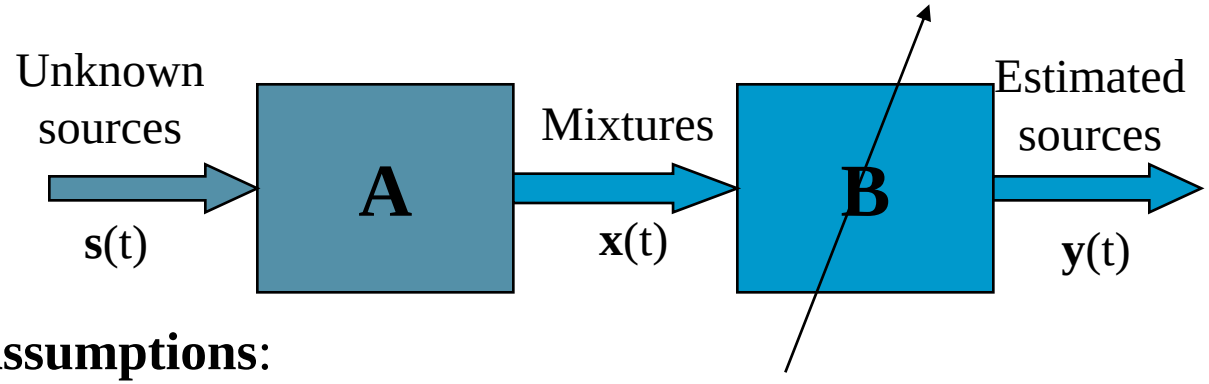
3. ICA for Linear instantaneous mixtures

Second order ?

Contrast functions

Mutual information

ICA for linear instant. mixtures



Assumptions:

- **A** is an unknown mixing matrix, assumed regular or full rank
- in the following, we assume **A** is a square matrix,
- **B** is the separating matrix
- sources are mutually independent

Principle:

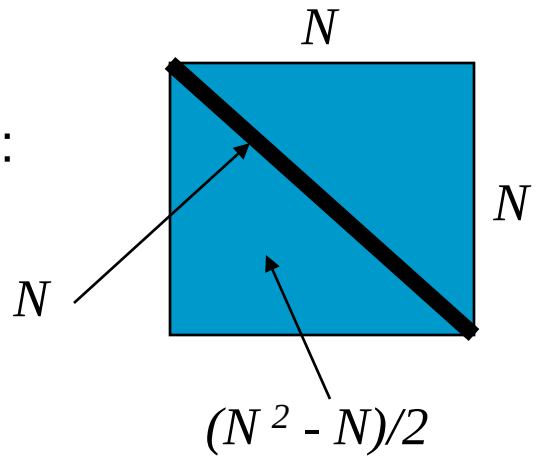
- Unsupervised : since **s**(t) is unknown, one cannot compare **y**(t) to **s**(t) !
- **B** is estimated so that **y**(t) becomes independent

3.1. Linear mixtures: second order

- If we only use second order statistics…?
- For 2 mixtures of 2 sources,
  - **B** has 4 parameters (unknowns),
  - there are only 3 equations: $E(Y_1^2), E(Y_2^2), E(Y_1 Y_2)$

  i.e. less equations than unknowns. Impossible !
- For N mixtures of N sources,
  - **B** has *N* x *N* parameters (unknowns),
  - there are only *N* + *N*(*N* - 1)/2 equations:

$$E(Y_1^2), E(Y_2^2), \cdots, E(Y_N^2),$$
$$E(Y_1 Y_2), E(Y_1 Y_3) \cdots E(Y_{N-1} Y_N)$$



Linear mixtures: second order

- At the second order 2… one could adjust **B** so that outputs become decorrelated:

$$b_{ij} = b_{ij} - \mu E[y_i y_j]$$

- The algorithm converges if: $E[y_i y_j] = 0, \forall i \neq j$
- One observes that: $b_{ij} = b_{ji}$
- The separation matrix **B** is then symmetric, and

cannot inverse any mixing matrix.

Linear mixtures: second order

- For 2 mixtures of 2 sources, solution is not unique. They live in a 1-D manifold defined by:

$$b_{21} = -\frac{b_{12}\left(a_{21}^2 + \left(\frac{\sigma_2}{\sigma_1}\right)^2\right) + a_{21} + a_{12}\left(\frac{\sigma_2}{\sigma_1}\right)^2}{\left(a_{21} + a_{12}\left(\frac{\sigma_2}{\sigma_1}\right)^2\right) b_{12} + 1 + a_{21}^2\left(\frac{\sigma_2}{\sigma_1}\right)^2}$$

- This is a set of hyperboles, depending of the ratio of variances and of the mixing matrix.

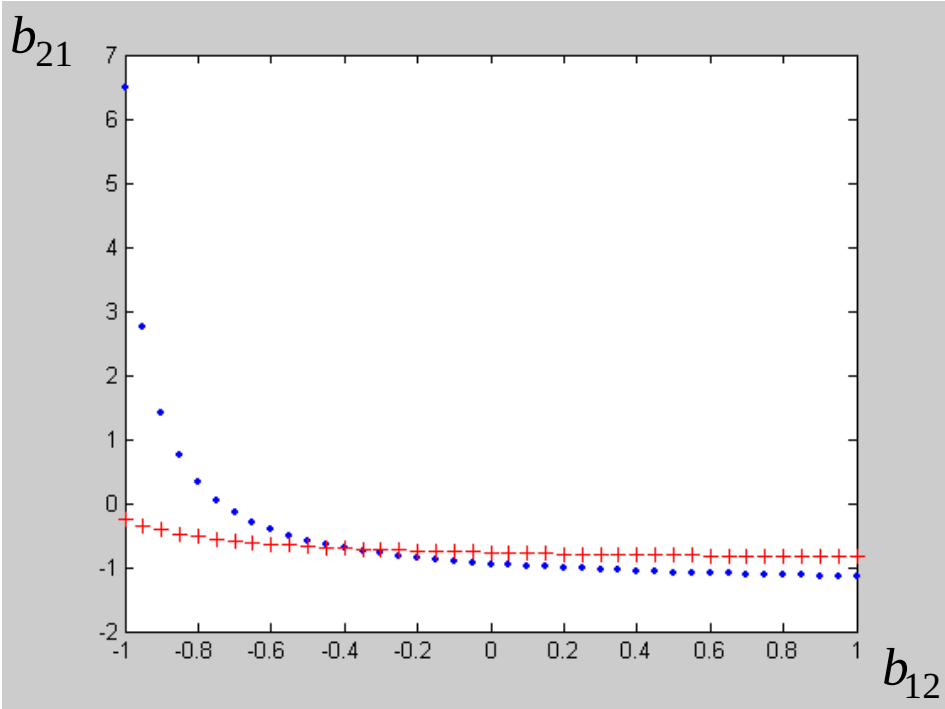
Linear mixtures: second order

- The family of hyperboles intersects in a point depending only on mixing matrix coefficients.

$$\mathbf{A} = \begin{pmatrix} 1 & 0.4 \\ 0.7 & 1 \end{pmatrix}$$

$$(\sigma_2 / \sigma_1)^2 = 1 \,(\text{in blue dots})$$

$$(\sigma_2 / \sigma_1)^2 = 2 \,(\text{in red} +)$$

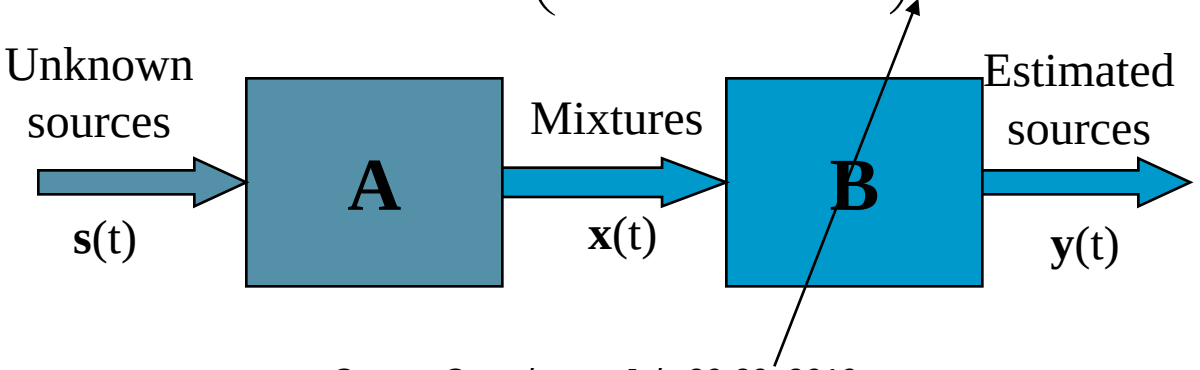


Exercice

- Assuming $\mathbf{A} = \begin{pmatrix} 1 & a_{12} \\ a_{21} & 1 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 1 & b_{12} \\ b_{21} & 1 \end{pmatrix}$

and source variance are denoted σ_1 and σ_2 , show that $E[y_1 y_2] = 0$ leads to

$$b_{21} = -\frac{b_{12}\left(a_{21}^2 + \left(\frac{\sigma_2}{\sigma_1}\right)^2\right) + a_{21} + a_{12}\left(\frac{\sigma_2}{\sigma_1}\right)^2}{\left(a_{21} + a_{12}\left(\frac{\sigma_2}{\sigma_1}\right)^2\right) b_{12} + 1 + a_{21}^2\left(\frac{\sigma_2}{\sigma_1}\right)^2}$$



A two step approach

- For linear mixtures, **B** can be factorized in 2 matrices
  - a whitening (or sphering) matrix **W**,
  - An orthogonal (rotation) matrix, **U**.

- In fact, **W** is computed with 2nd-order statistics so that : $E(\mathbf{Z}\mathbf{Z}^T) = \mathbf{I}$

$$E[\mathbf{WAS}(\mathbf{WAS})^T] = \mathbf{WA}\,E[\mathbf{S}\mathbf{S}^T](\mathbf{WA})^T = \mathbf{WA}\,(\mathbf{WA})^T = \mathbf{I}$$

- It means that **WA** is an orthogonal matrix, and thus **U** must be an orthogonal matrix.

A two step approach

- For linear mixtures, **B** can be factorized in 2 matrices
  - a whitening (or sphering) matrix **W**,
  - an orthogonal (rotation) matrix, **U**.

- From an algebraic point of view:
  - $\mathbf{W}E(\mathbf{X}\mathbf{X}^T)\mathbf{W}^T = \mathbf{I}$ involves $n(n+1)/2$ equations i.e. defines $n(n+1)/2$ parameters
  - the orthogonal matrix **U** is related to $n(n-1)/2$ elementary (plane) rotations (Givens rotations), i.e. requires $n(n-1)/2$ parameters
  - the total number of parameters is exactly $n \times n$

Demo: decorrelation not enough

- For linear mixtures, we show
  - joint distribution of sources (**s**), mixtures (**x**), mixtures after sphering (**z**) and **y** after a (non optimal) rotation
  - for uniform, Gaussian, sparse data

- Demo with dis_gau.m, dis_uni.m, dis_spa.m

2.2. Nonlinear decorrelation

- Higher (than 2) order moment leads to better independence approximation than decorrelation:

$$E[f(y_i)g(y_j)] = 0 \qquad f \neq g$$

$$E[y_i^3 y_j] = 0$$

- For instance, one proposes the following algorithm:

$$b_{ij} = b_{ij} - \mu E[f(y_i)g(y_j)]$$

- The algorithm converges if $E[f(y_i)g(y_j)] = 0$
- simplest case: $f(u) = u^3$, $g(v) = (v)$
  no longer symmetry: $E[y_i^3 y_j] \neq E[y_j^3 y_i]$
  as many equations as unknowns.
- Problem: how to choose optimally *f* and *g* ?

2.3. Contrast functions: definition

- Since **s**(t) is unknown, one cannot compare **y**(t) to **s**(t) and use a least square method
- One proposes to compute a contrast function which will be maximum when **y**(t) = **s**(t), but does not depend on **s**(t)

- (Comon, 1991 and SP 1994 ; Donoho, 1981)

  A contrast function is a function $\Psi(p)$ from the set of pdf to R with the following properties:

  1) $\Psi(p_{\mathbf{Px}}) = \Psi(p_{\mathbf{x}})$, for any permutation matrix **P,**

  2) $\Psi(p_{\mathbf{Dx}}) = \Psi(p_{\mathbf{x}})$, for any invertible diagonal matrix **D**,

  3) if **x** is a random vector with independent components and **A** is an invertible matrix, then $\Psi(p_{\mathbf{Ax}}) \leq \Psi(p_{\mathbf{x}})$,

  4) if $\Psi(p_{\mathbf{Ax}}) = \Psi(p_{\mathbf{x}})$, then **A** = **DP**, where **D** is diagonal matrix and **P** is a permutation matrix.

Contrast functions: examples

- The opposite of the absolute value of the cross-correlation is not a contrast function

$$\Psi(p_{\mathbf{y}}) = -\sum_{i,j} \left| Cum(y_i, y_j) \right|$$

  – For instance, for any rotation matrix **U,** $\Psi(p_{\mathbf{Ux}}) = \Psi(p_{\mathbf{x}})$ while $\mathbf{U} \neq \mathbf{DP}$**,** i.e. the 3rd property is not satisfied.

- The opposite of mutual information is a contrast function

$$\Psi(p_{\mathbf{y}}) = -I(\mathbf{Y})$$

Contrast functions: examples

- Define the 4th-order cross-cumulant of a zero-mean variable y

$$Cum_{ijkl}(\mathbf{y}) = Cum(y_i, y_j, y_k, y_l) =$$

$$= Ey_i y_j y_k y_l - Ey_i y_j \, Ey_k, y_l - Ey_i y_k \, Ey_j y_l - Ey_i y_l \, Ey_j y_k$$

- The sum of the squared 4th-order cumulants (after sphering):

$$\Psi^o_{ICA}(p_{\mathbf{y}}) = \sum_{ijkl \neq iiii} Cum^2_{ijkl}(\mathbf{y})$$

$$= -\sum_i Cum^2_{iiii}(\mathbf{y}) + cte = -\sum_i \kappa_4^2(y_i) + cte$$

- The sum of the absolute 4th-order cumulants:

$$\Psi^o(p_{\mathbf{y}}) = -\sum_i \left| \kappa_4(y_i) \right|$$

Contrast: Maximum likelihood

- According to the model **x** = **As**, we can write the likelihood:

$$p_{\mathbf{x}}(x|A,p_{\mathbf{S}}) = p_{\mathbf{S}}(A^{-1}x)\left|\det(A^{-1})\right|$$

- For T iid samples, $X_{1:T} = (x_1, x_2, \cdots, x_T)$, the log-likelihood is:

$$L(A,q) = \frac{1}{T}\ln p_{\mathbf{X}}(X_{1:T}|A,q) = \frac{1}{T}\ln \prod_{t=1:T} p_{\mathbf{S}}(A^{-1}x_t) - \ln\left|\det(A)\right|$$

$$= \frac{1}{T}\sum_{t=1}^{T} \ln p_{\mathbf{S}}(A^{-1}x_t) - \ln\left|\det(A)\right|$$

Contrast: Maximum likelihood

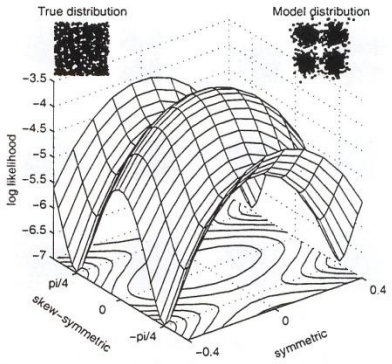


Fig. 5. Log-likelihood with a slightly misspecified model for source distribution: maximum is reached close to the true value.

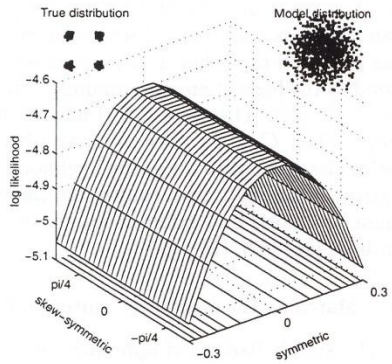


Fig. 6. Log-likelihood with a Gaussian model for source distribution: no 'contrast' in the skew-symmetric direction.

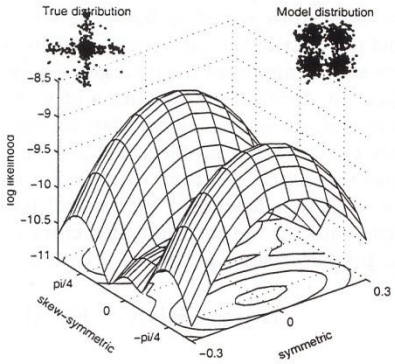


Fig. 7. Log-likelihood with a widely misspecified model for source distribution: maximum is reached for a mixing system.

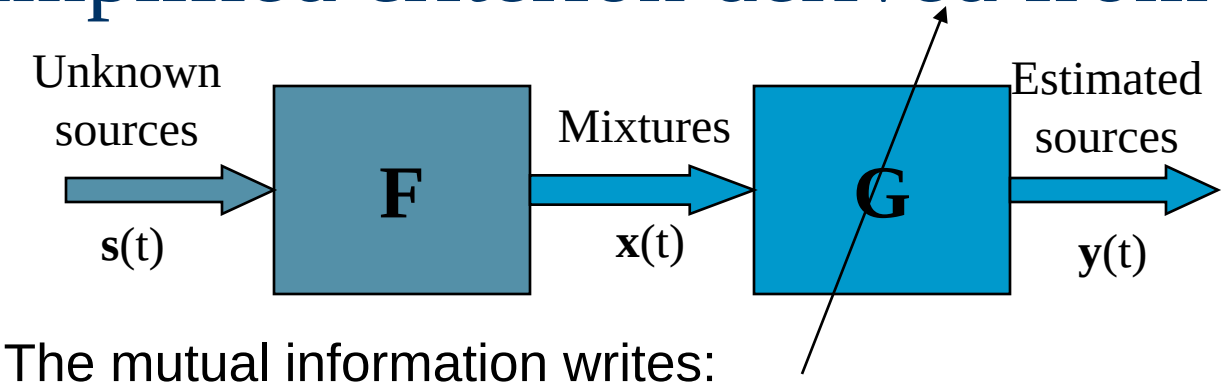
Fairly good match Gaussian model Bad match

Reprinted from J.-F. Cardoso, *Proceedings of the IEEE*, Vol. 9(10) 2009-25

Contrast functions: exercises

- Show that the opposite of mutual information is a contrast function.
  - The idea is to show -I(**Y**) is negative and equal to zero if and only if components of **Y** are independent (one will use $\log(u) \leq u - 1$)
- Compute the pdf $p_{\mathbf{Y}}(\mathbf{y})$ of the random vector **y**=**Cs**, where **C** is an invertible matrix, as a function of the (known) pdf $p_{\mathbf{S}}$ of the random vector **s**, assumed with independent components
- Show that mutual information is preserved by any invertible diagonal transform.

Simplified criterion derived from MI



- The mutual information writes:

$$I(\mathbf{Y}) = \sum_i H(Y_i) - H(\mathbf{Y}) = \sum_i H(Y_i) - H(\mathbf{X}) - E\ln\left|\det J_G\right|$$

- Since H(**X**) does not depend on the separating system, minimizing MI is equivalent to minimize:

$$J(\mathbf{Y}) = \sum_i H(Y_i) - E\ln\left|\det J_G\right|$$

- This criterion is simpler, since it doesn't require estimation of joint pdf.

MI and Maximum likelihood

- Asymptotically, ML tends toward:

$$L(\mathbf{A}, p_{\mathbf{S}}) = \frac{1}{T}\sum_{t=1}^{T} \ln p_{\mathbf{S}}(\mathbf{A}^{-1}\mathbf{x}_t) - \ln\left|\det \mathbf{A}\right| \xrightarrow[T\to+\infty]{} E \ln p_{\mathbf{S}}(\mathbf{A}^{-1}\mathbf{x}) - \ln\left|\det \mathbf{A}\right|$$

- Denoting **y** = **A**$^{-1}$ **x**,one can compute the KL divergence:

$$KL(p_{\mathbf{Y}} \mid p_{\mathbf{S}}) = \int\cdots\int p_{\mathbf{Y}}(\mathbf{u}) \ln \frac{p_{\mathbf{Y}}(\mathbf{u})}{p_{\mathbf{S}}(\mathbf{u})} \mathbf{du}$$

$$= -E \ln p_{\mathbf{S}}(\mathbf{u}) - H(\mathbf{Y})$$

$$= -E \ln p_{\mathbf{S}}(\mathbf{u}) - H(\mathbf{X}) - \ln\left|\det \mathbf{A}^{-1}\right|$$

$$= -L(\mathbf{A}, p_{\mathbf{S}}) + cte$$

- ML finds a matrix A such that **y** = **A**$^{-1}$ **x** is as close as possible (at the KL sense) of the hypothesized distribution of the sources.

MI and Maximum likelihood

- One can also consider the relationship between MI and ML by denoting :
  - $\tilde{\mathbf{y}}$ is the distribution with independent entries (joint pdf factorises) with the same marginal distribution than $\mathbf{y}$
  - $\mathbf{s}$ is the hypothetized distribution of the independent sources
- Then, one can write (using information theory relationships)

$$KL(p_{\mathbf{Y}}|p_{\mathbf{S}}) = KL(p_{\mathbf{Y}}|p_{\tilde{\mathbf{Y}}}) + KL(p_{\tilde{\mathbf{Y}}}|p_{\mathbf{S}})$$

ML

Independence of $\mathbf{y}$ (MI) does not depend on $\mathbf{s}$

Mismatch between estimated $\mathbf{y}$ and the model $\mathbf{s}$

MI and Gaussianity

- Intuitively: due to Large Number Law, sum of random variables is more Gaussian than each random variable
- For linear mixtures: 2-stage algorithm

- The separating matrix is split in two matrices: a whitening (or sphering) matrix **W** and a rotation matrix **U,** i.e.:

$$\mathbf{W} \text{ such that } E\mathbf{z}\mathbf{z}^T = \mathbf{I}$$

- The mutual information writes, with **B** = **UW**:

$$I(\mathbf{Y}) = \sum_i H(Y_i) - H(\mathbf{Y}) = \sum_i H(Y_i) - H(\mathbf{Z}) - \ln|\det \mathbf{U}|$$

$$= \sum_i H(Y_i) + cte$$

Doesn't depend on U

1 for orthogonal matrix

MI and Gaussianity

- Minimizing MI is equivalent
  - to minimize the sum of marginal entropies, with random variables with normalized variances,
  - to minimize gaussianity of each Y_i since entropy is maximum for Gaussian distribution
- Since Yi are unit variance random variables, on can use the neguentropy J(Yi). Negentropy is positive and equal to 0 iff Yi is Gaussian:

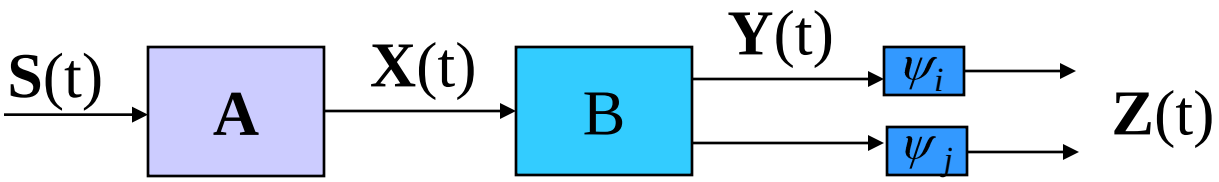
$$J(Y_i) = H(Y_{gi}) - H(Y_i)$$

$$\sum_i H(Y_i) + cte = \sum_i (H(Y_i) - H(Y_{gi})) + cte$$

$$= -\sum_i J(Y_i) + cte$$

- Minimize Gaussianity is equivalent to maximize neguentropy

MI and Infomax



- B is followed by component-wise NL mappings ψ_i , which are the cumulative density functions of Y_i
- Z_i is then uniformly distributed in [0, 1], hence: $H(Z_i) = cte$
- The mutual information writes:

$$I(\mathbf{Y}) = I(\mathbf{Z}) = \sum H(Z_i) - H(\mathbf{Z}) = cte - H(\mathbf{Z})$$

- Minimizing MI of Y, I(Y) = minimizing MI of Z, I(Z)
  = maximizing joint entropy of Z, H(Z)
- Infomax algorithm: Bell and Sejnowsky (Neural Comp., 1995)

4. Algorithms

Nonlinear decorrelation

MI minimization

FastICA

Joint diagonalization

4.1. Nonlinear decorrelation

- For instance, one proposes the following algorithm:

$$b_{ij} = b_{ij} - \mu E[f(y_i)g(y_j)] \qquad f \neq g$$

simplest case: $f(u) = u^3,\ g(v) = (v)$

other functions: $f(u) = u^3,\ g(v) = \tanh(v)$

4.2. MIM-based algorithm

- For linear mixtures, one has to estimate **B**
- We compute the criterion gradient with respect to the parameter : $\frac{\partial I}{\partial \mathbf{B}}$
- The separation matrix is updated according to:

$$\mathbf{B}(t+1) = \mathbf{B}(t) - \mu \frac{\partial I}{\partial \mathbf{B}}$$

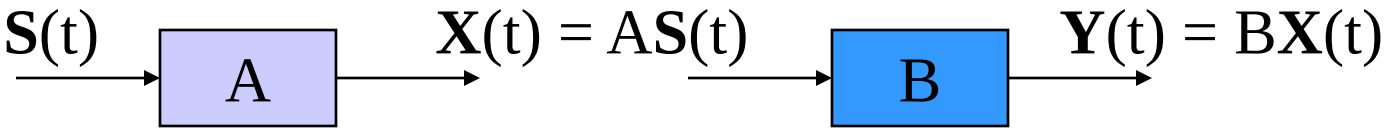
- The algorithm converges when $\frac{\partial I}{\partial \mathbf{B}}$ is equal to 0 (on the average), *i.e.* if I is minimum.

Mutual information algorithm (1/2)

- The mutual information writes:
$$I(\mathbf{Y}) = \sum_i H(Y_i) - H(\mathbf{Y}) = \sum_i H(Y_i) - H(\mathbf{X}) + \ln|\det \mathbf{B}|$$
- Estimation equation is then: $\frac{\partial I(Y)}{\partial \mathbf{B}} = 0$ where $\mathbf{B}$ represents the parameters
- It leads to the following equation:
$$\frac{\partial I(\mathbf{Y})}{\partial \mathbf{B}} = \sum_i \frac{\partial H(Y_i)}{\partial \mathbf{B}} + \frac{\partial \ln|\det \mathbf{B}|}{\partial \mathbf{B}}$$
$$= -\sum_i E\left\{\frac{d \ln p_{Y_i}(y_i)}{dy_i} \frac{\partial y_i}{\partial \mathbf{B}}\right\} + \frac{\partial \ln|\det \mathbf{B}|}{\partial \mathbf{B}} = 0$$

which needs estimation of pdf or score function (der. of log pdf).

Mutual information algorithm (2/2)



Since $\frac{d \ln|\det \mathbf{B}|}{d\mathbf{B}} = (\mathbf{B}^{-1})^T$, the derivative of MI is:

$$\frac{\partial I(\mathbf{Y})}{\partial \mathbf{B}} = -\sum_i E \frac{d \ln p_{Y_i}(y_i)}{dy_i} \frac{\partial y_i}{\partial \mathbf{B}} + \frac{\partial \ln|\det \mathbf{B}|}{\partial \mathbf{B}}$$

$$= E[\boldsymbol{\psi}_{\mathbf{Y}}(\mathbf{Y})\mathbf{X}^T] + (\mathbf{B}^{-1})^T$$

Multiplying by $\mathbf{B}^T$ and using $\mathbf{y} = \mathbf{Bx}$,

$$\frac{\partial I(\mathbf{Y})}{\partial \mathbf{B}} = 0 \Leftrightarrow E[\boldsymbol{\psi}_{\mathbf{Y}}(\mathbf{Y})\mathbf{X}^T] + (\mathbf{B}^{-1})^T = 0$$

$$\text{i.e. } E[\boldsymbol{\psi}_{\mathbf{Y}}(\mathbf{Y})\mathbf{Y}^T] + \mathbf{I} = 0$$

Mutual information algorithm (2/2)

This result leads to the following estimation equations (NL decorrelation):

$$E\left[-\frac{d\ln p_{Y_i}(Y_i)}{dY_i}Y_j\right]=0,\quad i\neq j$$

– For zero-mean Gaussian sources, $-\frac{d\ln p_{Y_i}(y_i)}{dy_i}=\frac{y_i}{\sigma_i^2}$,

i.e. the estimating equation only requires 2nd-order statistics,

– For non Gaussian sources : higher (than 2) order statistics,
– Priors or good estimates of source distribution leads to optimal statistics ; approximation of pdf leads to different algorithm implementations (2nd order, cumulants, etc.).

Independence criterion: pdf estimation

- Estimation equations require pdf's or score functions estimates
- pdf's can be estimated using various methods
  - expansion near Gaussianity: Gram-Charlier (Lacoume 91, Comon SP 94, Yang et al. SP 98), or Edgeworth expansions
  - kernel estimators (Pham IEEE Trans. SP 96, Taleb, Jutten IEEE SP 99) … then, score functions are estimated by derivation
- Score function can be estimated directly by minimizing MSE cost (Pham et al. EUSIPCO 92 ; Taleb, Jutten ICANN 97, IEEE SP 99)

$$J(\mathbf{w}) = \frac{1}{2} E\left[\hat{\psi}_Y(\mathbf{w}, y) - \psi_Y(y)\right]^2$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = E\left[\frac{\partial \hat{\psi}_Y(\mathbf{w}, y)}{\partial \mathbf{w}} \left(\hat{\psi}_Y(\mathbf{w}, y) - \psi_Y(y)\right)\right]$$

$$= E\left[\hat{\psi}_Y(\mathbf{w}, y) \frac{\partial \hat{\psi}_Y(\mathbf{w}, y)}{\partial \mathbf{w}} + \frac{\partial^2 \hat{\psi}_Y(\mathbf{w}, y)}{\partial y \partial \mathbf{w}}\right]$$

Independence criterion: pdf estimation

- Pdf estimation based on kernel estimate
  - kernel estimate is a well know statistical method (Silverman, etc.)
  - assuming N samples x_k of the random variable X, the pdf estimation is:

$$\hat{p}_X(x) = \frac{1}{N}\sum_{n=1}^{N} K_h(x - x_n)$$

  - The kernel must satisfy to a few conditions, and can have different shapes, e.g. Gaussian, triangle, square (Parzen window), etc.
  - Example of Gaussian kernels:

$$K_h(x) = \frac{1}{\sqrt{2\pi}h}\exp\left(-\frac{x^2}{2h^2}\right)$$

- Demonstration with different values of h (ker_est.m)

Independence criterion: score estimation

- Score function can be estimated directly by minimizing MSE cost (Pham et al. EUSIPCO 92 ; Taleb, Jutten ICANN 97, IEEE SP 99)

$$J(\mathbf{w}) = \frac{1}{2} E\left[\hat{\psi}_Y(\mathbf{w}, y) - \psi_Y(y)\right]^2$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = E\left[\frac{\partial \hat{\psi}_Y(\mathbf{w}, y)}{\partial \mathbf{w}} \left(\hat{\psi}_Y(\mathbf{w}, y) - \psi_Y(y)\right)\right]$$

$$= E\left[\hat{\psi}_Y(\mathbf{w}, y) \frac{\partial \hat{\psi}_Y(\mathbf{w}, y)}{\partial \mathbf{w}} + \frac{\partial^2 \hat{\psi}_Y(\mathbf{w}, y)}{\partial y \partial \mathbf{w}}\right]$$

$$\text{since } \psi_Y(y) = -\frac{d \ln p_Y(y)}{dy} = -\frac{p'_Y(y)}{p_Y(y)}$$

4.3. FastICA algorithm

- Main idea for estimating one source
  - consider first a whitening step, **W**, such that **z** = **W x** is spatially white and with unit variance ; **U** is a rotation matrix
  - estimating a vector **u** such that the estimated source $\mathbf{u}^T\mathbf{z}$

  has a maximum absolute value kurtosis, i.e.

$$\frac{\partial\left|kurt(\mathbf{u}^T\mathbf{z})\right|}{\partial\mathbf{u}} = 4sign(kurt(\mathbf{u}^T\mathbf{z}))\left[E\mathbf{z}(\mathbf{u}^T\mathbf{z})^3 - 3\mathbf{u}\|\mathbf{u}\|^2\right]$$

- Remark that, due to sphering of **z**, the gradient is maximum if **u** is colinear to the derivative of $F(\mathbf{u})$.

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

$$\frac{\partial L(\mathbf{u})}{\partial\mathbf{u}} = \frac{\partial F(\mathbf{u})}{\partial\mathbf{u}} + 2\lambda\mathbf{u} = 0 \quad \Rightarrow \mathbf{u} \propto \frac{\partial F(\mathbf{u})}{\partial\mathbf{u}}$$

FastICA algorithm

- We have then:

$$\mathbf{u} \propto E\mathbf{z}(\mathbf{u}^T\mathbf{z})^3 - 3\mathbf{u}$$

- The algorithm is then the fixed-point sequence:

$$\mathbf{u}(t+1) = E\mathbf{z}(\mathbf{u}(t)^T\mathbf{z})^3 - 3\mathbf{u}(t)$$
$$\mathbf{u}(t+1) = \mathbf{u}(t+1)/\left\|\mathbf{u}(t+1)\right\|$$

FastICA algorithm

- More generally, with a contrast function:

$$F(\mathbf{u}) = EG(\mathbf{u}^T\mathbf{z})$$

- Adding the constraint and writing the Lagrangian:

$$L(\mathbf{u}) = EG(\mathbf{u}^T\mathbf{z}) + \lambda(\|\mathbf{u}\|^2 - 1)/2$$

$$\frac{\partial L(\mathbf{u})}{\partial \mathbf{u}} = E\mathbf{z}g(\mathbf{u}^T\mathbf{z}) + \lambda\mathbf{u}$$

where g denotes the derivative of G

- Solving the Lagrangian by Newton method writes:

$$\mathbf{u}(t+1) = \mathbf{u}(t) - \left[\frac{\partial^2 L(\mathbf{u})}{\partial \mathbf{u}^2}\right]^{-1} \frac{\partial L(\mathbf{u})}{\partial \mathbf{u}}$$

- Computing the second derivative of Lagrangian leads to the matrix (Hessian):

$$\frac{\partial^2 L(\mathbf{u})}{\partial \mathbf{u}^2} = E\{\mathbf{z}\mathbf{z}^T g'(\mathbf{u}^T\mathbf{z})\} + \lambda\mathbf{I}$$

FastICA algorithm

$$\frac{\partial^2 L(\mathbf{u})}{\partial \mathbf{u}^2} = E\{\mathbf{z}^T g'(\mathbf{u}^T \mathbf{z})\} + \lambda \mathbf{I}$$

- The problem is the Hessian matrix should be inverted, which is cost computing.
- For avoiding the inversion, Hyvärinen and Oja proposed the approximation:

$$\frac{\partial^2 L(\mathbf{u})}{\partial \mathbf{u}^2} \approx E\mathbf{z}\mathbf{z}^T E g'(\mathbf{u}^T \mathbf{z}) + \lambda \mathbf{I} = E g'(\mathbf{u}^T \mathbf{z})\mathbf{I} + \lambda \mathbf{I} \text{ since } E\mathbf{z}\mathbf{z}^T = \mathbf{I}$$

- Using this approximation, the approximative Newton iteration becomes:

$$\mathbf{u}(t+1) = E\mathbf{z}g(\mathbf{u}^T \mathbf{z}) - E g'(\mathbf{u}^T \mathbf{z})\mathbf{u}$$

$$\mathbf{u}(t+1) = \mathbf{u}(t+1) / \|\mathbf{u}(t+1)\|$$

FastICA algorithm

- Coming back on the derivative of the contrast F(u)

$$\frac{\partial F(\mathbf{u})}{\partial \mathbf{u}} = E\mathbf{z}g(\mathbf{u}^T\mathbf{z}) = 0$$

- Denoting $\mathbf{u}^T\mathbf{z} = y$, the minima of the constrast function involve higher-order statistics:

$$\frac{\partial F(\mathbf{u})}{\partial \mathbf{u}} = E\mathbf{z}g(y) = 0$$

$$\text{or } E\mathbf{y}g(y) = 0$$

- This condition is similar to the one obtained with $\frac{\partial I(\mathbf{y})}{\partial \mathbf{y}} = 0$

- Consequently, the optimal function g is related to the unknown score function of y

FastICA algorithm: deflation scheme

- Idea is to extract one independent component, and then to remove it from the mixture, etc.
- The deflation algorithm
  1. Choose the number *m* of IC to extract, *p* = 1
  2. Initialize $\mathbf{u}_p$ (randomly)
  3. Do an one-unit iteration for computing $\mathbf{u}_p$
  4. Do the following (Gram-Schmidt like) orthogonalization

$$\mathbf{u}_p = \mathbf{u}_p - \sum_{j=1}^{p-1} (\mathbf{u}_p^T \mathbf{u}_j)\mathbf{u}_j$$

  5. Normalize $\mathbf{u}_p$ by dividing by its norm
  6. If $\mathbf{u}_p$ not converged, goto 3.
  7. Set $p=p+1$ and if $p < m+1$, goto 2, else stop.

FastICA algorithm: symmetric scheme

- Idea: to extract all the independent components simulataneously
- The symmetric algorithm
  1. Choose the number *m* of IC to extract, p=1
  2. Initialize $\mathbf{u}_p$ (randomly), $p = 1, \ldots m$
  3. Do an iteration for computing all the $\mathbf{u}_p$ in parallel
  4. Do the parallel orthogonalization (see Hyvärinen, Karhunen, Oja)

$$\mathbf{U} = (\mathbf{U}\mathbf{U}^T)^{-1/2}\mathbf{U}$$

  5. If not converged, goto 3, else stop

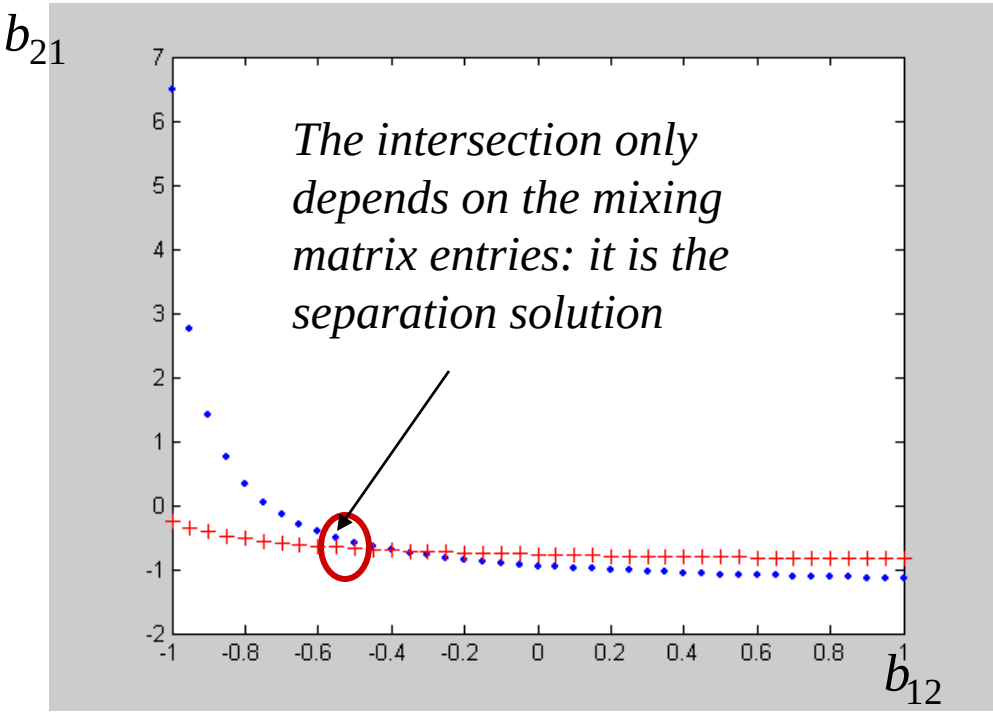
- It is easy to check that the orthonormalization step leads to an orthogonal matrix i.e. after orthonormalization $\mathbf{U}\mathbf{U}^T = \mathbf{I}$

4.4. Joint diagonalization algorithms

- Coming back to decorrelation

  – for different variance ratios, there is only one intersection between the curves $Ey_1y_2 = 0$,

  – the idea is to jointly diagonalize the two (or more) covariance matrices.



Joint diagonalization algorithms

- Non white sources:

$$Es_i(t)s_i(t-\tau) = \gamma_i(\tau) \neq \delta(\tau)$$

- Non white sources with different spectra:

$$\exists \tau \,/\, \gamma_i(\tau) \neq \gamma_j(\tau)$$

- Covariance matrix of delayed sources

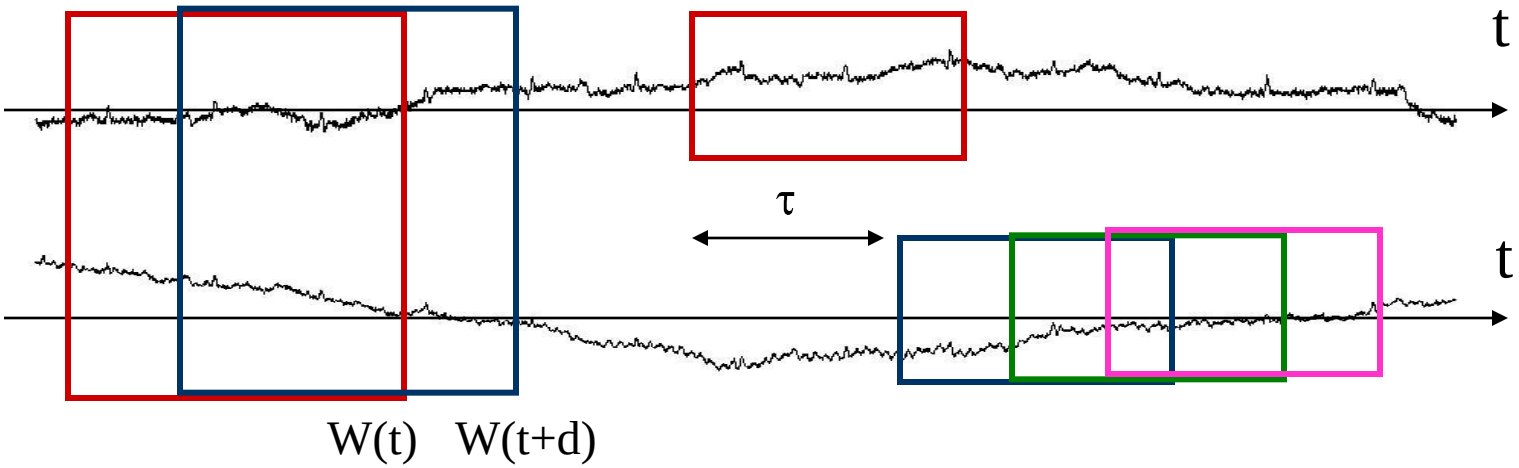
$$\mathbf{C_Y}(\tau) = E\mathbf{y}(t)\mathbf{y}(t-\tau) = \begin{pmatrix} Ey_1(t)y_1(t-\tau) & Ey_1(t)y_2(t-\tau) \\ Ey_2(t)y_1(t-\tau) & Ey_2(t)y_2(t-\tau) \end{pmatrix}$$

or its symetrized version

$$\frac{1}{2}\left[\mathbf{C_Y}(\tau) + \mathbf{C_Y}(-\tau)\right] = \begin{pmatrix} Ey_1(t)y_1(t-\tau) & (\gamma_{12}(\tau)+\gamma_{21}(\tau))/2 \\ (\gamma_{12}(\tau)+\gamma_{21}(\tau))/2 & Ey_2(t)y_2(t-\tau) \end{pmatrix}$$

Joint diagonalization algorithms

- Joint diagonalization of at least 2 (symetrized) covariance matrices of delayed sources, with different delays (AMUSE, Tong et al. 1990 ; SOBI, Belouchrani, Cardoso, 1995, etc.)
- Joint diagonalization of at least 2 (symetrized) covariance matrices of non stationary sources, computed on different windows (Matsuoka et al., 1995 ; Pham, Cardoso, 2001)



Joint diagonalization algorithms

- Algorithm principles
  - a first sphering step **W**, corresponding to diagonalizing the covariance matrix

$$\mathbf{C_Y}(0) = E\mathbf{y}(t)\mathbf{y}(t)^T = \begin{pmatrix} Ey_1(t)y_1(t) & Ey_1(t)y_2(t) \\ Ey_2(t)y_1(t) & Ey_2(t)y_2(t) \end{pmatrix} = \begin{pmatrix} Ey_1^2 & \gamma_{12}(0) \\ \gamma_{12}(0) & Ey_2^2 \end{pmatrix}$$

  - then, one has to determine the rotation matrix **U**, at higher order. With suited parameterization, this can be done very fast and easily, for any dimension.

Joint diagonalization algorithms

- Parameterizing **U**, with Givens rotation leads to very simple algorithms, for any dimensions :

$$U = \prod_{i=1,k>i}^{n-1} G(i,k,\theta_{ik})$$

$$\text{with } G(i,k,\theta_{ik}) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \cos\theta_{ik} & \sin\theta_{ik} & \vdots \\ \vdots & -\sin\theta_{ik} & \cos\theta_{ik} & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

- The idea is to estimate elementary rotations successively, by means of Jacobi rotations, which tends to cancel off-diagonal elements, 2 by 2

Joint diagonalization algorithms

- Jacobi rotation: find Q_{ik} such that:

$$\mathbf{A} \rightarrow \mathbf{Q}_{ik}^{T}\mathbf{A}\mathbf{Q}_{ik} = \mathbf{A}'$$

$$\begin{pmatrix} * & * & \cdots & * \\ * & a_{ii} & a_{ik} & \vdots \\ \vdots & a_{ki} & a_{kk} & * \\ * & \cdots & * & * \end{pmatrix} \rightarrow \begin{pmatrix} * & * & \cdots & * \\ * & a'_{ii} & 0 & \vdots \\ \vdots & 0 & a'_{kk} & * \\ * & \cdots & * & * \end{pmatrix}$$

where $\mathbf{Q}_{ik}$ is a Givens rotation matrix:

$$\begin{matrix} \\ \text{Row } i \\ \text{Row } k \\ \\ \end{matrix} \longrightarrow \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \cos\theta & \sin\theta & \vdots \\ \vdots & -\sin\theta & \cos\theta & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

It is easy to see that the product $\boldsymbol{Q}_{ik}{}^{T}\boldsymbol{A}\,\boldsymbol{Q}_{ik}$ only modify the entries belonging only to rows i or k and columns i or k. The condition $a'_{ik} = 0$ leads to a simple 2nd degree polynomial equation, and to simple algorithms.

Joint diagonalization algorithms

- In joint diagonalization algorithms, one iteration consists in a sweep of all the elementary Jacobi rotations, i.e. $n(n-1)/2$ for a n-dimensional matrix

.

- The algorithm can be used :
  - for any dimension (just the number of elementary rotations in each sweep changes),
  - for matrices computed as covariance matrices of delayed non white signals (SOBI or AMUSE), of non stationary signals on different windows, or of matrices deduced from 4th-order cumulant tensor (JADE),
  - in the time domain as well as in the frequency domain, since linear relation is preserved by Fourier transform:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \rightarrow \mathbf{X}(v) = \mathbf{A}\mathbf{S}(v)$$

Equivariance principle

- The speed and performance of first ICA algorithms were depending on the mixing matrix A.

.

- Instead of considering additive adaptation scheme, Cardoso and Laheld (IEEE SP 1996) proposed multiplicative scheme:

$$\mathbf{B} \leftarrow \mathbf{B} + \delta\mathbf{B}$$

$$\mathbf{B} \leftarrow (\mathbf{I} + \varepsilon)\mathbf{B}$$

- The adaptation scheme is related to the optimization of a criterion
- In this purpose, we have to compute the gradient of the criterion for the multiplicative scheme. It has been called "*relative gradient*".
- Note that Amari and Cichocki obtained the same results but with a quite different approach, and called "*natural gradient*".

Equivariance principle

- Denote J(**y**) the criterion to minimize. In multiplicative scheme, we want :

$$J(\mathbf{B}+\boldsymbol{\varepsilon}\mathbf{B}) = J(\mathbf{B}) + tr\left[\left(\frac{\partial J}{\partial \mathbf{B}}\right)^T \boldsymbol{\varepsilon}\mathbf{B}\right] + O(\|\boldsymbol{\varepsilon}\mathbf{B}\|)$$

$$= J(\mathbf{B}) + tr\left[\left(\frac{\partial J}{\partial \mathbf{B}}\mathbf{B}^T\right)^T \boldsymbol{\varepsilon}\right] + O(\|\boldsymbol{\varepsilon}\mathbf{B}\|)$$

- The variation is minimal if:

$$\boldsymbol{\varepsilon} = -\frac{\partial J}{\partial \mathbf{B}}\mathbf{B}^T = -\nabla_{rel} J(\mathbf{y}) = -H(\mathbf{y})$$

Note H(**y**) is not the entropy !

- It leads to the following iteration:

$$\mathbf{B} \leftarrow (\mathbf{I} - \mu H(\mathbf{y}))\mathbf{B}$$

Why Equivariance ?

- Multiplying the following iteration by **A**, the mixing matrix, and denote **C** = **BA**:

$$\mathbf{B}_{t+1} = (\mathbf{I} - \mu H(\mathbf{y}))\mathbf{B}_t$$

$$\mathbf{B}_{t+1}\mathbf{A} = (\mathbf{I} - \mu H(\mathbf{y}))\mathbf{B}_t\mathbf{A}$$

$$\mathbf{C}_{t+1} = (\mathbf{I} - \mu H(\mathbf{C}_t\mathbf{s}))\mathbf{C}_t$$

- The last rule shows that the trajectory does only depend on the global system **C**, and not on **A**. The mixing matrix only defines (according to the initial value of **B**) the initial point of the iteration since:

$$\mathbf{C}_0 = \mathbf{B}_0\mathbf{A}$$

- As a result, the performance of the algorithm does not depend on the mixing matrix **A**. In practice, the equivariance property is then very interesting.

5. Convolutive mixtures

Time domain

Frequency domain

Convolutive mixtures

- **Most of the works only concern 2 sources and 2 sensors**
- **Mainly 3 approaches:**
  - **methods based on linear time invariant filters,**
  - **methods based on a frequency approach,**
  - **methods in frequency-time domain.**
- **Independence x(t) and y(t) now concerns independence between random processes, i.e. independence between any part of the infinite series of samples**

$$Y_{t_1:t_2} \perp (y(t_1),\cdots y(t_2)) \text{ and } X_{t_3:t_4} \perp (x(t_3),\cdots x(t_4))$$

5.1. Convolutive mixtures (1/7) - Subspace

- *Method based on linear time invariant filters*
- The convolutive mixture

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

is modelled in the z-domain by:

$$\mathbf{x}(n) = \mathbf{A}(z)\,\underline{\mathbf{s}}(n)$$

- Subspace approach consists in writing the mixing:

$$\mathbf{x}(n) = \mathbf{A}(z)\,\underline{\mathbf{s}}(n) = \sum_{i=0}^{L} \mathbf{a}(i)\mathbf{s}(n-i), \qquad \mathbf{x}(n+1) = \sum_{i=0}^{L} \mathbf{a}(i)\mathbf{s}(n-i+1), \cdots$$

For T samples

$$\begin{pmatrix} \mathbf{x}(n) \\ \mathbf{x}(n-1) \\ \vdots \\ \mathbf{x}(n-T) \end{pmatrix} = \begin{pmatrix} \mathbf{a}(0) & \mathbf{a}(1) & \cdots & \mathbf{a}(L) & 0 & 0 & 0 \\ 0 & \mathbf{a}(0) & \mathbf{a}(1) & \cdots & \mathbf{a}(L) & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{a}(0) & \mathbf{a}(1) & & \mathbf{a}(L) \end{pmatrix} \begin{pmatrix} \mathbf{s}(n) \\ \mathbf{s}(n-1) \\ \vdots \\ \mathbf{s}(n-T-L) \end{pmatrix}$$

Convolutive mixtures (2/7) - Subspace

- Under mild conditions on the mixing filter and on the observation number, source can be separated
- The main drawbacks are:
  - large matrice has to be handled
  - the sensor number must be larger than the source number
- For more details, see Gürelli, Nikias 1995; Moulines et al., IEEE SP 1995 ; Abed-Meraim et al., IEEE IT 1997; Gorokhov, Loubaton, IEEE CAS 1997 and SP 1999; Mansour et al. IEEE SP 2000, etc.

Convolutive mixtures (3/7) - LTI filters

- *Methods based on linear time invariant filters*
- The convolutive mixture

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

  is modelled in the z-domain by**:**

$$\mathbf{x}(n) = \mathbf{A}(z)\,\underline{\mathbf{s}}(n)$$

- The methods consist in estimating an inverse matrix **B**(z) (up to **PD**(z)).
- For stability reason, **A**(z) have to be phase minimal.

Convolutive mixtures (4/7) - LTI filters

- Due to the indeterminacies (D(z)), one can propose simplified mixtures:

$$\mathbf{A}(z) = \begin{pmatrix} 1 & A_{12}(z) \\ A_{21}(z) & 1 \end{pmatrix}$$

This model is realist, if one source is dominant on each sensor

- Then, one can also impose structural constraints on B(z):

$$\mathbf{B}(z) = \begin{pmatrix} 1 & B_{12}(z) \\ B_{21}(z) & 1 \end{pmatrix} \qquad \mathbf{B}(z) = \frac{1}{1 - C_{12}(z)C_{21}(z)} \begin{pmatrix} 1 & C_{12}(z) \\ C_{21}(z) & 1 \end{pmatrix}$$

- With the 1st constraint a post-processing must be applied, since, at separation (when outputs are independent):

$$\mathbf{B}(z)\mathbf{A}(\mathbf{z}) = \begin{pmatrix} 1 - A_{12}(z)A_{21}(z) & 0 \\ 0 & 1 - A_{12}(z)A_{21}(z) \end{pmatrix}$$

Convolutive mixtures (5/7) - LTI filters

Optimization of **B**(z) can be done according to various methods:

- cancelling high-order cross-moments (Nguyen Thi, Jutten, SP 1995)

$$b_{ij}(k) = b_{ij}(k) - \mu Ef(y_i(n))y_j(n-k)$$

- cancelling 4-th order cross-cumulants (Nguyen Thi, Jutten, SP 1995)

$$b_{ij}(k) = b_{ij}(k) - \mu Cum(y_i(n), y_i(n), y_i(n), y_j(n-k))$$

- Minimizing the sum of squared cumulants (Simon et al., ICASSP 98)
- cancelling cross-trispectra (Yellin, Weinstein, IEEE SP, 1994)
- partial approximate joint diagonalisation (PAJOD) of cross-cumulant matrices (Comon et al. ICA 2001)

Convolutive mixtures (6/7) - LTI filters

Another simple approach, based on *linear prediction*, is possible for MA model of mixtures:

$$\mathbf{x}(n) = \mathbf{A}_0\mathbf{s}(n) + \sum_{k=1}^{L} \mathbf{A}_k\mathbf{s}(n-k)$$

Assuming A_0 invertible and denoting $\overline{\mathbf{A}}_k = \mathbf{A}_k\mathbf{A}_0^{-1}$ and $\bar{\mathbf{s}}(n-k) = \mathbf{A}_0\mathbf{s}(n-k)$

$$\mathbf{x}(n) = \bar{\mathbf{s}}(n) + \sum_{k=1}^{L} \overline{\mathbf{A}}_k\bar{\mathbf{s}}(n-k)$$

Matrices $\overline{\mathbf{A}}_k$ can be estimated, at the second order by linear prediction (Comon, TS 1990 in French ; AbedMeraim et al. IEEE SP 1997)

Then, the following equation (corresponding to instantaneous mixtures) $\bar{\mathbf{s}}(n-k) = \mathbf{A}_0\mathbf{s}(n-k)$ is solved by BSS at higher order

Convolutive mixtures (7/7) - Frequency

- Taking short-term FT, the model $\mathbf{x}(t) = \mathbf{A}(t) * \mathbf{s}(t)$ becomes: $\mathbf{x}(\nu, t) = \mathbf{A}(\nu)\mathbf{s}(\nu, t)$
- Around a frequency ν_0, one has

$$\mathbf{A}(\nu) = \mathbf{A}(\nu_0) \approx cst$$

- Then, on each narrowband, the problem reduces to a simple BSS problem in instantaneous mixtures, with complex-valued entries
- For reconstructing the large band sources, the reconstruction process must take into account the possible scale and permutation indeterminacies in each narrowband
- Various solutions, based on continuity and correlations between sources from a channel to the neighbors have been proposed: Capdevielle et al., ICASSP 94 ; Wu, Principe, ICA'99 ; Mejuto et al., ICA 2000 ; Dapena, Servière, ICA 2001, Pham, Servière, ICA03, etc.)