Unsupervised data decompositions

Slim ESSID

Télécom ParisTech

slim.essid@telecom-paristech.fr

Slides by Cédric Févotte (cfevotte@unice.fr)

Objectives

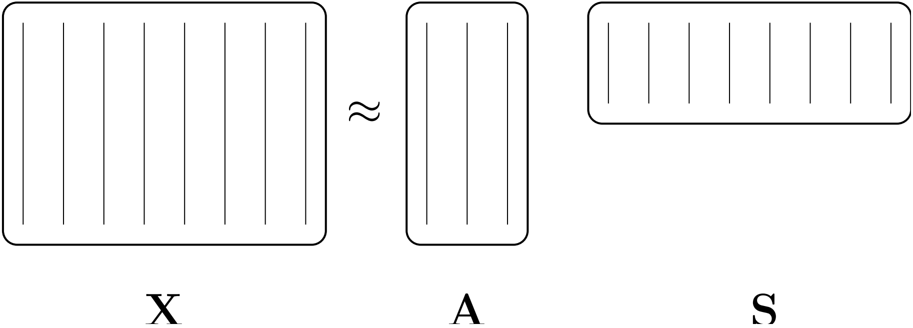
We search for **unsupervised decompositions** of data such that

$\mathbf{x}_n$	$\approx$	$\mathbf{A}$	$\mathbf{s}_n$
data vector		"explanatory variables"	"regressors"
		"basis", "dictionary"	"expansion coefficients"
		"patterns"	"activation coefficients"

and $\mathbf{A}$ is learnt from a set of data vectors $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_N]$.

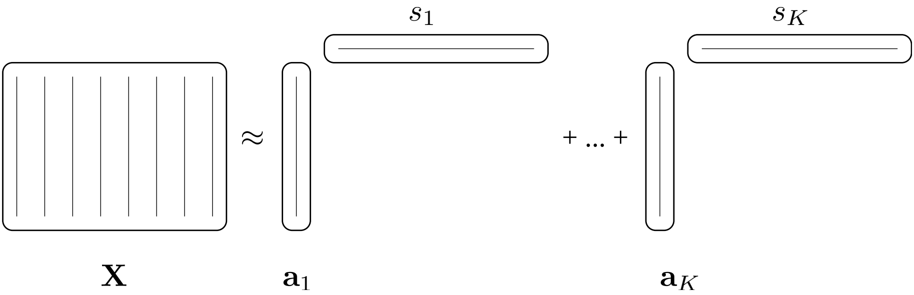
- $\mathbf{x}_n$ is a vector of size F
- $\mathbf{s}_n$ is a vector of size K
- $\mathbf{A}$ is a matrix of size $F \times K$, with usually $F \geq K$.

Example: dimensionality reduction



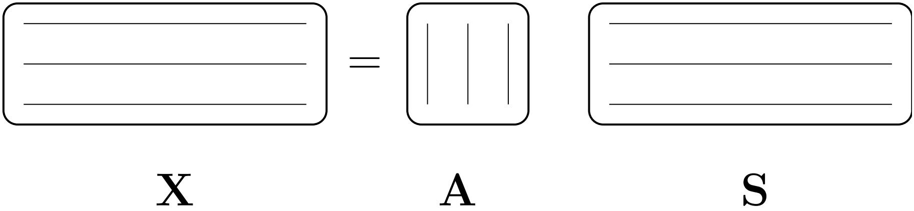
The $F \times N$ data matrix is approximated by $K(F+N)$ coefficients.

Example: dimensionality reduction (ctd)



The factorization is akin to a rank-K approximation.

Example: source separation



The rows of **X** are mixed signals, **A** is a mixing matrix and the sources form the rows of **S**.

Questions

In matrix form, we search for the following factorization

$$\mathbf{X} \approx \mathbf{AS}$$

- What should the “$\approx$” entail ?
- What constraints should be imposed on $\mathbf{A}$ and/or $\mathbf{S}$?

Recalling PCA

- The data is assumed real-valued ($\mathbf{x}_n \in \mathbb{R}^F$) and centered ($\mathrm{E}\{\mathbf{x}_n\} = 0$)
- PCA returns a dictionary $\mathbf{A}_{PCA} \in \mathbb{R}^{F \times K}$ such that the least squares error is minimized:

$$\mathbf{A}_{PCA} = \min_{\mathbf{A}} \frac{1}{N} \sum_n \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|_2^2 = \frac{1}{N} \|\mathbf{X} - \mathbf{A}\mathbf{A}^T \mathbf{X}\|_F^2$$

- The solution can be shown to be of the form

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

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

$$\mathbf{C}_\mathbf{x} = \mathrm{E}\{\mathbf{x}\mathbf{x}^T\} \approx \frac{1}{N} \sum_n \mathbf{x}_n \mathbf{x}_n^T$$

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

Recalling PCA

- The data is assumed real-valued ($\mathbf{x}_n \in \mathbb{R}^F$) and centered ($\mathrm{E}\{\mathbf{x}_n\} = 0$)
- PCA returns a dictionary $\mathbf{A}_{PCA} \in \mathbb{R}^{F \times K}$ such that the least squares error is minimized:

$$\mathbf{A}_{PCA} \quad = \quad \min_{\mathbf{A}} \frac{1}{N} \sum_n \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|_2^2 = \frac{1}{N} \|\mathbf{X} - \mathbf{A}\mathbf{A}^T\mathbf{X}\|_F^2$$

- The solution can be shown to be of the form

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

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

$$\mathbf{C}_\mathbf{x} = \mathrm{E}\{\mathbf{x}\mathbf{x}^T\} \approx \frac{1}{N} \sum_n \mathbf{x}_n \mathbf{x}_n^T$$

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

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

where $\{d_i\}_i$ are the eigenvalues of $\mathbf{C}_\mathbf{x}$, sorted in order of decreasing value.

PCA can be used for compression: the original $F \times N$ data matrix $\mathbf{X}$ can be approximated by $FK + KN$ coefficients of the matrices $\mathbf{A}_{PCA}$ and $\mathbf{S}_{PCA} = \mathbf{A}_{PCA}^T\mathbf{X}$.

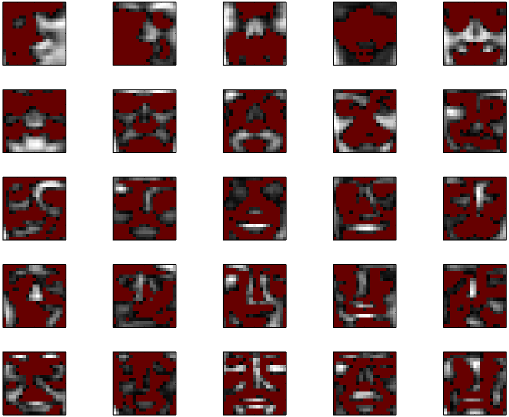
Uncorrelatedness

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

$$\begin{aligned} \mathbf{C_s} &= \mathrm{E}\{(\mathbf{A}^T\mathbf{x})(\mathbf{A}^T\mathbf{x})^T\} \\ &= \mathbf{A}^T\mathbf{C_x}\mathbf{A} \\ &= \mathrm{diag}\left([d_1, \ldots, d_K]\right) \\ &\overset{\text{def}}{=} \mathbf{D}_K \end{aligned}$$

49 images among 2429 from MIT’s CBCL face dataset

PCA dictionary with $K = 25$



(*Red pixels indicate negative values*)

1. Nonnegative matrix factorization (NMF)
   - Concept
   - Majorization-minimization algorithms
   - Audio examples

2. Independent Component Analysis (ICA)
   - Sphering
   - Concept
   - Nongaussian is independent
   - FastICA algorithms

Nonnegative matrix factorization (NMF)

Data is often nonnegative by nature

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

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

Nonnegative matrix factorization (NMF)

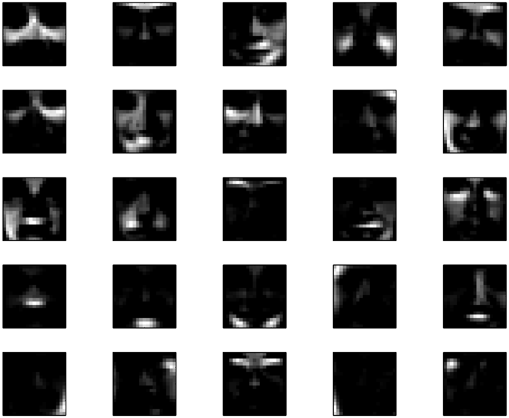
Given nonnegative data $\mathbf{X}$, NMF is the problem of finding a factorization

$$\mathbf{X} \approx \mathbf{AS}$$

such that $\mathbf{A}$ and $\mathbf{S}$ are *nonnegative* matrices of dimensions $F \times K$ and $K \times N$, respectively.

- **nonneg. of A** ensures *interpretability* of the dictionary (features $\mathbf{a}_k$ and data $\mathbf{x}_n$ belong to same space).
- **nonneg. of S** produces *part-based* representations because subtractive combinations are forbidden.

NMF dictionary with $K = 25$



(*as shown in* (Lee &Seung, 99))

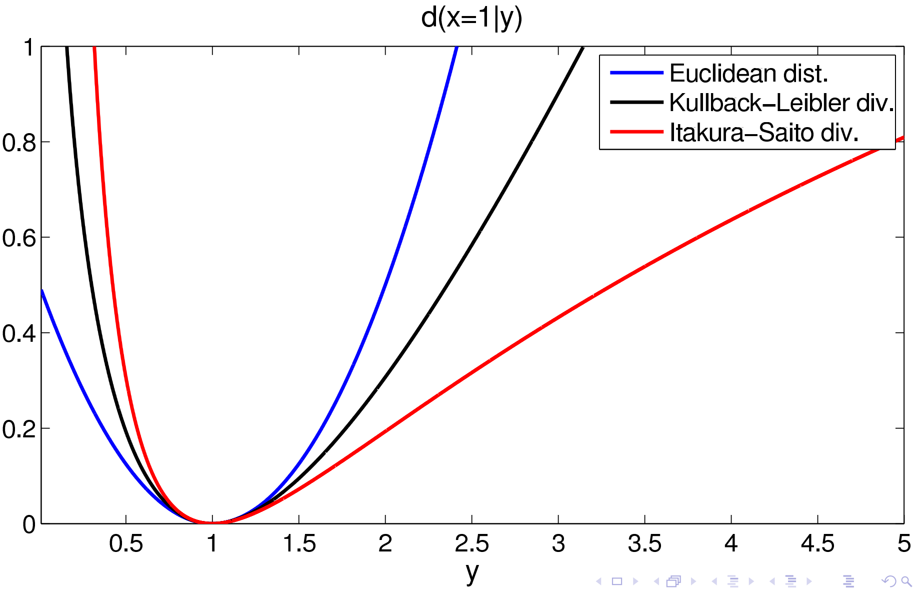
NMF is a constrained minimization problem

$$\min_{\mathbf{A},\mathbf{S}\geq\mathbf{0}} D(\mathbf{X}|\mathbf{AS}) = \sum_{fn} d([\mathbf{X}]_{fn}|[\mathbf{AS}]_{fn})$$

where $d(x|y)$ is a measure of fit. E.g:

Euclidean distance :	$d_{EUC}(x|y) = \frac{1}{2}(x-y)^2$
Kullback-Leibler (KL) divergence :	$d_{KL}(x|y) = x \log\frac{x}{y} - x + y$
Itakura-Saito (IS) divergence :	$d_{IS}(x|y) = \frac{x}{y} - \log\frac{x}{y} - 1$

Usual measures of fit



Variance to scale

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

The IS divergence is scale-invariant. Implies higher accuracy in the representation of data with large dynamic range, such as audio spectra.

Algorithms

We describe iterative algorithms that update $\mathbf{A}$ given $\mathbf{S}$ and $\mathbf{S}$ given $\mathbf{A}$.

The optimization problem is symmetric in $\mathbf{A}$ and $\mathbf{S}$ because $\mathbf{X} \approx \mathbf{AS} \iff \mathbf{X}^T \approx \mathbf{S}^T \mathbf{A}^T$.

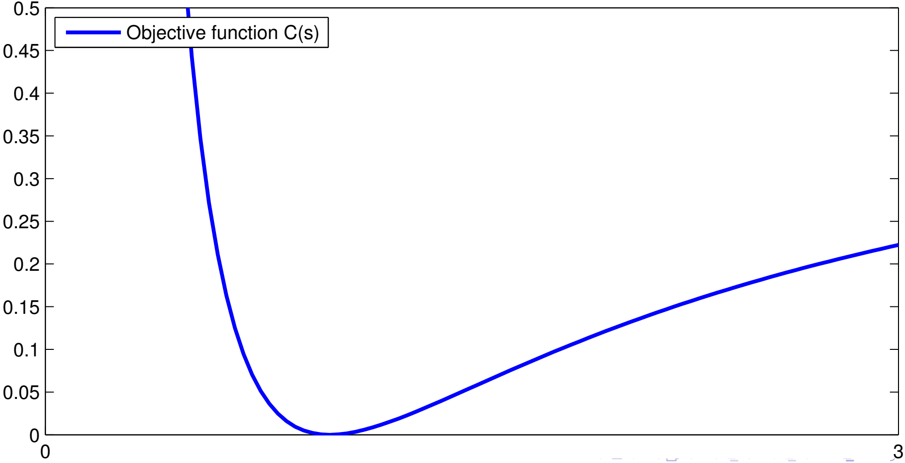
The optimization is separable : $D(\mathbf{X}|\mathbf{AS}) = \sum_n D(\mathbf{x}_n|\mathbf{As}_n)$.

We are essentially left with

$$\min_{\mathbf{s} \geq \mathbf{0}} C(\mathbf{s}) = D(\mathbf{x}|\mathbf{As}) = \sum_f d(x_f|[\mathbf{As}]_f)$$

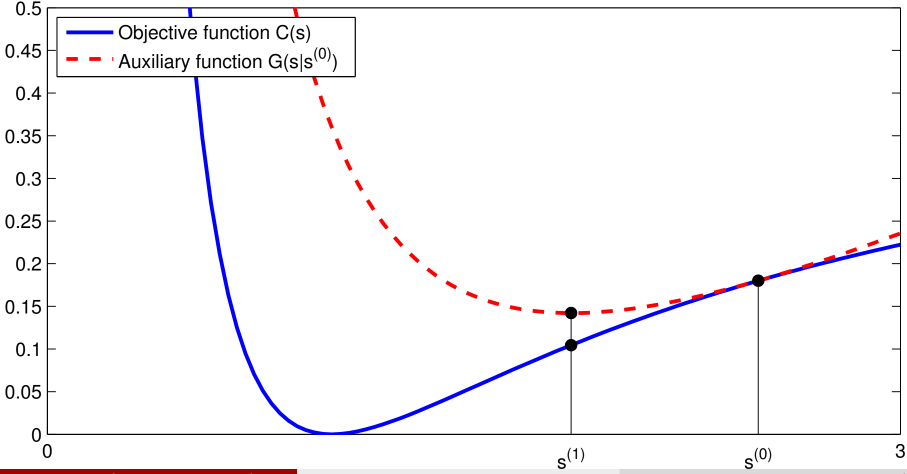
Majorization-minimization (MM)

Build $G(\mathbf{s}|\tilde{\mathbf{s}})$ such that $G(\mathbf{s}|\tilde{\mathbf{s}}) \geq C(\mathbf{s})$ and $G(\tilde{\mathbf{s}}|\tilde{\mathbf{s}}) = C(\tilde{\mathbf{s}})$.
Optimize (iteratively) $G(\mathbf{s}|\tilde{\mathbf{s}})$ instead of $C(\mathbf{s})$.



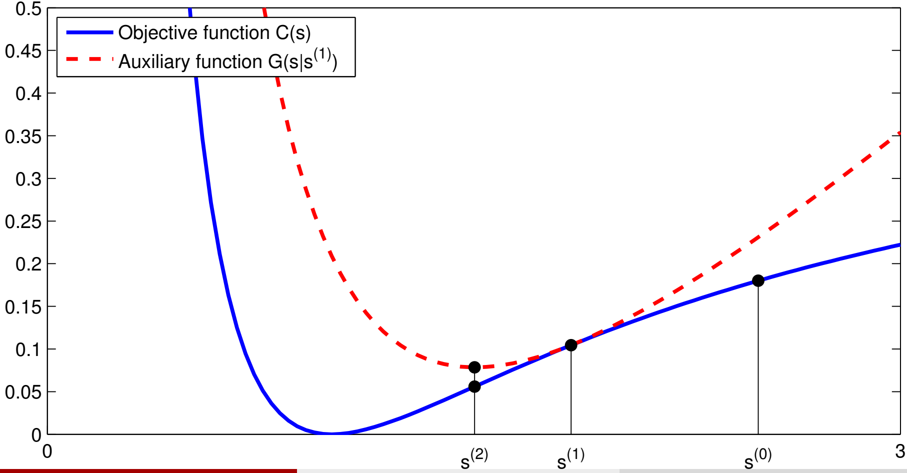
Majorization-minimization (MM)

Build $G(\mathbf{s}|\tilde{\mathbf{s}})$ such that $G(\mathbf{s}|\tilde{\mathbf{s}}) \geq C(\mathbf{s})$ and $G(\tilde{\mathbf{s}}|\tilde{\mathbf{s}}) = C(\tilde{\mathbf{s}})$.
Optimize (iteratively) $G(\mathbf{s}|\tilde{\mathbf{s}})$ instead of $C(\mathbf{s})$.



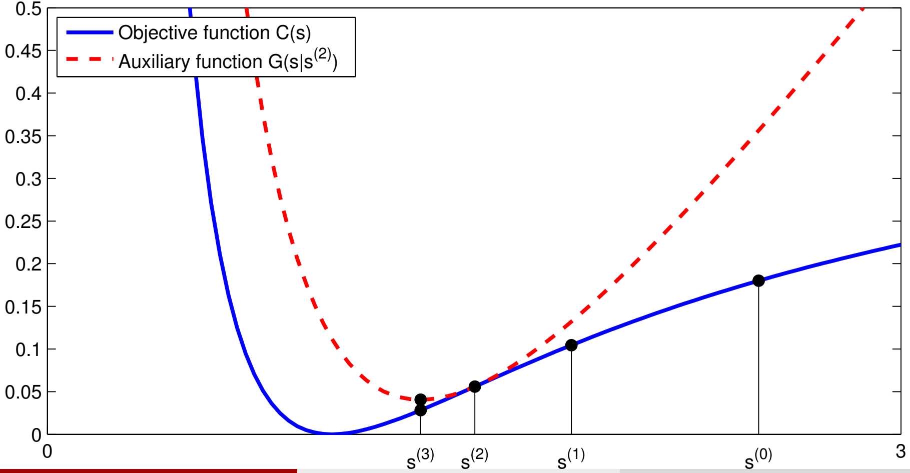
Majorization-minimization (MM)

Build $G(\mathbf{s}|\tilde{\mathbf{s}})$ such that $G(\mathbf{s}|\tilde{\mathbf{s}}) \geq C(\mathbf{s})$ and $G(\tilde{\mathbf{s}}|\tilde{\mathbf{s}}) = C(\tilde{\mathbf{s}})$.
Optimize (iteratively) $G(\mathbf{s}|\tilde{\mathbf{s}})$ instead of $C(\mathbf{s})$.



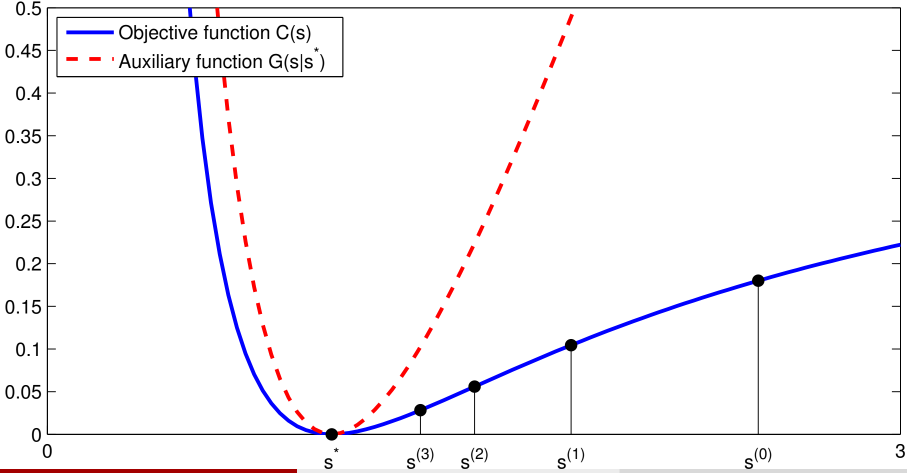
Majorization-minimization (MM)

Build $G(\mathbf{s}|\tilde{\mathbf{s}})$ such that $G(\mathbf{s}|\tilde{\mathbf{s}}) \geq C(\mathbf{s})$ and $G(\tilde{\mathbf{s}}|\tilde{\mathbf{s}}) = C(\tilde{\mathbf{s}})$.
Optimize (iteratively) $G(\mathbf{s}|\tilde{\mathbf{s}})$ instead of $C(\mathbf{s})$.



Majorization-minimization (MM)

Build $G(\mathbf{s}|\tilde{\mathbf{s}})$ such that $G(\mathbf{s}|\tilde{\mathbf{s}}) \geq C(\mathbf{s})$ and $G(\tilde{\mathbf{s}}|\tilde{\mathbf{s}}) = C(\tilde{\mathbf{s}})$.
Optimize (iteratively) $G(\mathbf{s}|\tilde{\mathbf{s}})$ instead of $C(\mathbf{s})$.



Majorization-minimization algorithms (ctd)

For convex cost functions $d(x|y)$ (w.r.t y), an auxiliary function can be constructed using Jensen's inequality. Let us denote $\tilde{\lambda}_{fk} = \frac{a_{fk}\tilde{s}_k}{[\mathbf{A}\tilde{\mathbf{s}}]_f}$, such that $\sum_k \tilde{\lambda}_{fk} = 1$. Then, we may write

$$\begin{aligned}
C(\mathbf{s}) &= D(\mathbf{x}|\mathbf{A}\mathbf{s}) \\
&= \sum_f d(x_f | \sum_k \tilde{\lambda}_{fk}.\frac{a_{fk}s_k}{\tilde{\lambda}_{fk}}) \\
&\leq \sum_{fk} \tilde{\lambda}_{fk}\, d(x_f | \frac{a_{fk}s_k}{\tilde{\lambda}_{fk}}) \\
&= \sum_{fk} \frac{a_{fk}\tilde{s}_k}{[\mathbf{A}\tilde{\mathbf{s}}]_f}\, d\left(x_f | [\mathbf{A}\tilde{\mathbf{s}}]_f \frac{s_k}{\tilde{s}_k}\right) \\
&= G(\mathbf{s}|\tilde{\mathbf{s}})
\end{aligned}$$

Majorization-minimization algorithms (ctd)

For the Itakura-Saito divergence, the convex part can be majorized with Jensen's inequality, and the concave part by the tangent of $C(\mathbf{s})$ in $\tilde{\mathbf{s}}$.

In the end, optimization of $G(\mathbf{s}|\tilde{\mathbf{s}})$ w.r.t to $\mathbf{s}$ leads to the following multiplicative update rules :

$$\mathbf{s} = \tilde{\mathbf{s}}.\frac{\mathbf{A}^T[(\mathbf{A}\tilde{\mathbf{s}})^{\cdot[\beta-2]}.\mathbf{x}]}{\mathbf{A}^T[(\mathbf{A}\tilde{\mathbf{s}})^{\cdot[\beta-1]}]}$$

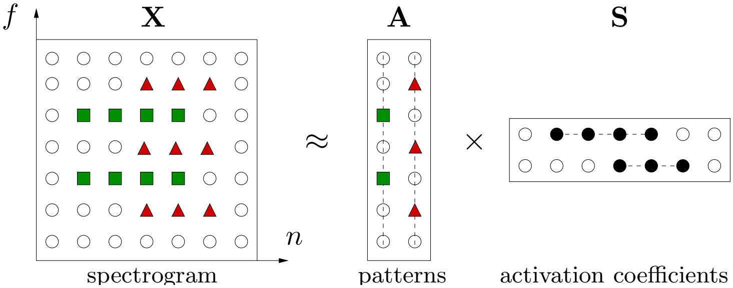
with

Cost	Euclidean	KL	IS
β	2	1	0

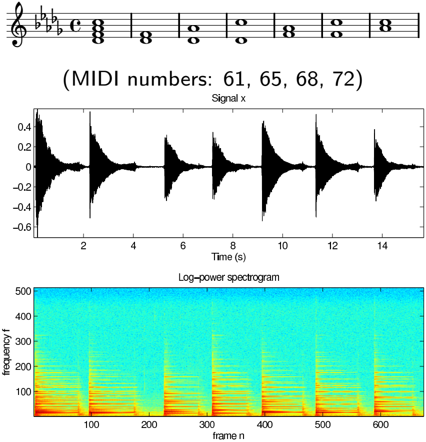
Majorization-minimization algorithms (ctd)

```
Input: nonnegative matrix $\mathbf{X}$
Output: nonnegative matrices $\mathbf{A}$ and $\mathbf{S}$ such that $\mathbf{X} \approx \mathbf{AS}$
Initialize $\mathbf{A}$ and $\mathbf{S}$ with nonnegative values
for $i = 1 : n_{iter}$ do
    $$\mathbf{S} \leftarrow \mathbf{S}.\frac{\mathbf{A}^T\left((\mathbf{AS})^{\cdot[\beta-2]}.\mathbf{X}\right)}{\mathbf{A}^T(\mathbf{AS})^{\cdot[\beta-1]}}$$
    $$\mathbf{A} \leftarrow \mathbf{A}.\frac{\left((\mathbf{AS})^{\cdot[\beta-2]}.\mathbf{X}\right)\mathbf{S}^T}{(\mathbf{AS})^{\cdot[\beta-1]}\,\mathbf{S}^T}$$
    Normalize $\mathbf{A}$ and $\mathbf{S}$
end for
```

Polyphonic music decomposition



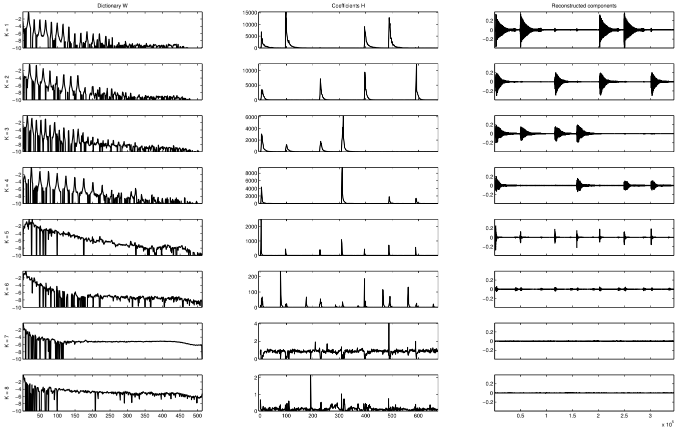
Small-scale example



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

Figure: Three representations of data.

IS-NMF on power spectrogram with $K = 8$



Pitch estimates: 65.0 68.0 61.0 72.0 0 0 0 0

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

References

- D. D. Lee and H. S. Seung. *Learning the parts of objects by non-negative matrix factorization*. Nature, 1999. [online]
- C. Févotte, N. Bertin and J.-L. Durrieu. *Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis*. Neural Computation, 2009. [online]

Independent Component Analysis (ICA)

Sphering (aka whitening)

Besides decorrelation, the variance of the entries of $\mathbf{s}$ can be normalized to 1. This achieved for $\mathbf{S}_{SPH} = \mathbf{A}_{SPH}^T \mathbf{X}$ where

$$\mathbf{A}_{SPH} = \mathbf{E}_{1:K} \mathbf{D}_K^{-\frac{1}{2}}$$

Remark

The sphering matrix $\mathbf{A}_{SPH}$ is not unique. Indeed, for any unitary matrix $\mathbf{U}$ of size $K \times K$, the matrix $(\mathbf{A}_{SPH}\mathbf{U})$ is also a sphering matrix, as we may write

$$\begin{aligned} \mathrm{E}\{(\mathbf{A}_{SPH}\mathbf{U})^T \mathbf{x}\mathbf{x}^T (\mathbf{A}_{SPH}\mathbf{U})\} &= \mathbf{U}^T \mathbf{A}_{SPH}^T \mathbf{C}_\mathbf{x} \mathbf{A}_{SPH} \mathbf{U} \\ &= \mathbf{U}^T \mathbf{U} \\ &= \mathbf{I} \end{aligned}$$

Sphering (aka whitening)

Besides decorrelation, the variance of the entries of $\mathbf{s}$ can be normalized to 1. This achieved for $\mathbf{S}_{SPH} = \mathbf{A}_{SPH}^T\mathbf{X}$ where

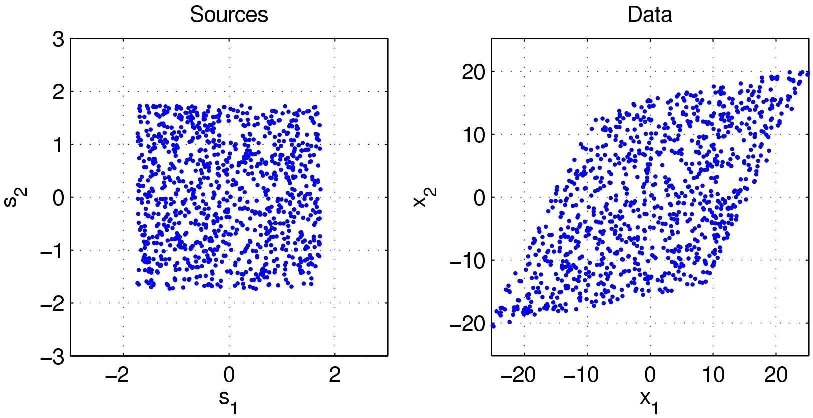
$$\mathbf{A}_{SPH} = \mathbf{E}_{1:K}\mathbf{D}_K^{-\frac{1}{2}}$$

Remark

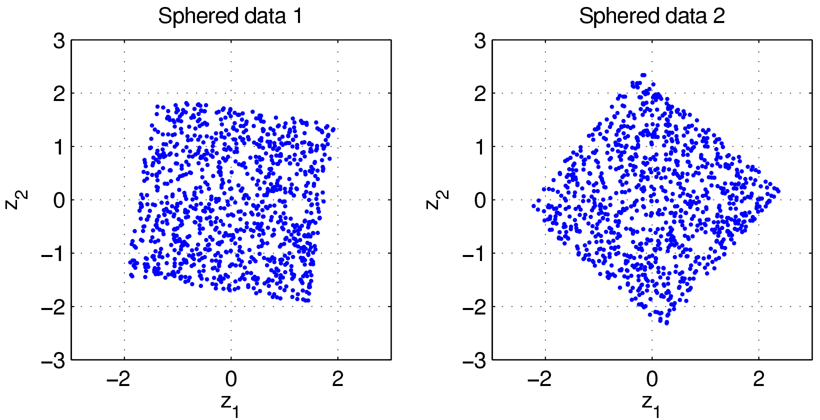
The sphering matrix $\mathbf{A}_{SPH}$ is not unique. Indeed, for any unitary matrix $\mathbf{U}$ of size $K \times K$, the matrix $(\mathbf{A}_{SPH}\mathbf{U})$ is also a sphering matrix, as we may write

$$\begin{aligned}\mathrm{E}\{(\mathbf{A}_{SPH}\mathbf{U})^T\mathbf{x}\mathbf{x}^T(\mathbf{A}_{SPH}\mathbf{U})\} &= \mathbf{U}^T\mathbf{A}_{SPH}^T\mathbf{C}_\mathbf{x}\mathbf{A}_{SPH}\mathbf{U} \\ &= \mathbf{U}^T\mathbf{U} \\ &= \mathbf{I}\end{aligned}$$

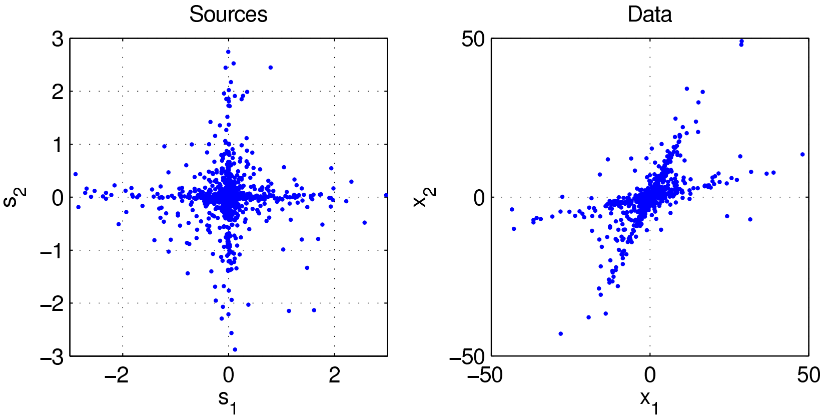
Example : uniform coefficients



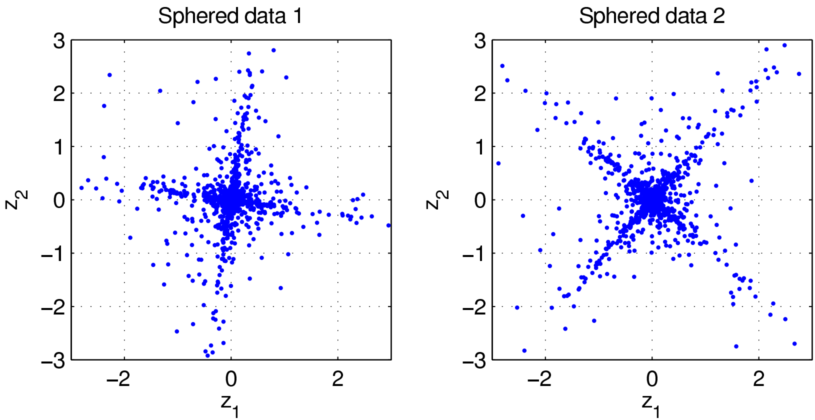
Example : uniform coefficients (ctd)



Example : sparse coefficients



Example : sparse coefficients (ctd)



Concept

Sphering returns coefficients $\mathbf{z} = \mathbf{A}_{SPH}^T\mathbf{x}$ that are uncorrelated and with unit variance, i.e., $\mathrm{E}\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}$.

Sphering is not unique as any rotation $\mathbf{Uz}$ is also white. Hence, one may choose the arbitrary rotation $\mathbf{U}$ so that $\mathbf{Uz}$ satisfies an additional criterion.

ICA aims at finding $\mathbf{U}_{ICA}$ so that the components of

$$\mathbf{s}_{ICA} = \mathbf{U}_{ICA}\mathbf{z} = \mathbf{U}_{ICA}\mathbf{A}_{SPH}^T\mathbf{x}$$

are sphered and **mutually independent**.

Concept (ctd)

Assume for simplicity that $F = K$. In other words, ICA decomposes the data as

$$\mathbf{x} = \mathbf{A}_{ICA}\,\mathbf{s}$$

such that the entries of $\mathbf{s}$ are mutually independent :

$$p(\mathbf{s}) = \prod_k p(s_k)$$

Given what precedes, ICA can be achieved in two steps :

1) Sphere the observations as $\mathbf{z} = \mathbf{A}_{SPH}^T\,\mathbf{x}$,
2) Find $\mathbf{U}_{ICA}$ such that the entries of $\mathbf{U}_{ICA}\,\mathbf{z}$ are mutually independent.

Concept (ctd)

Hence, in practice, given sphered data $\mathbf{z}_n = \mathbf{A}_{SPH}^T \mathbf{x}_n$ we need to

1) Construct a numerical criterion $C(\mathbf{Y})$ measuring the independence of the entries of the random vector $\mathbf{y}$ given realizations $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$.
2) Solve the following optimization problem

$$\max_{\mathbf{U}} C(\mathbf{UZ})$$

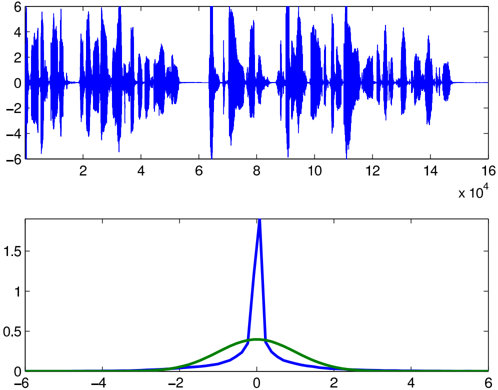
Nongaussian is independent

The Central Limit Theorem tells us that the distribution of the sum of independent random variables tends towards a gaussian distribution.

Basically, it implies that the sum of two random variables is “more gaussian” than the original random variables.

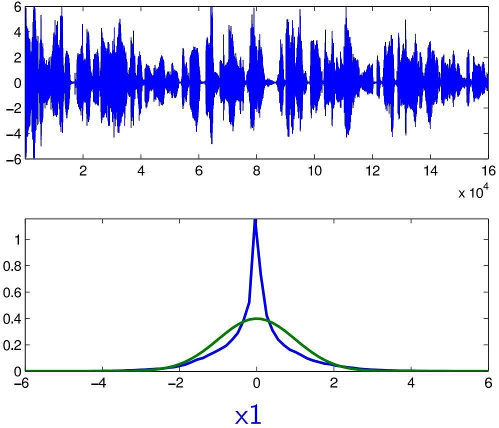
This suggests that the entries of $\mathbf{y} = \mathbf{U}\mathbf{z}$ should be searched as *nongaussian* as possible.

Nongaussian is independent (ctd)

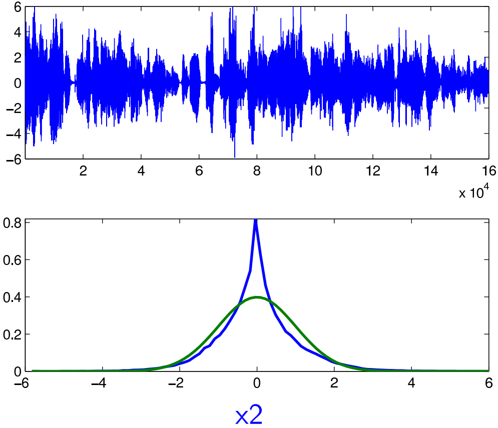


s1

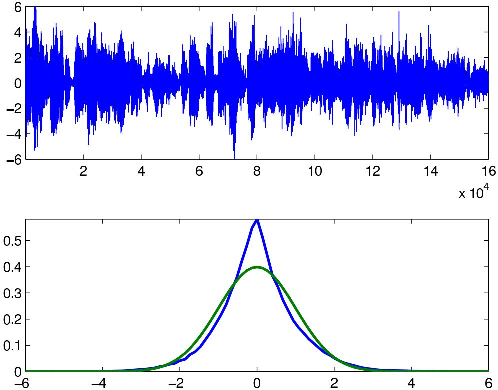
Nongaussian is independent (ctd)



Nongaussian is independent (ctd)

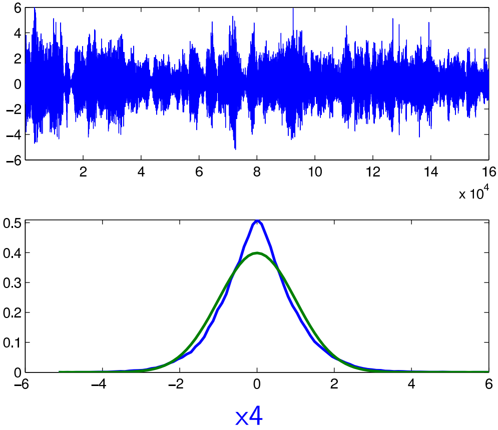


Nongaussian is independent (ctd)

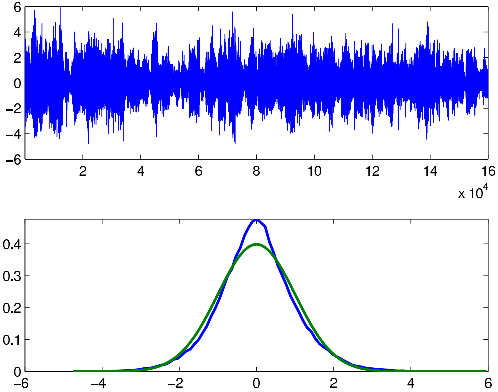


x3

Nongaussian is independent (ctd)

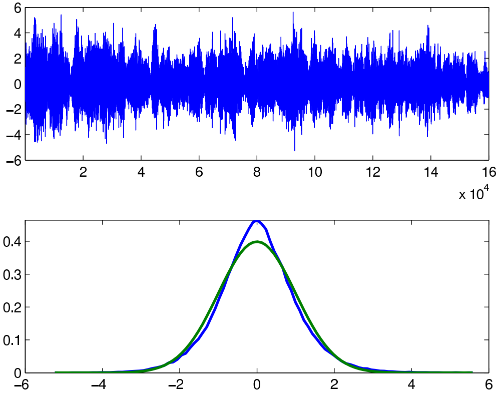


Nongaussian is independent (ctd)



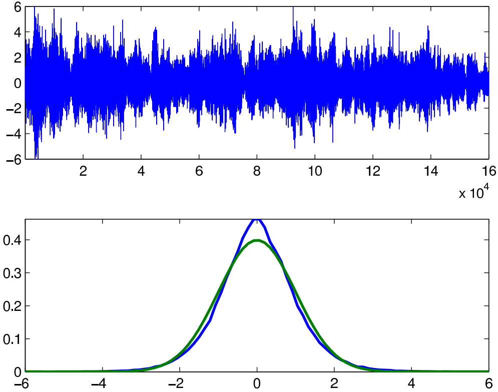
x5

Nongaussian is independent (ctd)



x6

Nongaussian is independent (ctd)



x7

Nongaussian is independent (ctd)

This intuition can be made rigorous via the properties of *mutual information*, defined as

$$\begin{aligned} I\{\mathbf{y}\} &= KL[p(\mathbf{y})|\prod_f p(y_f)] \\ &= \int p(\mathbf{y}) \log \frac{p(\mathbf{y})}{\prod_f p(y_f)} d\mathbf{y} \\ &= \sum_f H\{y_f\} - H\{\mathbf{y}\} \end{aligned}$$

where $H\{\mathbf{y}\} = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y}$ denotes *differential entropy*.

Nongaussian is independent (ctd)

The mutual information of $\mathbf{y} = \mathbf{U}\mathbf{z}$ can be written

$$\begin{aligned} I\{\mathbf{y}\} &= \sum_f H\{y_f\} - H\{\mathbf{z}\} - \log|\det \mathbf{U}| \\ &= \sum_f H\{y_f\} + cst \end{aligned}$$

Hence, because the Gaussian is the distribution with highest entropy (for given variance), minimizing $I\{\mathbf{y}\}$ (i.e., enforcing mutual independence) is equivalent to minimizing $\sum_f H\{y_f\}$ (i.e., enforcing nongaussianity).

Identifiability of ICA

This discussion implies that ICA cannot separate gaussian sources.

This is because the sum of gaussian random variables is itself gaussian.

The ICA model $\mathbf{X} = \mathbf{AS}$ (with $F = K$) is identifiable (up to scale and order ambiguities) when at most one source is gaussian.

Measures of nongaussianity

A quantitative measure of nongaussianity is the *kurtosis*, defined by

$$\mathrm{kurt}\{y\} = \mathrm{E}\{y^4\} - 3\mathrm{E}\{y^2\}^2$$

- The Gaussian distribution has zero kurtosis
- Distributions "flatter" than the Gaussian are called *subgaussian* and have kurtosis < 0
- Distributions "peakier" than the Gaussian are called *supergaussian* and have kurtosis > 0.

Another common measure of nongaussianity is the *negentropy*, defined by

$$J\{y\} = H\{y_G\} - H\{y\}$$

where y_G denotes a Gaussian variable with same variance as y.

Measures of nongaussianity

A quantitative measure of nongaussianity is the *kurtosis*, defined by

$$\text{kurt}\{y\} = \text{E}\{y^4\} - 3\text{E}\{y^2\}^2$$

- The Gaussian distribution has zero kurtosis
- Distributions “flatter” than the Gaussian are called *subgaussian* and have kurtosis < 0
- Distributions “peakier” than the Gaussian are called *supergaussian* and have kurtosis > 0.

Another common measure of nongaussianity is the *negentropy*, defined by

$$J\{y\} = H\{y_G\} - H\{y\}$$

where y_G denotes a Gaussian variable with same variance as y.

FastICA algorithms

Using the kurtosis as a quantitative measure of nongaussianity, we are left with the following optimization problem

$$\max_{\mathbf{U}} \sum_{k} \left|\text{kurt}\{[\mathbf{U}^T\mathbf{z}]_k\}\right| \quad \text{subject to} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

For simplicity, let's first consider the problem of finding only one maximally nongaussian component, i.e, solve

$$\max_{\mathbf{u}} \; C(\mathbf{u}) = \left|\text{kurt}\{\mathbf{u}^T\mathbf{z}\}\right| \quad \text{subject to} \quad \mathbf{u}^T\mathbf{u} = 1$$

FastICA algorithms

Using the kurtosis as a quantitative measure of nongaussianity, we are left with the following optimization problem

$$\max_{\mathbf{U}} \sum_k |\text{kurt}\{[\mathbf{U}^T\mathbf{z}]_k\}| \quad \text{subject to} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}$$

For simplicity, let's first consider the problem of finding only one maximally nongaussian component, i.e, solve

$$\max_{\mathbf{u}} \; C(\mathbf{u}) = |\text{kurt}\{\mathbf{u}^T\mathbf{z}\}| \quad \text{subject to} \quad \mathbf{u}^T\mathbf{u} = 1$$

FastICA algorithms

For sphered, centered data $\mathbf{z}$, the criterion writes

$$C(\mathbf{u}) = |\mathrm{E}\{(\mathbf{u}^T\mathbf{z})^4\} - 3|$$

Its gradient thus writes

$$\nabla_{\mathbf{u}} C(\mathbf{u}) = 4\,\mathrm{sign}(\mathrm{E}\{(\mathbf{u}^T\mathbf{z})^4\} - 3)\,\mathrm{E}\{(\mathbf{u}^T\mathbf{z})^3\mathbf{z}\}$$

FastICA algorithms (ctd)

A suitable projected gradient ascent algorithm writes

```
Initialize $\mathbf{u}^{(0)}$
for $i = 1 : n_{iter}$ do
    $\mathbf{u}^{(i)} \leftarrow \mathbf{u}^{(i-1)} + \alpha^{(i)} \nabla_{\mathbf{u}} C(\mathbf{u}^{(i-1)})$
    $\mathbf{u}^{(i)} \leftarrow \frac{\mathbf{u}^{(i)}}{\|\mathbf{u}^{(i)}\|}$
end for
```

where $\alpha^{(i)}$ is a sequence of positive step sizes.

FastICA algorithms (ctd)

A faster algorithm, free of tuning parameters, may be obtained by observing that a stationary point of the criterion must point in the direction of the gradient.
Indeed the Lagrangian to the original problem

$$\max_{\mathbf{u}} \; C(\mathbf{u}) = |\text{kurt}\{\mathbf{u}^T\mathbf{z}\}| \quad \text{subject to} \quad \mathbf{u}^T\mathbf{u} = 1$$

writes

$$L(\mathbf{u}, \lambda) = C(\mathbf{u}) + \lambda(1 - \|\mathbf{u}\|^2)$$

so that a stationary point $\mathbf{u}^\star$ must satisfy $\nabla_{\mathbf{u}} C(\mathbf{u}^\star) = 2\lambda\mathbf{u}^\star$.

FastICA algorithms (ctd)

Hence, a fast fixed point algorithm can be obtained as

```
Initialize $\mathbf{u}^{(0)}$
for $i = 1 : n_{iter}$ do
    $\mathbf{u}^{(i)} \leftarrow \frac{\nabla_{\mathbf{u}} C(\mathbf{u}^{(i-1)})}{\|\nabla_{\mathbf{u}} C(\mathbf{u}^{(i-1)})\|}$
end for
```

Though based on a heuristic, the convergence of this algorithm to a stationary point of the original constrained problem can be shown.

FastICA algorithms (ctd)

In practice, the expectation appearing in the gradient is replaced by sample averages, i.e,

$$\mathrm{E}\{(\mathbf{u}^T\mathbf{z})^3\mathbf{z}\} \approx \frac{1}{N}\sum_n (\mathbf{u}^T\mathbf{z}_n)^3\mathbf{z}_n$$

The estimation may however be quite sensitive to outliers so that other algorithms based on robust approximations of the negentropy should be used.

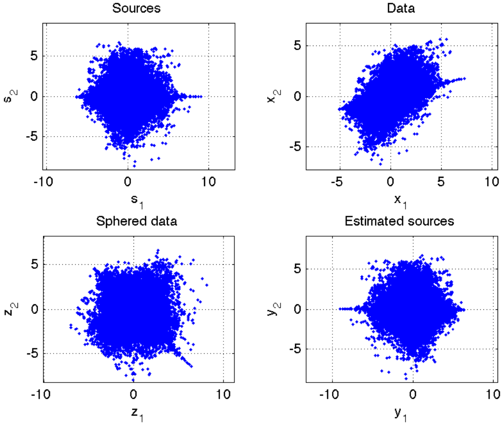
However, the optimization concepts hold, and lead to the family of FastICA algorithms.

Fast ICA algorithms (ctd)

The "one-unit" optimization can be generalized to optimization of the whole matrix $\mathbf{U}$ through orthogonalization.

```
Initialize $\mathbf{u}_1^{(0)}, \dots, \mathbf{u}_K^{(0)}$ (randomly)
for $i = 1 : n_{iter}$ do
    Do one iteration of a one-unit algorithm on every $\mathbf{u}_k$ in parallel
    Orthogonalize the set of vectors $\mathbf{u}_1^{(i)}, \dots, \mathbf{u}_K^{(i)}$
        $\mathbf{U} \leftarrow (\mathbf{U}\mathbf{U}^T)^{-\frac{1}{2}}\mathbf{U}$
end for
```

2 x 2 audio example



x_1 x_2 z_1 z_2 $\hat{s}_1$ $\hat{s}_2$

References

- A. Hyvärinen, J. Karhunen and E. Oja. *Independent Component Analysis*. Wiley, 2001.
- A. Hyvärinen and E. Oja. *Independent Component Analysis : Algorithms and Applications*. Neural Networks, 2000. [online]
- The FastICA package for MATLAB. `http://www.cis.hut.fi/projects/ica/fastica/`
- J.-F. Cardoso. *Blind Signal Separation: Statistical Principles*. Proceeding of the IEEE, 1998. [online]