

Introduction to Independent Component Analysis

Barnabás Póczos

University of Alberta

Nov 26, 2009



Contents

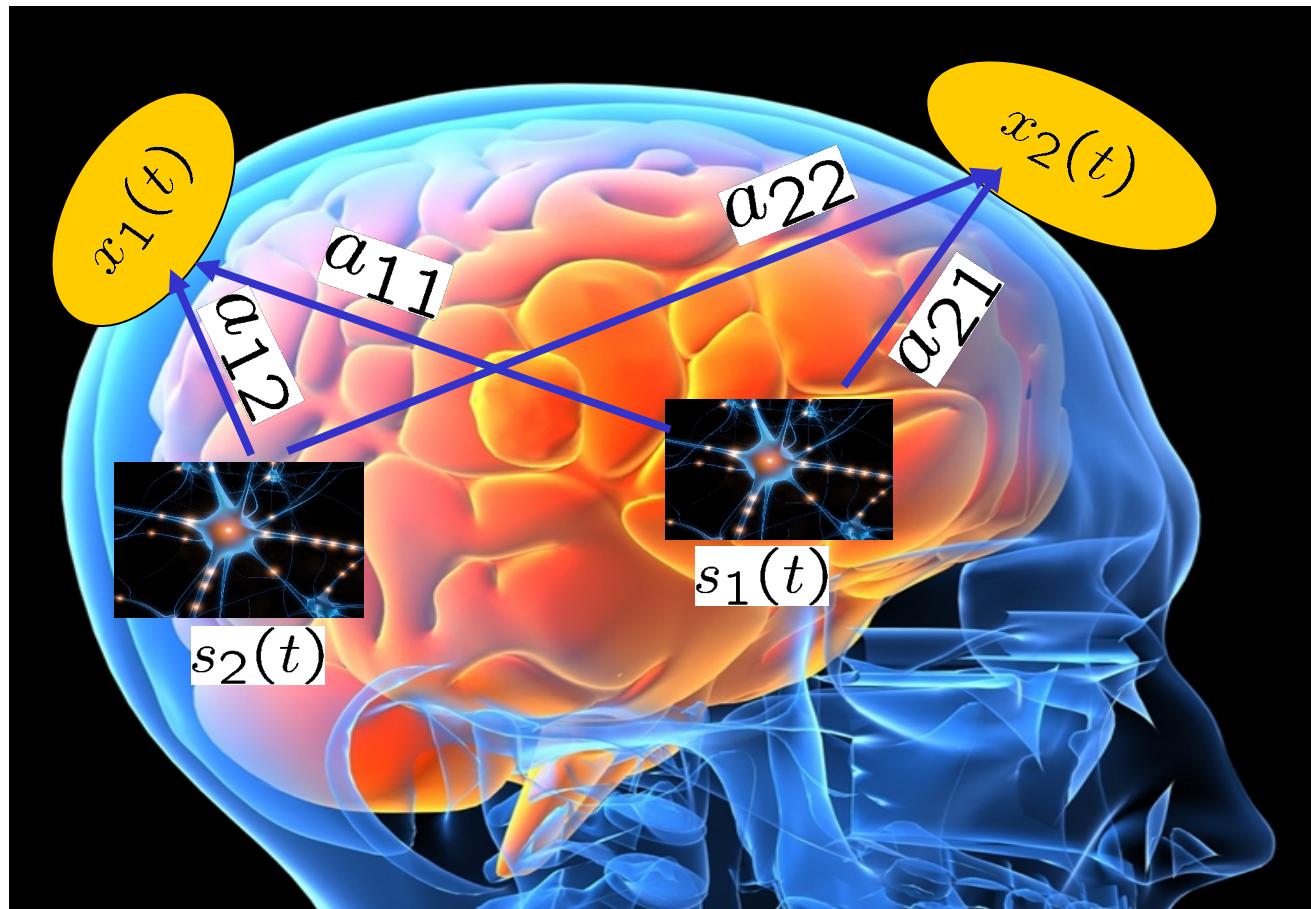
- Independent Component Analysis
 - ICA model
 - ICA applications
 - ICA generalizations
 - ICA theory
- Independent Subspace Analysis
 - ISA model
 - ISA theory
 - ISA results

Independent Component Analysis

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$$

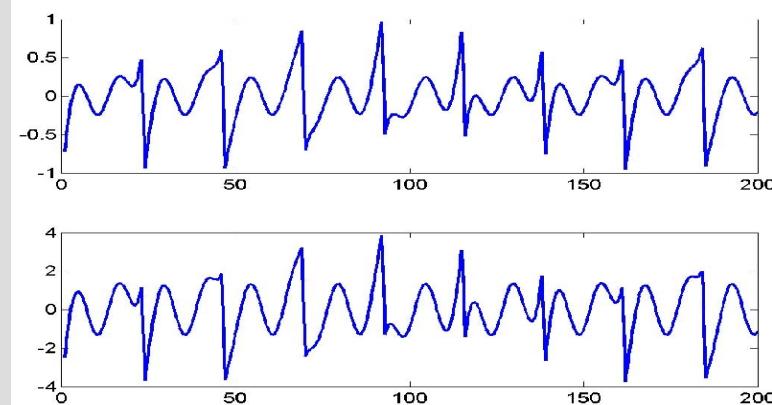
Goal: Estimate $\{s_i(t)\}$,
(and also $\{a_{ij}\}$)



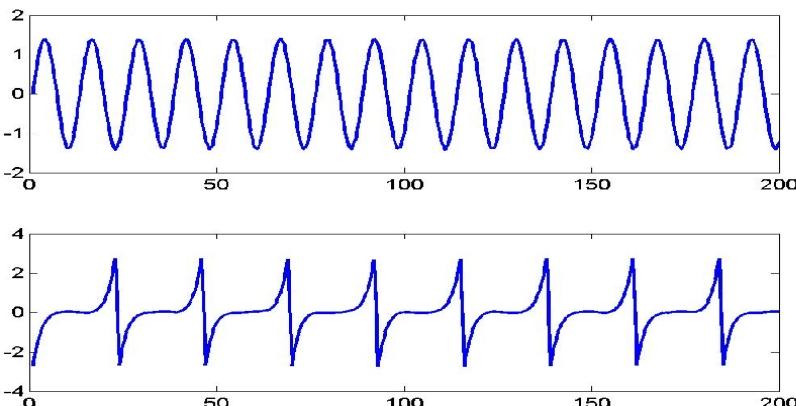
Independent Component Analysis

$$\begin{aligned}x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) \\x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t)\end{aligned}$$

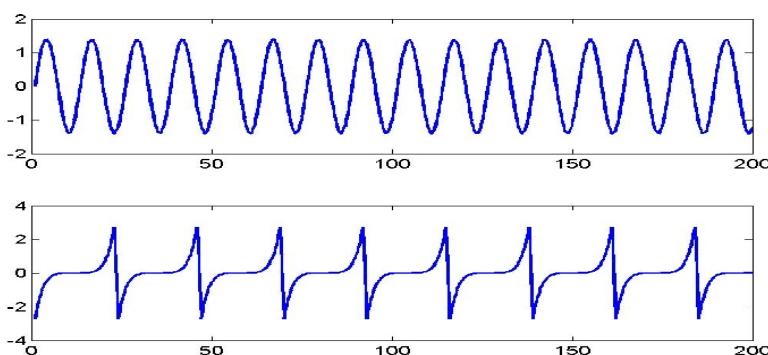
Model



Observations (Mixtures)



ICA estimated signals



original signals

Independent Component Analysis

Model

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$$

We observe

$$\begin{pmatrix} x_1(1) \\ x_2(1) \end{pmatrix}, \begin{pmatrix} x_1(2) \\ x_2(2) \end{pmatrix}, \dots, \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

We want

$$\begin{pmatrix} s_1(1) \\ s_2(1) \end{pmatrix}, \begin{pmatrix} s_1(2) \\ s_2(2) \end{pmatrix}, \dots, \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix}$$

But we don't know $\{a_{ij}\}$, nor $\{s_i(t)\}$

Goal: Estimate $\{s_i(t)\}$, (and also $\{a_{ij}\}$)

ICA vs PCA, Similarities

- Perform linear transformations
- Matrix factorization

PCA: *low rank* matrix factorization for *compression*

$$N \left\{ \begin{matrix} X \\ \vdots \\ M \end{matrix} \right. = \begin{matrix} U \\ \vdots \\ M \end{matrix} \left[\begin{matrix} S \\ \vdots \\ M \end{matrix} \right] \right\} M < N$$

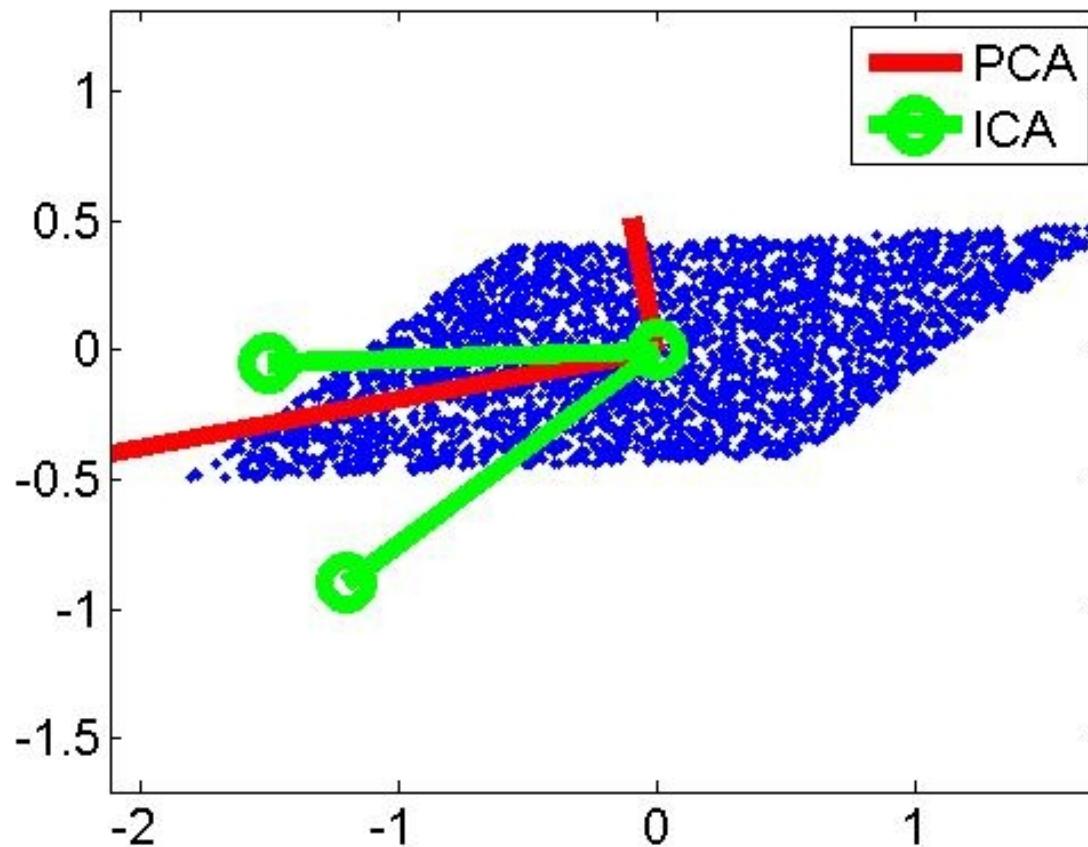
ICA: *full rank* matrix factorization to *remove dependency* between the rows

$$N \left\{ \begin{matrix} X \\ \vdots \\ N \end{matrix} \right. = \begin{matrix} A \\ \vdots \\ N \end{matrix} \left[\begin{matrix} S \\ \vdots \\ N \end{matrix} \right] \right\}$$

ICA vs PCA, Differences

- PCA: $\mathbf{X} = \mathbf{US}$, $\mathbf{U}^T \mathbf{U} = \mathbf{I}$
- ICA: $\mathbf{X} = \mathbf{AS}$
- PCA **does** compression
 - $M < N$
- ICA does **not** do compression
 - same # of features ($M = N$)
- PCA just removes correlations, **not** higher order dependence
- ICA removes correlations, **and** higher order dependence
- PCA: some components are **more important** than others
(based on eigenvalues)
- ICA: components are **equally important**

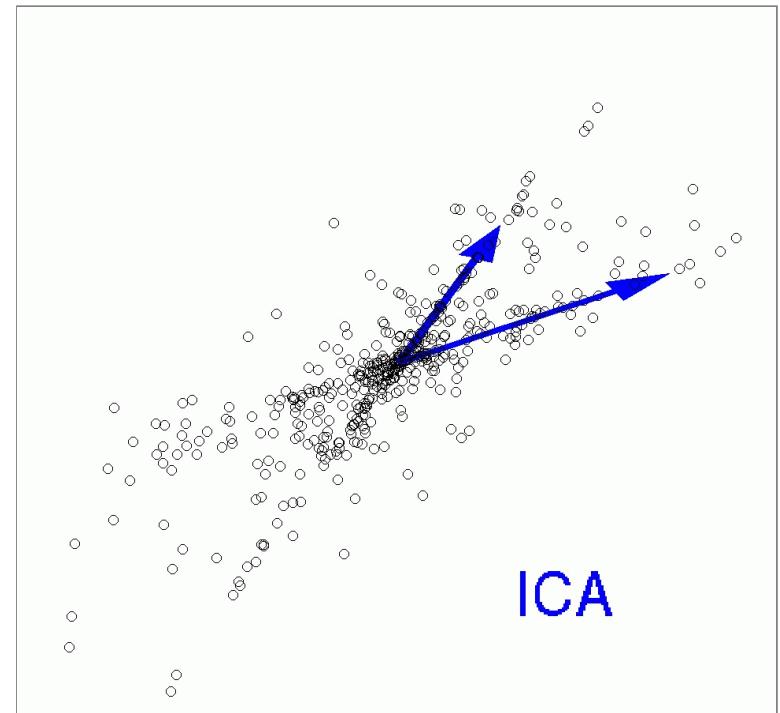
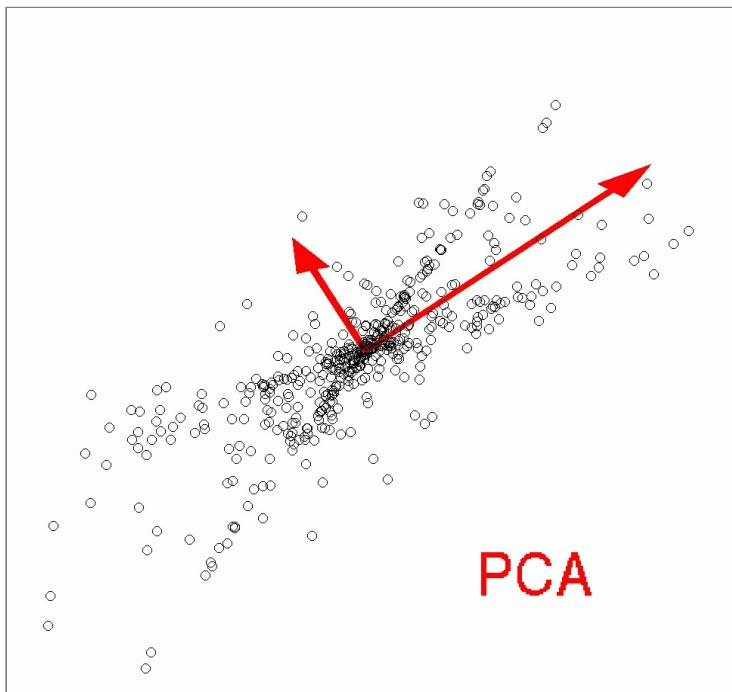
ICA vs PCA



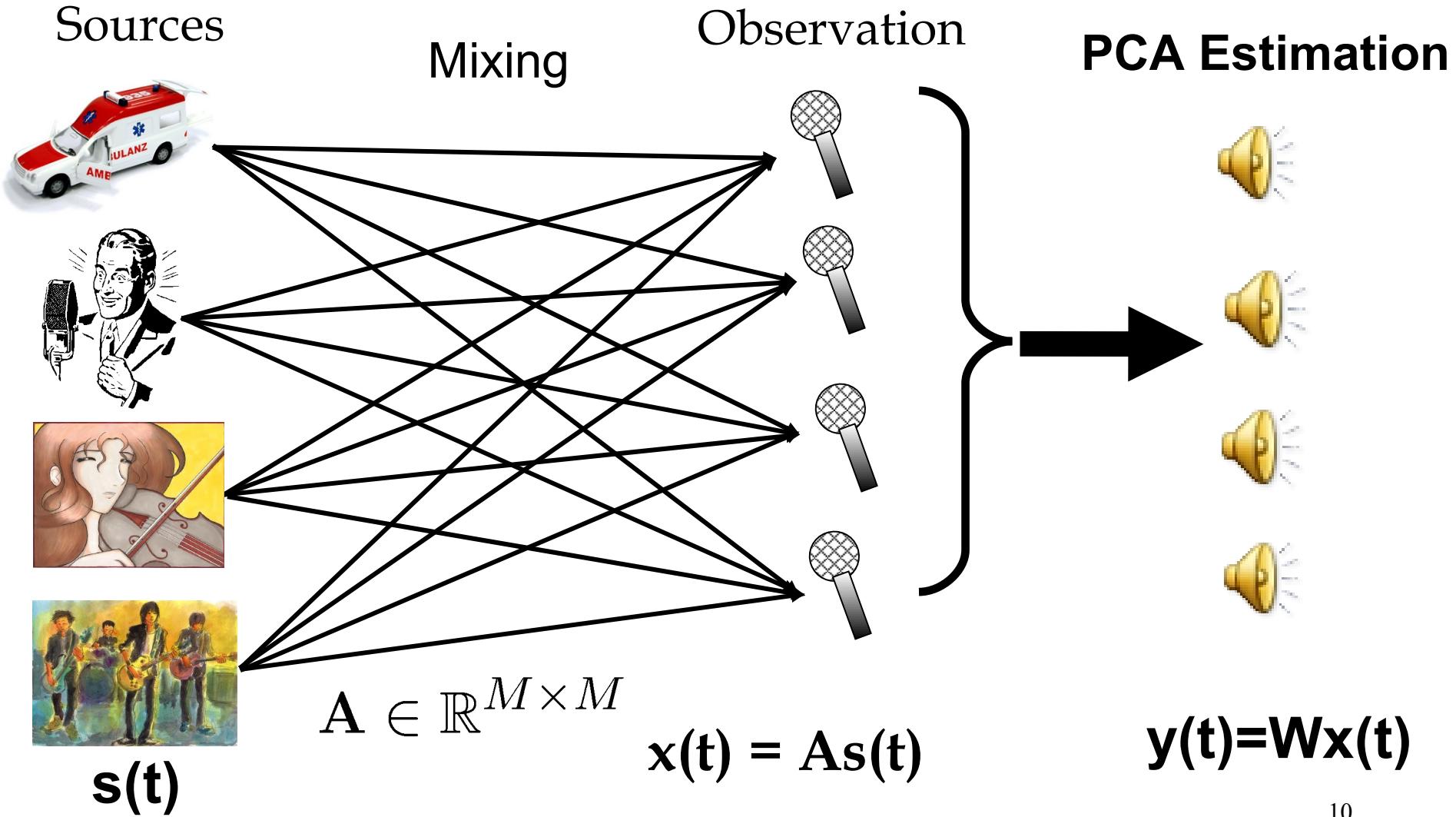
Note

- PCA vectors are orthogonal
- ICA vectors are **not** orthogonal

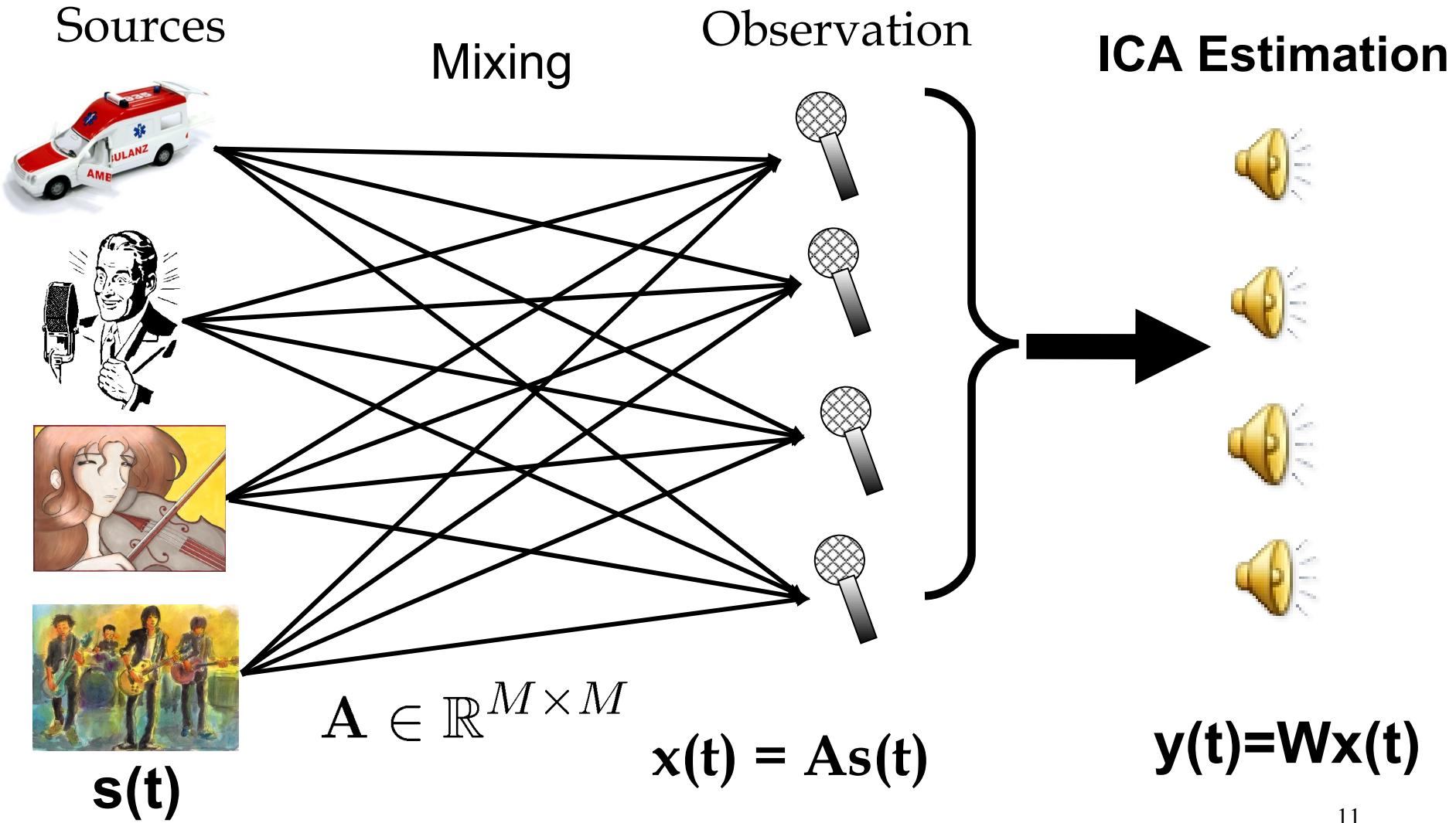
PCA vs ICA



The Cocktail Party Problem **SOLVING WITH PCA**



The Cocktail Party Problem **SOLVING WITH ICA**



Some ICA Applications

STATIC

- Image denoising
- Microarray data processing
- Decomposing the spectra of galaxies
- Face recognition
- Facial expression recognition
- Feature extraction
- Clustering
- Classification

TEMPORAL

- Medical signal processing – fMRI, ECG, EEG
- Brain Computer Interfaces
- Modeling of the hippocampus, place cells
- Modeling of the visual cortex
- Time series analysis
- Financial applications
- Blind deconvolution

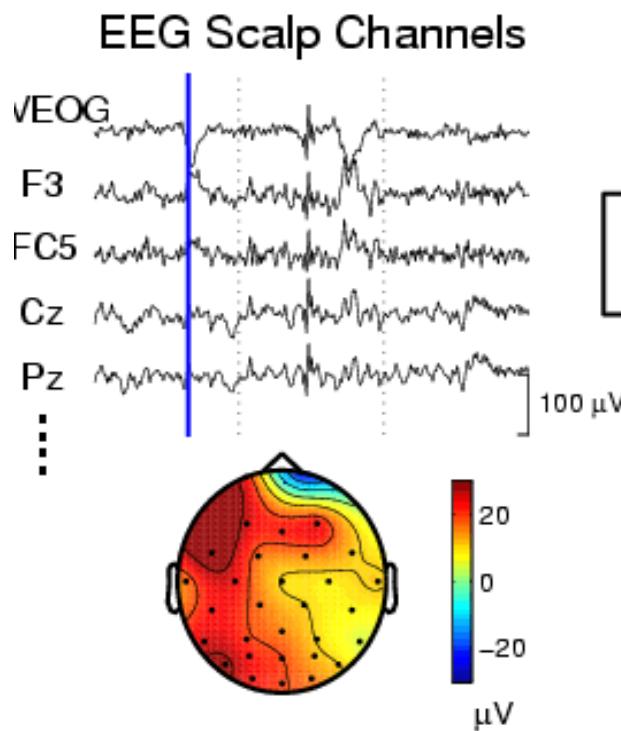
ICA Application, Removing Artifacts from EEG

- EEG \sim *Neural cocktail party*
- Severe **contamination** of EEG activity
 - eye movements
 - blinks
 - muscle
 - heart, ECG artifact
 - vessel pulse
 - electrode noise
 - line noise, alternating current (60 Hz)
- ICA can improve signal
 - effectively **detect, separate and remove** activity in EEG records from a wide variety of artifactual sources.
(Jung, Makeig, Bell, and Sejnowski)
- ICA weights help find **location** of sources





ICA decomposition



unmixing
(W)

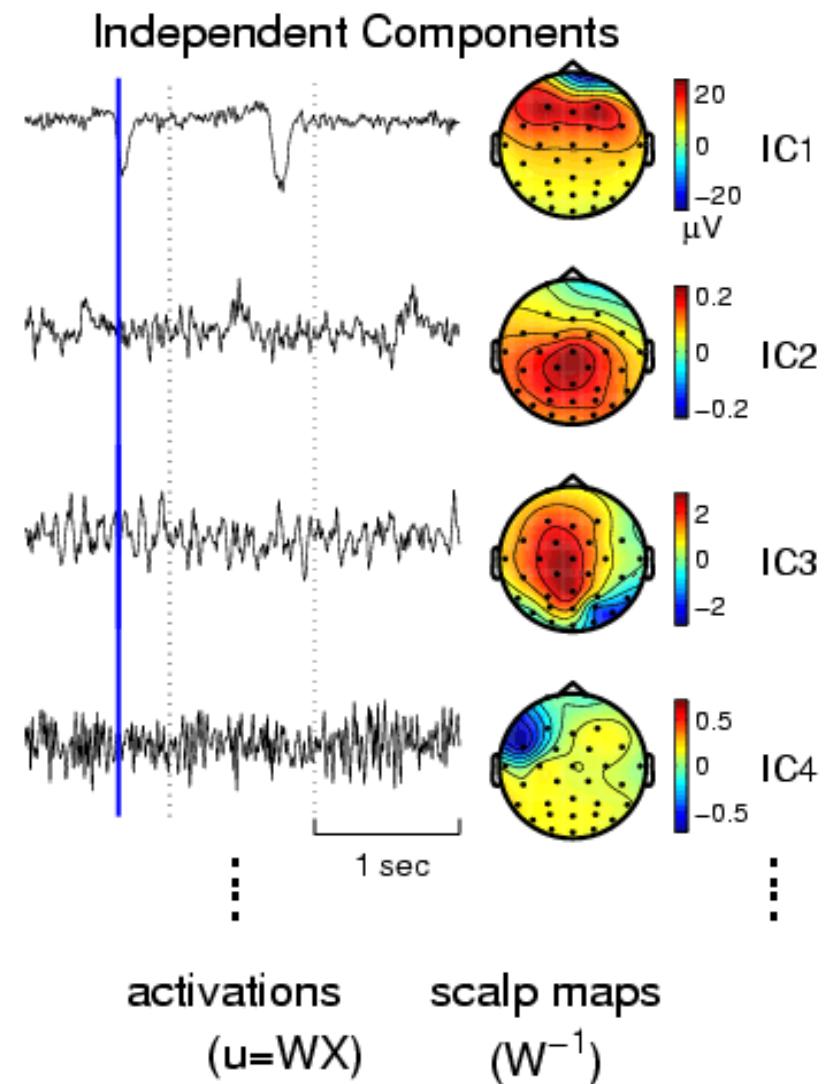
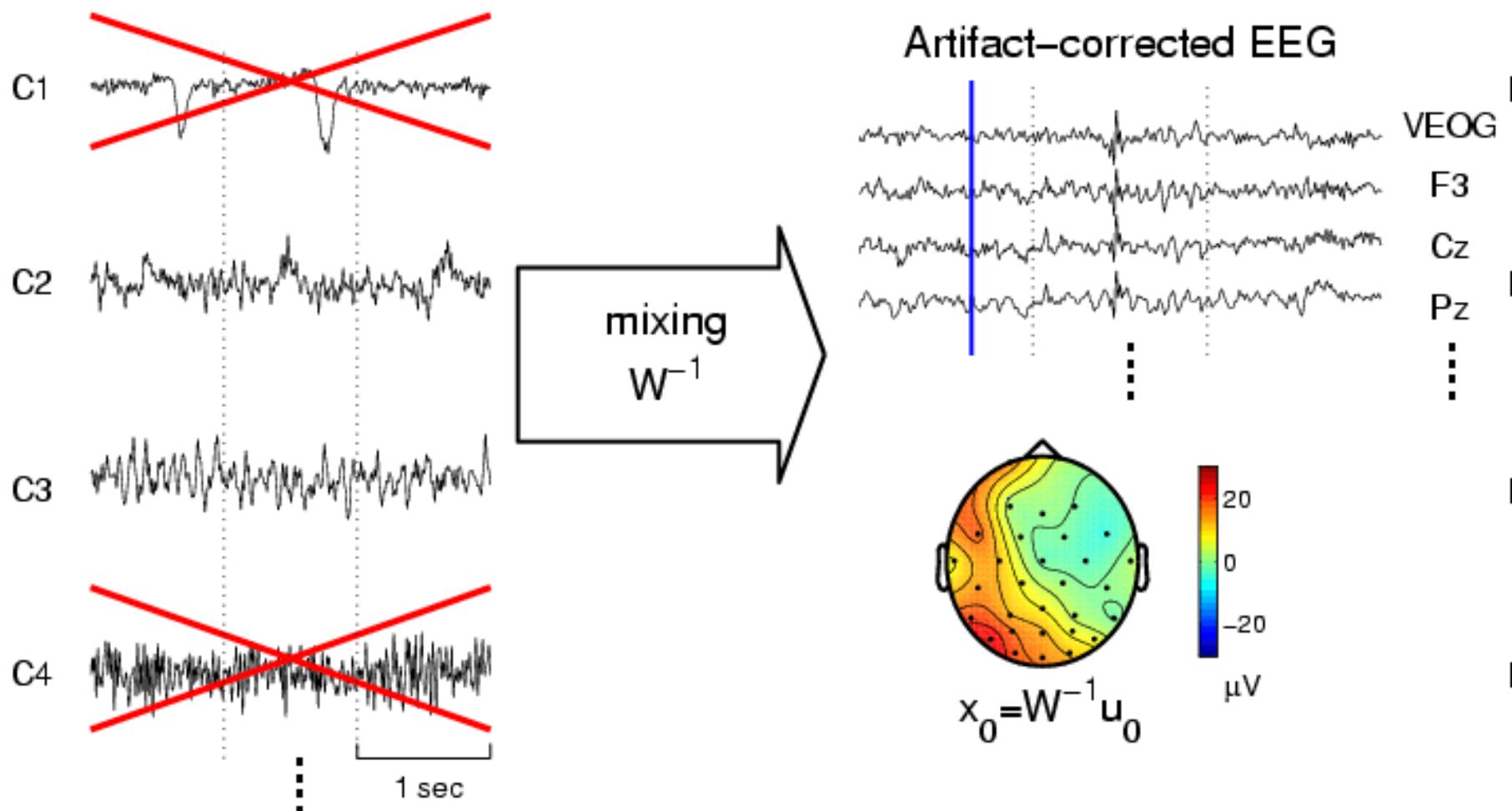


Fig from Jung

Summed Projection of Selected Components



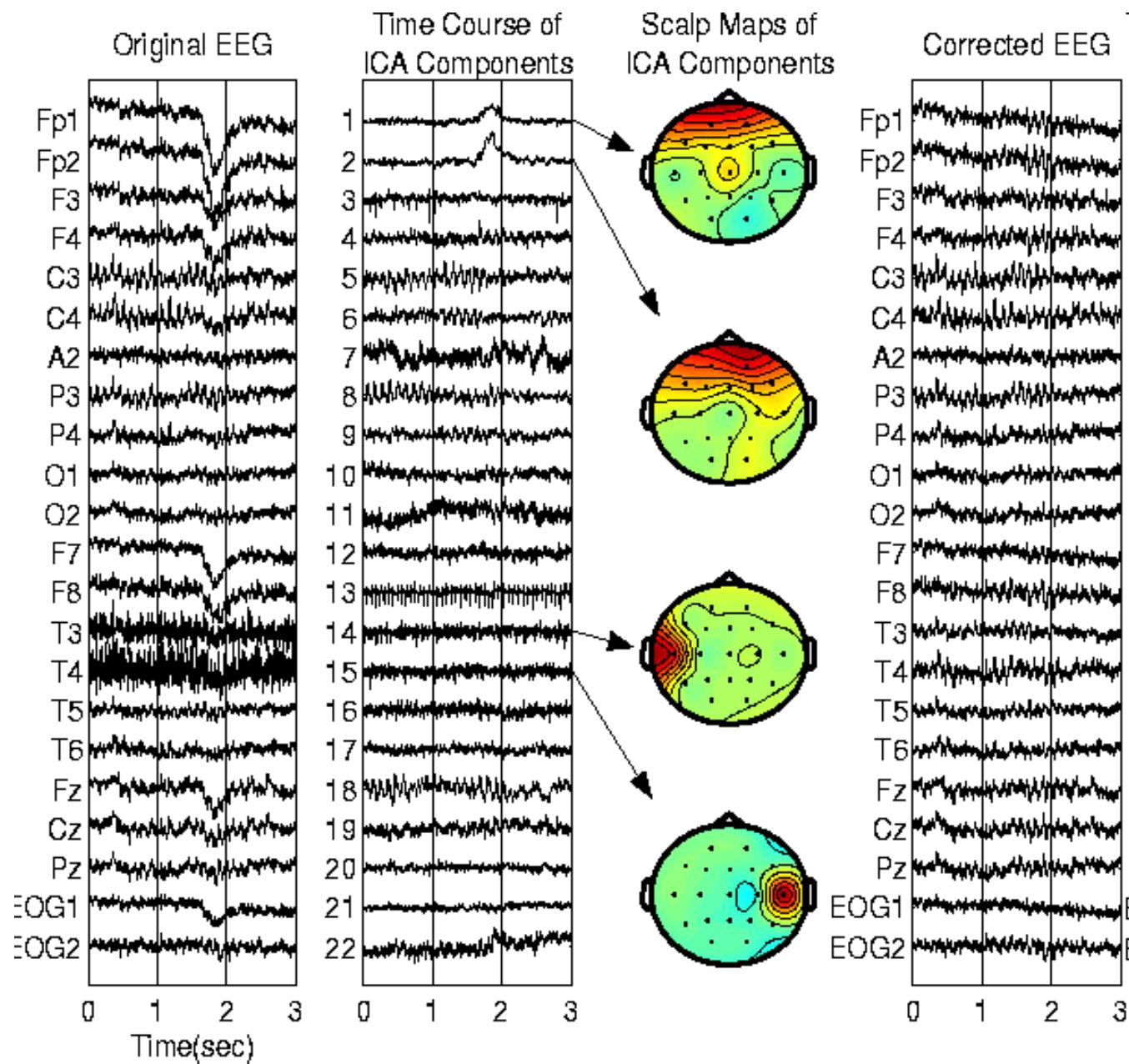


Fig from Jung

PCA+ICA for Microarray data processing

$$X^T = \begin{matrix} \text{labels} \\ \downarrow \\ \text{matrix of gene expression levels} \end{matrix} = A \begin{pmatrix} a_1 & a_2 & \dots & a_k \end{pmatrix}^T * S$$

Assumption:

- each experiment is a mixture of **independent expression modes** (s_1, \dots, s_K).
- some of these modes (e.g. s_k) can be related to the difference between the classes.
- $\rightarrow a_k$ correlates with the class labels

PCA alone can estimate
US only \Rightarrow doesn't work

ICA for Microarray data processing

(Schachtner et al, ICA07)

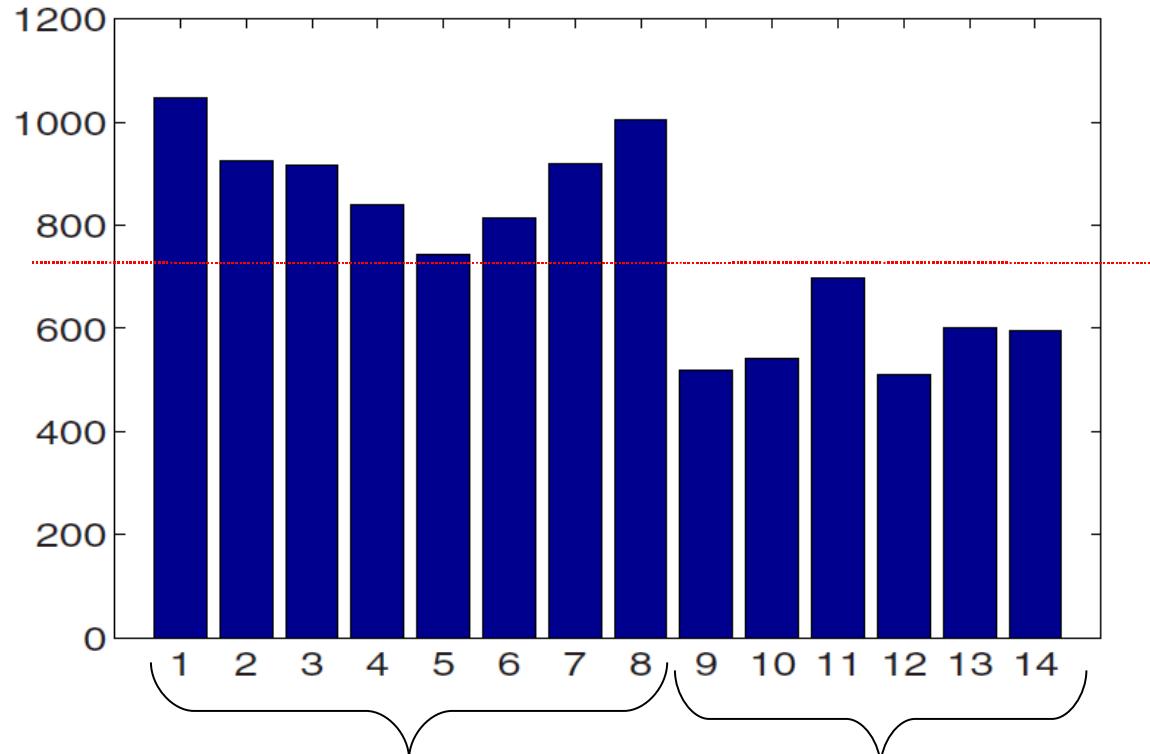
Breast Cancer Data set

M=14 patients

N=22283 genes

2 classes

9th column of A:



$|\text{Corr}(\mathbf{a}_9, \mathbf{d})|=0.89$, where
 \mathbf{d} is the vector of class labels

Class 1,
weak metastasis

Class 2,
strong metastasis

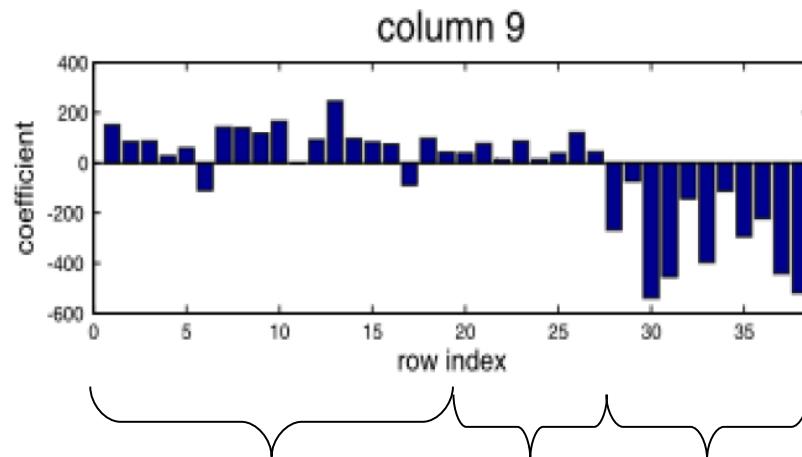
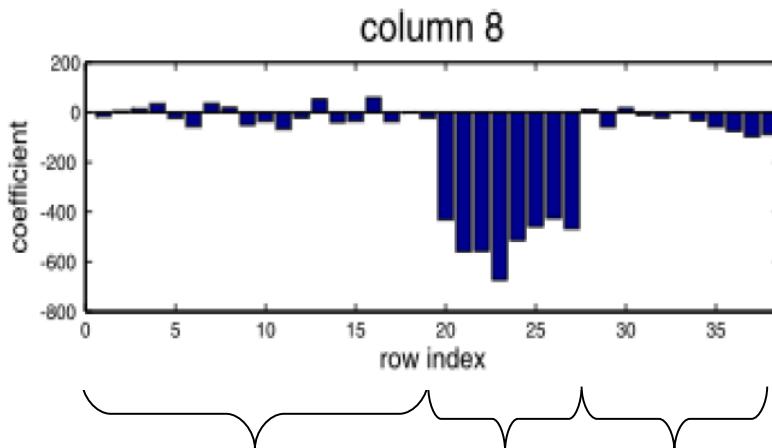
ICA for Microarray data processing

(Schachtner et al, ICA07)

Leukemia Data set $M=38$ Patients

$N=5000$ genes

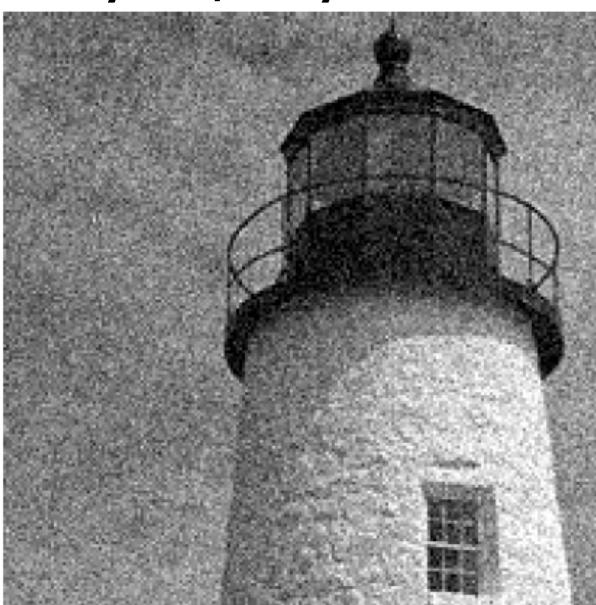
3 classes: ALL-B, ALL-T, AML



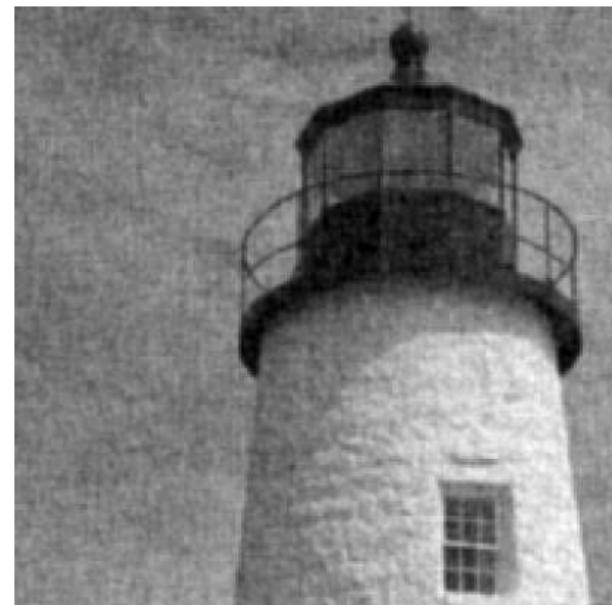
ICA for Image Denoising (Hoyer, Hyvarinen)



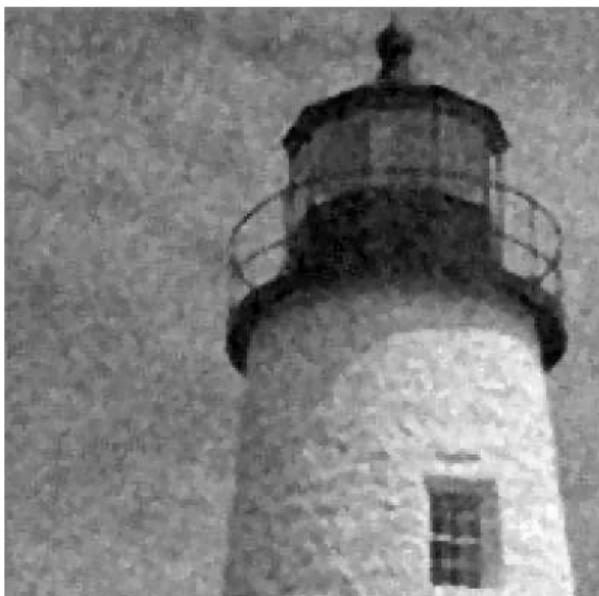
original



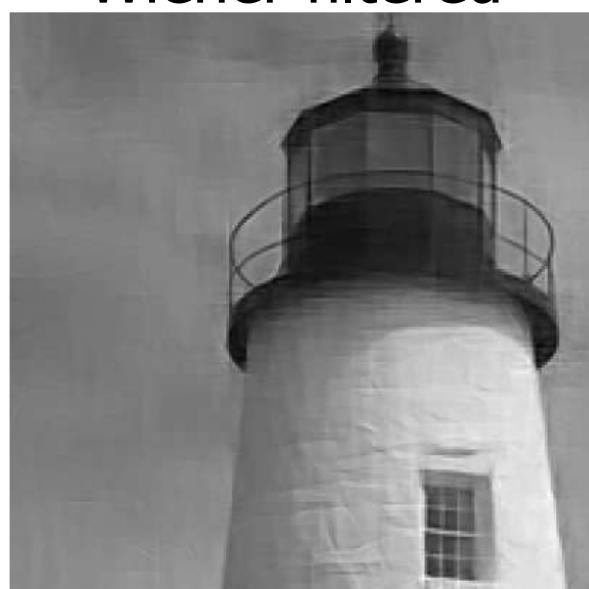
noisy



Wiener filtered



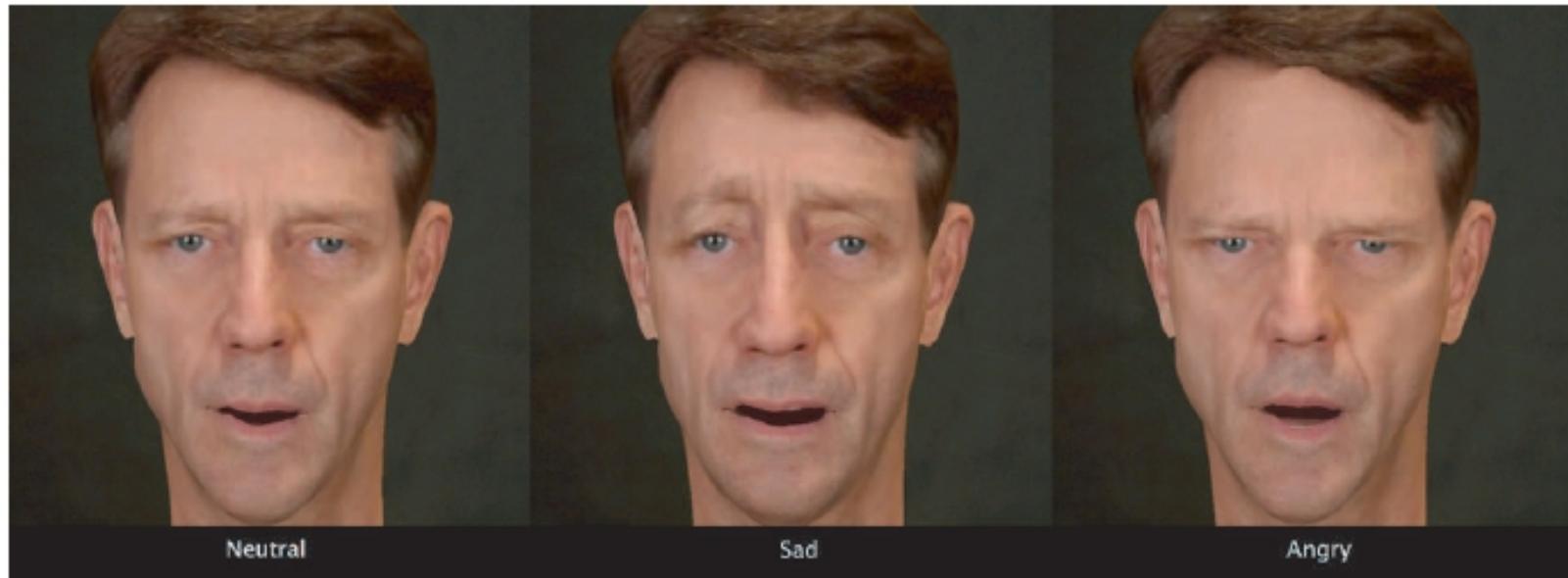
median filtered



ICA for Motion Style Components

(Mori & Hoshino 2002, Shapiro et al 2006, Cao et al 2003)

- Method for analysis and synthesis of human motion from motion captured data
- Provides perceptually meaningful components
- 109 markers, 327 parameters
 ⇒ 6 independent components (emotion, content,...)





Neutral



53% Sad + 46% Neutral



46% Sad + 53% Neutral



Sad



Neutral



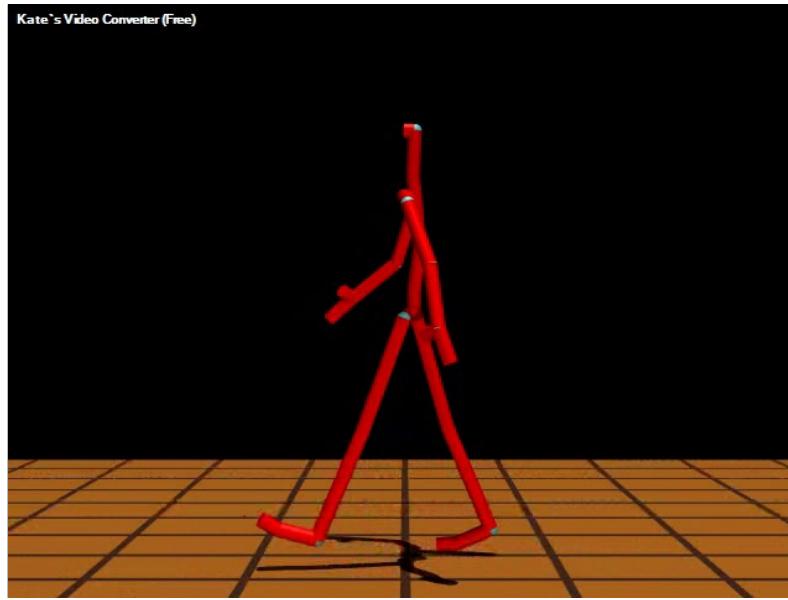
53% Sad + 46% Neutral



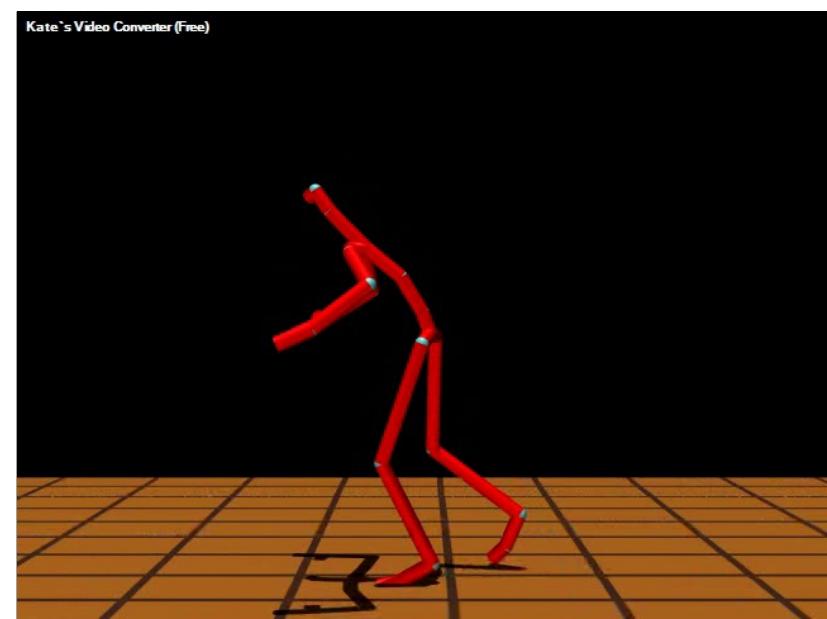
46% Sad + 53% Neutral



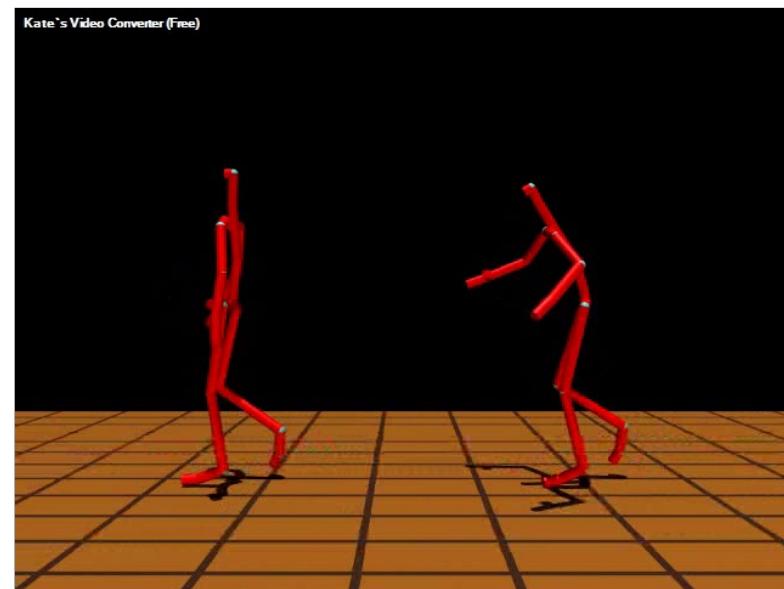
Sad



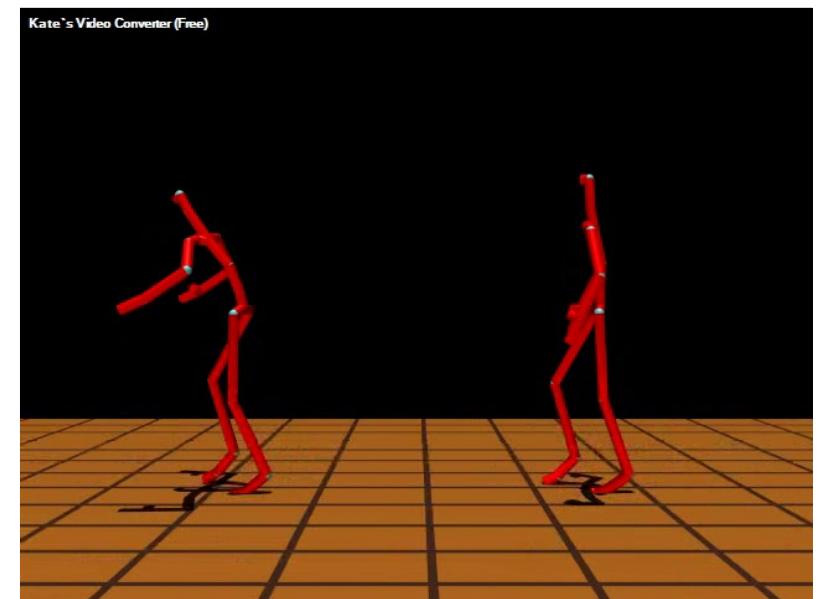
walk



sneaky

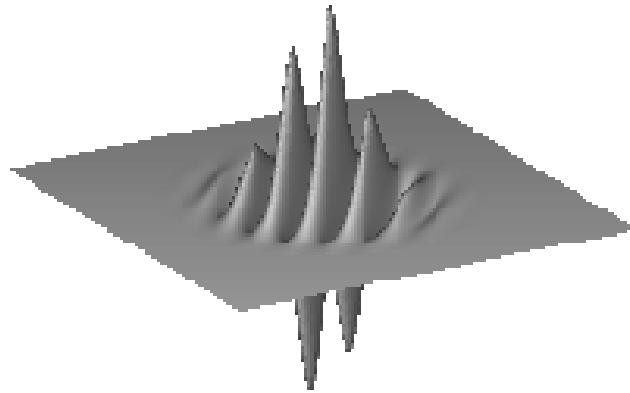
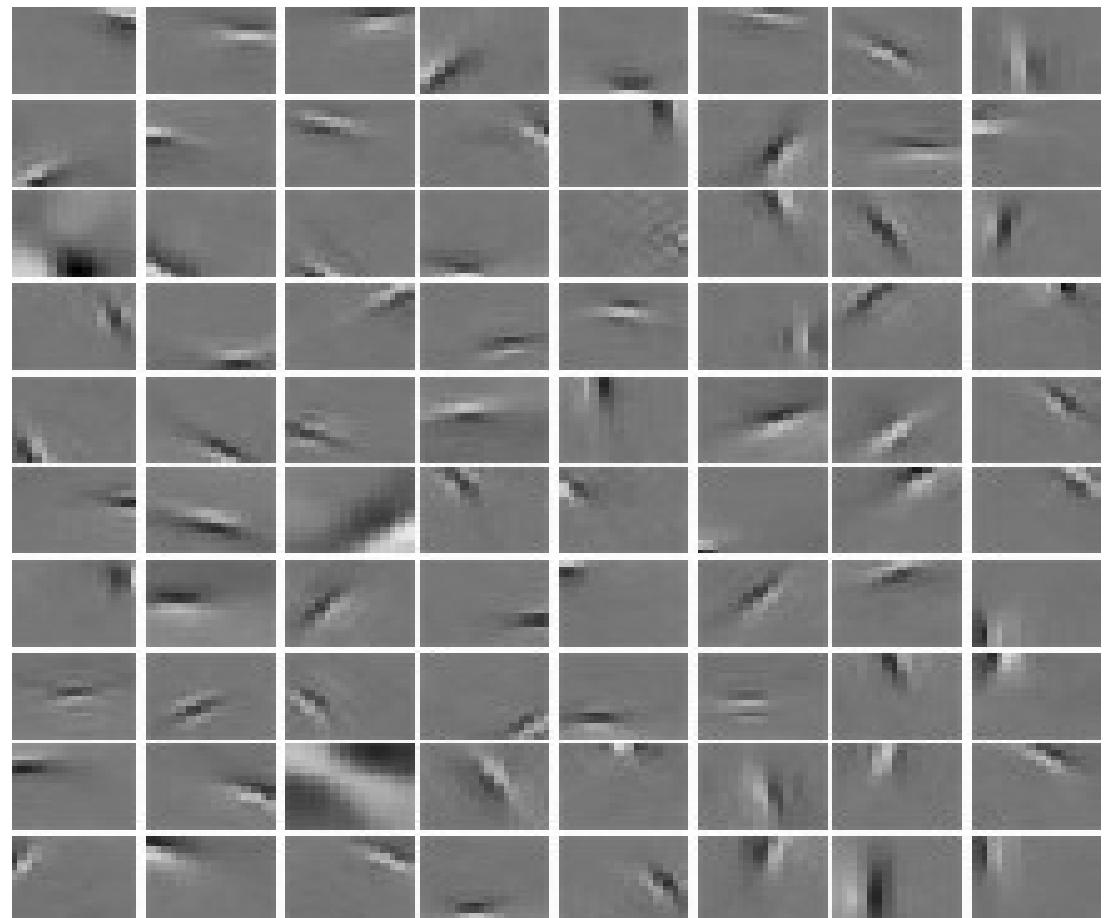


walk with sneaky



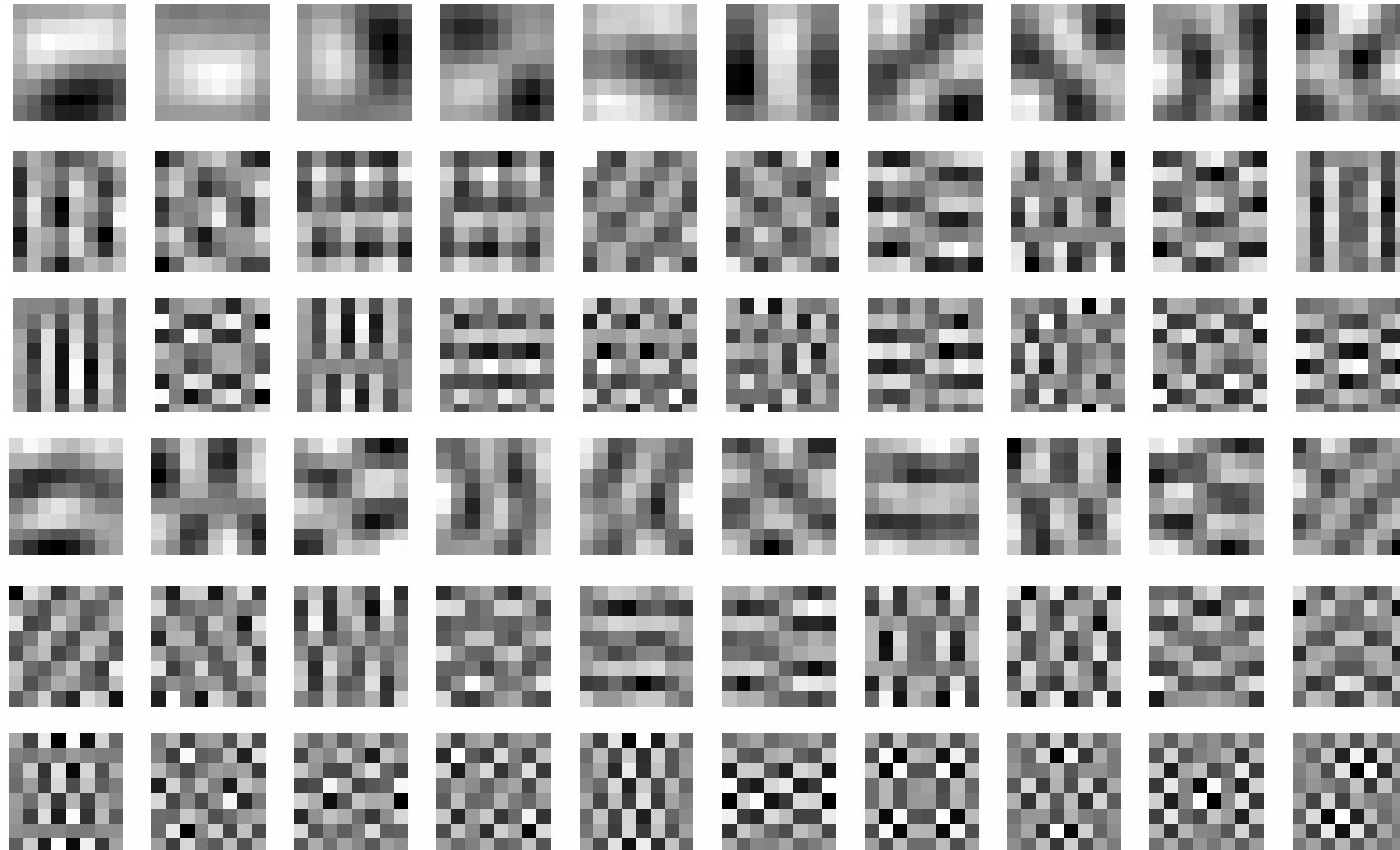
sneaky with walk

ICA basis vectors extracted from natural images



Gabor wavelets,
edge detection,
receptive fields of V1 cells...

PCA basis vectors extracted from natural images



Using ICA for classification

Activity distributions of

- within-category test images are much narrower
- off-category is closer to the Gaussian distribution

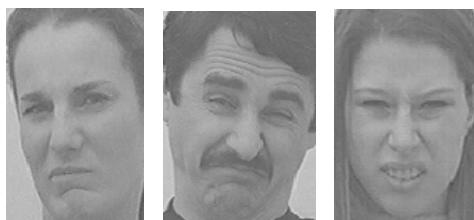
Happy



ICA basis
→ [Happy]

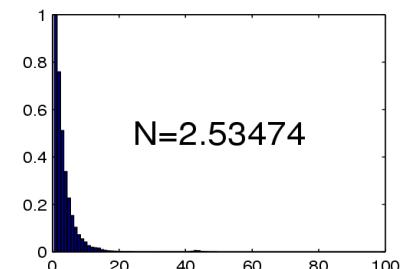
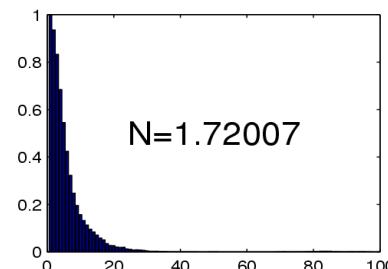
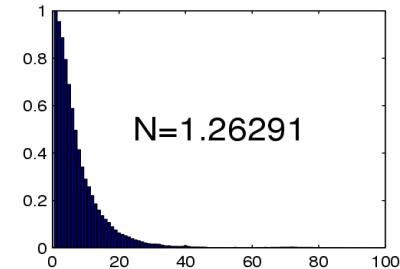
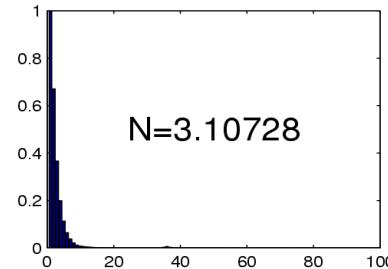
Train data

Disgust



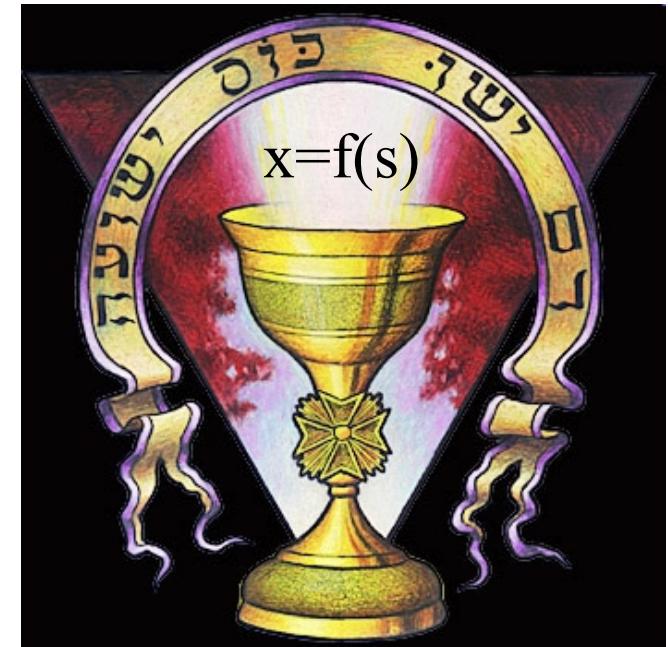
ICA basis
[Disgust]

Test data



ICA Generalizations

- Independent Subspace Analysis
- Multilinear ICA
- Blind Source Deconvolution
- Blind SubSpace Deconvolution
- Nonnegative ICA
- Sparse Component Analysis
- Slow Component Analysis
- Noisy ICA
- Undercomplete, Overcomplete ICA
- Varying mixing matrix
- Online ICA
- (Post) Nonlinear ICA



The Holy Grail

ICA Theory

R L
&
A I



Basic terms, definitions

- uncorrelated and independent variables
- entropy, joint entropy, neg_entropy
- mutual information
- Kullback-Leibler divergence

Statistical (in)dependence

Definition:

Y_1, Y_2 are independent $\Leftrightarrow p(y_1, y_2) = p(y_1)p(y_2)$

Lemma:

Let h_1, h_2 be arbitrary functions.

Y_1, Y_2 are independent \Rightarrow

$$\mathbb{E}[h_1(Y_1) h_2(Y_2)] = \mathbb{E}[h_1(Y_1)] \mathbb{E}[h_2(Y_2)]$$

Proof: Homework

Correlation

Definition:

$$\text{corr}(Y_1, Y_2) = \frac{\mathbb{E}[(Y_1 - \mathbb{E}[Y_1])(Y_2 - \mathbb{E}[Y_2])]}{\text{var}(Y_1)^{1/2} \text{var}(Y_2)^{1/2}}$$

Lemma:

$$\text{corr}(Y_1, Y_2) = 0 \Leftrightarrow \mathbb{E}[Y_1 Y_2] = \mathbb{E}[Y_1] \mathbb{E}[Y_2]$$

Proof: Homework

Lemma:

$$Y_1, Y_2 \text{ are independent} \stackrel{\Rightarrow}{\Leftarrow} Y_1, Y_2 \text{ are uncorrelated}$$

Proof: Homework

Lemma: If (Y_1, Y_2) are jointly Gaussian, then
 Y_1, Y_2 are independent $\Leftrightarrow Y_1, Y_2$ are uncorrelated

Proof: Homework

Mutual Information, Entropy

Definition (Mutual Information)

$$\begin{aligned} 0 \leq I(Y_1, \dots, Y_M) &\doteq \int p(y_1, \dots, y_M) \log \frac{p(y_1, \dots, y_M)}{p(y_1) \dots p(y_M)} dy \\ &= KL(p(y_1, \dots, y_M) || p(y_1) \dots p(y_M)) \\ &= \sum_{i=1}^M H(Y_i) - H(Y_1, \dots, Y_M) \end{aligned}$$

Definition (Shannon entropy)

$$H(Y) \doteq H(Y_1, \dots, Y_m) \doteq - \int p(y_1, \dots, y_m) \log p(y_1, \dots, y_m) dy.$$

Definition (KL divergence)

$$0 \leq KL(f||g) = \int f(x) \log \frac{f(x)}{g(x)} dx$$

Solving the ICA problem with i.i.d. sources

ICA problem: $\mathbf{x} = \mathbf{A}\mathbf{s}$, $\mathbf{s} = [s_1; \dots; s_M]$ are jointly independent.

1

Ambiguity:

$\mathbf{s} = [s_1; \dots; s_M]$ sources can be recovered only up to
sign, scale and permutation.

Proof:

- \mathbf{P} = arbitrary permutation matrix,
- Λ = arbitrary diagonal scaling matrix.

$$\Rightarrow \mathbf{x} = [\mathbf{A}\mathbf{P}^{-1}\Lambda^{-1}][\Lambda\mathbf{P}\mathbf{s}]$$

Solving the ICA problem with i.i.d. sources

Lemma:

We can assume that $E[\mathbf{s}] = 0$.

Proof:

Removing the mean does not change the mixing matrix.

$$\mathbf{x} - E[\mathbf{x}] = \mathbf{A}(\mathbf{s} - E[\mathbf{s}]).$$

In what follows we assume that $E[\mathbf{s}\mathbf{s}^T] = \mathbf{I}_M$, $E[\mathbf{s}] = 0$.

Whitening

Let $\mathbf{A} \in \mathbb{R}^{N \times M}$ with full rank, $N \geq M$, and $\mathbf{x} = \mathbf{As}$

Theorem (Whitening)

1

$\exists \mathbf{Q} \in \mathbb{R}^{M \times N}$ such that
if $\mathbf{x}^* \doteq \mathbf{Qx} = \mathbf{QAs} = \mathbf{A}^*\mathbf{s}$, $\mathbf{A}^* \doteq \mathbf{QA}$

$$\Rightarrow E[\mathbf{x}^* \mathbf{x}^{*T}] = \mathbf{I}_M, \text{ and } \mathbf{A}^* \mathbf{A}^{*T} = \mathbf{I}_M.$$

Definitions

- $\mathbf{x}^* = \mathbf{Qx}$ transformation is the *whitening* transformation.
- \mathbf{Q} is the *whitening matrix*
- $\boxed{\mathbf{x}^* \doteq \mathbf{A}^*\mathbf{s}}$ is the *whitened ICA task*.

Note

After whitening we need only to consider
orthogonal matrices for (de)mixing. (\mathbf{A}^* is orthogonal)

Proof of the whitening theorem

We can use PCA for whitening!

- Let $\Sigma \doteq cov(\mathbf{x}) = E[\mathbf{xx}^T] = \mathbf{A}E[\mathbf{ss}^T]\mathbf{A}^T = \mathbf{AA}^T$.
- Do **PCA**: $\Sigma \in \mathbb{R}^{N \times N}$, $rank(\Sigma) = M$,
 $\Rightarrow \Sigma = \mathbf{U}\mathbf{D}\mathbf{U}^T$,
where $\mathbf{U} \in \mathbb{R}^{N \times M}$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}_M$, **Principal vectors**
 $\mathbf{D} \in \mathbb{R}^{M \times M}$, diagonal with rank M . **Principal values**
- Let $\mathbf{Q} \doteq \mathbf{D}^{-1/2}\mathbf{U}^T \in \mathbb{R}^{M \times N}$ *whitening matrix*
- Let $\mathbf{A}^* \doteq \mathbf{Q}\mathbf{A}$
- $\mathbf{x}^* \doteq \mathbf{Q}\mathbf{x} = \mathbf{Q}\mathbf{As} = \mathbf{A}^*\mathbf{s}$ is our new (*whitened*) ICA task.

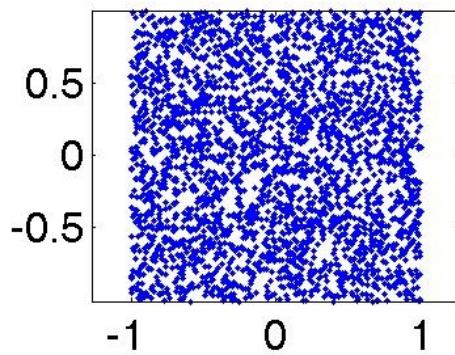
$$\Rightarrow E[\mathbf{x}^*\mathbf{x}^{*T}] = \mathbf{I}_M, \text{ and } \mathbf{A}^*\mathbf{A}^{*T} = \mathbf{I}_M.$$

Whitening solves half of the ICA problem

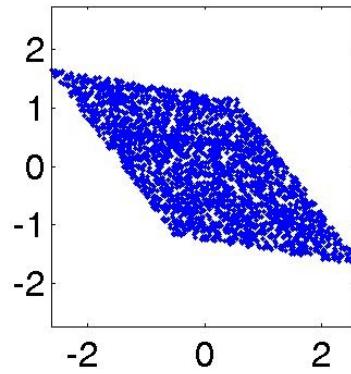
Note:

The number of free parameters of an N by N orthogonal matrix is $(N-1)(N-2)/2$.

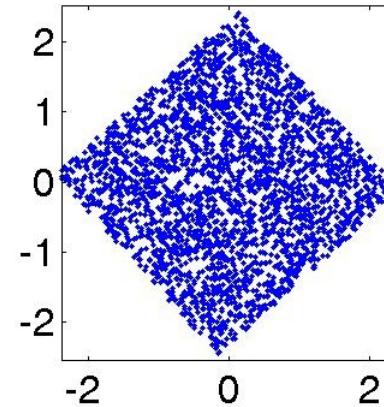
⇒ whitening solves **half** of the ICA problem



original



mixed



whitened

After whitening it is enough to consider
orthogonal matrices only for separation.

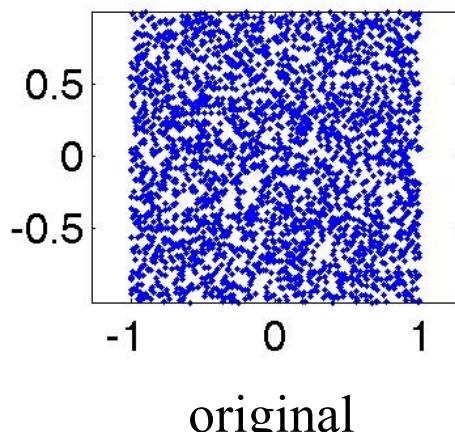
Solving ICA

ICA task: Given \mathbf{x} ,

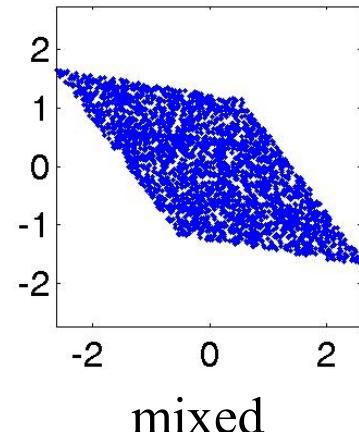
- find \mathbf{y} (the estimation of \mathbf{s}),
- find \mathbf{W} (the estimation of \mathbf{A}^{-1})

ICA solution: $\mathbf{y} = \mathbf{W}\mathbf{x}$

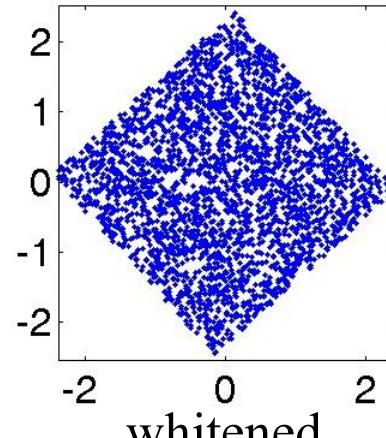
- Remove mean, $E[\mathbf{x}] = 0$
- Whitening, $E[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$
- Find an orthogonal \mathbf{W} optimizing an objective function
 - Sequence of 2-d Jacobi (Givens) rotations



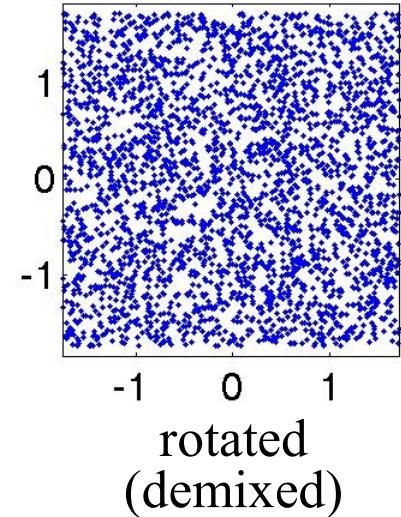
original



mixed



whitened



rotated
(demixed)

Optimization Using Jacobi Rotation Matrices

$$\mathbf{G}(p, q, \theta) \doteq \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & \cos(\theta) & \dots & -\sin(\theta) & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & \sin(\theta) & \dots & \cos(\theta) & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \begin{matrix} \leftarrow \mathbf{p} \\ \in \mathbf{R}^{M \times M} \\ \leftarrow \mathbf{q} \end{matrix}$$

$\uparrow \quad \quad \quad \uparrow$
 $\mathbf{p} \quad \quad \quad \mathbf{q}$

Observation : $\mathbf{x} = \mathbf{A}\mathbf{s}$

Estimation : $\mathbf{y} = \mathbf{W}\mathbf{x}$

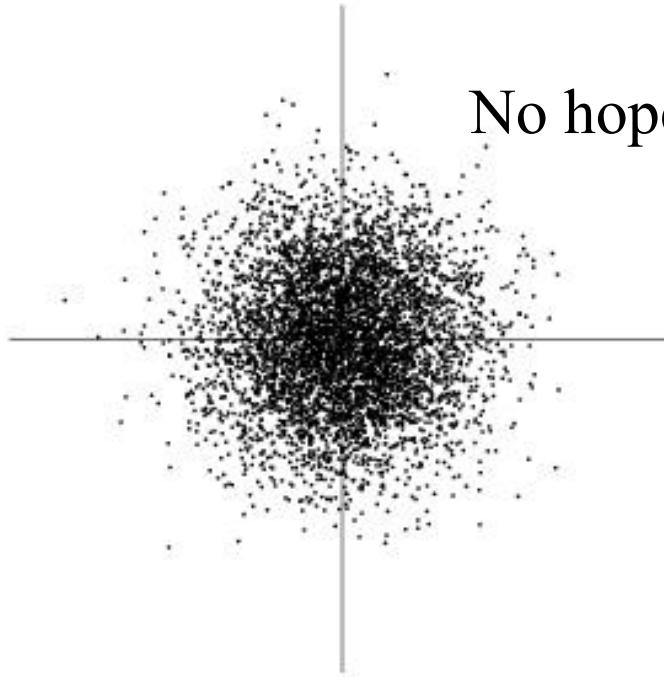
$$\mathbf{W} = \arg \min_{\tilde{\mathbf{W}} \in \mathcal{W}} J(\tilde{\mathbf{W}}\mathbf{x}),$$

where $\mathcal{W} = \{\mathbf{W} | \mathbf{W} = \prod_i G(p_i, q_i, \theta_i)\}$

Gaussian sources are problematic

The Gaussian distribution is ***spherically symmetric***.

Mixing it with an orthogonal matrix... produces the same distribution...



No hope for recovery... ☹

However, this is the only ‘nice’ distribution that we cannot recover! ☺

ICA Cost Functions

Let $\mathbf{y} \doteq \mathbf{W}\mathbf{x}$, $\mathbf{y} = [y_1; \dots; y_M]$, and let us measure the dependence using Shannon's mutual information:

$$J_{ICA_1}(\mathbf{W}) \doteq I(y_1, \dots, y_M) \doteq -\int p(y_1, \dots, y_M) \log \frac{p(y_1, \dots, y_M)}{p(y_1) \dots p(y_M)} dy,$$

Let $H(\mathbf{y}) \doteq H(y_1, \dots, y_m) \doteq -\int p(y_1, \dots, y_m) \log p(y_1, \dots, y_m) dy$.

$H(\mathbf{W}\mathbf{x}) = H(\mathbf{x}) + \log |\det \mathbf{W}|$, thus

$$\begin{aligned} I(y_1, \dots, y_M) &= \int p(y_1, \dots, y_M) \log \frac{p(y_1, \dots, y_M)}{p(y_1) \dots p(y_M)} \\ &= -H(y_1, \dots, y_M) + H(y_1) + \dots + H(y_M) \\ &= -H(x_1, \dots, x_M) - \log |\det \mathbf{W}| + H(y_1) + \dots + H(y_M). \end{aligned}$$

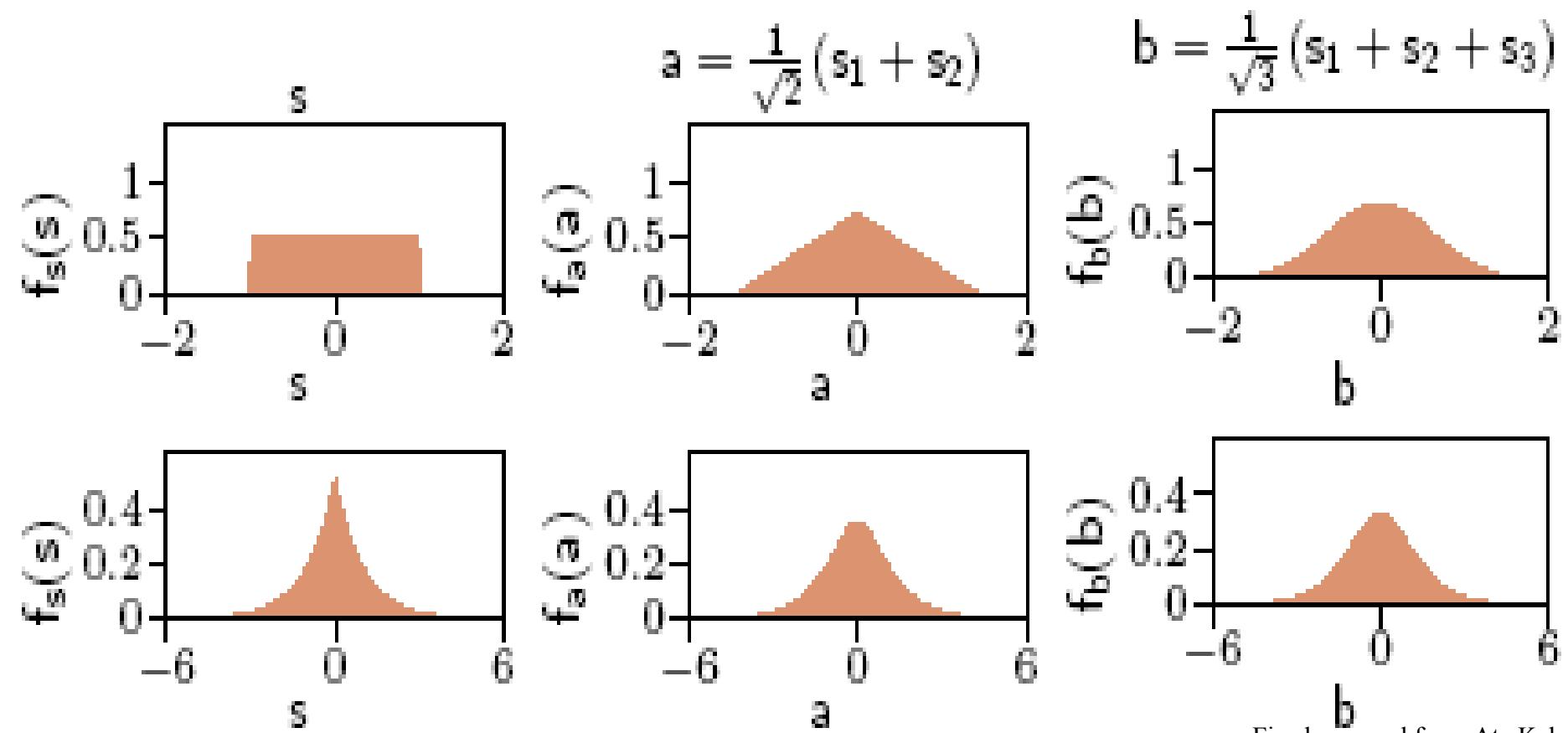
$H(x_1, \dots, x_M)$ is constant, $\log |\det \mathbf{W}| = 0$, thus

$$J_{ICA_2}(\mathbf{W}) \doteq H(y_1) + \dots + H(y_M)$$

\Rightarrow go away from normal distribution

Central Limit Theorem

The sum of independent variables converges to the normal distribution
⇒ For separation go far away from the normal distribution
⇒ **Negentropy, |kurtozis| maximization**



Figs borrowed from Ata Kaban

Algorithms

There are more than 100 different ICA algorithms...

- Mutual information (MI) estimation
 - Kernel-ICA [[Bach & Jordan, 2002](#)]
- Entropy, negentropy estimation
 - Infomax ICA [[Bell & Sejnowski 1995](#)]
 - RADICAL [[Learned-Miller & Fisher, 2003](#)]
 - FastICA [[Hyvarinen, 1999](#)]
 - [[Girolami & Fyfe 1997](#)]
- ML estimation
 - KDICA [[Chen, 2006](#)]
 - EM-ICA [[Welling](#)]
 - [[MacKay 1996; Pearlmutter & Parra 1996; Cardoso 1997](#)]
- Higher order moments, cumulants based methods
 - JADE [[Cardoso, 1993](#)]
- Nonlinear correlation based methods
 - [[Jutten and Herault, 1991](#)]

ICA ALGORITHMS

R L
&
A I



Maximum Likelihood ICA Algorithm

- simplest approach
- requires knowing densities of hidden sources $\{f_i\}$

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \quad \mathbf{s}(t) = \mathbf{W}\mathbf{x}(t), \quad \text{where } \mathbf{A}^{-1} = \mathbf{W} = [\mathbf{w}_1; \dots; \mathbf{w}_M] \in \mathbb{R}^{M \times M}$$

rows of \mathbf{W}

$$L = \sum_{t=1}^T \log p(\mathbf{x}(t)) = \sum_{t=1}^T \log p(\mathbf{A}\mathbf{s}(t)) = \sum_{t=1}^T \log |\mathbf{W}| + \log p(\mathbf{s}(t))$$

$$= T \log |\mathbf{W}| + \sum_{t=1}^T \sum_{i=1}^M \log f_i(\underbrace{\mathbf{w}_i \mathbf{x}(t)}_{s_i(t)})$$

$$\Rightarrow \max_{\mathbf{W}} L \Rightarrow \frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial w_{ij}} T \log |\mathbf{W}| + \frac{\partial L}{\partial w_{ij}} \sum_{t=1}^T \sum_{k=1}^M \log f_k(\underbrace{\mathbf{w}_k \mathbf{x}(t)}_{s_k(t)})$$

$$\Rightarrow \frac{\partial L}{\partial w_{ij}} \approx T(\mathbf{W}^T)^{-1}_{ij} + \sum_{t=1}^T \frac{f'_i(s_i(t))}{f_i(s_i(t))} x_j(t)$$

$$\Rightarrow \boxed{\Delta \mathbf{W} \propto [\mathbf{W}^T]^{-1} + \frac{1}{T} \sum_{t=1}^T g(\mathbf{W}\mathbf{x}(t))\mathbf{x}^T(t)}, \quad \text{where } g_i = f'_i/f_i$$

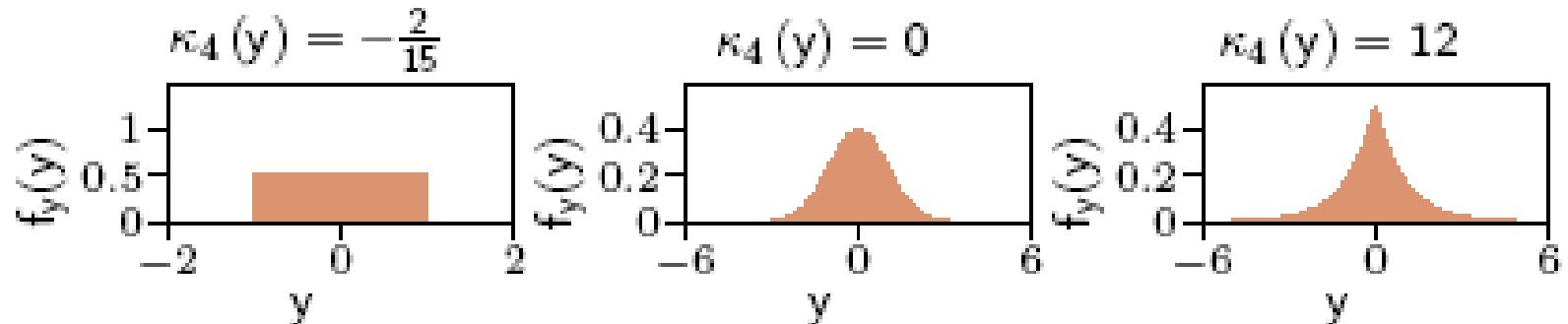
ICA algorithm based on Kurtosis maximization

Kurtosis = 4th order cumulant

Measures

- the distance from normality
- the degree of peakedness

$$\bullet \kappa_4(y) = E\{y^4\} - \underbrace{3(E\{y^2\})^2}_{= 3 \text{ if } E\{y\} = 0 \text{ and whitened}}$$



The Fast ICA algorithm (Hyvarinen)

- Given whitened data \mathbf{z}
- Estimate the 1st ICA component:

Probably the most famous ICA algorithm

- * $y = \mathbf{w}^T \mathbf{z}, \|\mathbf{w}\| = 1,$ $\Leftarrow \mathbf{w} = 1^{st}$ row of \mathbf{W}
- * maximize kurtosis $f(\mathbf{w}) \doteq \kappa_4(y) \doteq \mathbb{E}[y^4]$
with constraint $h(\mathbf{w}) = \|\mathbf{w}\|^2 - 1 = 0$
- * At optimum $f'(\mathbf{w}) + \lambda h'(\mathbf{w}) = \mathbf{0}^T$
 $\Rightarrow 4\mathbb{E}[(\mathbf{w}^T \mathbf{z})^3 \mathbf{z}] + 2\lambda \mathbf{w} = 0$
- * After calculating λ we arrive at the following iteration:

Let \mathbf{w}_1 be the fix pont of:

$$\tilde{\mathbf{w}}(k+1) = \mathbb{E}[(\mathbf{w}(k)^T \mathbf{z})^3 \mathbf{z}] - 3\mathbf{w}(k)$$

$$\mathbf{w}(k+1) = \frac{\tilde{\mathbf{w}}(k+1)}{\|\tilde{\mathbf{w}}(k+1)\|}$$

- Estimate the 2nd ICA component similarly
using the $\mathbf{w} \perp \mathbf{w}_1$ additional constraint... and so on ...

Dependence Estimation Using Kernel Methods

The Kernel ICA Algorithm

Kernel covariance (KC)

A. Gretton, R. Herbrich, A. Smola, F. Bach, M. Jordan

Let $\mathbf{x} \in \mathbb{R}^{d_x}$, $\mathbf{y} \in \mathbb{R}^{d_y}$ stochastic variables.

We want to measure their dependence.

1

$$J_{KC} \doteq \sup_{\substack{f \in \mathcal{F}^x, g \in \mathcal{F}^y \\ \|f\| \leq 1, \|g\| \leq 1}} |E\{[f(\mathbf{x}) - Ef(\mathbf{x})][g(\mathbf{y}) - Eg(\mathbf{y})]\}|$$

$$J_{KC}^{emp} \doteq \sup_{\substack{f \in \mathcal{F}^x, g \in \mathcal{F}^y \\ \|f\| \leq 1, \|g\| \leq 1}} \left| \frac{1}{m} \sum_{l=1}^m \{ [f(\mathbf{x}_l) - \frac{1}{m} \sum_{j=1}^m f(\mathbf{x}_j)][g(\mathbf{y}_l) - \frac{1}{m} \sum_{j=1}^m g(\mathbf{y}_j)] \} \right|$$

where $\mathbf{x}_1, \dots, \mathbf{x}_m$, and $\mathbf{y}_1, \dots, \mathbf{y}_m$ are m pieces of i.i.d. samples and \mathcal{F}^x , \mathcal{F}^y are sets of real valued functions.

*The calculation of the supremum over function sets is extremely difficult.
Reproducing Kernel Hilbert Spaces make it easier.*

RKHS construction for x, y stochastic variables.

Let $K^x(\cdot, \cdot) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \rightarrow \mathbb{R}$, $K^y(\cdot, \cdot) \in \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \rightarrow \mathbb{R}$ kernel functions.

These kernels define the following RKHS:

$$\mathcal{F}^x \doteq \{f : f = \sum_{j=1}^{\infty} \Psi_j \Phi_j^x(\cdot), \sum_{j=1}^{\infty} \frac{\Psi_j^2}{\lambda_j^x} < \infty\},$$

$$\mathcal{F}^y \doteq \{f : f = \sum_{j=1}^{\infty} \Psi_j \Phi_j^y(\cdot), \sum_{j=1}^{\infty} \frac{\Psi_j^2}{\lambda_j^y} < \infty\},$$

where $\Phi_j^x(\cdot)$, $\Phi_j^y(\cdot)$, λ_j^x , λ_j^y are eigenfunctions and eigenvalues corresponding to the $K^x(\cdot, \cdot)$, $K^y(\cdot, \cdot)$ Hilbert spaces.

The Representer Theorem

Theorem: $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, Mercer kernel on \mathcal{X}

$z = (x_1, y_1), \dots, (x_m, y_m) \in (\mathcal{X} \times \mathcal{Y})^m$ training sample

$g_{emp} : (\mathcal{X} \times \mathcal{Y} \times \mathbb{R})^m \rightarrow \mathbb{R} \cup \{\infty\}$

$g_{reg} : \mathbb{R} \rightarrow [0, \infty)$ strictly increasing function

\mathcal{F} : RKHS induced by $k(\cdot, \cdot)$

$$\Rightarrow f^* = \arg \min_{f \in \mathcal{F}} R_{reg}[f, z]$$

$$\doteq \arg \min_{f \in \mathcal{F}} \underbrace{g_{emp}[(x_i, y_i, f(x_i))_{i \in \{1 \dots m\}}]}_{\text{1st term, empirical loss}} + \underbrace{g_{reg}(\|f\|)}_{\text{2nd term, regularization}}$$

admits the following representation:

$$f^*(\cdot) = \sum_{i=1}^m c_i k(x_i, \cdot), \quad c = (c_1, \dots, c_m) \in \mathbb{R}^m$$

Kernel covariance (KC)

Yay! We can use the representer theorem for our problem 😊

The optimal f, g can be found in these forms:

$$f^*(\cdot) = \sum_{i=1}^m c_i k(x_i, \cdot), \quad \mathbf{c} = (c_1, \dots, c_m) \in \mathbb{R}^m$$

$$g^*(\cdot) = \sum_{i=1}^m d_i k(x_i, \cdot), \quad \mathbf{d} = (d_1, \dots, d_m) \in \mathbb{R}^m$$

Kernel covariance (KC)

$f(\mathbf{x}) = \langle f, K^x(\cdot, \mathbf{x}) \rangle_{\mathcal{F}^x}$ and $f(\cdot) = \sum_{j=1}^m c_j K^x(\cdot, \mathbf{x}_j) + f^\perp(\cdot)$, thus

$$f(\mathbf{x}_i) = \langle f, K^x(\cdot, \mathbf{x}_i) \rangle_{\mathcal{F}^x} = \left\langle \sum_{j=1}^m c_j K^x(\cdot, \mathbf{x}_j) + f^\perp(\cdot), K^x(\cdot, \mathbf{x}_i) \right\rangle_{\mathcal{F}^x} = \sum_{j=1}^m c_j K^x(\mathbf{x}_j, \mathbf{x}_i).$$

\Rightarrow

$$[f(\mathbf{x}_1) - \frac{1}{m} \sum_{i=1}^m f(\mathbf{x}_i), \dots, f(\mathbf{x}_m) - \frac{1}{m} \sum_{i=1}^m f(\mathbf{x}_i)] = \mathbf{c}^T \widetilde{\mathbf{K}}^x$$

$$[g(\mathbf{y}_1) - \frac{1}{m} \sum_{i=1}^m g(\mathbf{y}_i), \dots, g(\mathbf{y}_m) - \frac{1}{m} \sum_{i=1}^m g(\mathbf{y}_i)] = \mathbf{d}^T \widetilde{\mathbf{K}}^y$$

Where $\mathbf{K}^x = \{K(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j}$, $\mathbf{H} \doteq \mathbf{I}_m - \frac{1}{m} \mathbf{1}_m \mathbf{1}_m^T$, and $\widetilde{\mathbf{K}}^x \doteq \mathbf{H} \mathbf{K}^x \mathbf{H}$

Thus, for the estimation of J_{KC}^{emp} we have to calculate the maximum of $\mathbf{c}^T \widetilde{\mathbf{K}}^x \widetilde{\mathbf{K}}^y \mathbf{d}$ over $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ subject to $\mathbf{c}^T \widetilde{\mathbf{K}}^x \mathbf{c} = 1$, $\mathbf{d}^T \widetilde{\mathbf{K}}^y \mathbf{d} = 1$.

Amari Error for Measuring the Performance

- Measures how close a square matrix is to a permutation matrix

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

↑ ↑
demixing mixing

$$r(\mathbf{B}) = \frac{1}{2M(M-1)} \sum_{i=1}^M \left(\frac{\sum_{j=1}^M |b_{ij}|}{\max_j |b_{ij}|} - 1 \right) + \frac{1}{2M(M-1)} \sum_{j=1}^M \left(\frac{\sum_{i=1}^M |b_{ij}|}{\max_i |b_{ij}|} - 1 \right)$$

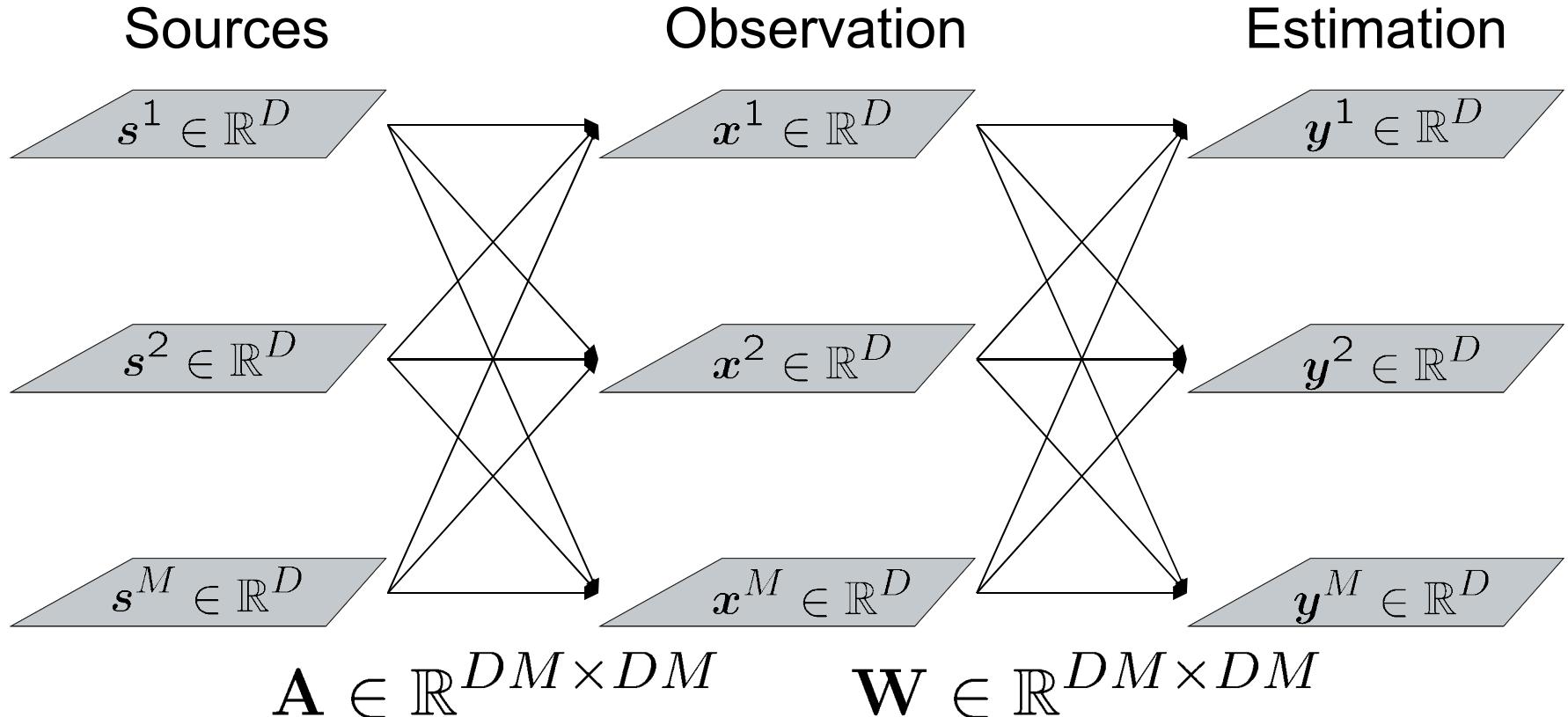
$$r(\mathbf{B}) \in [0, 1], \quad r(\mathbf{B}) = 0 \Leftrightarrow \mathbf{B} \text{ is a permutation matrix}$$

Independent Subspace Analysis

R L
&
A I



Independent Subspace Analysis (ISA, The Woodstock Problem)



Find \mathbf{W} , recover \mathbf{Wx}

Independent Subspace Analysis

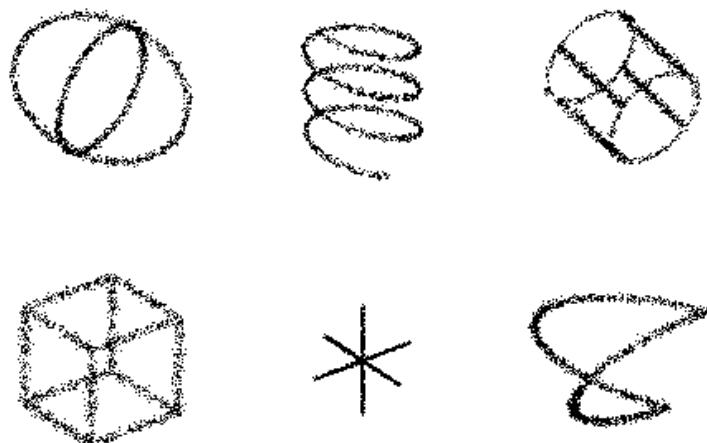
Original



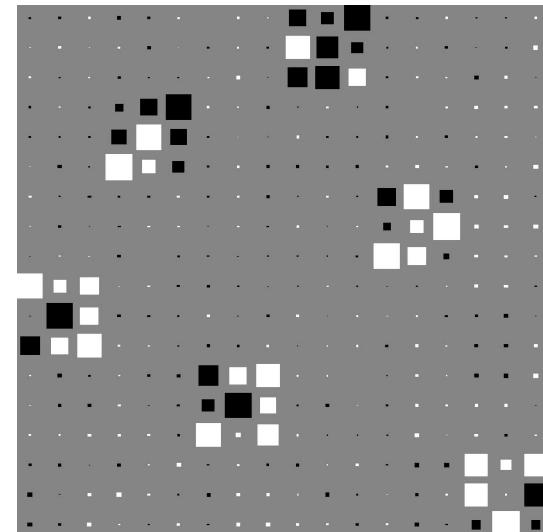
Mixed



Separated



Hinton diagram



ISA Cost Functions

Mutual Information: $I(\mathbf{y}^1, \dots, \mathbf{y}^m) = \int \log \frac{p(\mathbf{y})}{p(\mathbf{y}^1) \dots p(\mathbf{y}^m)} d\mathbf{y}$

Shannon-entropy: $H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y}$

Assume $\mathbf{y} = \mathbf{Wx}$. Then

$$H(\mathbf{y}) = H(\mathbf{y}^1, \dots, \mathbf{y}^m) = H(\mathbf{Wx}) = H(\mathbf{x}) + \log |\mathbf{W}|$$

$$I(\mathbf{y}^1, \dots, \mathbf{y}^m) = -H(\mathbf{x}) - \log |\mathbf{W}| + \sum_{i=1}^m H(\mathbf{y}^i)$$

$$I(\mathbf{y}^1, \dots, \mathbf{y}^m) = -H(\mathbf{y}^1, \dots, \mathbf{y}^m) + \sum_{j=1}^m \sum_{i=1}^d H(y_i^j) - \sum_{j=1}^m I(y_1^j, \dots, y_d^j)$$

$$H(\mathbf{y}^j) = H(y_1^j, \dots, y_d^j) = \sum_{i=1}^d H(y_i^j) - I(y_1^j, \dots, y_1^j)$$

and we get the following ISA cost functions:

ISA Cost Functions



$$J_{ISA_1}(\mathbf{W}) \doteq I(\mathbf{y}^1, \dots, \mathbf{y}^m)$$

$$J_{ISA_2}(\mathbf{W}) \doteq H(\mathbf{y}^1) + \dots + H(\mathbf{y}^m)$$

$$J_{ISA_3}(\mathbf{W}) \doteq \sum_{j=1}^m \sum_{i=1}^d H(y_i^j) - \sum_{j=1}^m I(y_1^j, \dots, y_d^j)$$

$$J_{ISA_4}(\mathbf{W}) \doteq I(y_1^1, \dots, y_d^m) - \sum_{j=1}^m I(y_1^j, \dots, y_d^j)$$

Multidimensional Entropy Estimation

Multi-dimensional Entropy Estimations, Method of Kozahenko and Leonenko

Let $\{\mathbf{z}(1), \dots, \mathbf{z}(n)\}$ denote n i.i.d. samples drawn from the distribution of $\mathbf{z} \in \mathbf{R}^d$.

Let $\mathcal{N}_{1,j}$ be the nearest neighbour of $\mathbf{z}(j)$ in the sample set.

Then the nearest neighbor entropy estimation:

$$\hat{H}(\mathbf{z}) = \frac{1}{n} \sum_{j=1}^n \log(n\|\mathcal{N}_{1,j} - \mathbf{z}(j)\|) + \ln(2) + C_E,$$

where $C_E = - \int_0^\infty e^{-t} \ln(t) dt$ is the Euler-constant.

This estimation is means-square consistent, but not robust.
Let us try to use more neighbors!

Multi-dimensional Rényi's Entropy Estimations

Let us apply Rényi's-
entropy for estimating
the Shannon-entropy:

$$H_\alpha = \frac{1}{1-\alpha} \log \int f^\alpha(\mathbf{z}) d\mathbf{z}$$
$$\lim_{\alpha \rightarrow 1} H_\alpha = - \int f(\mathbf{z}) \log f(\mathbf{z}) d\mathbf{z}$$

Let us use

- ***K-nearest neighbors***
- ***minimum spanning trees***

for estimating the multi-dimensional Rényi's entropy.
(It could be much more general...)

Beardwood - Halton - Hammersley Theorem for kNN graphs

Let $\{\mathbf{z}(1), \dots, \mathbf{z}(n)\}$ denote n i.i.d. samples drawn from the distribution of $\mathbf{z} \in \mathbf{R}^d$.

Let $\mathcal{N}_{k,j}$ be the k nearest neighbours of $\mathbf{z}(j)$ in the sample set.

Let $\gamma = d - d\alpha$, then

$$\frac{1}{1-\alpha} \log \left(\frac{1}{kn^\alpha} \sum_{j=1}^n \sum_{\mathbf{v} \in \mathcal{N}_{k,j}} \|\mathbf{v} - \mathbf{z}(j)\|^\gamma \right) \rightarrow H_\alpha(\mathbf{z}) + c,$$

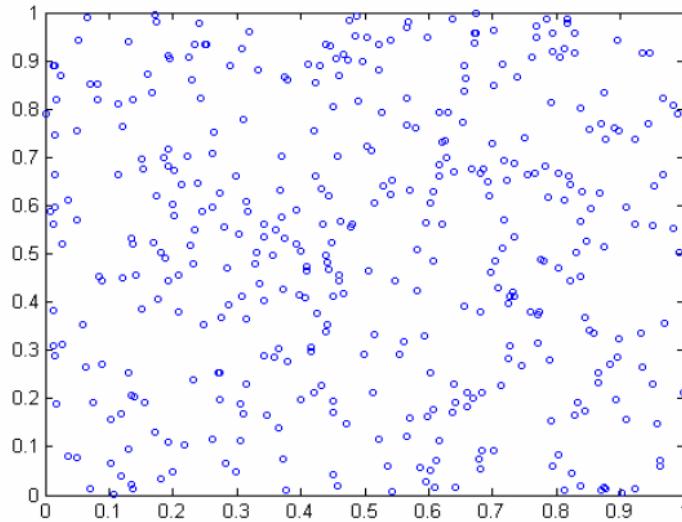
as $n \rightarrow \infty$

Lots of other graphs, e.g. MST, TSP, minimal matching, Steiner graph...etc could be used as well.

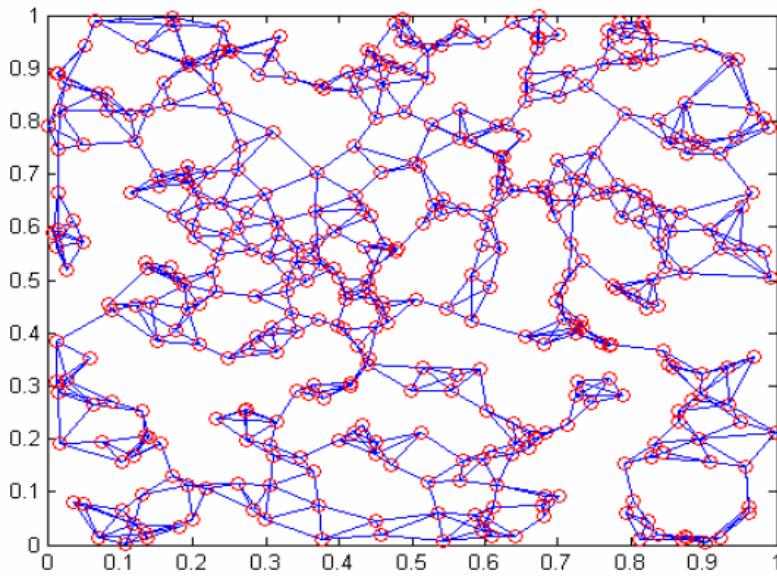
Examples

(J. A. Costa and A. O. Hero)

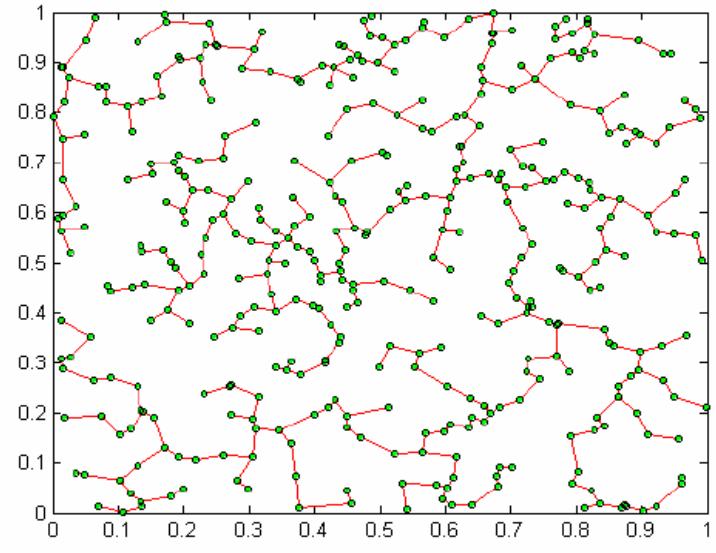
Uniform on unit square: $n = 400$ samples



4-NNG on 2D uniform: $\gamma = 1$

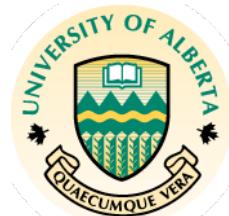


MST on 2D uniform: $\gamma = 1$



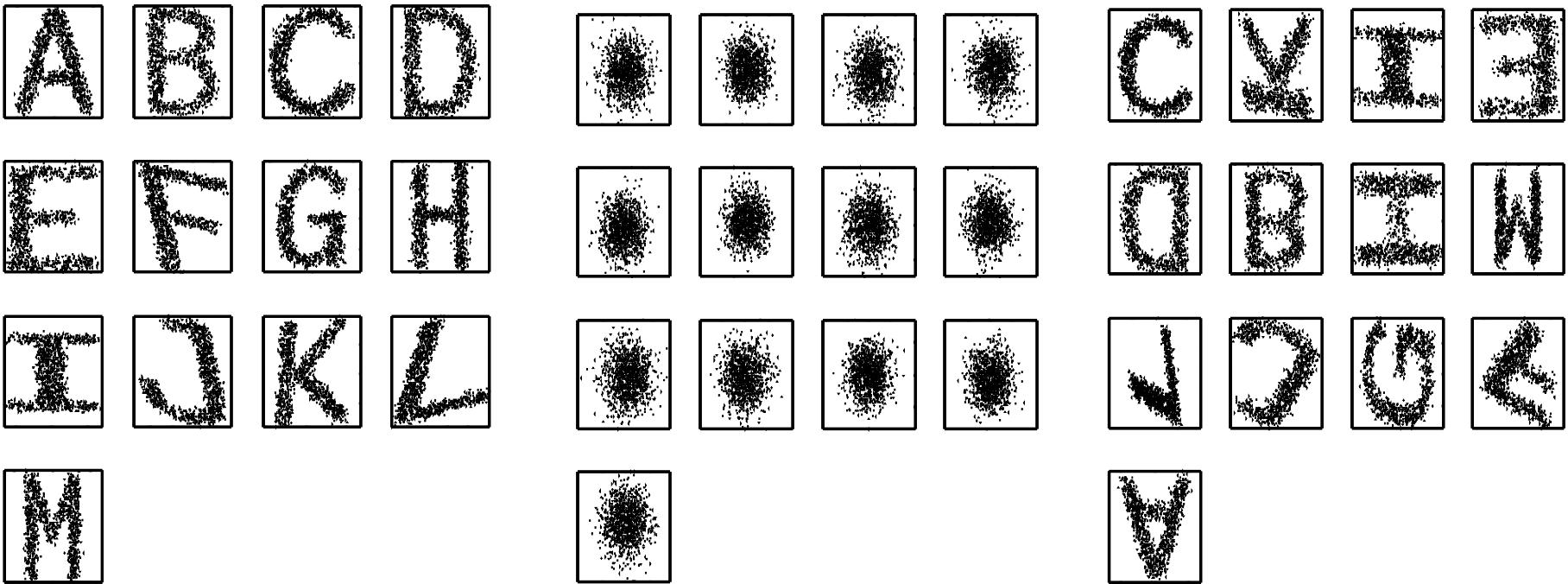
Independent Subspace Analysis Results

R L
&
A I



Numerical Simulations

2D Letters (i.i.d.)

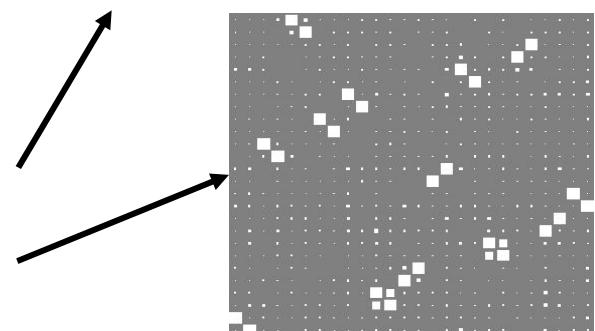


Sources

Observation

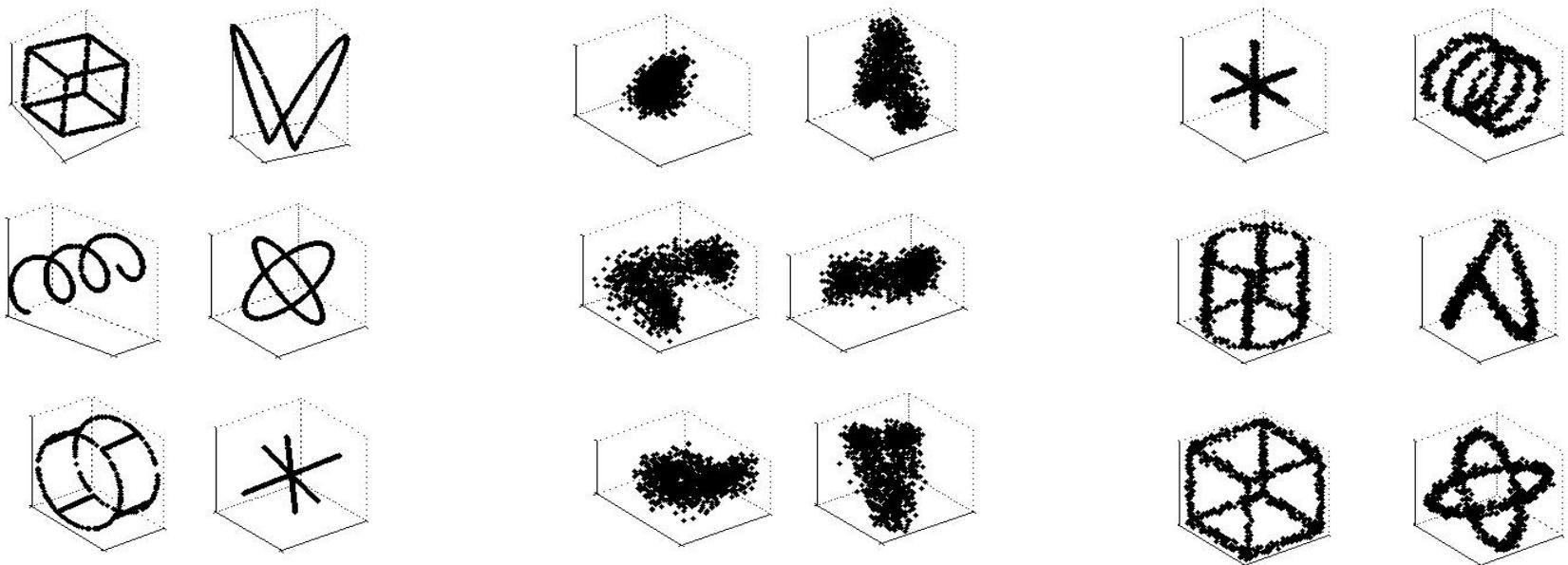
Estimated sources

Performance matrix



Numerical Simulations

3D Curves (i.i.d.)

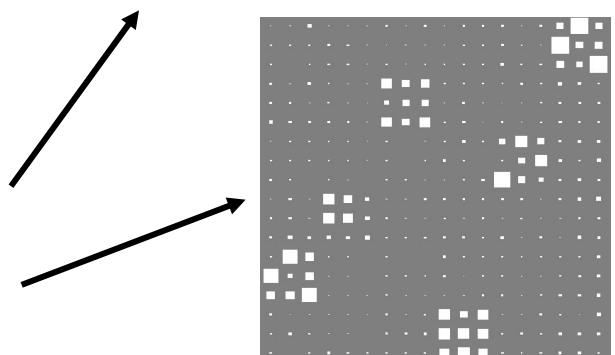


Sources

Observation

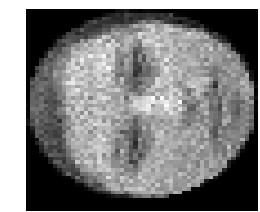
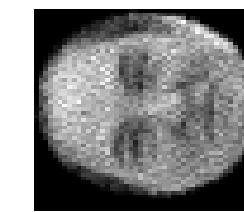
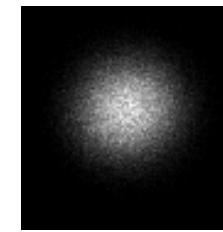
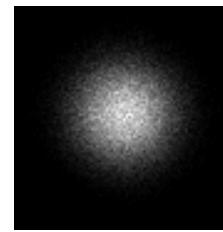
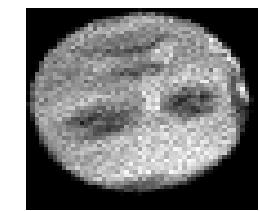
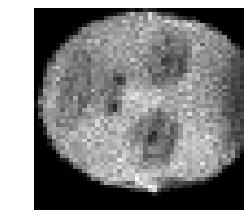
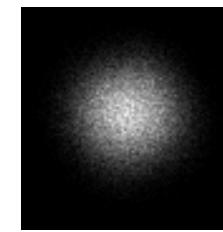
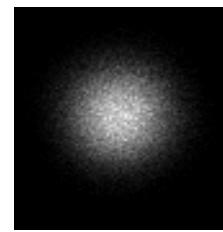
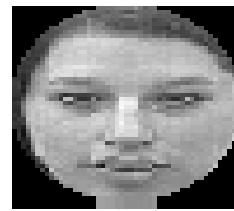
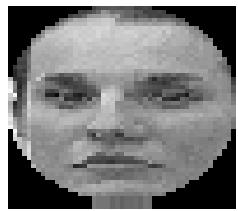
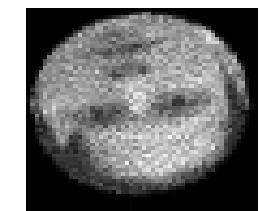
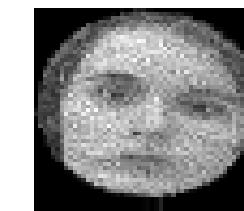
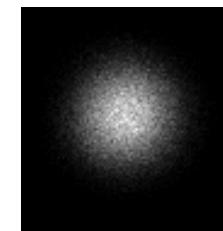
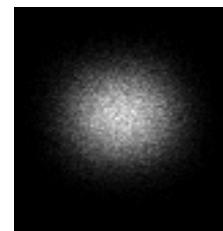
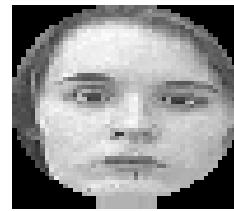
Estimated sources

Performance matrix



Numerical Simulations

Facial images (i.i.d.)

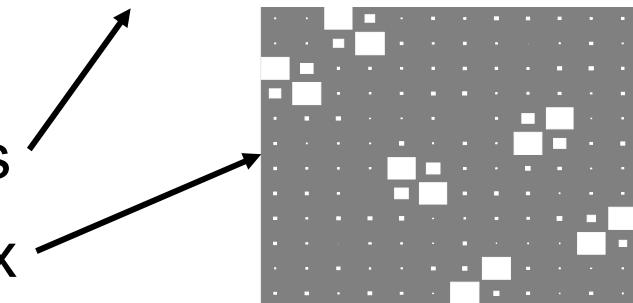


Sources

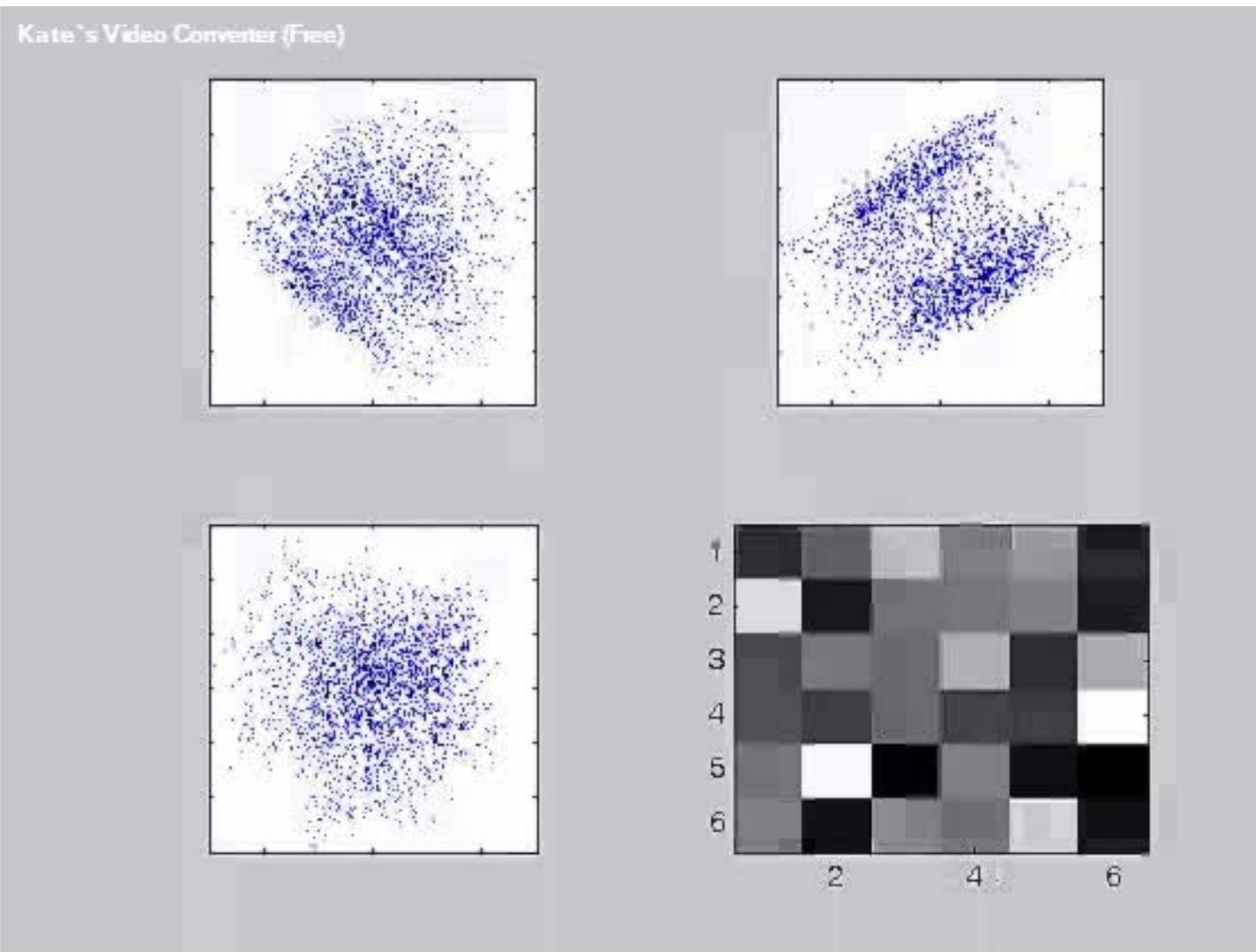
Observation

Estimated sources

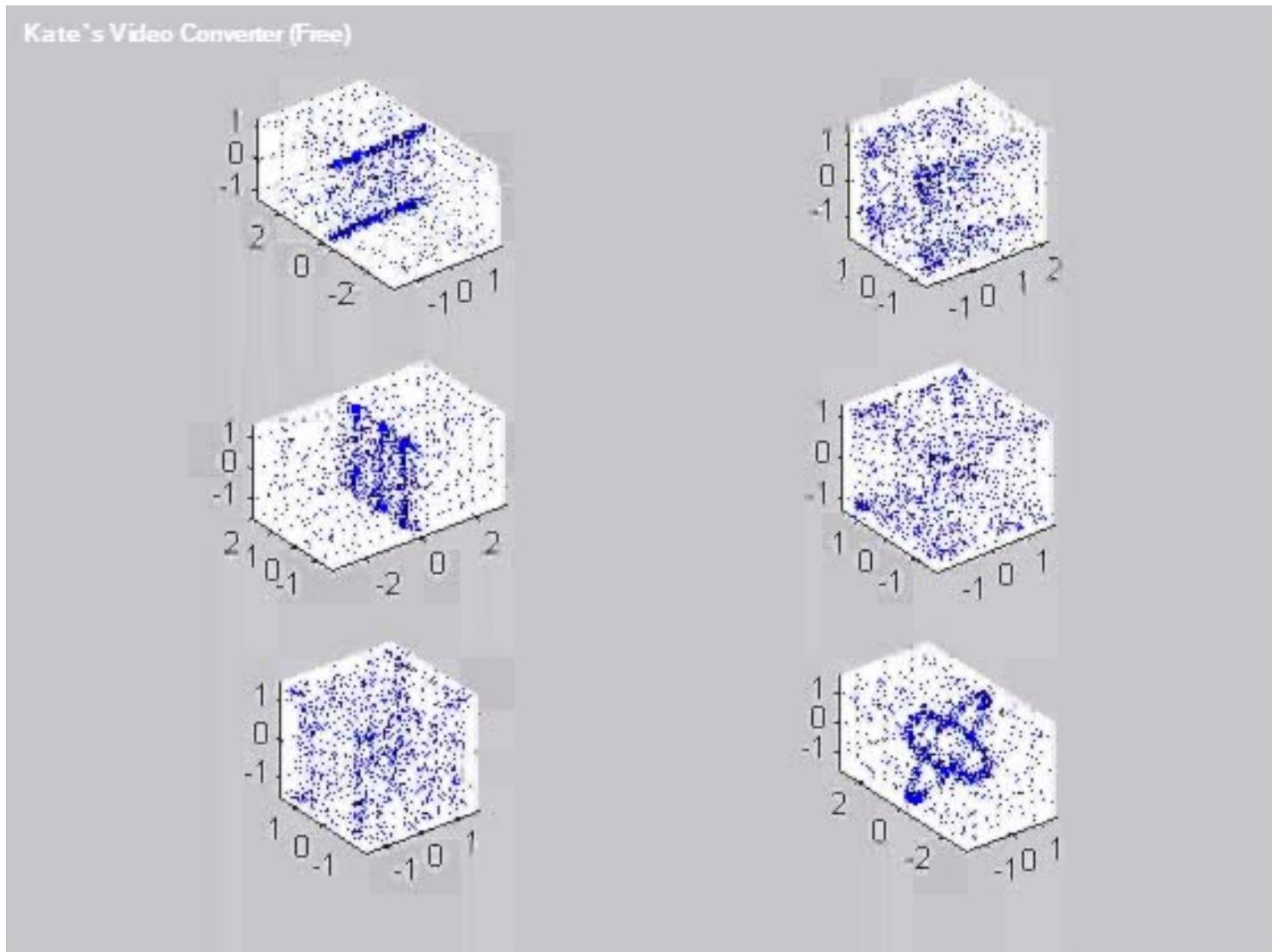
Performance matrix



ISA 2D



ISA 3D after ICA preprocessing



Thanks for the Attention! ☺

