Independent component analysis

PottsICA
Extraction of fetal ECGs

Papers

ScienceDirect - Neural Networks : Independent component analysis based on marginal density estimation using weighted Parzen Windows (2008)

ScienceDirect - Neurocomputing : Blind separation of fetal electrocardiograms by annealed expectation maximization (2008)

IEEE Trans on Neural Networks: Independent component analysis using Potts models (2001)

ICA for Mixed Images

- Load and display mixed faces
- Apply JadeICA for independent component analysis of mixed faces
- Display attained independent components

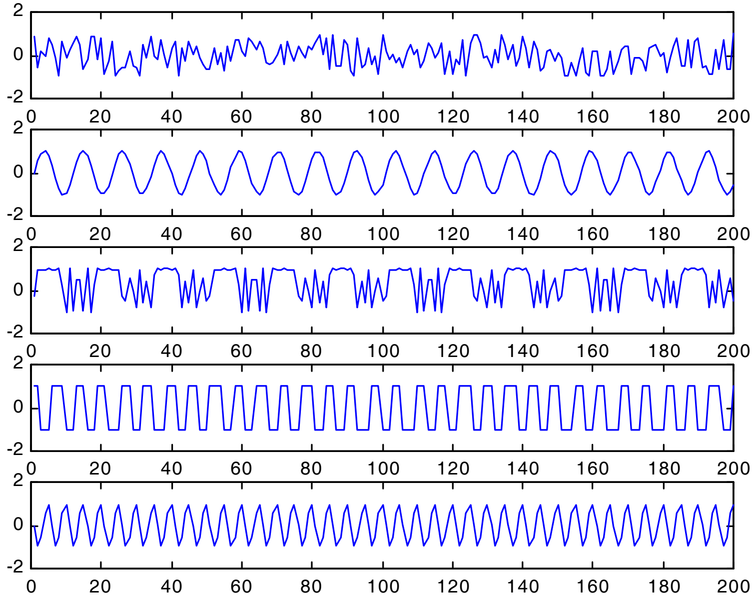
ICA for Mixed Images

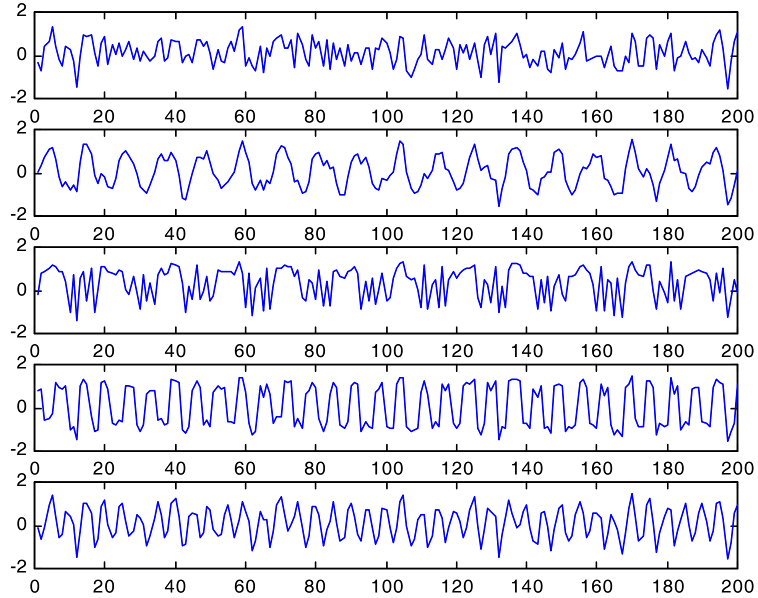
- Load and display mixed faces
- Apply FastICA for independent component analysis of mixed faces
- Display attained independent components

Outlines

1. Blind source separation
2. Independent component analysis
3. Previous works, JadeICA, fastICA
4. ICA using Potts models, PottsICA
5. A comparison on the three methods
6. Conclusions and future works

Blind Source Separation (BSS)



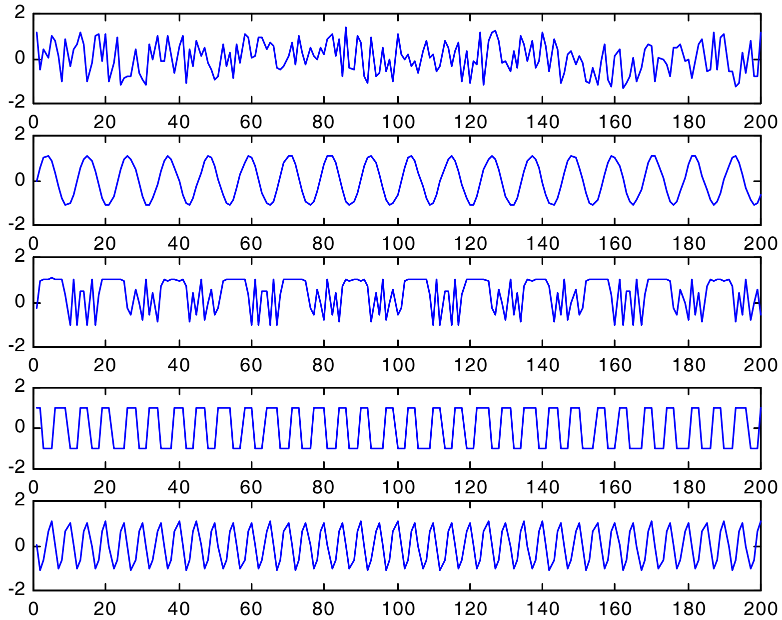


Sources → Unknown Mixing Structure → Observations

PottsICA
BSS by PottsICA

IEEE Trans. On Neural Networks(2001)

- Observations ⟶ **PottsICA**

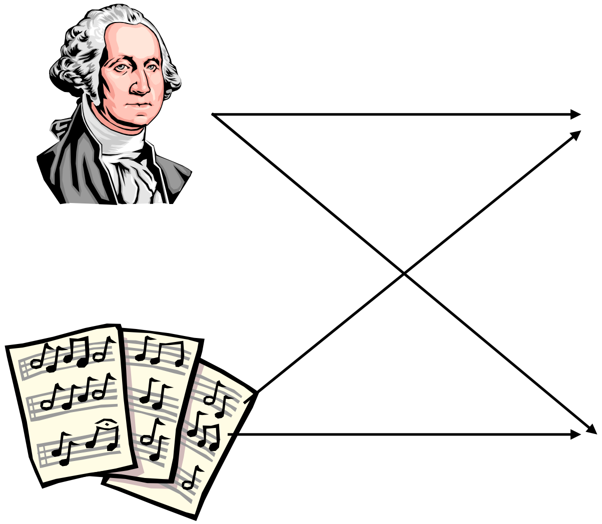


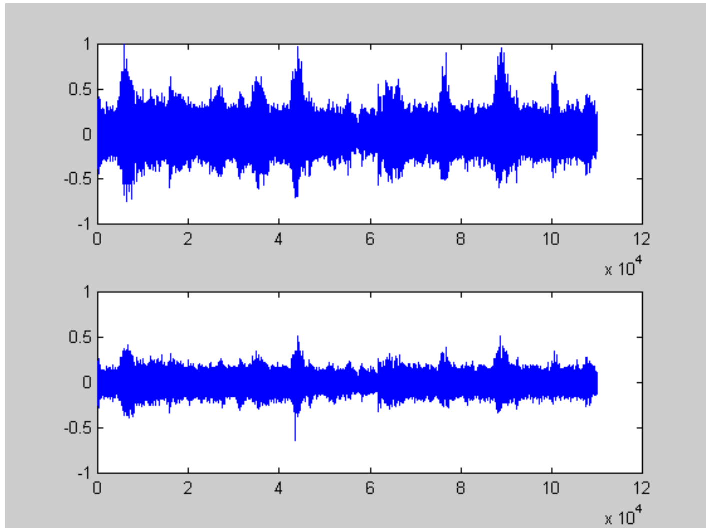
Recovered

sources

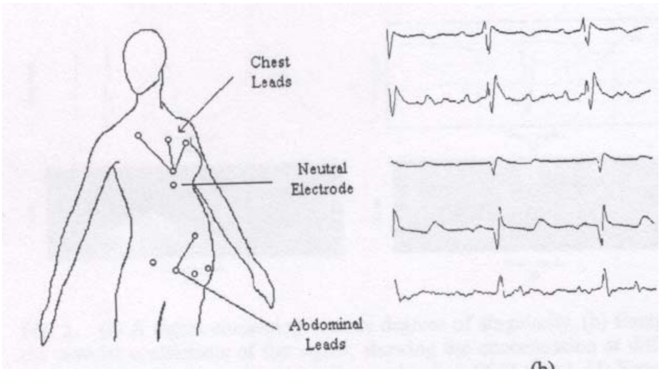
Voice-music separation:

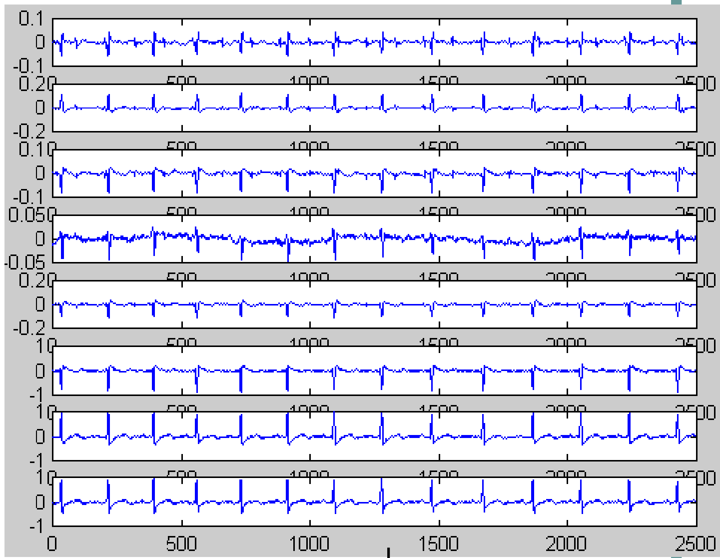
⌘BSS for voice-music separation



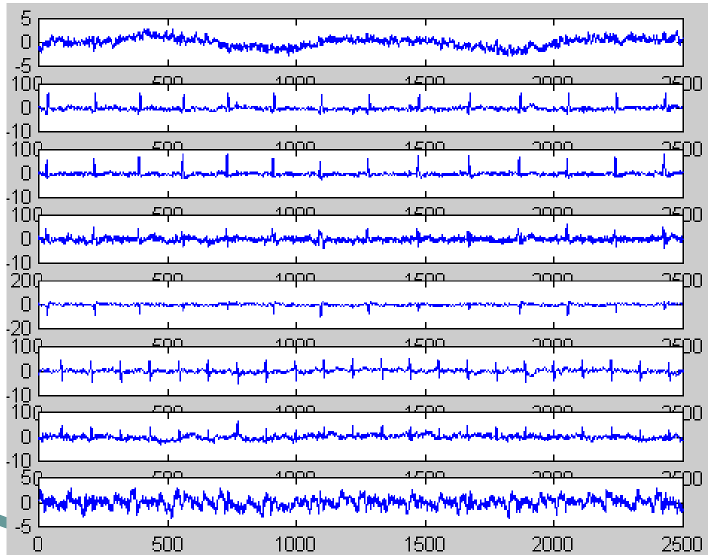


Blind Source separation : Fetal ECGs





ICA



Previous works

- FastICA:Helsinki University of Technology
- JadeICA:by JF Cardoso
- InformaxICA: Salk Institute
- The others…

Linear Mixtures

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$$

Unknown mixing structure: $\boldsymbol{A}$ is an $N \times M$ scalar matrix

Unkown statistically independent sources: S= $[s_1, \ldots, s_M]'$

Observations: $\boldsymbol{x} = [x_1, \ldots, x_N]'$

Goal of ICA

Recover independent sources by a de-mixing process

$$\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{A}\mathbf{s}$$

such that

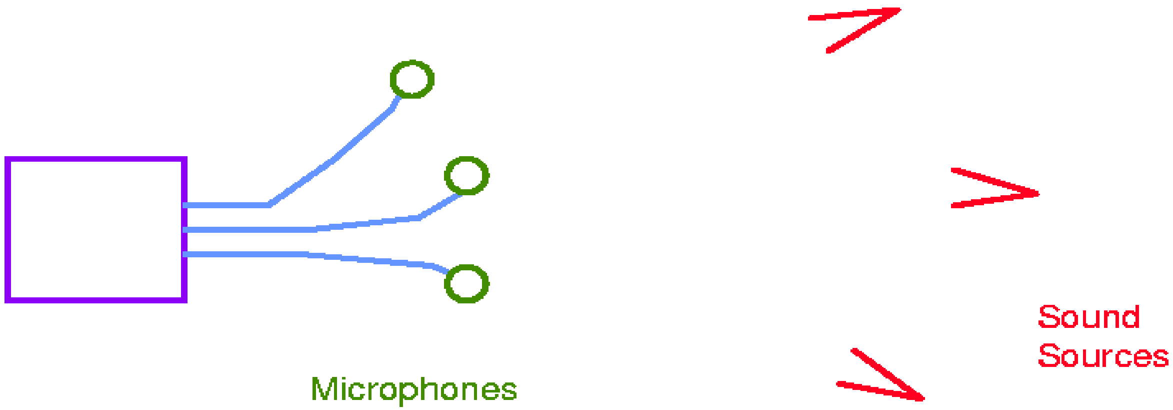
$$p(\mathbf{y}) = \prod_{i=1}^{M} p_i(y_i)$$

$p_i(y_i)$ denotes the marginal distribution

The joint distribution p is as close as possible to the product of marginal distributions

Cocktail Party problem

Information Theory for Signal Separation



- The Cocktail Party Problem
- Many Speakers -- the signals a hopelessly mixed

The Unmixing Problem

$$Y = U\,X = U(AS)$$

- We would like to undo the mixing...
- Signals were independent -- Mut Info was zero.
- Search for U which minimizes Mut Info.

$$\underset{U}{\text{minimize}} \quad MI(y_i, y_j) \ \forall i, j$$
$$MI\left([UX]_i, [UX]_j\right)$$

De-mixing

$$\mathbf{W} = \mathbf{A}^{-1};$$

or

$$\mathbf{W} = \mathbf{\Lambda P A}^{-1}$$

where $\mathbf{P}$ is a permutation matrix and $\mathbf{\Lambda}$ is a nonsingular diagonal matrix for arbitrary scaling.

Kullback-Leibler Divergence

$$D(\mathbf{y}) = \int p(\mathbf{y}) \log \frac{p(\mathbf{y})}{\prod_{i=1}^{N} p_i(y_i)} d\mathbf{y}.$$

$$D(\mathbf{y}) = -H(\mathbf{y}) + \sum_{i=1}^{N} H_i(y_i)$$

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

$$H_i(y_i) = -\int p_i(y_i) \log p_i(y_i) dy_i$$

Joint entropy

Since $\mathbf{y} = \mathbf{W}\mathbf{x}$,

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

Then

$$D(\mathbf{y}) = -H(\mathbf{x}) - \log|\det(\mathbf{W})| + \sum_{i=1}^{N} H_i(y_i).$$

Normalized K-bin histograms

- Let $\{y[t]\}_{t=1}^{T}$ denote a sample from a random variable y
- Let $\{h_k\}_{k=1}^{K}$ denote a set of knots for partition of the range of y into non-overlapping intervals

Non-overlapping intervals

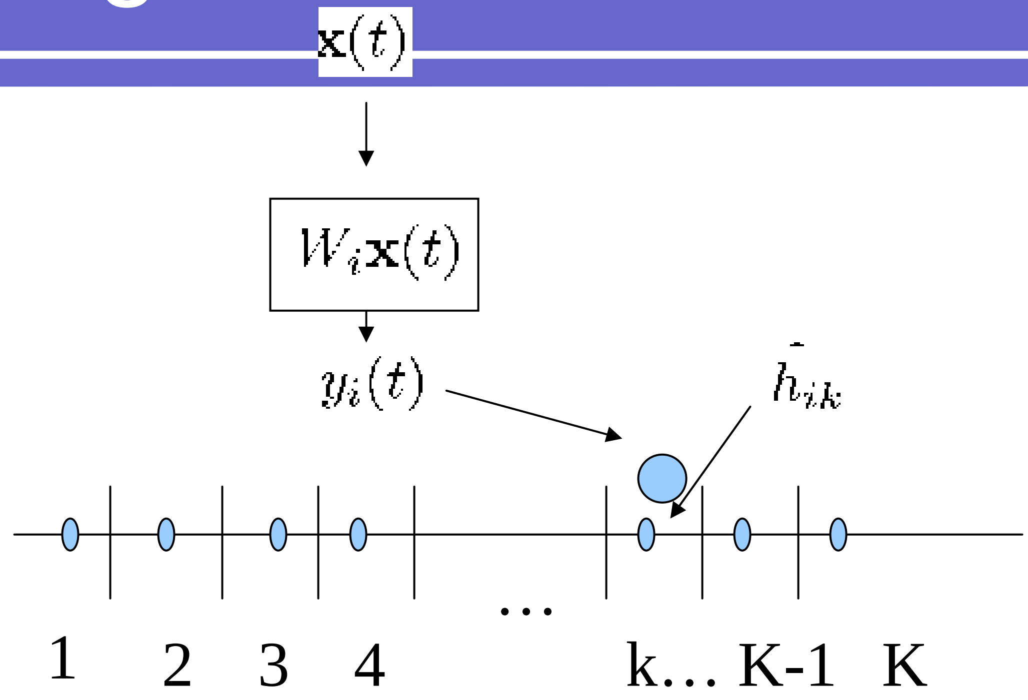
$$B_k = \left\{ y \mid k = \arg\min_{l} |y - h_l|, y \in R \right\}$$

Bins

Potts encoding

Output component i

Bin or state number



Membership vector

$\xi[t] = (\xi_1[t], \ldots, \xi_K[t])'$ = [0 0 0 0 … 1…0 0]

the kth bit

Potts variable

$$\xi[t] = (\xi_1[t], \ldots, \xi_K[t])',$$

$$\sum_{k=1}^{K} \xi_k[t] = 1,$$

$$\xi_k[t] \in \{0, 1\} \quad \text{for all } k.$$

Normalized Histograms

$$\Pr(y \text{ in } B_k) = \frac{1}{T}\sum_{t=1}^{T} \xi_k[t],$$

Math Programming

Potts encoding of membership vectors by $\xi[t]$ is equivalently to minimize

$$E(\{\boldsymbol{\xi}[t]\}_t) = \frac{1}{2}\sum_{t=1}^{T}\sum_{k=1}^{K}\xi_k[t]|y[t] - h_k|^2.$$

$$\xi[t] = (\xi_1[t], \ldots, \xi_K[t])',$$

Subject to

$$\sum_{k=1}^{K}\xi_k[t] = 1,$$

$$\xi_k[t] \in \{0, 1\} \quad \text{for all } k.$$

Potts encoding of Multiple components

For encoding marginal entropies of multiple components, we extend membership vectors with a subindex

$$\xi_i[t] = (\xi_{i1}[t], \ldots, \xi_{iK}[t])'$$

Math programming of Multiple components

Minimize

$$E(\xi, \mathbf{W}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} \xi_{ik}[t]|\mathbf{w}_i\mathbf{x}[t] - h_k|^2$$

Subject to

$$\sum_{k=1}^{K} \xi_{ik}[t] = 1 \quad \text{for all } i, t,$$

$$\xi_{ik}[t] \in \{0, 1\} \quad \text{for all } i, k, t.$$

Joint entropy

$$S(\mathbf{y}) = S(\mathbf{x}) + \log|\det(\mathbf{W})|,$$

where the first term is negligible.

Marginal entropies

$$S(y_i) \approx -\sum_{k=1}^{K} \Pr(y_i \in B_k) \ln \Pr(y_i \in B_k)$$

$$= -\frac{1}{T}\sum_{k=1}^{K}\sum_{t=1}^{T} \xi_{ik}[t] \ln\left(\sum_{t=1}^{T} \xi_{ik}[t]\right) + \ln T,$$

for approximating the marginal entropy of y_i.

Discrete marginal entropy

$$H_i(y_i) = -\int p_i(y_i) \log p_i(y_i) dy_i$$

$$\approx -\sum_{k=1}^{K} p_{ik} \log p_{ik}.$$

Math programming for minimization of KL divergence

$$L(\xi, \mathbf{W}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} \xi_{ik}[t]|\mathbf{w}_i\mathbf{x}[t] - h_k|^2$$

$$- c\log|\det(\mathbf{W})|$$

$$- \frac{c}{T}\sum_{k=1}^{K}\sum_{t=1}^{T} \xi_{ik}[t]\ln\left(\sum_{t=1}^{T}\xi_{ik}[t]\right)$$

$$\sum_{k=1}^{K} \xi_{ik}[t] = 1 \quad \text{for all } i, t,$$

$$\xi_{ik}[t] \in \{0, 1\} \quad \text{for all } i, k, t.$$

Hopfiel-like energy function

Hopfield-like energy function:

$$L(\xi, \mathbf{W}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \xi_{ik}[t] |\mathbf{w}_i \mathbf{x}[t] - h_k|^2$$

$$- c \log |\det(\mathbf{W})|$$

$$- \frac{c}{T} \sum_{k=1}^{K} \sum_{t=1}^{T} \xi_{ik}[t] \ln \left(\sum_{t=1}^{T} \xi_{ik}[t] \right)$$

E-step

- Replace $L(\mathbf{W}, \xi)$ by $L(\mathbf{W}, \mathbf{v})$
- $\mathbf{v}$ denotes the mean configuration of ξ

$$u_{ik}[t] = \frac{-\partial L(\mathbf{v}, \mathbf{W})}{\partial v_{ik}[t]},$$

$$v_{ik}[t] = \frac{\exp(\beta u_{ik}[t])}{\sum_l \exp(\beta u_{il}[t])},$$

Mean field equations

$$u_{itk} = -\frac{1}{2}\left\|\mathbf{W}_i\mathbf{x}(t) - h_{ik}\right\|^2 + \frac{C_2}{T}\log\left(\frac{1}{T}\sum_{t=1}^{T} v_{itk}\right)$$

$$v_{itk} = \frac{\exp\left(\beta u_{itk}\right)}{\sum_{l}\exp\left(\beta u_{itl}\right)}$$

Boltzmann assumption

Boltzmann distribution

$$\Pr(\boldsymbol{\delta}) \propto \exp(-\beta L(\boldsymbol{\delta}))$$

Use mean field equations to find the mean configuration at each β

$$\lim_{\beta \to \infty} \Pr(\boldsymbol{\delta}^*) = 1$$

$$L(\boldsymbol{\delta}^*) = \min_{\boldsymbol{\delta}} L(\boldsymbol{\delta})$$

β denotes the inverse of a temperature-like parameter

M step

$$\Delta \mathbf{W} = -\eta(\nabla \mathbf{W})\mathbf{W}'\mathbf{W},$$

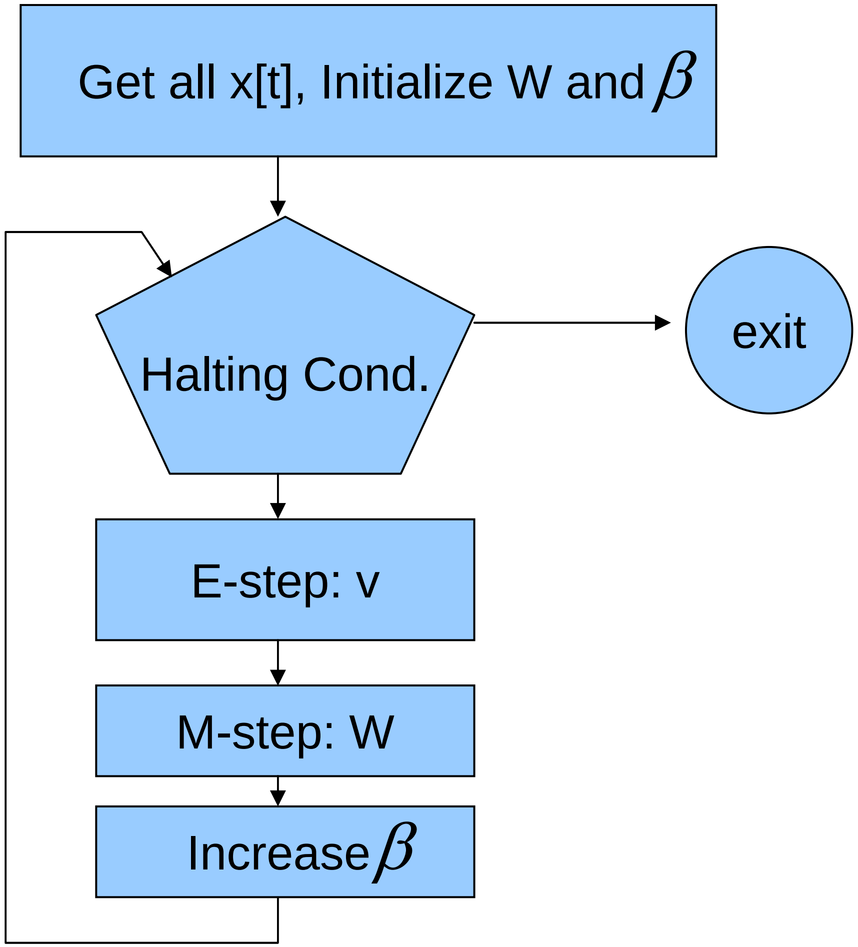
where η denotes the step size

$$\nabla \mathbf{W} = \frac{\mathrm{d}L(\mathbf{v}, \mathbf{W})}{\mathrm{d}\mathbf{W}}$$

Annealed EM

1. Input all $\mathbf{x}[t]$, set $\mathbf{W}$ as an identity matrix, $c = 1$, $\beta = \frac{1}{2.5}$ and $\eta = 0.015$, and initialize $v_{ik}[t]$ near $1/K$ randomly and β as a sufficiently low value.
2. Iteratively update all $u_{ik}[t]$ and $v_{ik}[t]$ using Eqs. (17) and (18) to a stationary point.
3. Iteratively update $\mathbf{W}$ using Eqs. (13) and (19).
4. If the value of $\sum_{i,k,t} v_{ik}[t]^2$ is less than a predetermined threshold, set β to $\beta/0.995$ and go to step 2, otherwise halt.

Flow chart

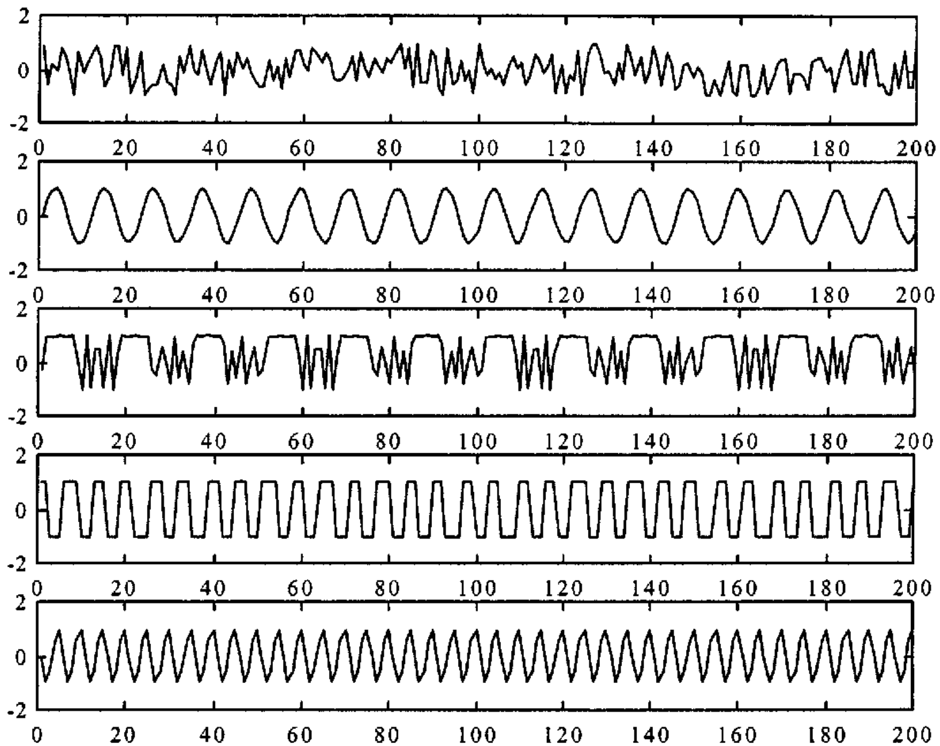


Get all x[t], Initialize W and β
Halting Cond.
exit
E-step: v
M-step: W
Increase β

Numerical simulations

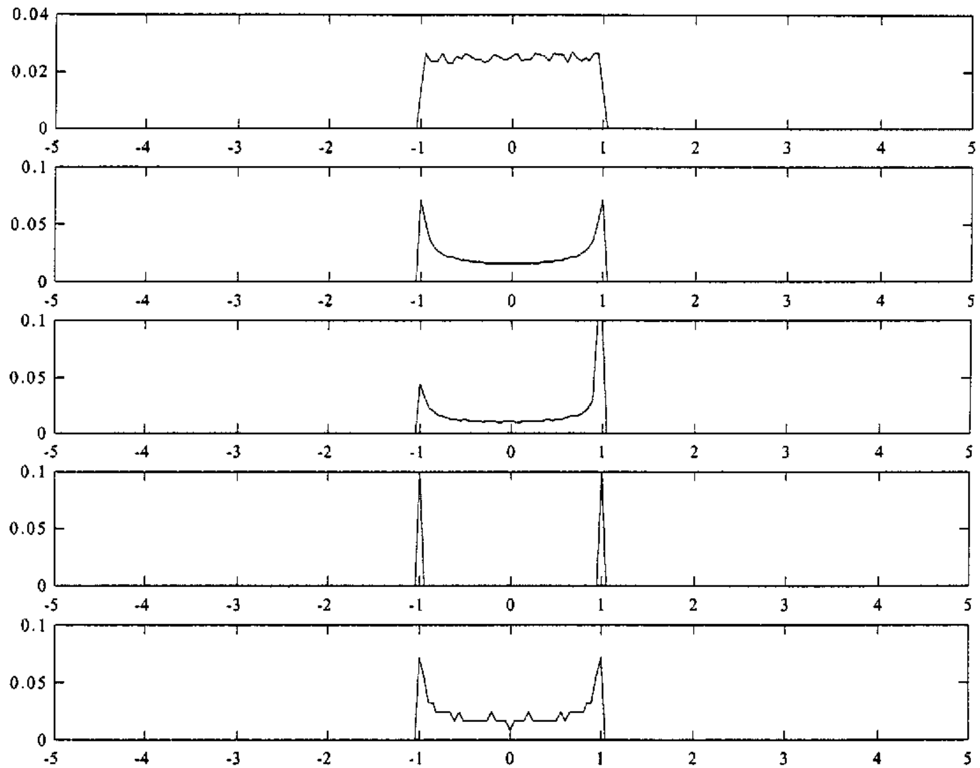
Artificial problem I by Amari *et al.*

Independent sources



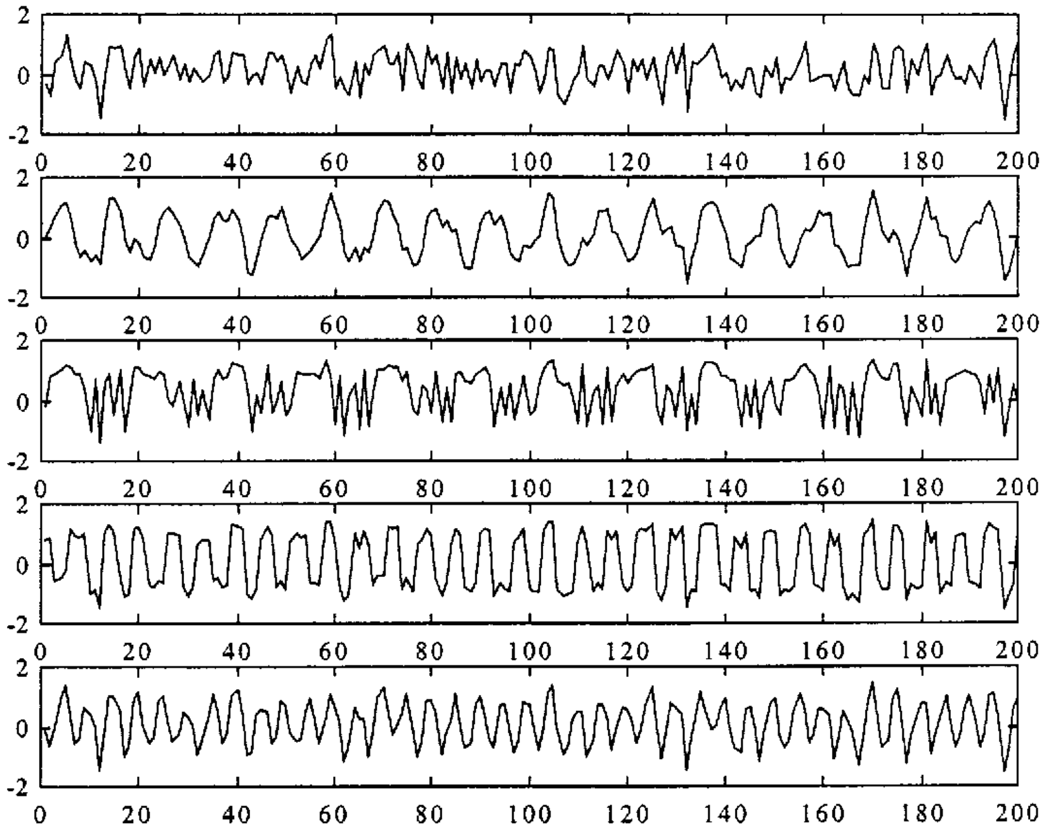
Numerical simulations

The normalized histogram of the five sources



Numerical simulations

The mixed signals of the five sources using
a randomly generated mixing matrix



Performance evaluations by Amari

$$E = \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \frac{|q_{ij}|}{\max_k |q_{ik}|} - 1 \right) + \sum_{j=1}^{N} \left(\sum_{i=1}^{N} \frac{|q_{ij}|}{\max_k |q_{kj}|} - 1 \right) \quad (28)$$

where q_{ij} denotes the joint element of the ith row and the jth column of the product of the mixing matrix and the demixing matrix.

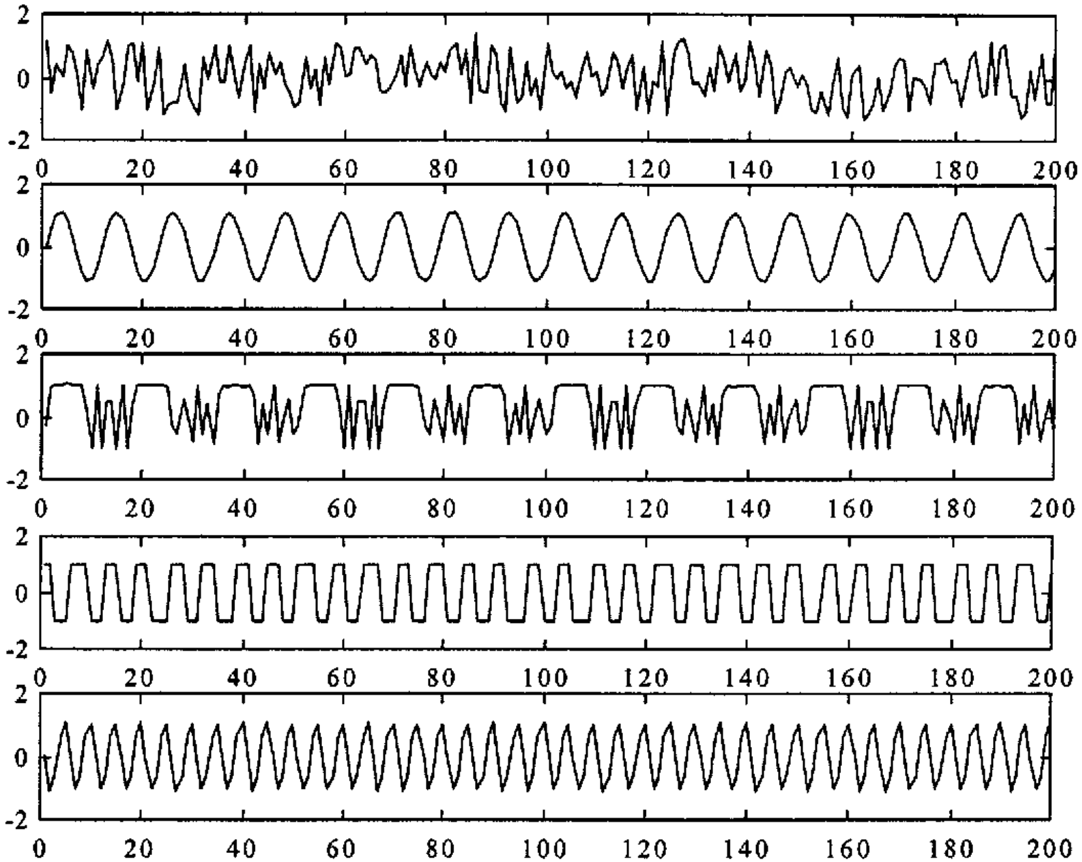
Performance evaluations

THE PERFORMANCE OF THE THREE ALGORITHMS FOR THE TESTS

mean E	PottsICA	JadeICA	FastICA	cpu-time(PottsICA)
example 1(N=5)	0.28	0.60	0.75	314 secs
example 2(N=6)	1.28	3.02	1.97	400 secs
example 3(N=8)	4.40	15.30	11.07	566 secs

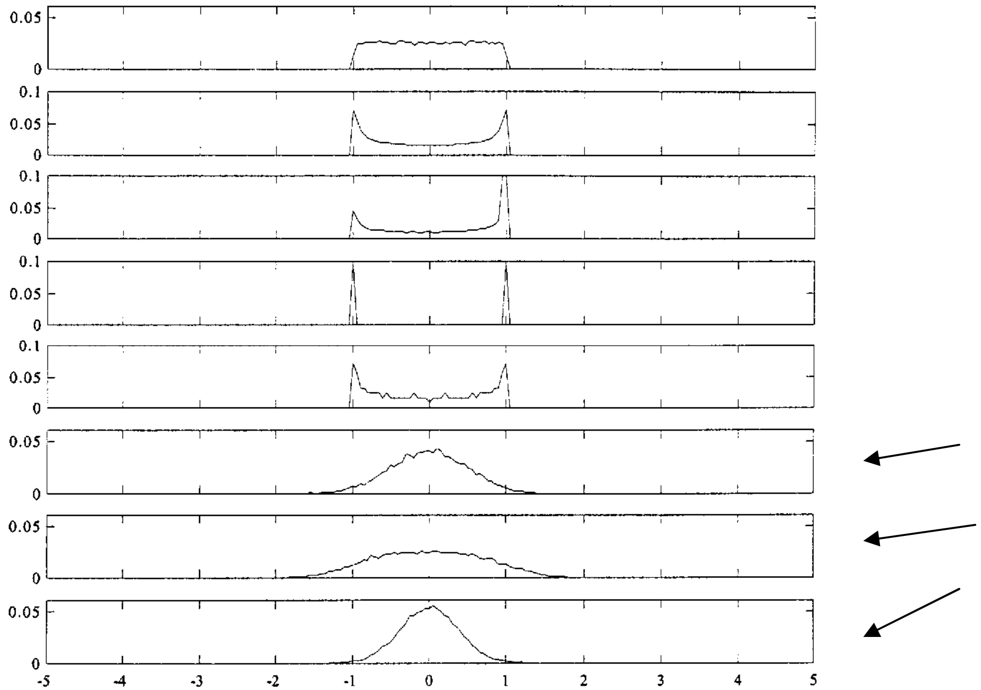
Recovered signals by PottsICA

The recovered signals by the PottsICA from the mixed signals



Numerical simulations

The normalized histograms of the eight sources



Gaussian
Sub-Gaussian
Super-Gaussian

Performance

TABLE II
TEST PERFORMANCE OF THE THREE ALGORITHMS FOR DIFFERENT PROBLEM SIZES WITH ALL SOURCES IN UNIFORM DISTRIBUTIONS

mean E	PottsICA	JadeICA	FastICA	cpu-time (secs) of PottsICA
N=2	0.13	0.17	0.18	153
N=3	0.38	0.48	0.64	224
N=4	0.84	0.91	1.13	317
N=5	1.48	1.66	2.45	386
N=6	2.00	2.73	3.77	460
N=7	3.19	3.64	5.50	556
N=8	4.88	6.54	7.35	619
N=9	6.41	13.20	11.26	716
N=10	7.98	33.85	16.91	782
N=11	9.64	77.96	24.04	861
N=12	13.37	115.26	38.82	950
N=13	15.74	141.02	40.64	1040
N=14	18.00	166.29	55.52	1123
N=15	21.82	183.66	76.51	1199
N=16	25.81	208.93	*	1322
N=17	29.24	233.51	*	1363
N=18	34.67	260.71	*	1437
N=19	39.53	292.50	*	1525
N=20	44.63	323.31	*	1614

Performance

TABLE III
TEST PERFORMANCE OF THE THREE ALGORITHMS FOR DIFFERENT PROBLEM SIZES WITH ONE GAUSSIAN SOURCE AND $N-1$ UNIFORM SOURCES

mean E	PottsICA	JadeICA	FastICA	cpu-time(secs) of PottsICA
N=2	0.31	0.27	0.31	145
N=3	0.77	0.73	0.92	216
N=4	1.35	1.37	1.38	298
N=5	2.21	2.69	2.66	381
N=6	3.21	5.16	4.26	439
N=7	3.95	5.09	5.68	531
N=8	5.48	17.96	8.87	605
N=9	7.28	19.75	15.20	688
N=10	9.02	48.03	20.67	763
N=11	11.18	74.30	22.30	853
N=12	14.27	110.57	36.10	949
N=13	17.49	133.84	55.21	1024
N=14	20.72	155.62	72.72	1107
N=15	23.86	173.66	76.04	1188
N=16	27.38	201.70	*	1309
N=17	32.80	235.56	*	1364
N=18	36.86	255.16	*	1422
N=19	44.28	285.16	*	1534
N=20	47.52	313.09	*	1625

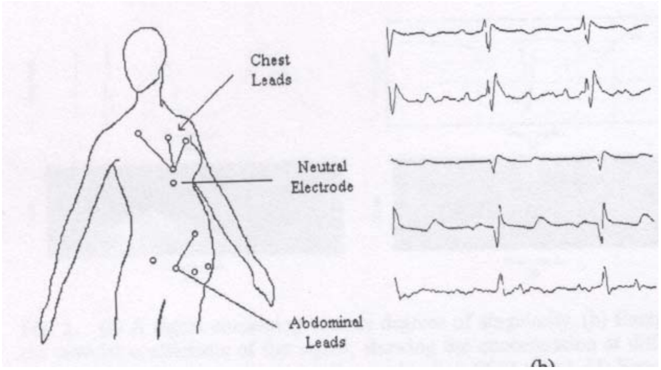
Conclusions

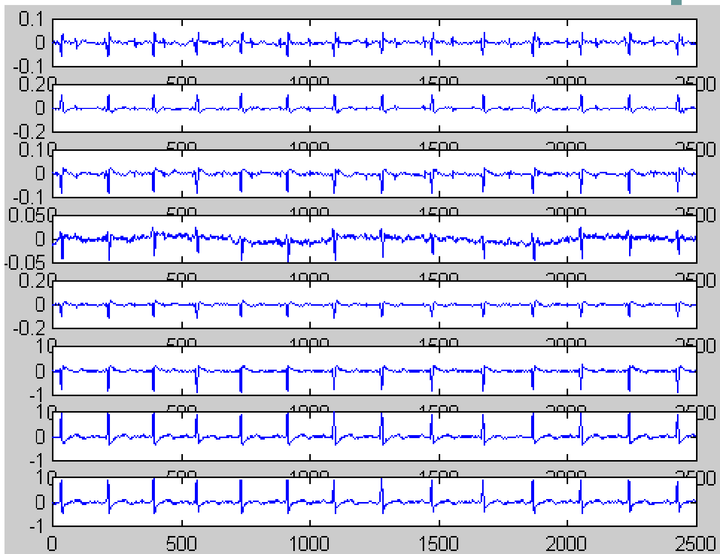
- Potts encoding is successfully applied to develop novel approaches for ICA
- PottsICA directly deals with minimization of KL divergence for ICA
- Marginal pdfs are well approximated by normalized histograms

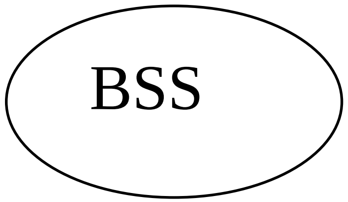
Conclusions

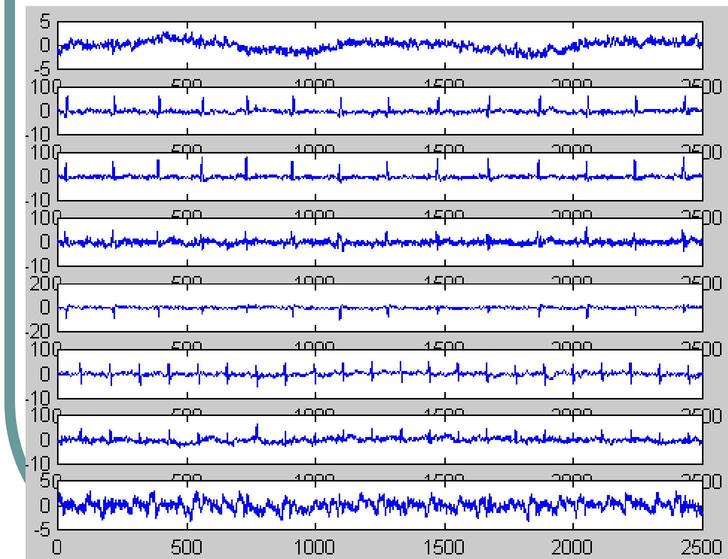
- PottsICA behaves better than JadeICA and FastICA
- PottsICA is potential for real world applications

Blind source separation – fetal ECG









Dataset

```
XX =load('data\\foetal_ecg.dat','ASCII');
```

Exercise

Plot MECG

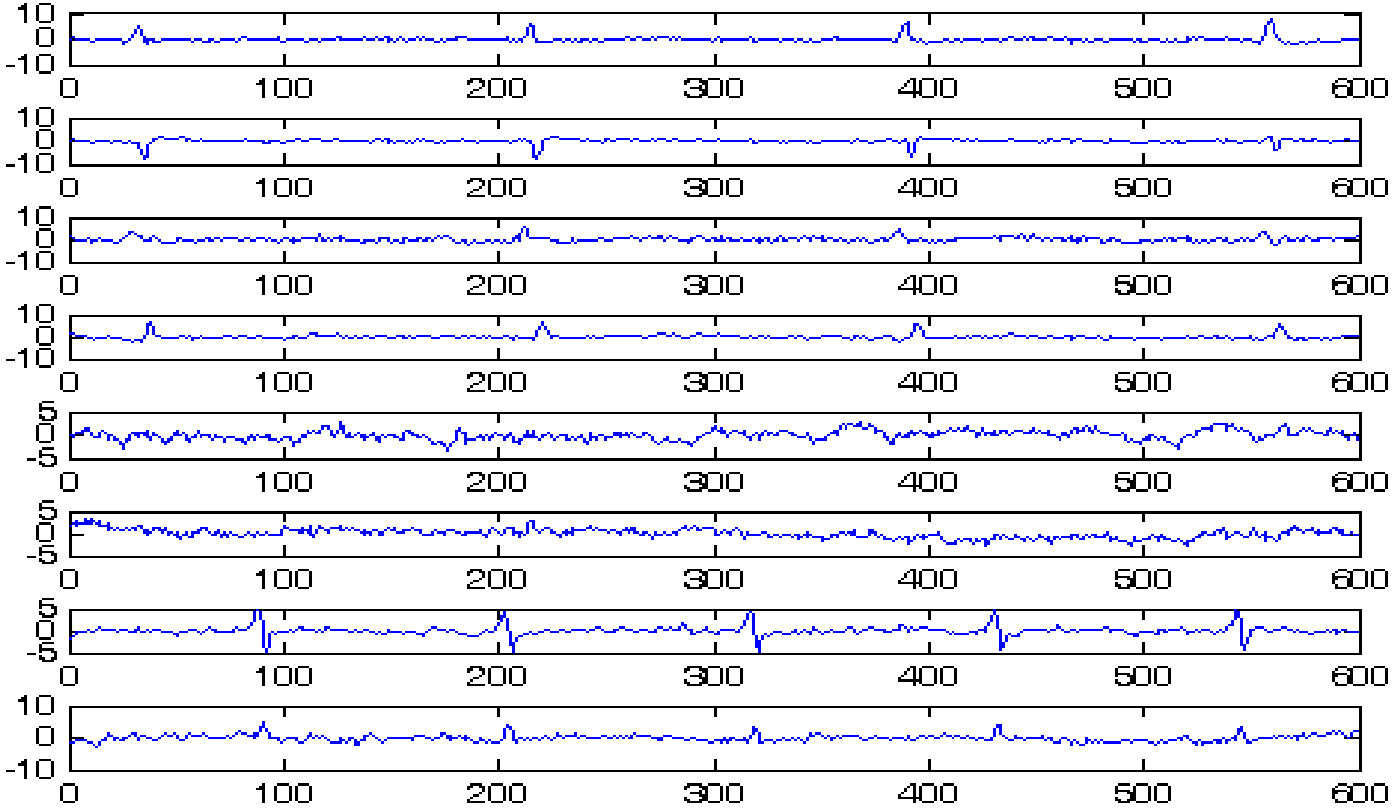
fECG extraction- JadeICA

- Process MECG by JadeICA
- Display recovered independent components

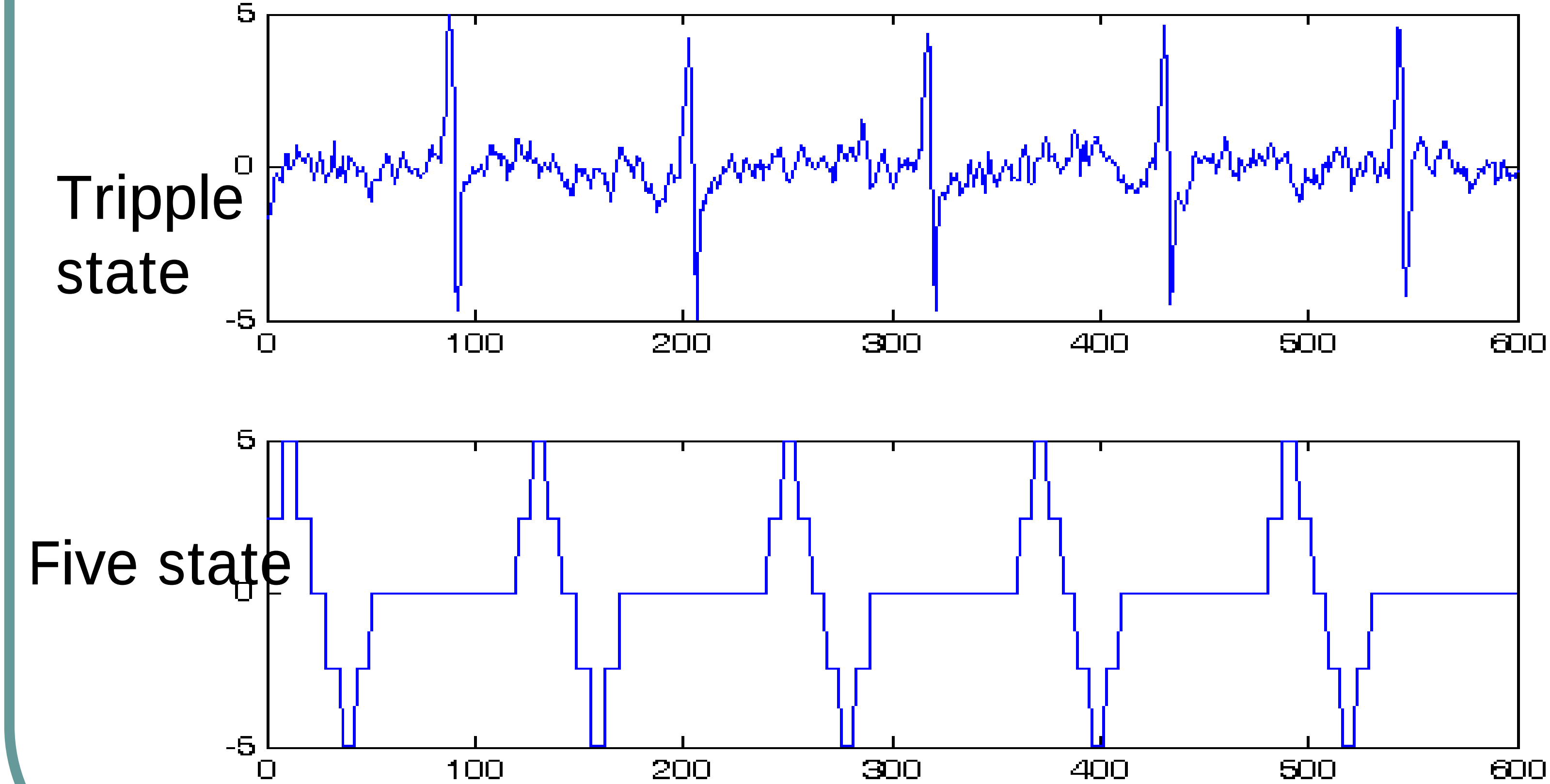
fECG extraction- FastICA

- Process MECG by FastICA
- Display recovered independent components

ICs extracted by JadeICA



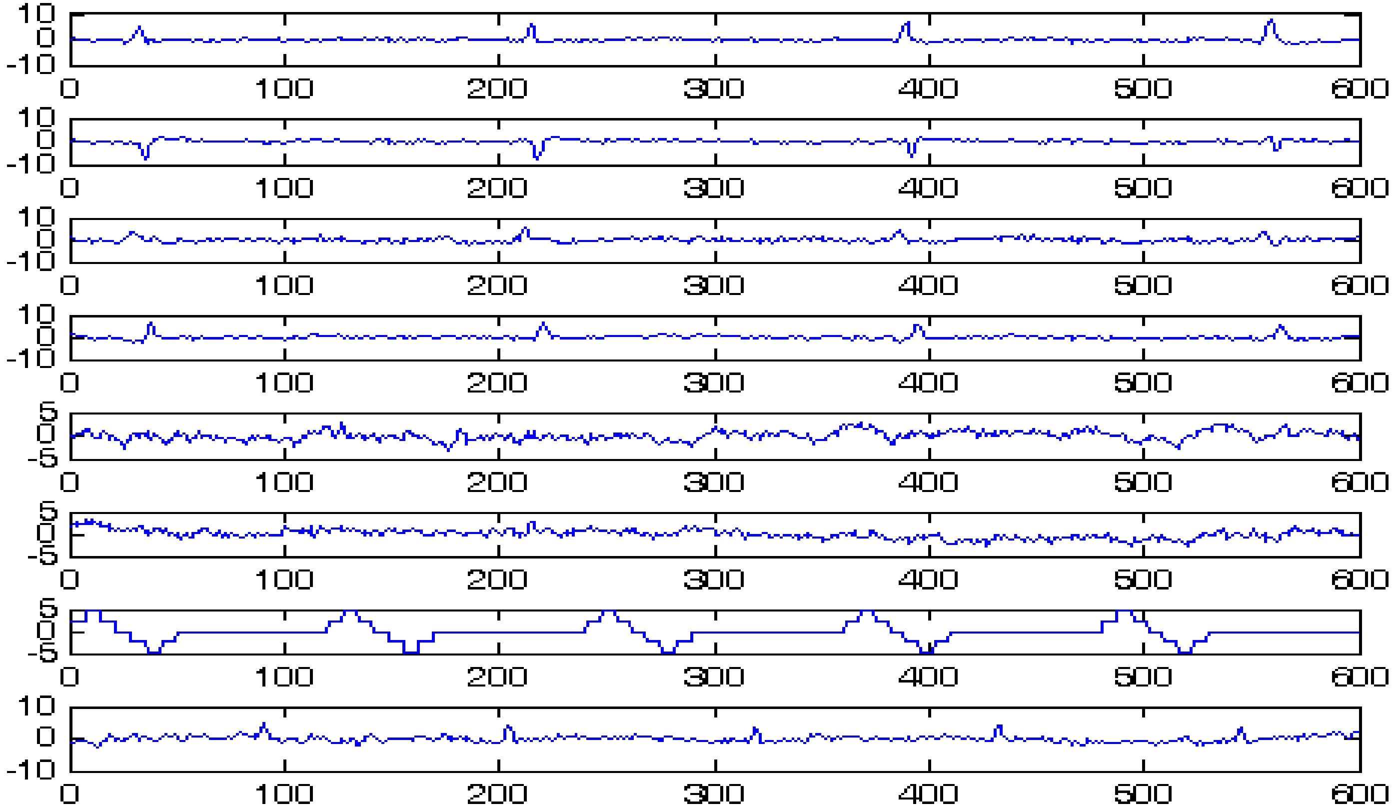
Weak target sources



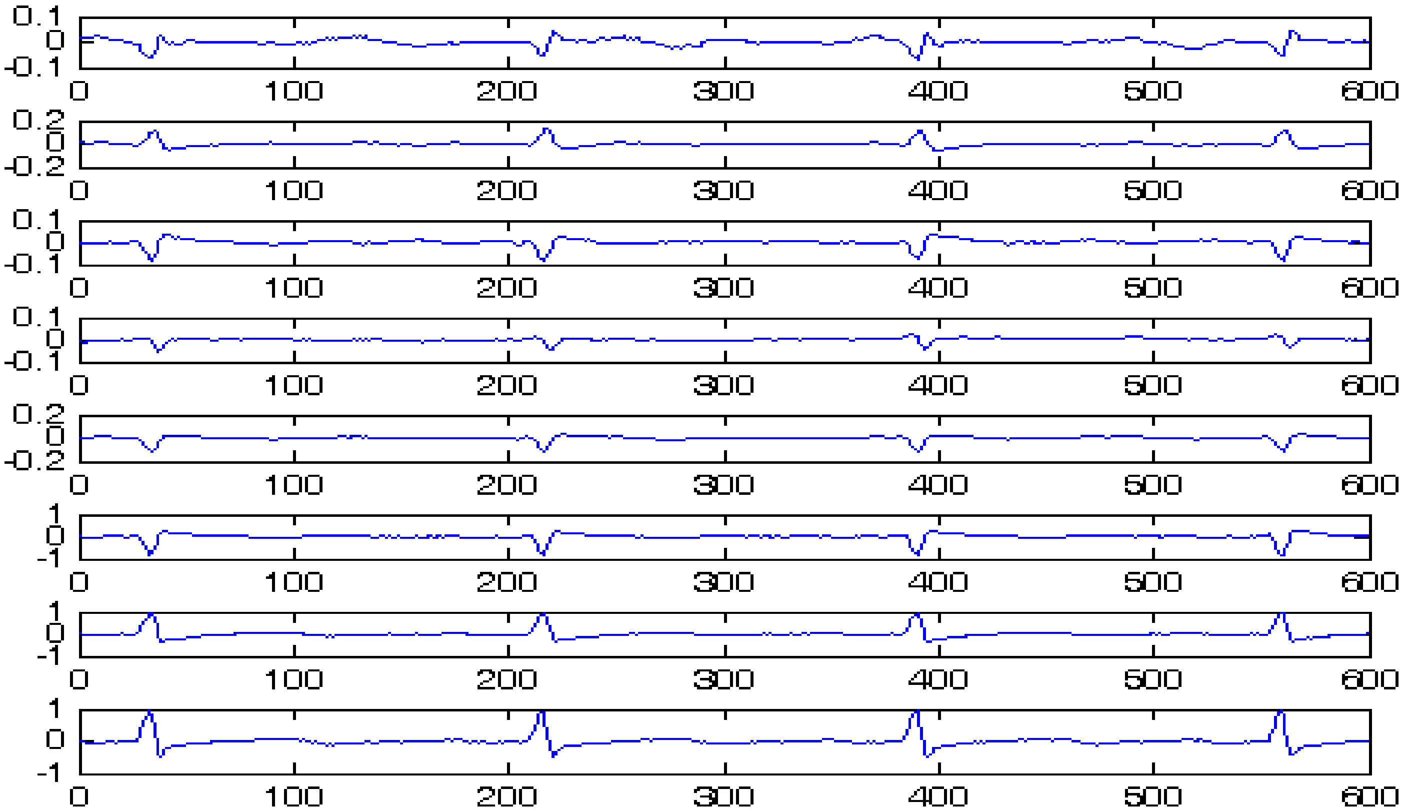
Weak source

- Replace the 7th component of JadeICA with a five-state source to attain eight sources
- Multiply the mixing matrix of JadeICA to eight sources to form artificially created MECGs

Artificially created eight sources

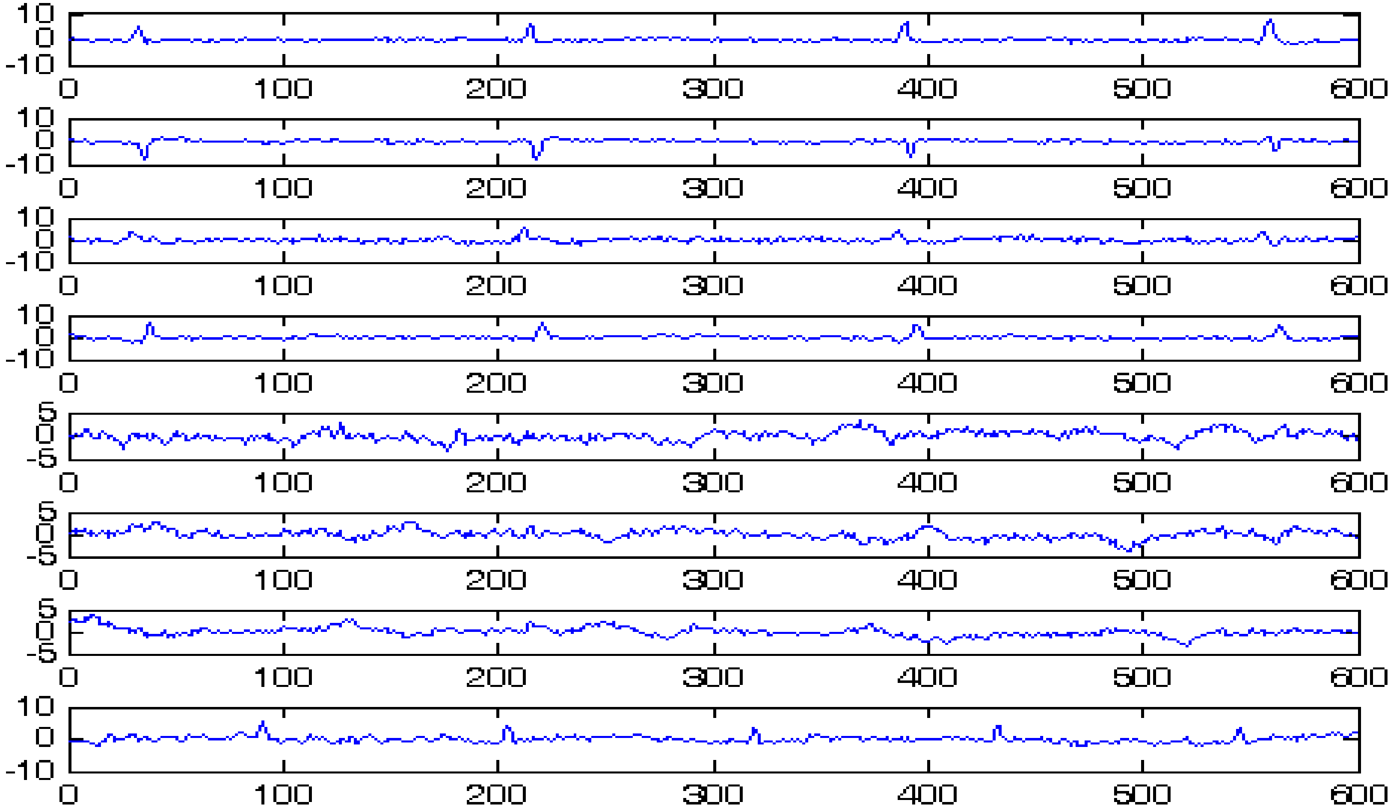


Artificially created MECG

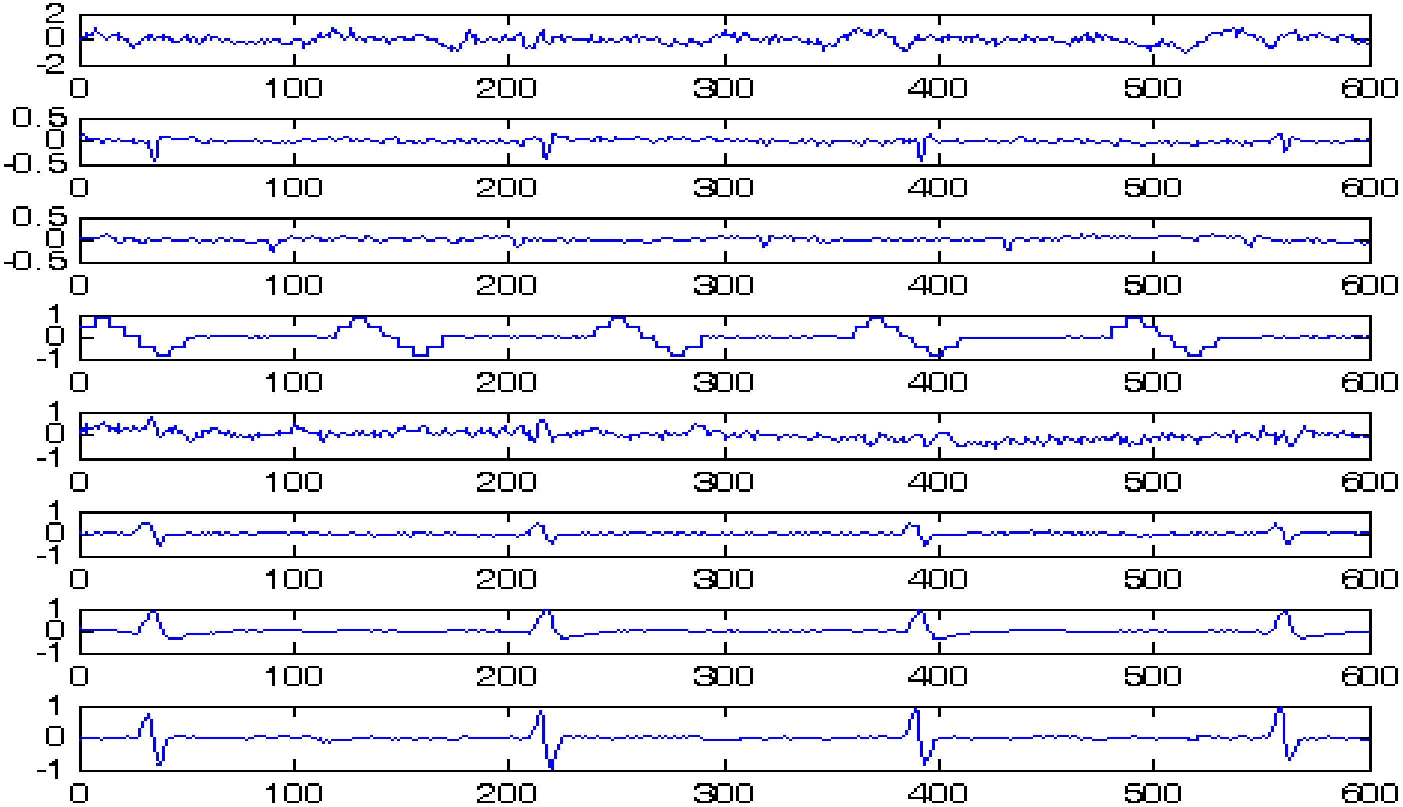


Ics extracted by JadeICA

Apply JadeICA to process artificially Created MECGs



Ics extracted by PottsICA



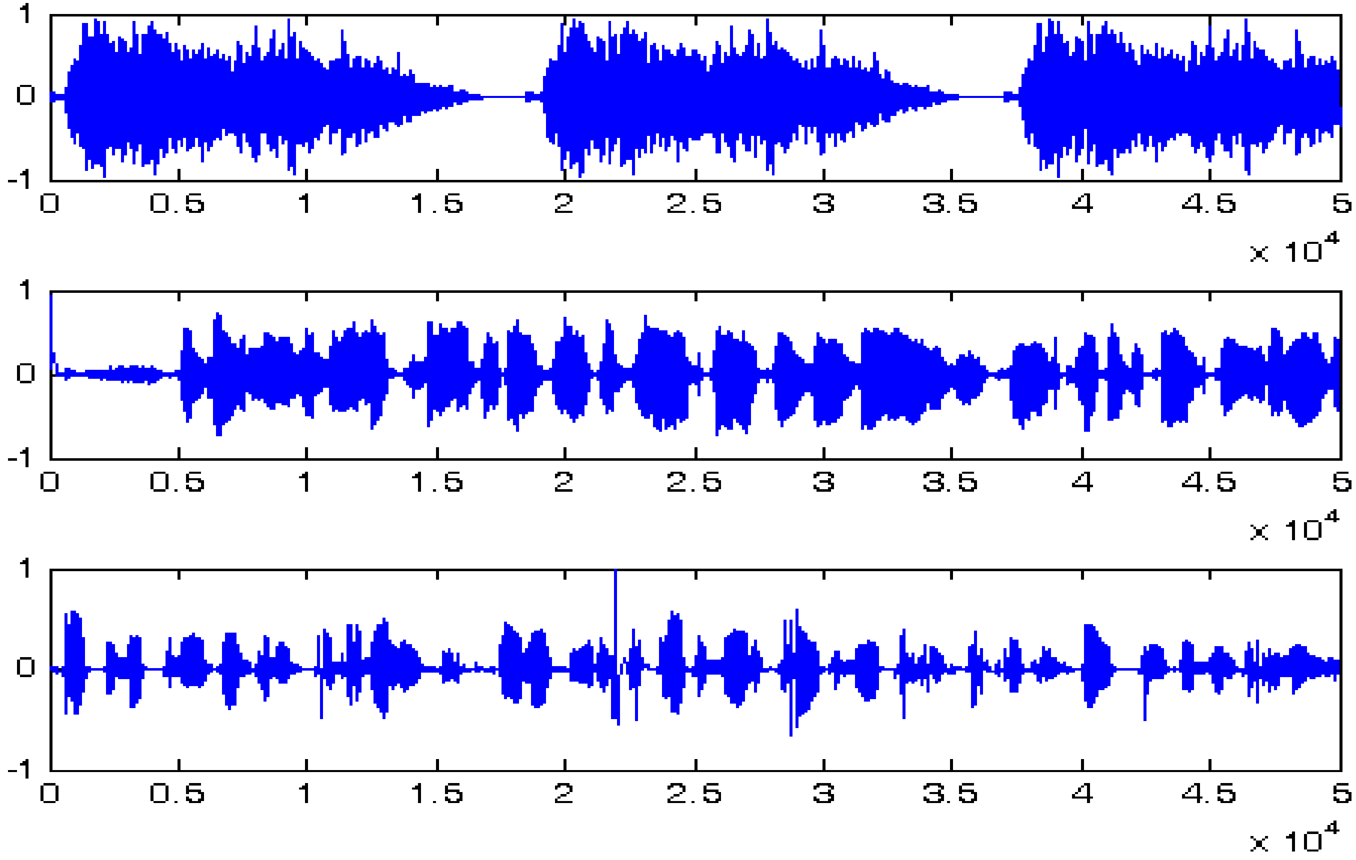
Exercise

- Process MECG by JadeICA
- Form an artificially created MECGs
  - Replace the 7th component of JadeICA with a five-state source to form a set of eight sources
  - Multiply the mixing matrix estimated by JadeICA to the eight sources
- Apply JadeICA to process artificially created MECGs

wavread

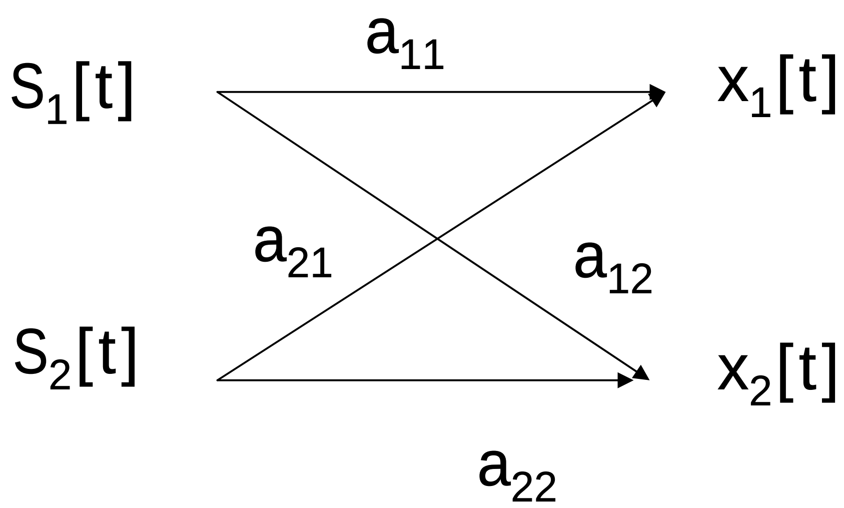
```
[s1,fs]=wavread('source1.wav');
plot(1:1:length(s1),s1);
wavplay(s1,fs);
```

Sounds



1
0
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
x 10^4
1
0
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
x 10^4
1
0
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
x 10^4

Linear mixtures



$$\begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} s_1[t] \\ s_2[t] \end{bmatrix}$$

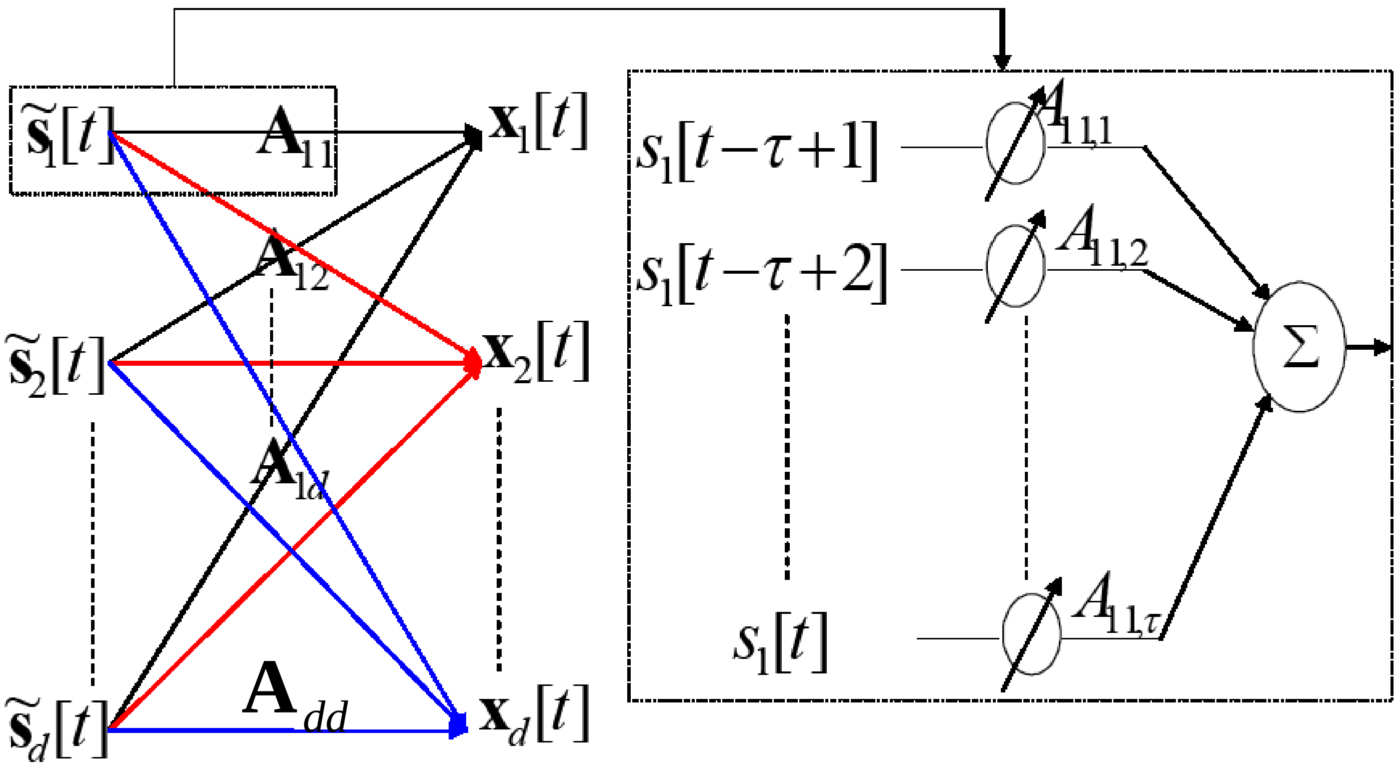
Linear mixtures of sound recordings

- Input sound recordings
- Create a random matrix
- Form linear mixtures of sound recordings
- Play and plot linear mixtures of sound recordings

ICA of linear mixtures of sound recordings

- Apply JadeICA to process linear mixtures of sound recordings

Convolutive mixing



Convolutive mixtures

$$\begin{bmatrix} x_1[t] \\ x_2[t] \end{bmatrix} = \begin{bmatrix} a_{11}(0) & a_{12}(0) \\ a_{21}(0) & a_{22}(0) \end{bmatrix} \begin{bmatrix} s_1[t] \\ s_2[t] \end{bmatrix}$$

$$+ \begin{bmatrix} a_{11}(1) & a_{12}(1) \\ a_{21}(1) & a_{22}(1) \end{bmatrix} \begin{bmatrix} s_1[t-1] \\ s_2[t-1] \end{bmatrix} + \cdots$$

$$+ \begin{bmatrix} a_{11}(\tau) & a_{12}(\tau) \\ a_{21}(\tau) & a_{22}(\tau) \end{bmatrix} \begin{bmatrix} s_1[t-\tau] \\ s_2[t-\tau] \end{bmatrix}$$

Convolutive mixtures of sound recordings

- Input sound recordings
- Create random matrices
- Form convolutive mixtures of sound recording
- Play and plot mixed sounds

ICA of convolutive mixtures of sound recordings

- Apply JadeICA to process convolutive mixtures of sound recordings

ICA of mixed-sound recordings

- Read convolutive mixtures of sound recordings
- Apply JadeICA for blind source separation of convolutive mixtures of sound recordings
- Plot and play independent components