

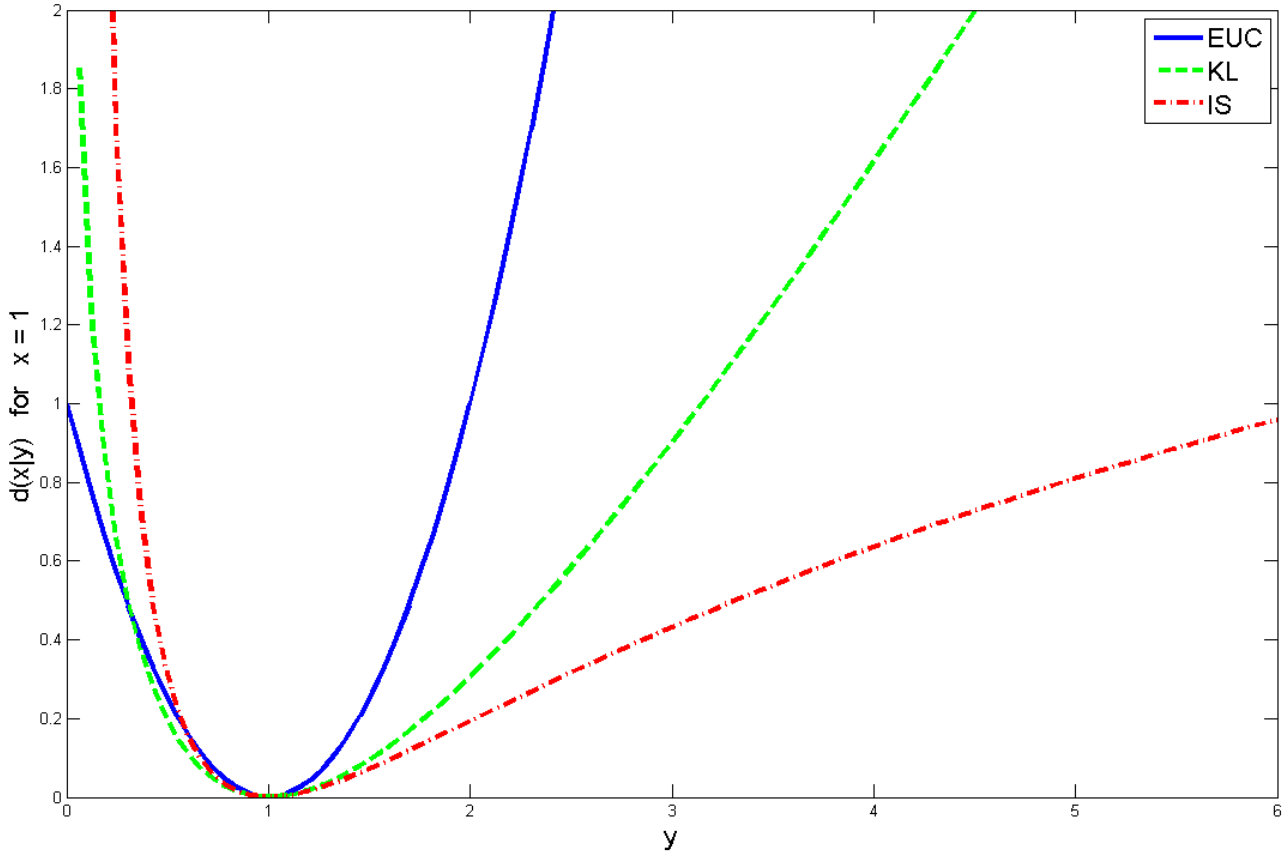
$$d_{KL}(x, y) = x \log \frac{x}{y} - x + y$$

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]

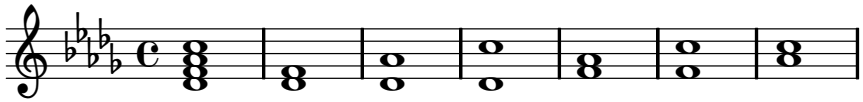
$$
\begin{aligned}
d_{EUC}(\lambda x|\lambda y) &= \lambda^2\, d_{EUC}(x|y) \\
d_{KL}(\lambda x|\lambda y) &= \lambda\, d_{KL}(x|y) \\
d_{IS}(\lambda x|\lambda y) &= d_{IS}(x|y)
\end{aligned}
$$

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

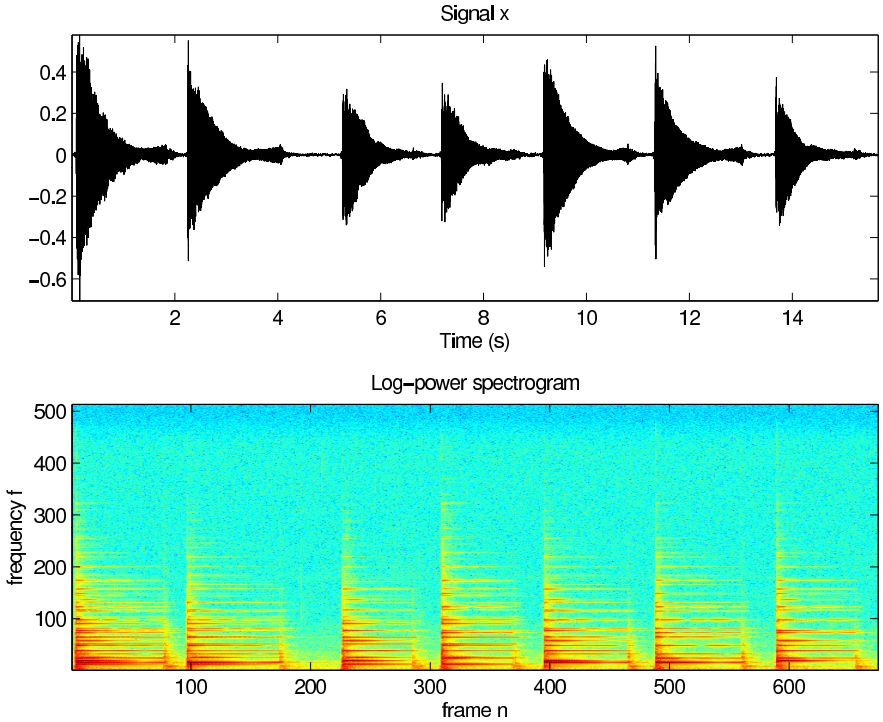
[4] slide adapted from (Févotte, 2012).

Music transcription demo

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



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

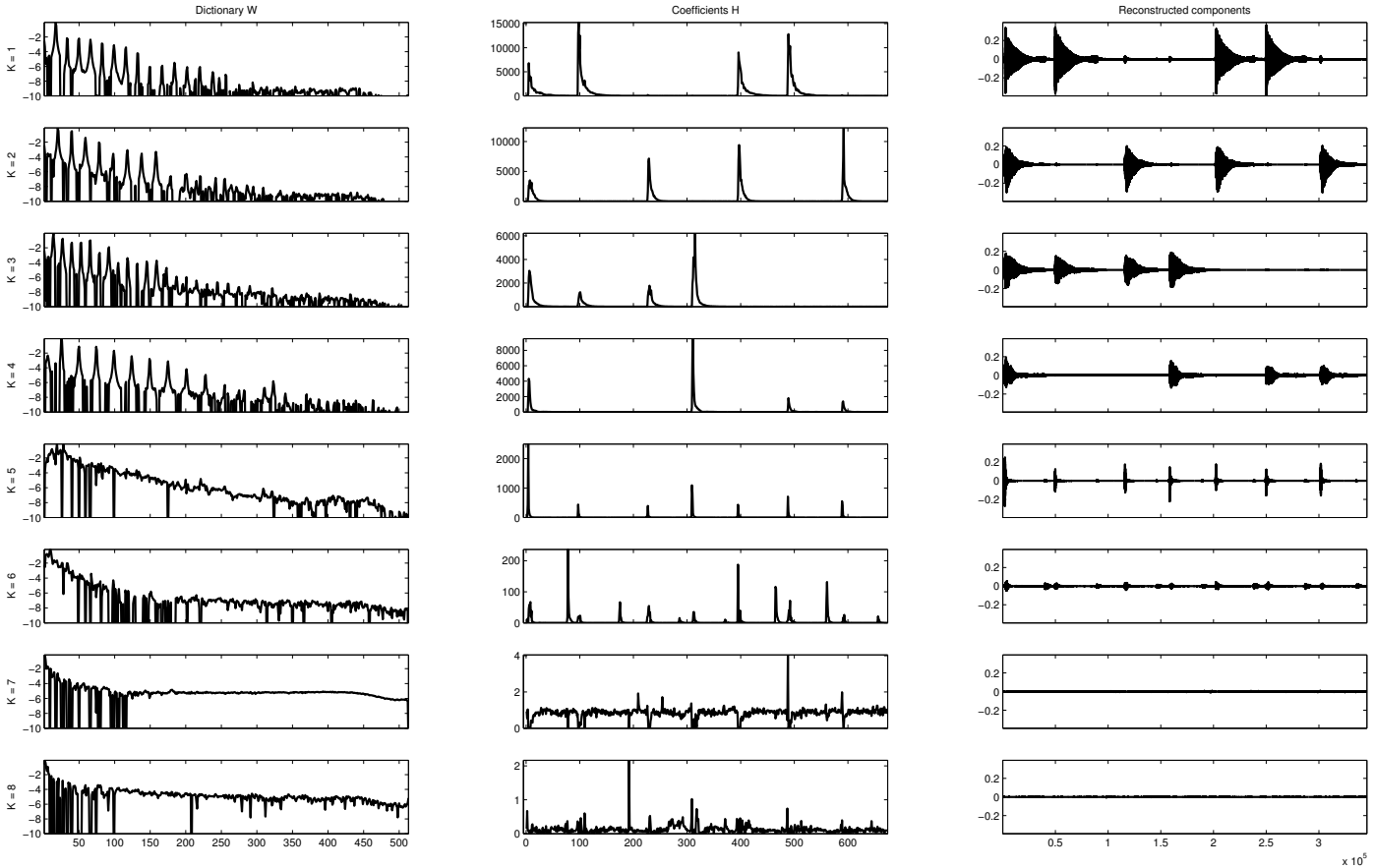


Three representations of the data.

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 as follows:

- by **intuition** or from some **prior knowledge of the application goal** (*e.g.*, NMF is used for predicting the unseen data while minimizing the mean squared error $\Longrightarrow$ EUC distance) or **invariances** (*e.g.*, scale invariance for music analysis with IS divergence) ;
- from some **probabilistic considerations** (presented in the upcoming section);
- **optimize the divergence** (*e.g.* from some parametric family) on some development data within a particular application.

Statistical viewpoint

For many divergences a probabilistic formulation is possible: the divergence minimization becomes equivalent to a maximum likelihood criterion (Févotte et al., 2009; Cemgil, 2009b):

$$D(\mathbf{V}|\widehat{\mathbf{V}}) = -\log p(\mathbf{V}|\widehat{\mathbf{V}}) + const$$

Examples:

Divergence $D(\mathbf{V}\|\widehat{\mathbf{V}})$		Probability distribution	p.d.f. $p(\mathbf{V}\|\widehat{\mathbf{V}})$
EUC	$\sum_{f,n} (v_{fn} - \hat{v}_{fn})^2$	$v_{fn} \sim \text{Gaussian}\left(\hat{v}_{fn}, \sigma^2\right)$	$\prod_{f,n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(v_{fn}-\hat{v}_{fn})^2}{2\sigma^2}\right)$
KL	$\sum_{f,n} \left(v_{fn} \log \frac{v_{fn}}{\hat{v}_{fn}} - v_{fn} + \hat{v}_{fn}\right)$	$v_{fn} \sim \text{Poisson}\left(\hat{v}_{fn}\right)$	$\prod_{f,n} \frac{1}{\Gamma(v_{fn}+1)} \hat{v}_{fn}^{v_{fn}} \exp\left(-\hat{v}_{fn}\right)$
IS	$\sum_{f,n} \left(\frac{v_{fn}}{\hat{v}_{fn}} - \log \frac{v_{fn}}{\hat{v}_{fn}} - 1\right)$	$v_{fn} \sim \text{Exponential}\left(\frac{1}{\hat{v}_{fn}}\right)$	$\prod_{f,n} \frac{1}{\hat{v}_{fn}} \exp\left(-\frac{v_{fn}}{\hat{v}_{fn}}\right)$

Statistical viewpoint

Numerous advantages of a probabilistic NMF formulation:

- possibility of using efficient **probabilistic inference algorithms** such as the Expectation-Maximization (EM) algorithm (Févotte et al., 2009) and the Monte Carlo methods (Cemgil, 2009b; Schmidt et al., 2009);
- possibility of **introducing various constraints** into NMF modeling via prior distributions (Arngren et al., 2011);
- possibility of learning the NMF from **partially missing** (Roux et al., 2011) or **noisy** (Arberet et al., 2012) data;
- possibility of combining the NMF with **different probabilistic models** (Ozerov et al., 2012), *e.g.*, the hidden Markov models (HMMs) (Ozerov et al., 2009).

Weighted NMF

Conventional NMF optimization criterion (separable divergence case):

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

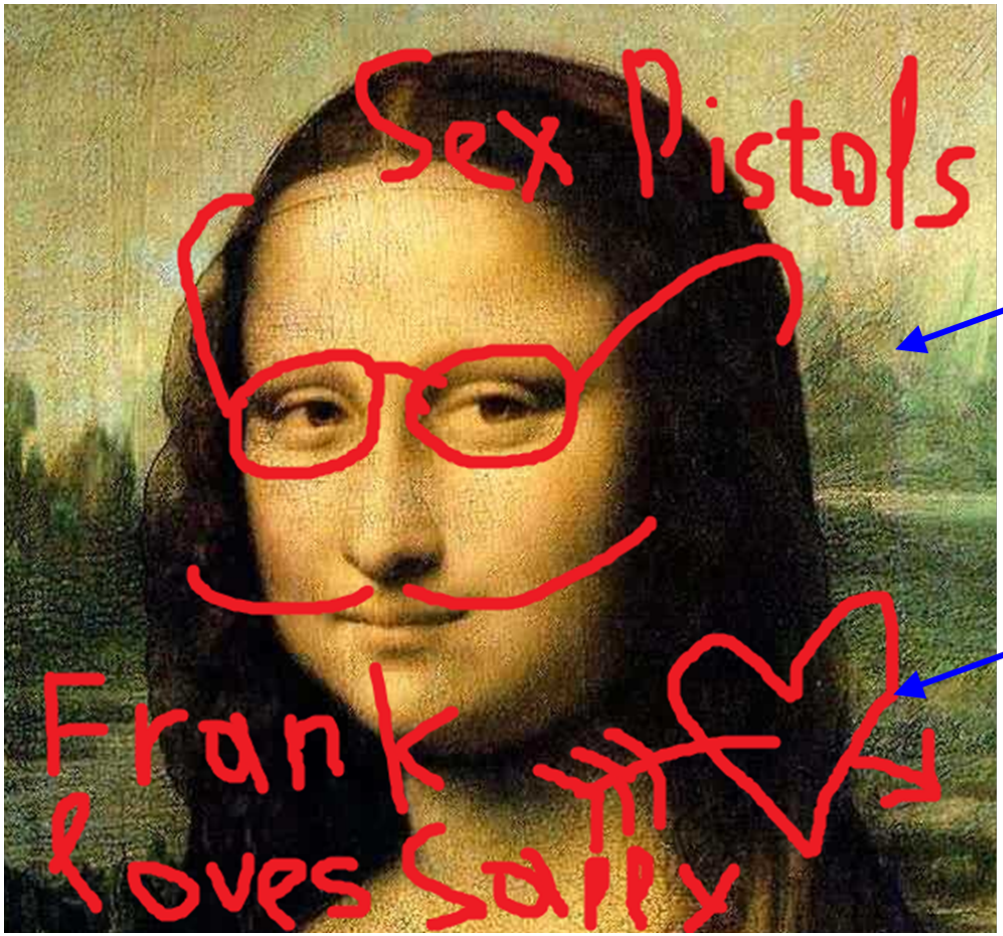
Weighted NMF optimization criterion:

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

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

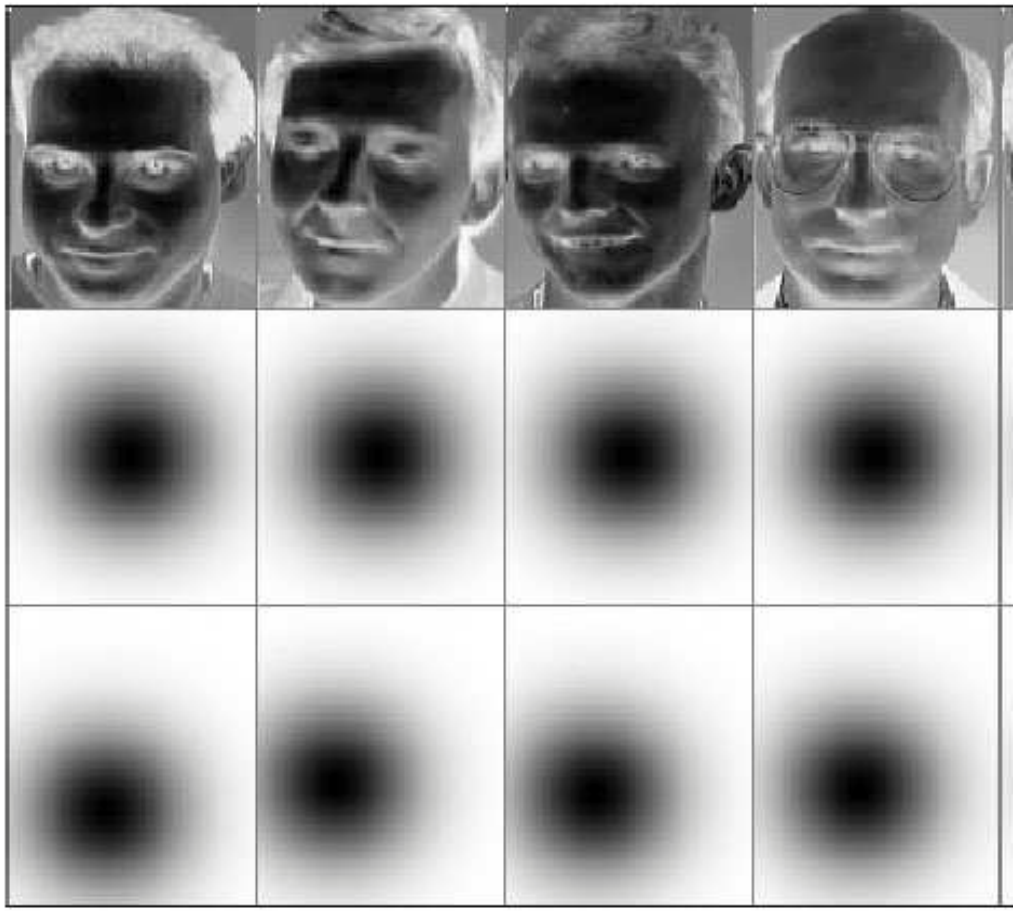
$b_{fn} = 1$

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)
- NMF models
- Algorithms for solving NMF
  - Preliminaries
  - Multiplicative update rules
  - Model order selection, initialization and stopping criteria
- Constrained NMF schemes
- Multi-stream and cross-modal NMF schemes
- Applications

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.

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):

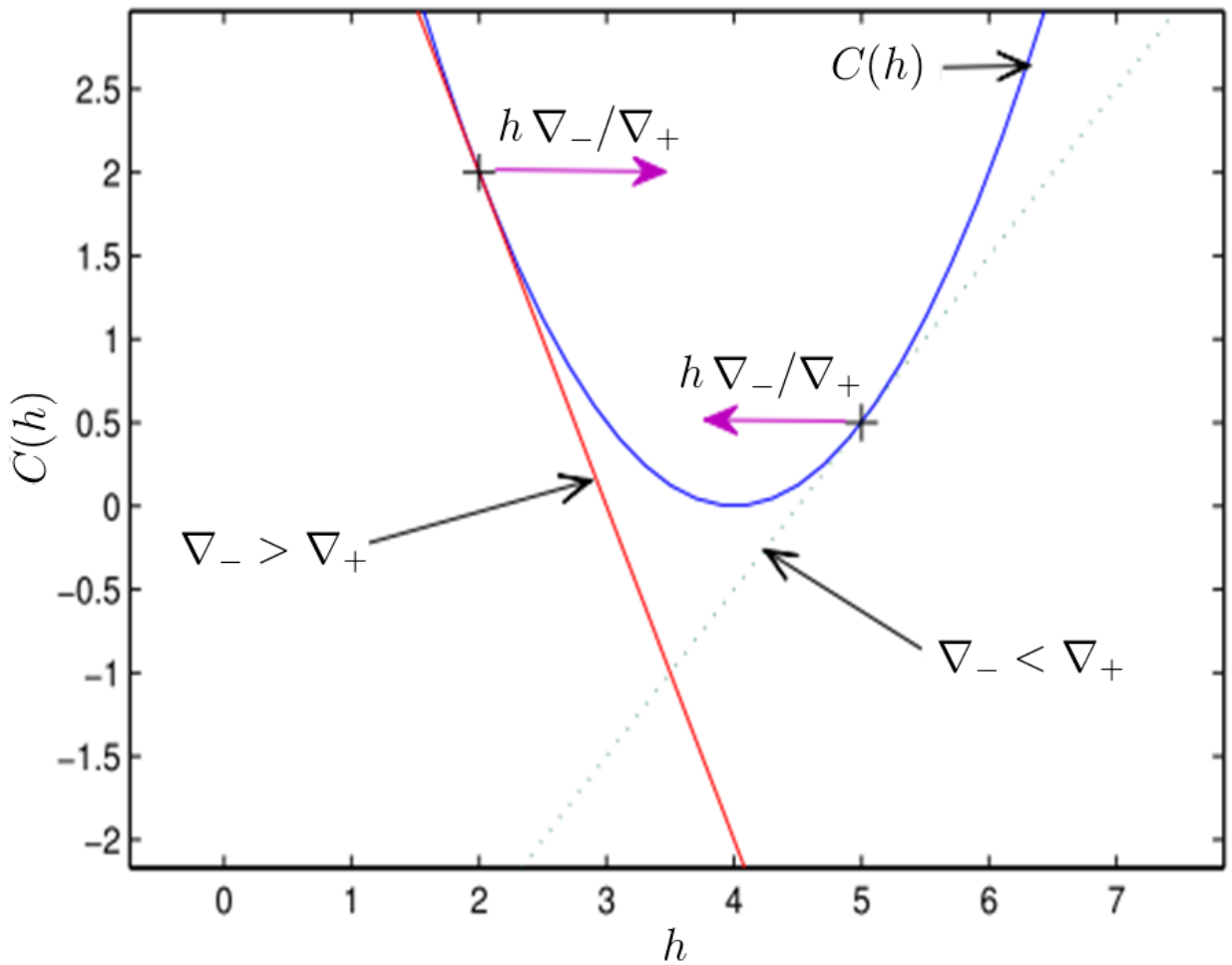
$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T \left((\mathbf{WH})^{\cdot[\beta-2]} \odot \mathbf{V} \right)}{\mathbf{W}^T (\mathbf{WH})^{\cdot[\beta-1]}},$$

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

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

Intuitive explanation

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



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 the other algorithm 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 $C(s)$.

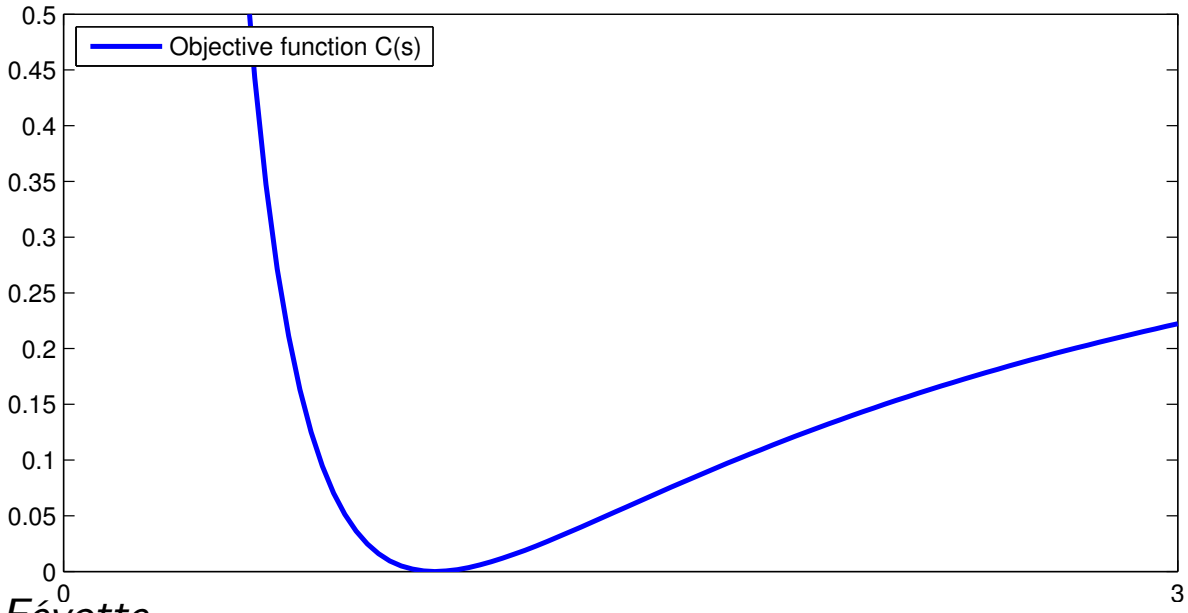


Illustration by C. Févotte

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 $C(s)$.

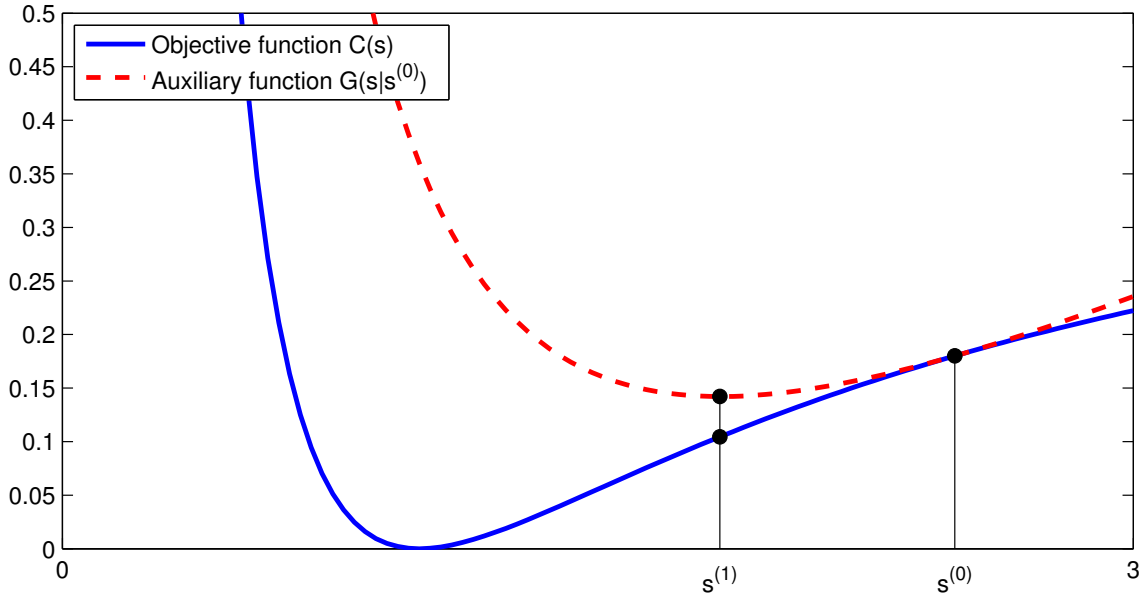


Illustration by C. Févotte

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 $C(s)$.

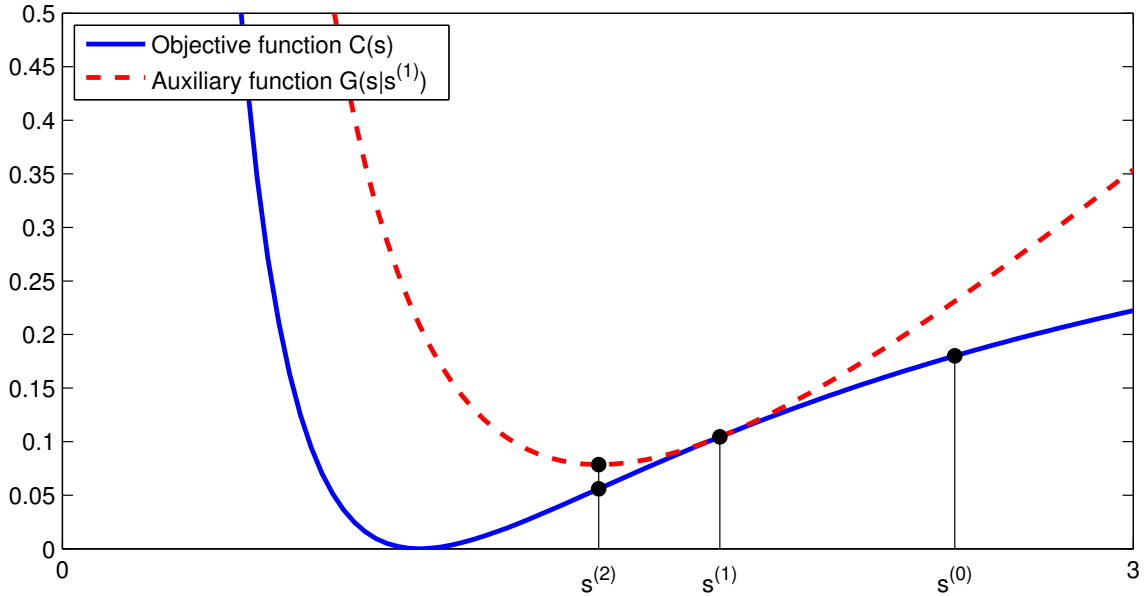


Illustration by C. Févotte

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 $C(s)$.

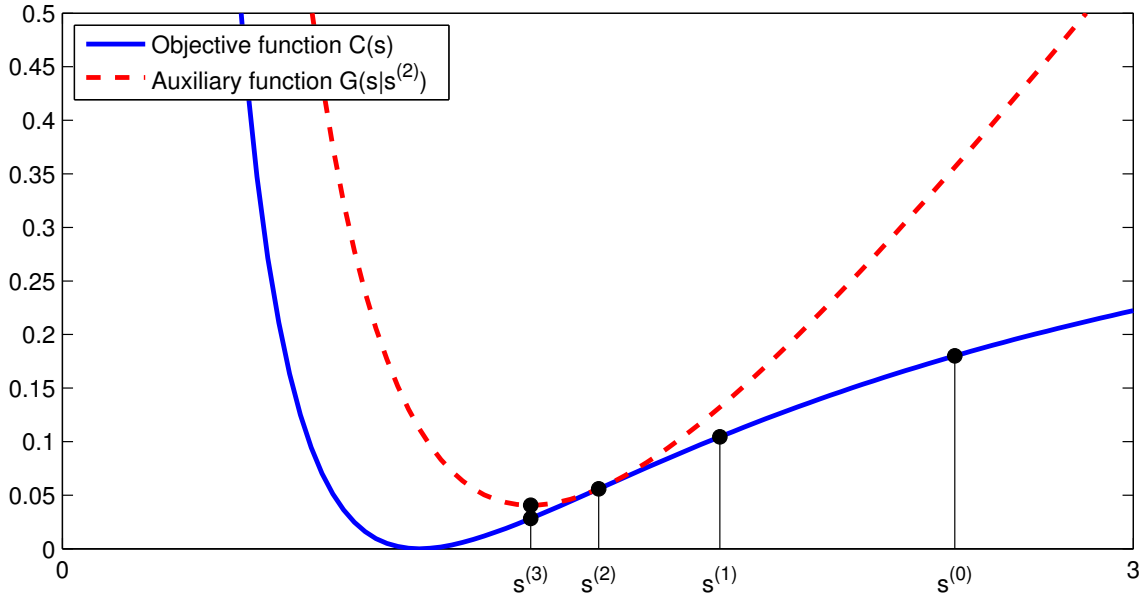


Illustration by C. Févotte

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 $C(s)$.

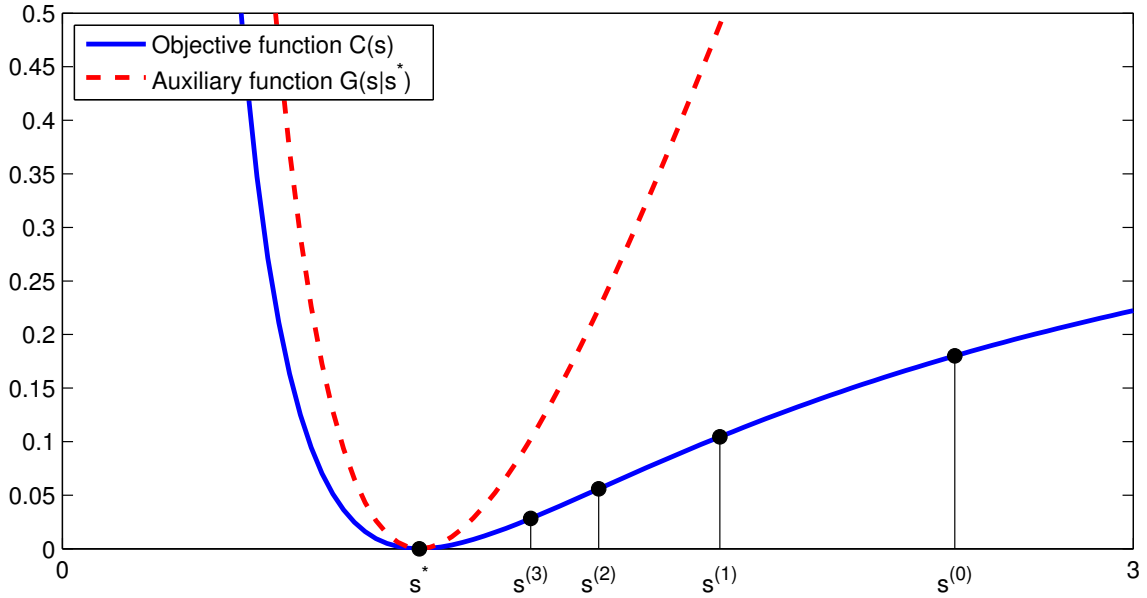


Illustration by C. Févotte

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 $\mathbf{W}$ and $\mathbf{H}$ 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, 2009a)

- 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?

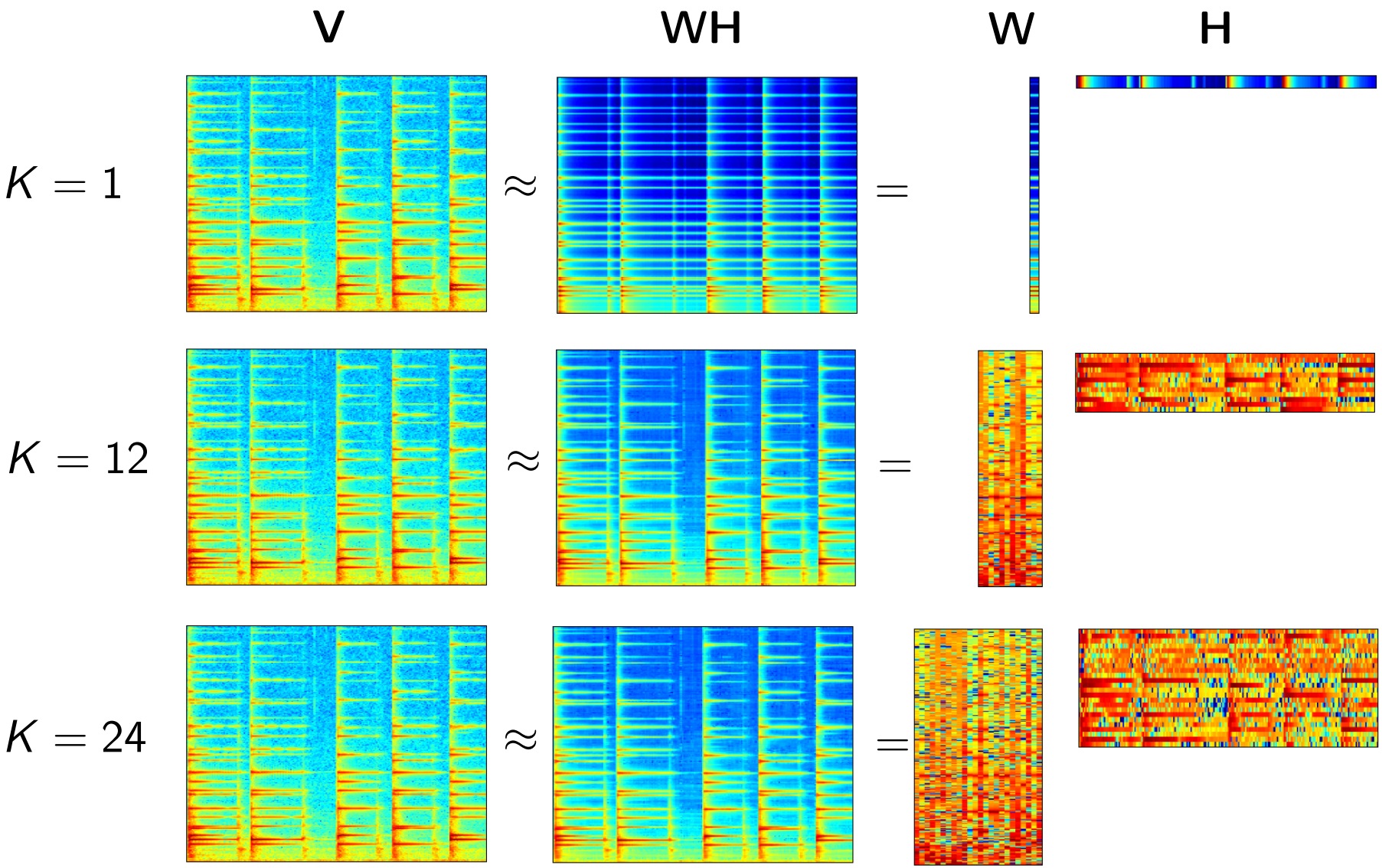
A right **model order choice is important** and it depends on the data $\mathbf{V}$ and on the application.

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

- **Model order K is fixed** during the NMF decomposition, and it was
  - either chosen by intuition,
  - either chosen based on some prior knowledge (*e.g.*, known number of clusters for clustering),
  - or trained on some development data within a particular application.
- **Model order K is estimated 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);
- etc ...