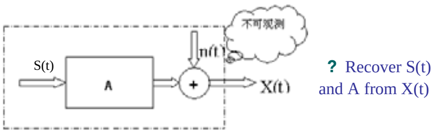
Blind Source Separation using Infomax algorithm

Yandong Li & Wenkai Lu
Intelligent Signal Processing Lab,
Institute of Information Technology
Tsinghua University
liyd02@mails.tsinghua.edu.cn
2004年12月14日星期二

Table of Contents

Part I. An overview of BSS

- **Blind Source Separation**: recover source signals from the mixtures without a priori knowledge about S(t) and A.



? Recover S(t) and A from X(t)

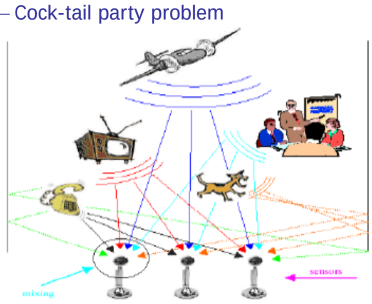
- Blind : source signal ***can not be observed***

  mixing process is unkown

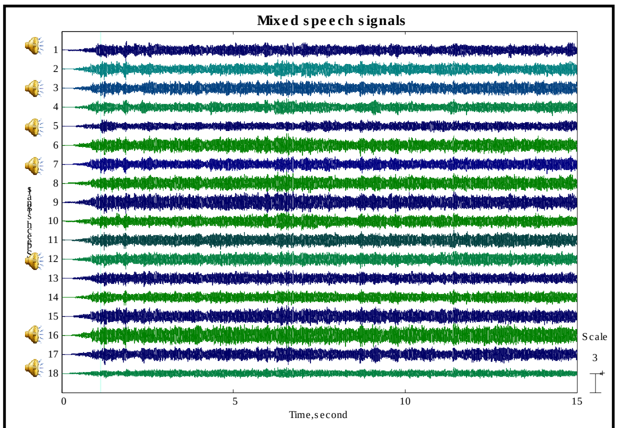
Part I. An overview of BSS
— Applications of BSS

- Speech separation
- Communication signal processing
- Biomedical signal processing
- Geophysical data processing
- Image recovery

Part I. An overview of BSS
— Cock-tail party problem

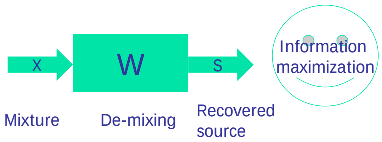


- Recovering source speech signals from their mixtures



Part I. An overview of BSS
— Cock-tail party problem

How to solve this problem ?



Part I. An overview of BSS
— Cock-tail party problem

Back

Mathematical Formulation:

$$\begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_n(t) \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \quad t = 1, \ldots, N$$

X(t) is a linear mixture of S(t)

Part II. Theoretical Analysis of Infomax Algorithm

Back

2.1 What is information maximization?

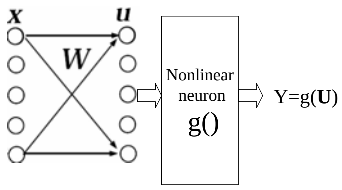
- In 1925, Fisher put forward the concept of "**information**"(Fisher information).
- In 1948, Shannon put forward the concept of **entropy** and **mutual information**.
- The mutual information I(X,Y)
- I(X,Y) = H(Y) – H(X|Y)

I(X,Y) can be used to measure the statistical dependency of random variable X and Y.

2.1 What is information maximization?

Back

- Infomax algorithm is to maximize the information transferred in a network of non-linear units.



- Adjusting W to maximize the mutual information of Y about input X

2.2 Mathematical Backgroud
— Transformation of the random variable

- Prerequisite: pdf (probability density function) of the function of a random variable X
- Suppose Y = g(X), then the pdf of the random variable Y could be expressed by the pdf of X as

$$f_y(Y) = \frac{f_x(X)}{|J|}$$

- The transformation g() should have an unique inverse.

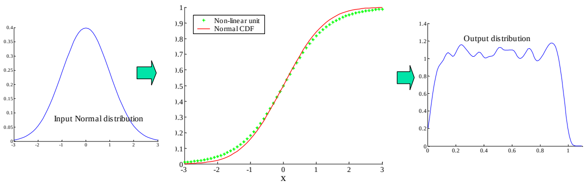
[Back](#)

2.2 Mathematical Backgroud
— Transformation of the random variable

- J is the Jacobian matrix,
- |J| is the Jacabian determinant

$$|J| = \det\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix}$$

2.2 Mathematical Backgroud
—Transformation the Random varible



$$y = \frac{1}{1+\exp(-[wx+w_0])}$$

[Back](#)

2.2 Mathematical Backgroud
— Differentiation

- Differentiation of the composite function

$$y = \frac{1}{1+\exp(-u])}, u = wx + w_0$$

$$\frac{\partial y}{\partial x} = wy(1-y)$$

$$\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) = y(1-y)(1+wx(1-2y))$$

[Back](#)

2.2 Mathematical Backgroud
— Differentiation

- Differentiation of the matrix determinant

$$\frac{\partial}{\partial W}\ln|\det W|$$

$$\frac{\partial}{\partial w_{ij}}\ln|\det W| = \frac{\text{cof } w_{ij}}{\det W}$$

$$\frac{\partial}{\partial W}\ln|\det W| = \frac{(adj W)^T}{\det W} = \left[W^T\right]^{-1}$$

[Back](#)

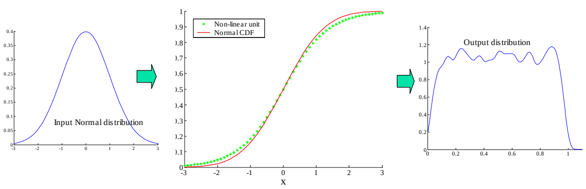
Part II Infomax Algorithm
3. Deduce the learning rule

- Maximize the mutual information of Y about input X

  I(Y,X)=H(Y)-H(Y|X)

  - H(Y) is entropy of output Y
  - H(Y|X) is conditional entropy, which is entropy of Y that didn't come from the input
  - When W and g() are deterministic, H(Y|X) is zero, thus maximizing I(Y,X) is equivalent to maximize H(Y)
- Method: Adjust the weights W of the network

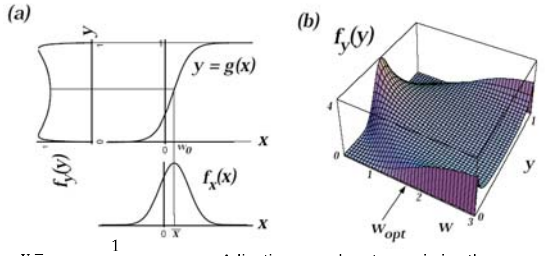
Part II Infomax Algorithm
3. Deduce the learning rule



$$y = \frac{1}{1+\exp(-[wx+w_0])}$$

Adjusting w and w_0 to maximize the entropy of Y

Part II Infomax Algorithm
3. Deduce the learning rule



$$y = \frac{1}{1+\exp(-[wx+w_0])}$$

Adjusting w and w_0 to maximize the entropy of output y

Back

Part II Infomax Algorithm
3. Deduce the learning rule

- Maximize the output entropy H(y) using the gradient method
- Calculation of H(Y):

$$H(Y)=\int f_Y(Y)\ln f_Y(Y)dY$$

$$f_Y(Y)=\frac{f_X(X)}{|J|}$$

J is the Jacobian of the transformation from X to Y

Part II Infomax Algorithm
3. Deduce the learning rule

$$H(Y) = -\int f_Y(Y)\ln f_Y(Y)dY$$
$$= \int f_Y(Y)\ln|J|dY - \int f_Y(Y)\ln f_X(X)dY$$
$$= E[\ln|J|] + E[\ln f_X(X)]$$

1 2

- 2 does not depends on W
- Maximizing H(Y) is equivalent to maximize

$$E[\ln|J|] = \int f_Y(Y)\ln|J|dY$$

Back

Part II Infomax Algorithm
3. Deduce the learning rule

- Using stochastic gradient method

$$\Delta W \propto \frac{\partial H}{\partial W} = \frac{\partial}{\partial W}(\ln|J|)$$

- Suppose $y=g(\mathbf{u})$, $u=WX+W_0$, $g(u)=(1+e^{-\mathbf{u}})^{-1}$
- Jacobian matrix J of the transformation from X to Y

$$J = diag(\frac{\partial y_i}{\partial u_i}) * W$$

$$\ln|J| = \ln|\det W| + \ln\prod_i \frac{\partial y_i}{\partial u_i}$$

Part II Infomax Algorithm
3. Deduce the learning rule

$$\frac{\partial}{\partial W}\ln|\det W| = \left[W^T\right]^{-1}$$

$$\frac{\partial}{\partial W}\ln\prod_i \frac{\partial y_i}{\partial u_i} = (1-2Y)X^T$$

$$\Delta W \propto \frac{\partial}{\partial W}\ln|J| = \frac{\partial}{\partial W}\ln|\det W| + \frac{\partial}{\partial W}\ln\prod_i \frac{\partial y_i}{\partial u_i}$$

- We obtain the learning rule

$$\Delta W \propto \left[W^T\right]^{-1} + (1-2y)x^T$$
$$\Delta w_0 \propto 1-2y$$

Back

Part II Infomax Algorithm
3. Deduce the learning rule

- Speed up the learning process
  - Natural gradient extension
- Multiplying the previous learning rule by W^TW
- The new learning rule is

$$\Delta W \propto \left[I + (I-2y)u^T\right]W$$

$$W_{i+1} = W_i + \mu\Delta W$$

  - μ is the learning rate

Back

Part II. Infomax Algorithm

4. Implementation of Infomax Algorithm

- Software package:
  - EEGLAB
  - Designed for EEG signal analysis.
- Two function file:
  - runica.m & binica.m
  - binica() calls C version of runica.m to speed up the execution.

Back

Part II. Infomax Algorithm

4. Implementation of Infomax Algorithm

- Parameter setting in Infomax algorithm:
- Optional inputs:
- 'extended' = 0 [default: don't look for subGaussian components]
- 'pca' = 0 [default: don't reduce data dimension]
- 'blocksize' = 2048 [default: heuristic dependent on data size]
- 'lrate' = 1e-4 ' stop ' 1e-6
- 'maxsteps' = 256 ' annealstep ' 0.98 [range 0-1]
- 'annealdeg' = 60 ' momentum ' 0 [range 0-1]
- 'sphering' = 'on' ' bias ' on
- 'posact' = 'on' ' verbose ' on

Back

Part II. Infomax Algorithm

4. Implementation of Infomax Algorithm

- Parameter setting in Infomax algorithm:
- Optional inputs:
- 'extended' = 0 [default: don't look for subGaussian components]
- 'pca' = 0 [default: don't reduce data dimension]
- 'blocksize' = 2048 [default: heuristic dependent on data size]
- 'lrate' = 1e-4 ' stop ' 1e-6
- 'maxsteps' = 256 ' annealstep ' 0.98 [range 0-1]
- 'annealdeg' = 60 ' momentum ' 0 [range 0-1]
- 'sphering' = 'on' ' bias ' on
- 'posact' = 'on' ' verbose ' on

Back

Part II. Infomax Algorithm

4. Implementation of Infomax Algorithm

- For binica.m, two files need to be edited,
  - 1. icadefs.m: edit the path of the ica.exe file
  - 2. binica.sc: stores the needed parameters, users need to edit this file and update the corresponding parameters.

Back

Part II. Infomax Algorithm

4. Implementation of Infomax Algorithm

- 1. Preprocessing
  - mean removal and sphering
    - $X = X\text{-E}(X)$
    - $Y = QX$, st. $\text{E}[YY^T] = \text{I}$
- 2. Training process
  - Iterate to update the estimated separating matrix $\mathbf{W}$
    - $\mathbf{W}_{i+1} = \mathbf{W}_i + \mu\Delta\mathbf{W}$
    - μ is the learning rate

Back

Part II. Infomax Algorithm

4. Implementation — MATLAB code for the training process

```
while step < maxsteps,
  permute = randperm(datalength);
  for t = 1:block:lastt,
    u=wts * data(:,permute(t:t+block-1));
    y = tanh(u);
    wts = wts + lrate*(BI-signs*y*u'-u*u')*wts;
    -- Estimate the number of sub-Gaussian sources
    -- Change the learning rate
    -- Judge the exit condition
  end
end
```

Back

Part III. Extended Infomax algorithm

- The original infomax algorithm can only deal with the case where the distribution of all the source signals are super-Gaussian.
- The Extended Infomax algorithm can separate mixed signals with ***sub- and supergaussian*** distributions (Lee T. W.,1999).

Part III. Extended Infomax algorithm

- Model the symmetric subgaussian density
- Model the supergaussian density

Back

Part III. Extended Infomax algorithm

- Model the symmetric subgaussian density:

$$p(u)=\frac{1}{2}\left(N(\mu,\sigma^2)+N(-\mu,\sigma^2)\right) \qquad \text{(2-1)}$$

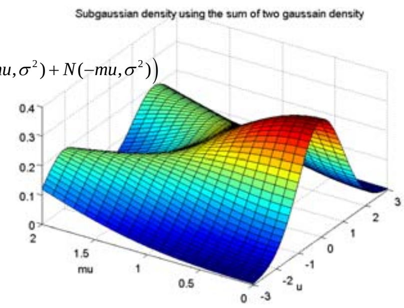
  - $N(\mu,\sigma^2)$ is normal density function
- The kurtosis k_4 of p(u) is

$$k_4=\frac{-2\mu^4}{(\mu^2+\sigma^2)^2}$$

- Thus equation (2-1) defines a strictly subgaussian when $\mu>0$.

Part III. Extended Infomax algorithm

$$p(u)=\frac{1}{2}\left(N(mu,\sigma^2)+N(-mu,\sigma^2)\right)$$



Back

Part III. Extended Infomax algorithm

—— learning rule for subgaussian sources

- p(u) is defined by equation (2-1), choose the nonlinear transformation g(u) in the Infomax algorithm as

$$g(u)=-\frac{\frac{\partial p(u)}{\partial u}}{p(u)}=\frac{u}{\sigma^2}-a\left(\frac{exp(au)-\exp(-au)}{exp(au)+\exp(-au)}\right)$$

$$=\frac{u}{\sigma^2}-a\tanh(au)$$

$$a=\frac{\mu}{\sigma^2}$$

Back

Part III. Extended Infomax algorithm

—— learning rule for subgaussian sources

$$g(u)=u-\tanh(u)\ ,\ \mu=1,\sigma^2=1$$

- The learning rule for subgaussian sources is

$$\Delta W \propto \left[I-g(u)u^T\right]W=\left[I+\tanh(u)u^T-uu^T\right]W$$

Back

Part III. Extended Infomax algorithm
— Learning rule for supergaussian sources

- For unimodal supergaussian source, we can use density model

$$p(u) \propto p_G(u)\operatorname{sech}^2(u)$$

- The nonlinearity g(u) is

$$g(u) = -\frac{\frac{\partial p(u)}{\partial u}}{p(u)} = u + \tanh(u)$$

- The learning rule is

$$\Delta W \propto \left[I - g(u)u^T\right]W = \left[I - \tanh(u)u^T - uu^T\right]W$$

Back

Part III. Extended Infomax algorithm
— Learning rule for supergaussian sources

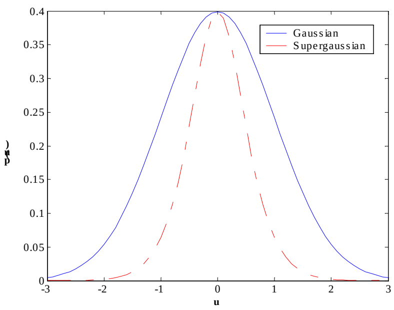
- For unimodal supergaussian source, we can use density model

$$p(u) \propto p_G(u)\operatorname{sech}^2(\beta u)$$

  - $p_G(u)$ is normal density N(0,1)
  - sech(u) is hyperbolic secant
  - β can control the sharpness (kurtosis) of the supergaussian density

Back

Part III. Extended Infomax algorithm



Back

Part III. Extended Infomax algorithm
— Switching between nonlinearities

- The switching between the sub- and supergaussian learning rule is

$$\Delta W \propto \left[I - K\tanh(u)u^T - uu^T\right]W$$

$$\text{for} \quad \begin{Bmatrix} k_i = 1: \text{ supergaussian} \\ k_i = -1: \text{subgaussian} \end{Bmatrix}$$

  - k_i are elements of the N-dimensional diagonal matrix K.

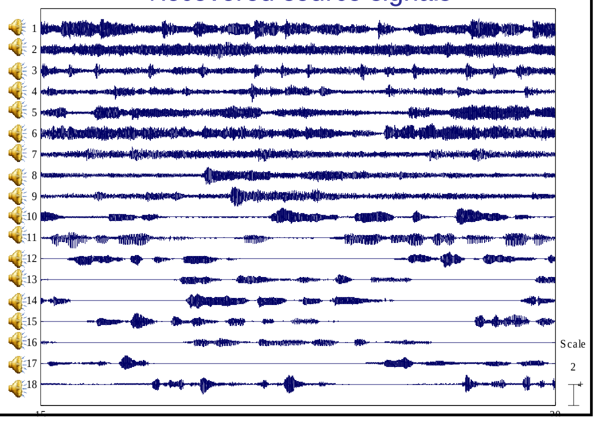
Back

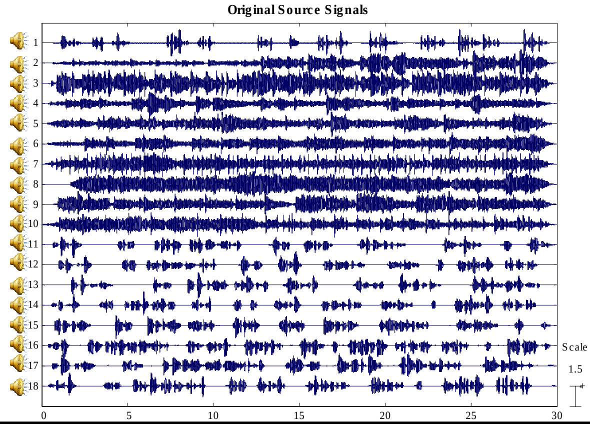
Part IV. Experimental Results

Part IV. Experimental Results

- Data set: 18 signals，9 music signals, 9 speech signals
- Mixing by a randomly generated mixing matrix
- Parameter setting:
  - Block size: 4096
  - Learning rate: 0.0001
  - Maximum iteration step: 256

Recovered source signals





Part IV. Experimental Results

— Evaluation

- **Permutation**
- **Scale**
- $x = As = \mathbf{a}_1 s_1 + \ldots + \mathbf{a}_n s_n$
- Order between the independent components:
  - Use the variance of the independent components
  - Use the nongaussianity of the independent components

2004-12-14 45/54

Part IV. Experimental Results

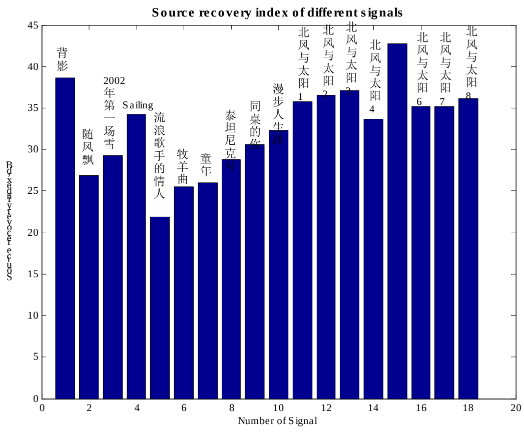
— Evaluation

- Performance Index
- Source recovery SNR:

$$\mathrm{SNR}=10\log\frac{\left\|S_i-\hat{S}_i\right\|^2}{\left\|S_i\right\|^2},\ i=1,\ldots,N$$

- Eliminate scale and permutation ambiguity
- Mixing matrix recovery
- P = WA, $\mathrm{PI}=\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{|p_{ij}|}{\max_k |p_{ik}|}-1\right)+\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{|p_{ij}|}{\max_k |p_{kj}|}-1\right)$

2004-12-14 46/54



- Order of the separated independent components

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S	刀郎	泰坦尼克	童年	邓丽君	随风飘	同桌的你	牧羊曲	远航	流浪歌手	北风7	北风6	北风1	北风4	北风5	北风8	北风2	北风3	背影

- Order of the variance of each signal

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S	刀郎	同桌的你	泰坦尼克	童年	邓丽君	北风5	北风6	流浪歌手	北风7	远航	牧羊曲	北风1	北风2	背影	北风3	北风4	北风8	随风飘

2004-12-14 48/54

Part IV. Experimental Results

— Order of the independent component

Back

- Infomax algorithm reorders the independent component according to the mean projected variance
  - Xproj = A'(:,i) * S'(i,:)
  - Var(i) = mean(sum(Xproj.*Xproj)/(N-1))
  - A' is the estimated mixing matrix, S' is the separated independent components
  - i = 1,…,N,N is number of the sources
- Order the independent components according to Var

2004-12-14 49/54

Part IV. Experimental Results

— Comparison between the Infomax and FastICA

Back

- 18 source signal data set
  - Execution time:
    - Infomax 469 s
    - FastICA 2051ss
  - Mean Reconstruction SNR
    - Infomax 32.7dB(using runica)
    - FastICA 36.3
- Iteration number
  - Infomax 63

2004-12-14 50/54

Part V Summary and Perspective

- Summary:
  - BSS has become a useful tool in signal processing.
  - Information maximization provides a general framework for BSS problem
  - By modeling the sub- and supergaussian densities, the Extended Infomax algorithm are able to separate mixed signals with sub- and supergaussian densities
  - Expermental results verified the effectiveness of the Infomax algorithm

2004-12-14 51/54

Part V Summary and Perspective

- Perspective:
  - Consider more practical mixing models
  - Exploiting the intrinsic statistical properties of the audio signals to enhance the performance of BSS

2004-12-14 52/54

References

- Bell A. J. & Sejnowski T. J. (1995). An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, **7**, 1004-1034
- Lee T. W., M. Girolami and Sejnowski T. J. (1995). Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Subgaussian and Supergaussian Sources. Neural Computation, 11, 417-441.

2004-12-14 53/54

Thank you very much!

Giving your criticisms and suggestions!