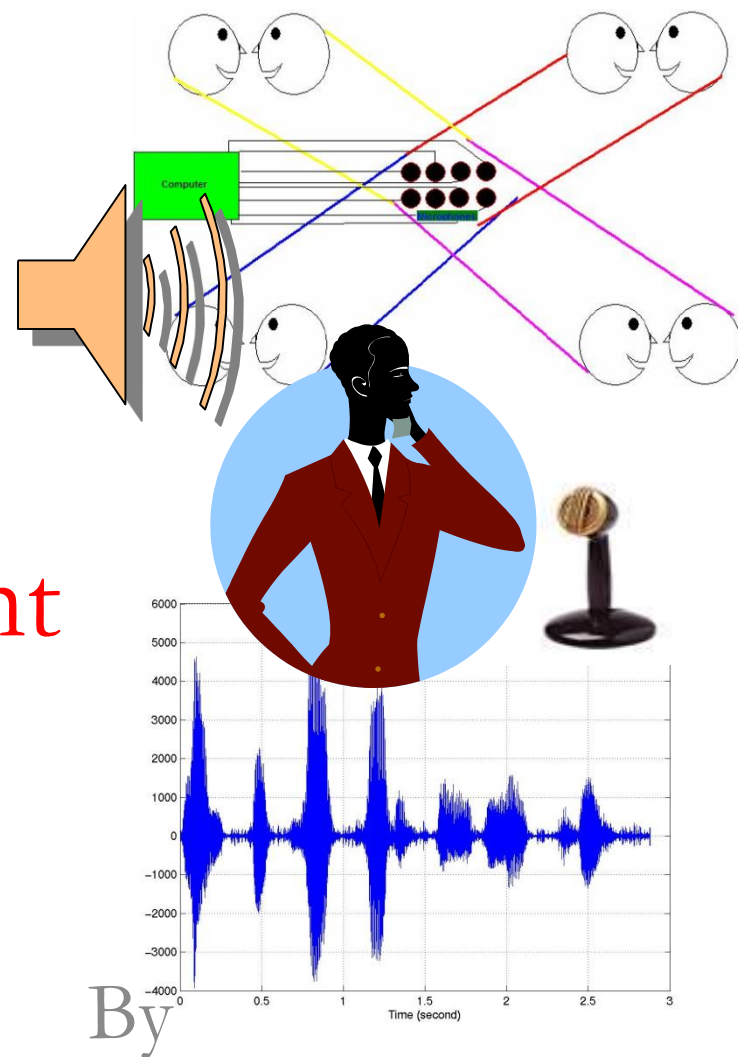


A Tutorial on Data Reduction

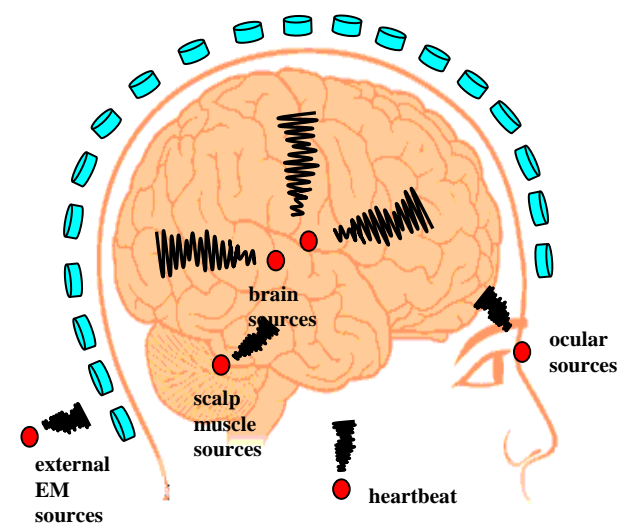
Independent Component Analysis (ICA)



By

Shireen Elhabian and Aly Farag
University of Louisville, CVIP Lab

September 2009

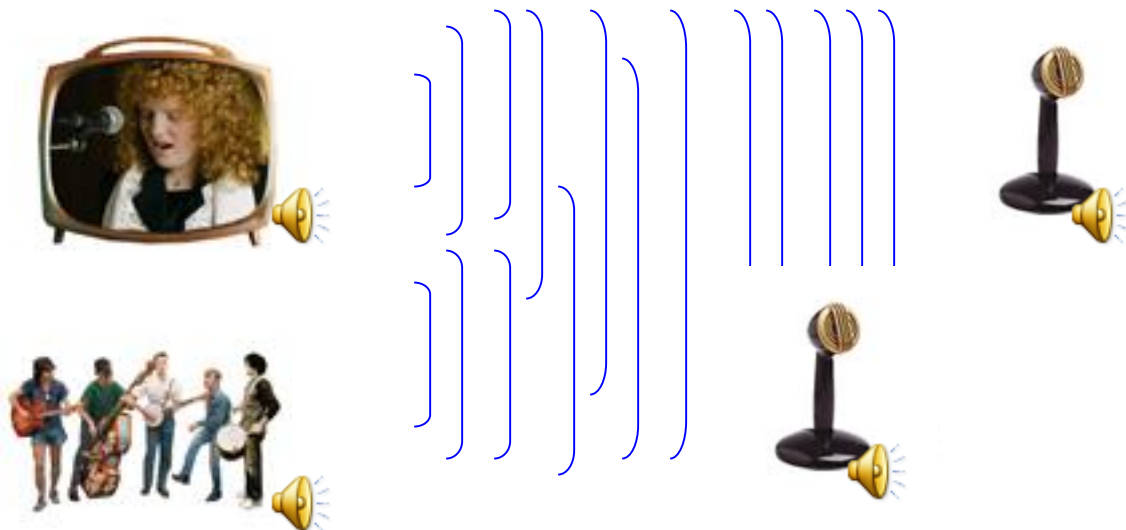


Outline

- Motivation – Cocktail-party problem
- ICA versus PCA
- Definition of ICA
- ICA Assumptions
- BSS – Blind Source Separation
- Ambiguities of ICA
- Statistical illustration of ICA
- Problem Formulation
 - What is Independence?
 - Uncorrelated does not mean independent
- Gaussian-variables are forbidden, Why?!!!
- Non-Gaussianity estimation
- Principles of ICA
- Preprocessing of ICA
- ICA – Examples of Algorithms
- Let's do it ...

Motivation - Cocktail-Party Problem

- Simple scenario:
 - Two people speaking simultaneously in a room.
 - Speeches are recorded by two microphones in separate locations.



Motivation - Cocktail-Party Problem

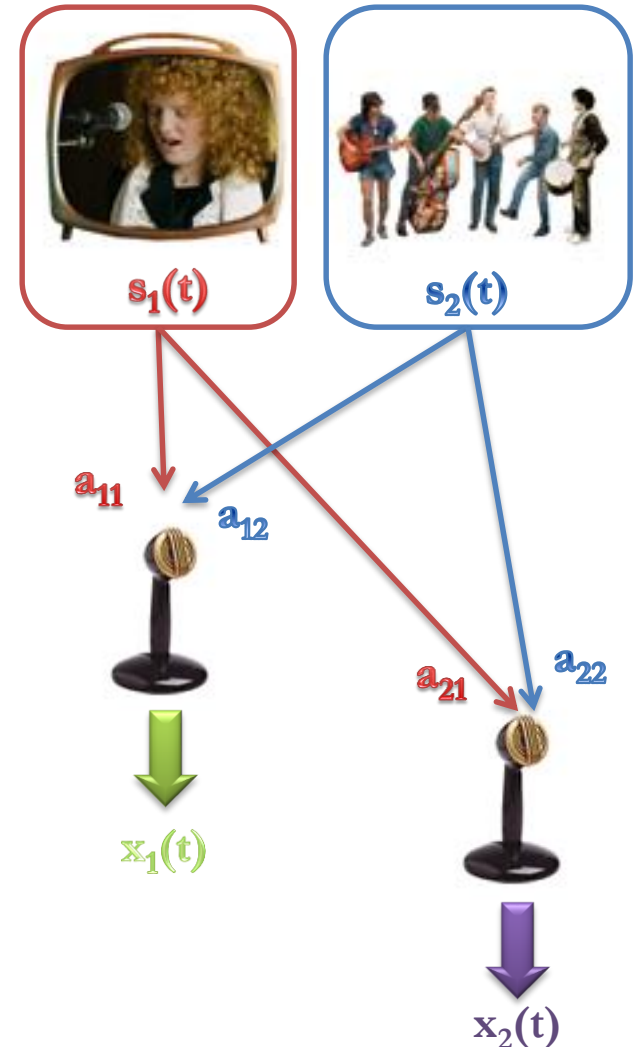
- Let $s_1(t)$, $s_2(t)$ be the speech signals emitted by the two speakers.
- Recorded time signals, by the two microphones, are denoted by $x_1(t)$, $x_2(t)$.
- The recorded time signals can be expressed as a linear equation:

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

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

where parameters in matrix \mathbf{A} depend on distances of the microphones to the speaker, along with other microphone properties

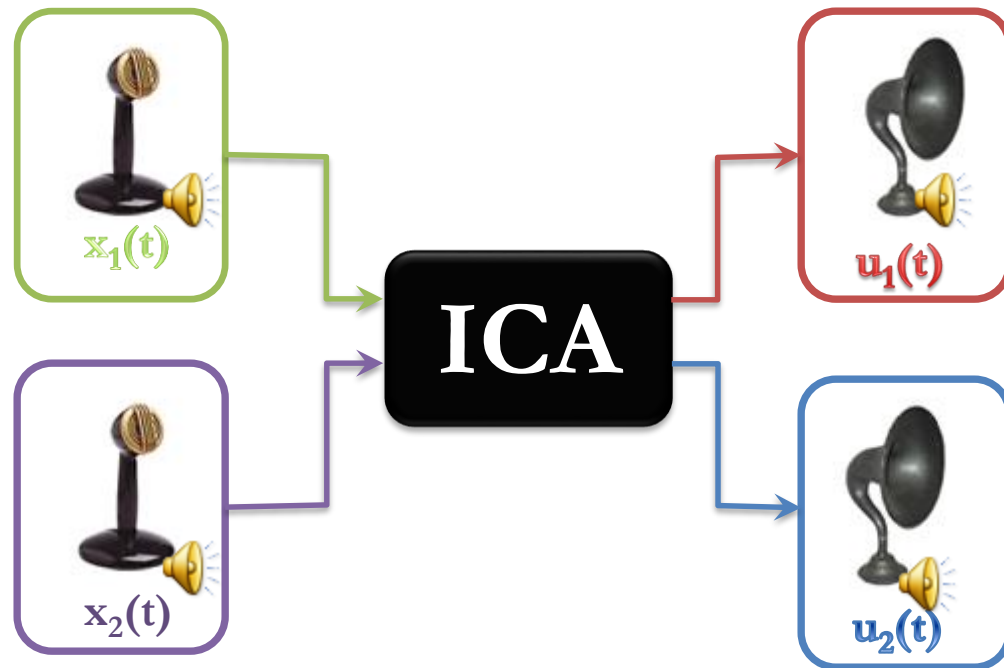
- Assume $s_1(t)$ and $s_2(t)$ are *statistically independent*.



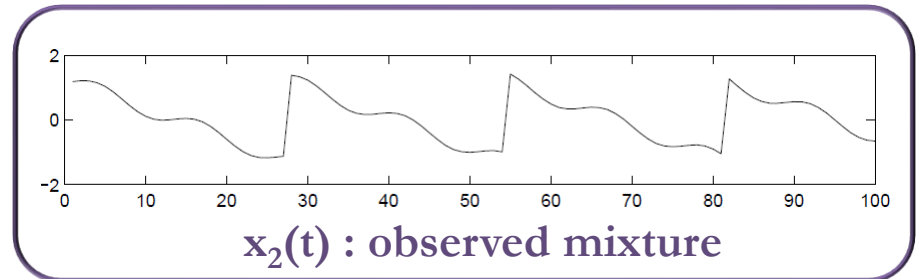
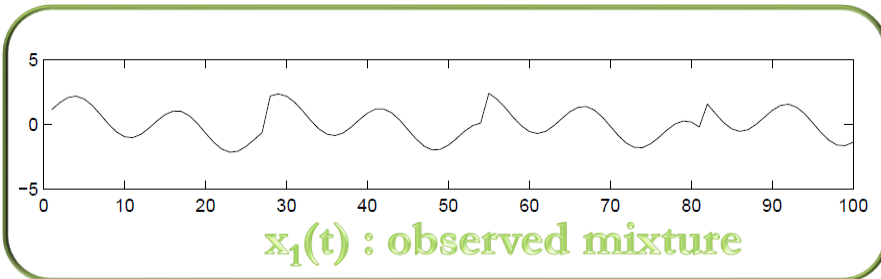
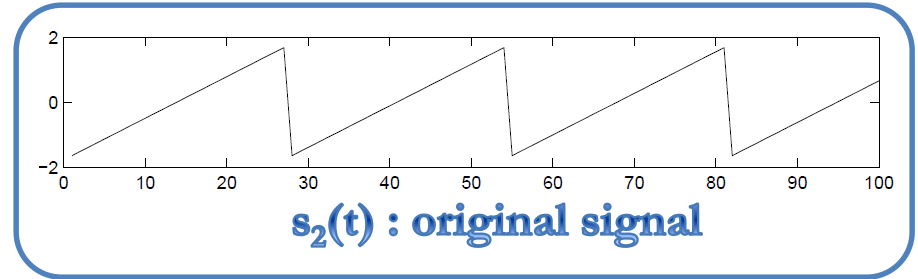
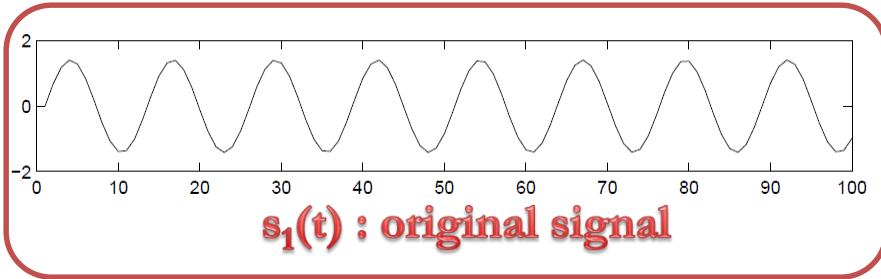
Motivation - Cocktail-Party Problem

Goal:

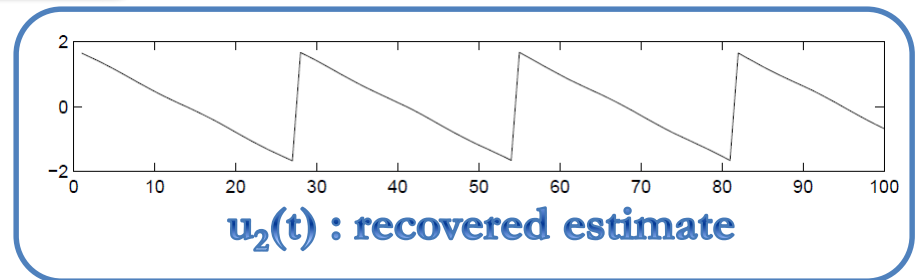
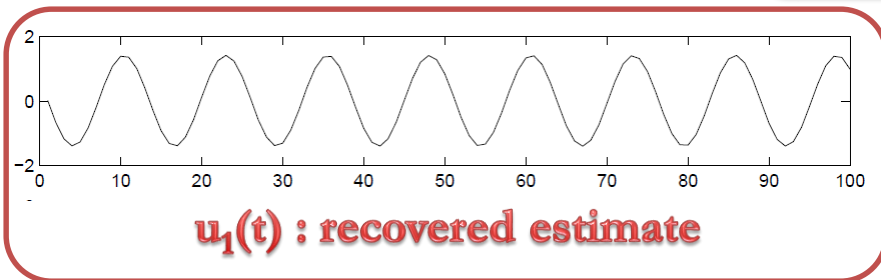
- Recover the unmixed speech signals, best estimate $u_i(t)$, without knowing \mathbf{A} or $\mathbf{s}_i(t)$.



Motivation - Cocktail-Party Problem



ICA

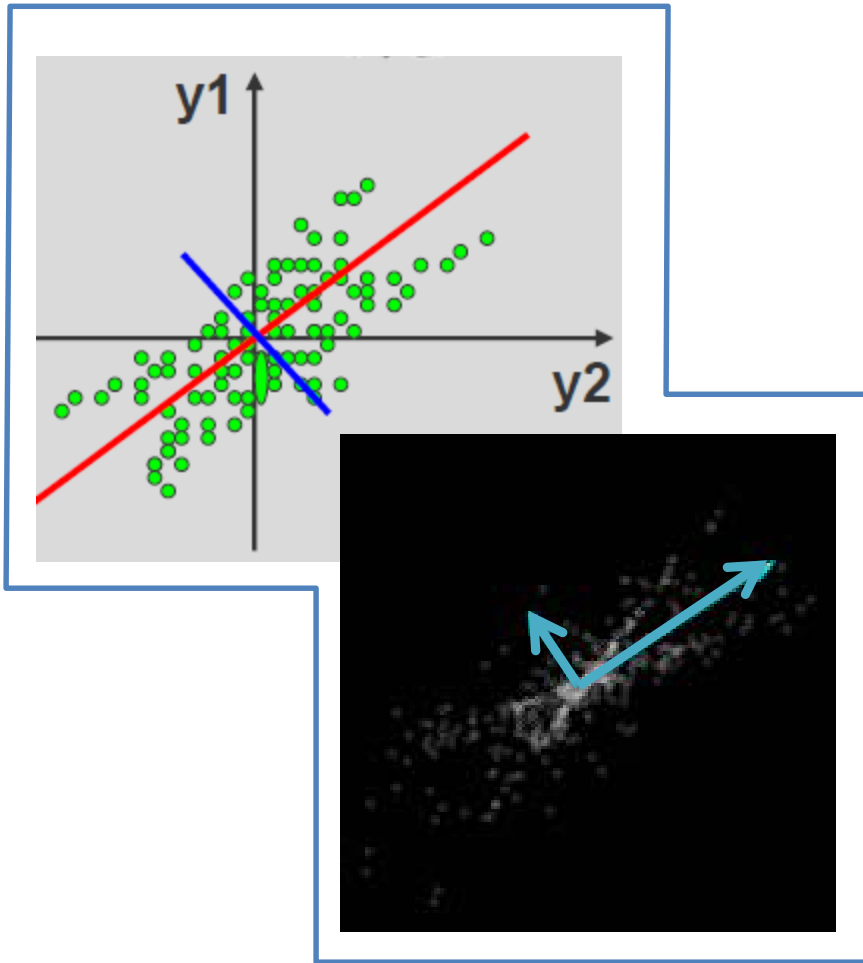


The original signals were very accurately estimated, up to multiplicative signs

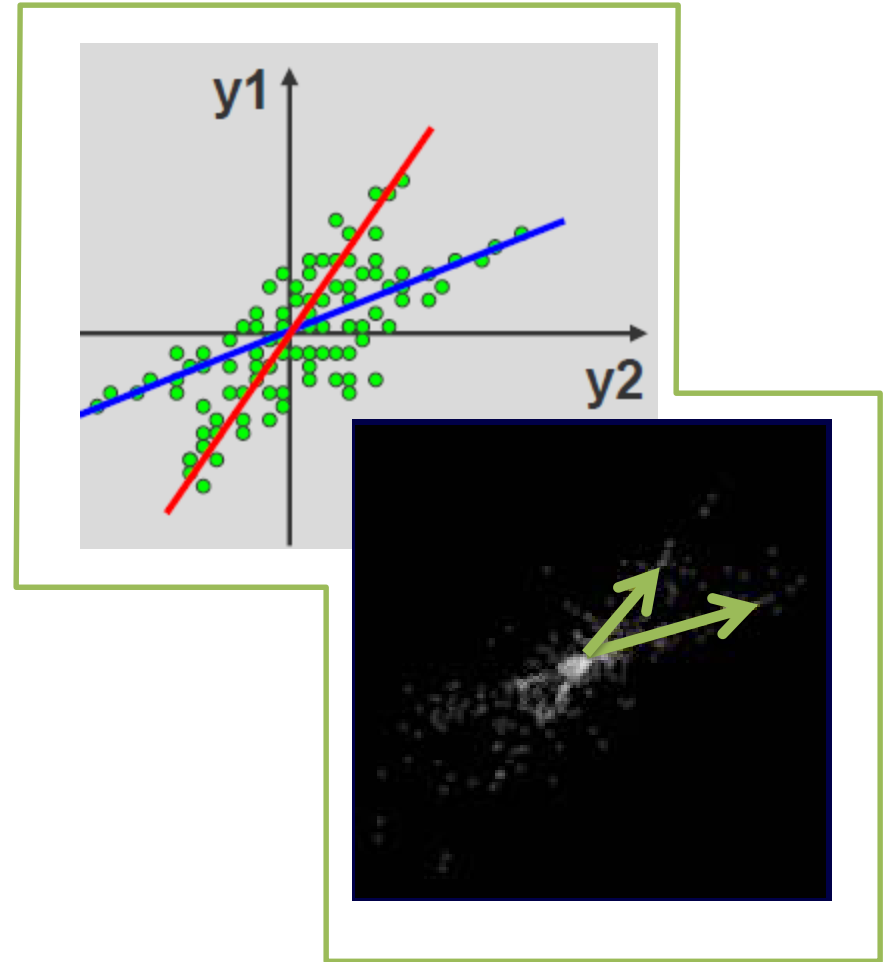
ICA versus PCA

- Similarity
 - Feature extraction
 - Dimension reduction
- Difference
 - PCA uses up to *second order moments* of the data to produce uncorrelated components.
 - ICA strives to generate components as independent as possible through minimizing both the second-order and higher-order dependencies in the given data.

ICA versus PCA



PCA finds directions of maximal variance (using second order statistics)



ICA finds directions which maximize independence (using higher order statistics)

Definition of ICA

- Assume that we have n mixtures $\mathbf{x}_1, \dots, \mathbf{x}_n$ of n independent components:

$$\mathbf{x}_j = \mathbf{a}_{j1}\mathbf{s}_1 + \mathbf{a}_{j2}\mathbf{s}_2 + \dots + \mathbf{a}_{jn}\mathbf{s}_n \quad \text{for all } j$$

The time index t has dropped in ICA model, since we assume that each mixture and individual components are random variables instead of a proper time signal. Thus the observed values $x_j(t)$, e.g. the microphone signals in the cocktail party problem, are then a sample/realization of this random variable.

- Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean.

If this is not true, then the observable variables x_j can always be centered by subtracting the sample mean, which makes the model zero-mean.

Definition of ICA

- The equation can be expressed using vector-matrix notation,

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

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s_1 \\ \cdot \\ \cdot \\ s_n \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} a_{11} & \cdot & \cdot & a_{1n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & \cdot & \cdot & a_{nn} \end{bmatrix}$$

\mathbf{x} : random vector whose elements are the mixtures x_1, \dots, x_n

\mathbf{s} : random vector whose elements are the sources s_1, \dots, s_n

\mathbf{A} : mixing matrix with elements a_{ij}

- Expression in *columns* of matrix \mathbf{A} ,
$$\mathbf{x} = \sum_{i=1}^n a_i s_i$$

Definition of ICA

- This statistical model is called *independent component analysis*, or **ICA** model.
- ICA model is a *generative* model, since it describes how the recorded data are generated by mixing the individual components.

ICA – Assumptions

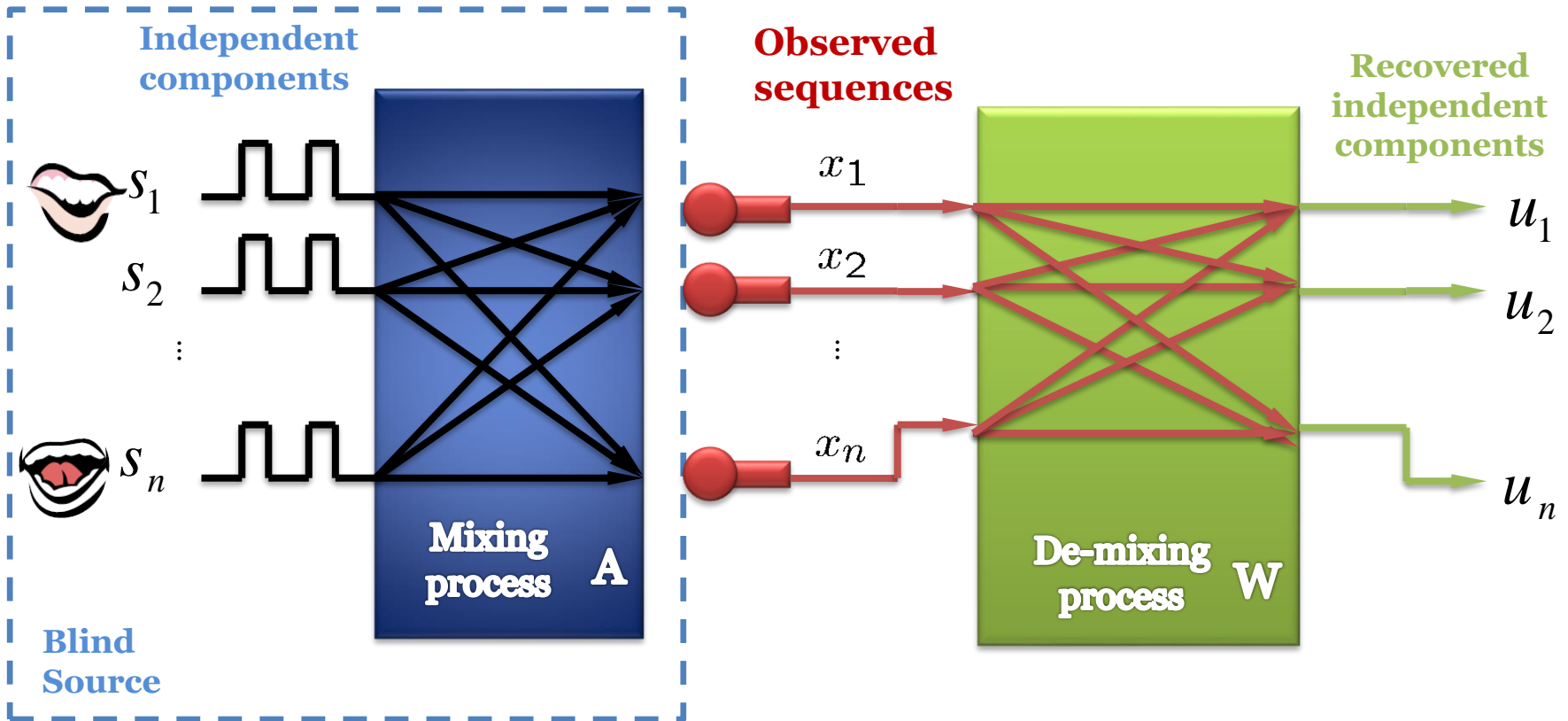
- The starting point for ICA is the very simple assumption that the components \mathbf{s}_i are *statistically independent* – explained later.
- It will be shown that we must also assume that the independent component must have *nongaussian distributions*. However, in the basic model we do not assume these distributions known (if they are known, the problem is considerably simplified.)
- For simplicity, we are also assuming that the unknown mixing matrix is square, but this assumption can be sometimes relaxed.
- Then, after estimating the matrix \mathbf{A} , we can compute its inverse, say \mathbf{W} , and obtain the independent component simply by:

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{x} = \mathbf{W}\mathbf{x}$$

BSS - Blind Source Separation

- ICA is very closely related to the method called *blind source separation (BSS)* or *blind signal separation*.
- A “**source**” means here an original signal, i.e. independent component, like the speaker in a cocktail party problem.
- “**Blind**” means that we know very little, if anything, on the mixing matrix **A**, and make little assumptions on the source signals.
- ICA is one method, perhaps the most widely used, for performing blind source separation.

BSS - Blind source separation



Ambiguities of ICA

Two major ambiguities:

1. The variances (energies) of the independent components \mathbf{s}_i cannot be determined.

- Since both \mathbf{s} and \mathbf{A} are unknown, any scalar multiplier of source \mathbf{s}_i can be cancelled by dividing the corresponding column \mathbf{a}_i of \mathbf{A} with the same scalar value.
- As a consequence, we may quite as well fix the magnitudes of the independent components; as they are random variables, the most natural way to do this is to assume that each has unit variance: $E\{s_i^2\} = 1$.
- Note that this still leaves the ambiguity of the sign: we could multiply the an independent component by -1 without affecting the model. This ambiguity is, fortunately, insignificant in most applications.

Ambiguities of ICA

Two major ambiguities:

2. The order of the independent components cannot be determined.
 - Again, since \mathbf{s} and \mathbf{A} are unknown, order of the terms in the model can be changed freely, and we can call any of the independent components the first one.

Statistical Illustration of ICA

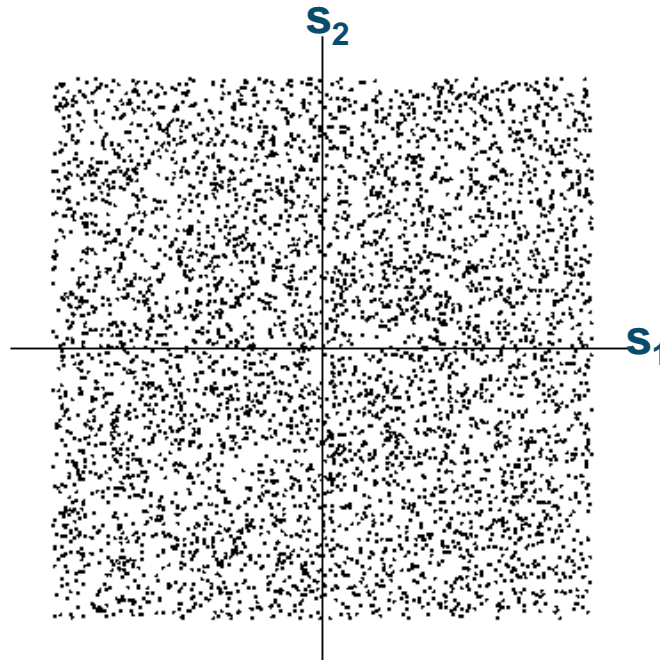
- Consider two independent components have the following uniform distributions,

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \leq \sqrt{3} \\ 0 & \text{otherwise} \end{cases}$$

- This uniform distribution has zero mean and the variance is equal to 1

Statistical Illustration of ICA

- The graphical view of the joint distribution is shown as following,



Statistical Illustration of ICA

- Assume that the two individual components are mixed by the following mixing matrix,

$$A_0 = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

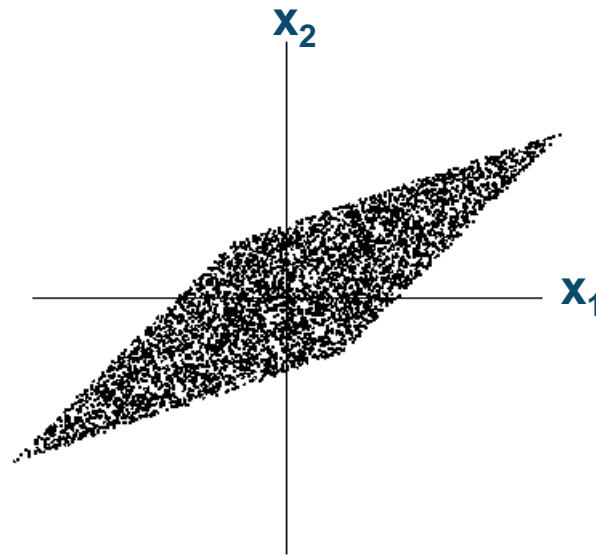
- The mixed variables \mathbf{x} can then be generated using the ICA model,

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

i.e.
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \Rightarrow \begin{cases} x_1 = 2s_1 + 3s_2 \\ x_2 = 2s_1 + 1s_2 \end{cases}$$

Statistical Illustration of ICA

- The following shows joint distribution of the mixtures \mathbf{x}_1 and \mathbf{x}_2 ,



Note that the random variables \mathbf{x}_1 and \mathbf{x}_2 are *not independent any more*; an easy way to see this is to consider, whether it is possible to predict the value of one of them, say \mathbf{x}_2 , from the value of the other. Clearly if \mathbf{x}_1 attains one of its maximum or minimum values, then this completely determines the value of \mathbf{x}_2 . *They are therefore not independent.*

Notice anything interesting ?!!!

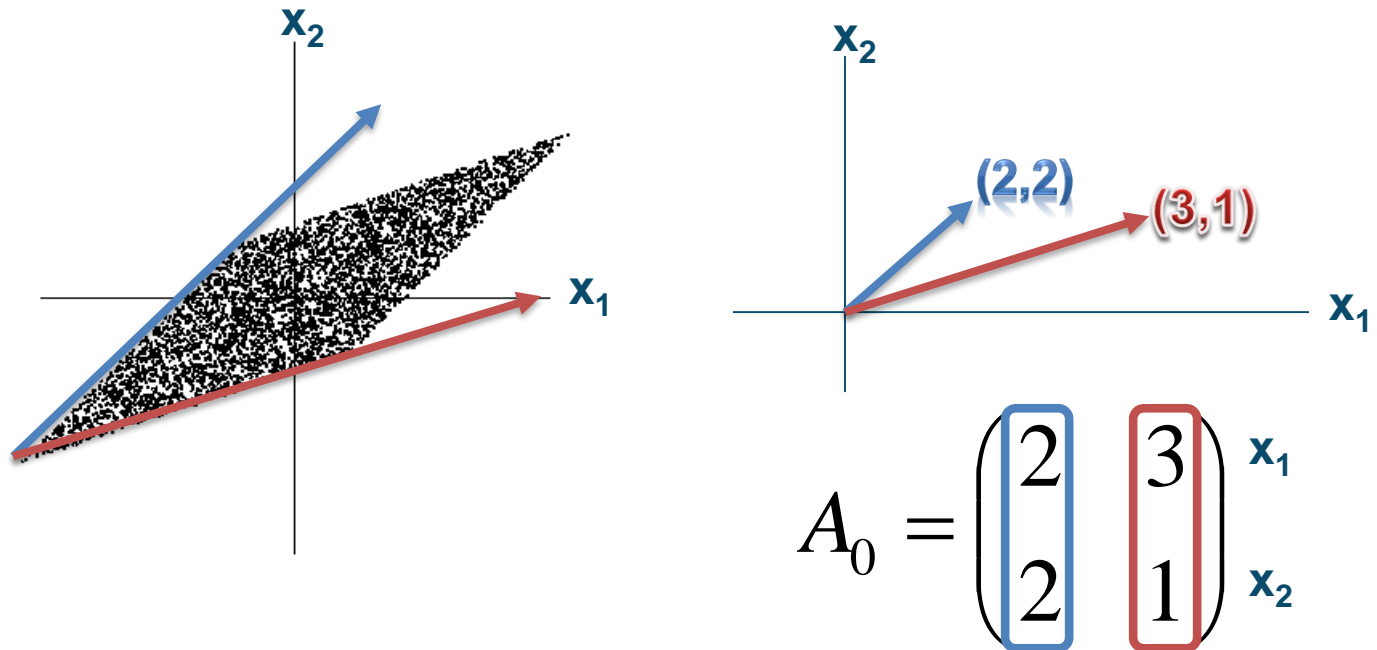
(Hint: Related to mixing matrix A_0)

$$A_0 = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

Statistical Illustration of ICA

- Answer:

- The edges of the parallelogram are the directions of the columns of A_0 .



This means that we could, *in principle*, estimate the ICA model by first estimating the joint density of \mathbf{x}_1 and \mathbf{x}_2 , and then *locating the edges*. So, the problem seems to have a solution.

Statistical Illustration of ICA

- HOWEVER, this method works poorly in reality because it only works with variables that has *uniform* distributions.
- Moreover, it would be computationally quite complicated.
- What we need is a method that works for any distributions of the independent components, and works fast and reliably.

Problem Formulation

The goal of ICA is to find a linear mapping \mathbf{W} such that the unmixed sequences \mathbf{u} ,

$$\mathbf{u}(t) = \mathbf{W} \mathbf{x}(t) = \mathbf{W} \mathbf{A} \mathbf{s}(t)$$

are maximally *statistically independent*.

What is Independence ?!!!

- To define the concept of independence, consider two scalar-valued random variables \mathbf{y}_1 and \mathbf{y}_2 .
 - Basically, the variables \mathbf{y}_1 and \mathbf{y}_2 are said to be independent if information on the value of \mathbf{y}_1 does not give any information on the value of \mathbf{y}_2 , and vice versa.
- Technically, independence can be defined by the probability densities.
 - Let us denote by $p(y_1, y_2)$ the joint probability density function (pdf) of \mathbf{y}_1 and \mathbf{y}_2 .
 - Let us further denote by $p_1(y_1)$ the marginal pdf of \mathbf{y}_1 , i.e. the pdf of \mathbf{y}_1 when it is considered alone, likewise, $p_2(y_2)$ the marginal pdf of \mathbf{y}_2

$$p_1(y_1) = \int p(y_1, y_2) dy_2 \quad \text{and} \quad p_2(y_2) = \int p(y_1, y_2) dy_1$$

- Then, we define \mathbf{y}_1 and \mathbf{y}_2 are independent if and only if the joint pdf is factorizable in the following way:

$$p(y_1, y_2) = p_1(y_1)p_2(y_2)$$

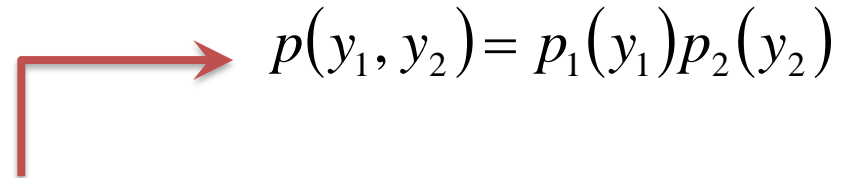
- This definition extends naturally for any number n of random variables, in which case the joint density must be a product of n terms.

What is Independence ?!!!

- The definition can be used to derive the most important property of independent random variables.
- Given two functions, h_1 and h_2 , we always have:

$$E\{h_1(y_1)h_2(y_2)\} = E\{h_1(y_1)\}E\{h_2(y_2)\}$$

- This can be proved as follows:


$$p(y_1, y_2) = p_1(y_1)p_2(y_2)$$

$$\begin{aligned} E\{h_1(y_1)h_2(y_2)\} &= \iint h_1(y_1)h_2(y_2)p(y_1, y_2)dy_1dy_2 \\ &= \iint h_1(y_1)p(y_1)h_2(y_2)p(y_2)dy_1dy_2 \\ &= \int h_1(y_1)p(y_1)dy_1 \int h_2(y_2)p(y_2)dy_2 \\ &= E\{h_1(y_1)\}E\{h_2(y_2)\} \end{aligned}$$

Uncorrelated does not mean Independent

- A weaker form of independence is uncorrelatedness. Two random variables y_1 and y_2 are said to be uncorrelated, if their covariance is zero:

$$\begin{aligned} C(y_1, y_2) &= E\{(y_1 - E\{y_1\})(y_2 - E\{y_2\})\} \\ &= E\{y_1 y_2\} - E\{y_1\}E\{y_2\} = 0 \end{aligned}$$

- If the variables are independent, they are uncorrelated,

$$E\{y_1 y_2\} = E\{y_1\}E\{y_2\} \Rightarrow E\{y_1 y_2\} - E\{y_1\}E\{y_2\} = 0$$

- On the other hand, uncorrelatedness does not imply independence.

- For example, assume that (y_1, y_2) are discrete valued and follow such a distribution that the pair are with probability 1/4 equal to any of the following values: (0,1), (0,-1), (1,0), (-1,0). Then y_1 and y_2 are *uncorrelated*, but *not independent*.

$$E\{y_1^2 y_2^2\} = 0 \neq 1/4 = E\{y_1^2\}E\{y_2^2\}$$

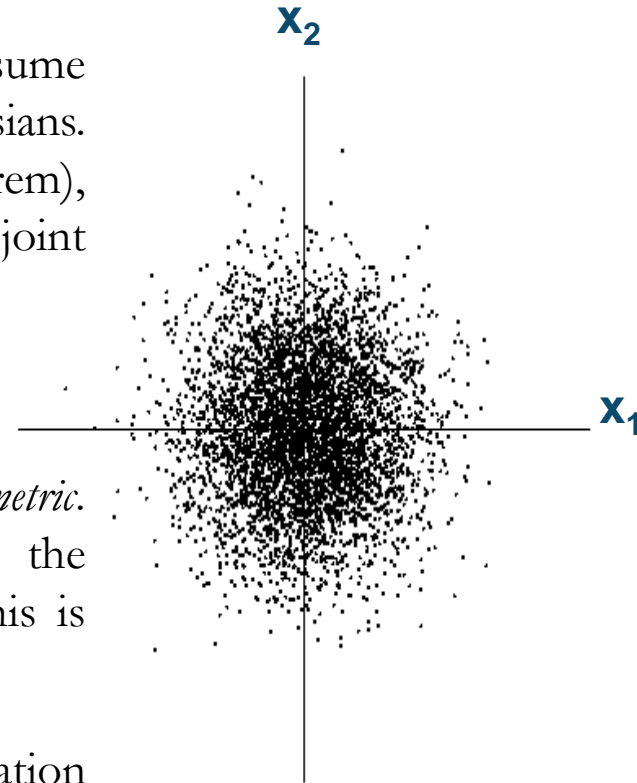
- Since independence implies uncorrelatedness, many ICA methods constrain the estimation procedure so that it always gives uncorrelated estimates of the independent components. This reduces the number of free parameters, and simplifies the problem.

Gaussian variables are forbidden, Why?!!!

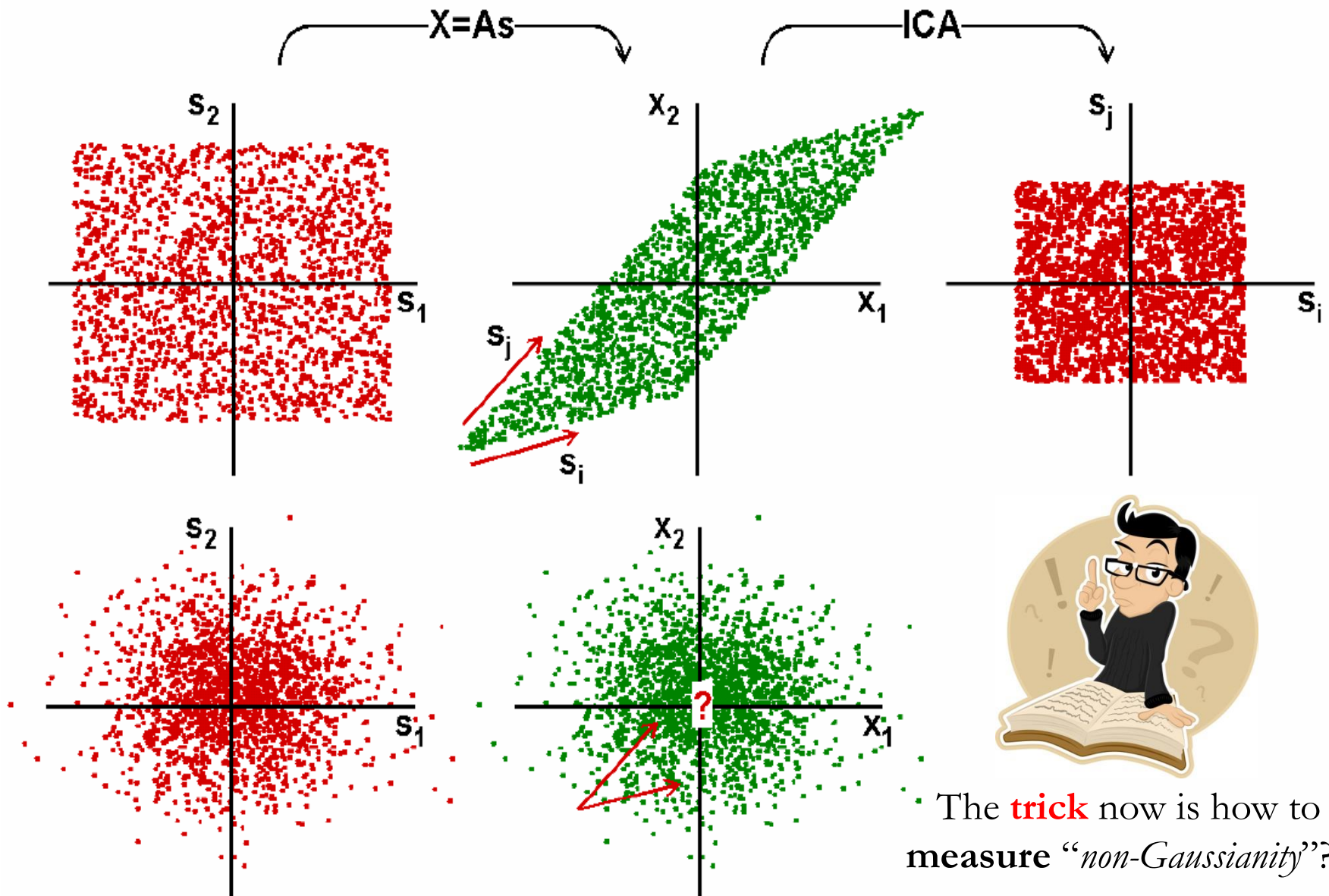
- The fundamental restriction in ICA is that the independent components must be *non-gaussian* for ICA to be possible.
- To see why gaussian variables make ICA impossible, assume that the mixing matrix is orthogonal and the \mathbf{s}_i are Gaussians. Then \mathbf{x}_1 and \mathbf{x}_2 are Gaussians too (by central limit theorem), they are uncorrelated, and of unit variance. Their joint density is given by;

$$p(x_1, x_2) = \frac{1}{2\pi} e^{-\frac{x_1^2 + x_2^2}{2}}$$

- The figure shows that the density is completely *symmetric*. Therefore, it does not contain any information on the directions of the columns of the mixing matrix \mathbf{A} . This is why \mathbf{A} cannot be estimated.
- Moreover, the distribution of any orthogonal transformation of the Gaussian $(\mathbf{x}_1, \mathbf{x}_2)$ has exactly the same distribution as $(\mathbf{x}_1, \mathbf{x}_2)$.
- Thus, in the case of Gaussian variables, we can only estimate the ICA model up to an orthogonal transformation.



Gaussian variables are forbidden, Why?!!!



Non-Gaussianity Estimation

- The Central Limit Theorem
 - Distribution of a sum of independent random variables tends toward a Gaussian distribution.
 - Thus, a sum of two independent random variables usually has a distribution that is closer to gaussian than any of the two original random variables.

Non-Gaussianity Estimation

- Let us now assume that the data vector \mathbf{x} is distributed according to the ICA data model, i.e. a mixture of independent components.
- For simplicity, let us assume that all the independent components have identical distributions.
- To estimate one of the independent components, we consider a linear combination of the \mathbf{x}_i , let's denote this by \mathbf{y} ;

$$\mathbf{y} = \mathbf{w}^T \mathbf{x}$$

where \mathbf{w} is a vector to be determined, and it's one row of the inverse of \mathbf{A} , i.e. \mathbf{W}

Non-Gaussianity Estimation

- Define $\mathbf{z} = \mathbf{A}^T \mathbf{w}$ and then we have,

$$\mathbf{y} = \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \mathbf{A} \mathbf{s} = \mathbf{z}^T \mathbf{s}$$

- This linear combination would actually equal one of the independent components.
- The question is now:
 - How could we use the Central Limit Theorem to determine \mathbf{w} so that it would equal one of the rows of the inverse of \mathbf{A} ?
 - In practice, we cannot determine such \mathbf{w} exactly, because we have no knowledge of matrix \mathbf{A} , but we can find an estimator that gives a good approximation.
- $\mathbf{z}^T \mathbf{s}$ is more Gaussian than any of the \mathbf{s}_i , and it is least Gaussian (i.e. non-gaussian) if it is equal to one of the \mathbf{s}_i
- Maximizing the non-Gaussianity of $\mathbf{w}^T \mathbf{x}$ will give us one of the independent components.

Non-Gaussianity Estimation

Measurement of non-Gaussianity

- **Kurtosis**

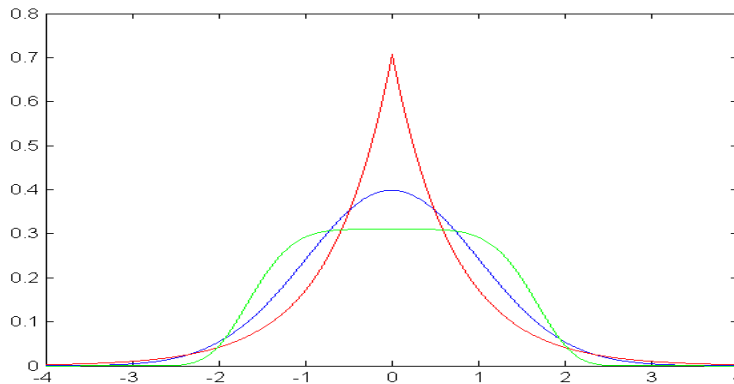
- Defined by: $\mathbf{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2$
- Since y is of unit variance, the kurtosis equation simplifies to $E\{y^4\} - 3$. Therefore, the kurtosis can be considered as the normalized version of the fourth moment $E\{y^4\}$.
- The kurtosis for a Gaussian is zero because the fourth moment is equal to $3(E\{y^2\})^2$.
- For most nongaussian random variables, the value for kurtosis is nonzero.
- However Kurtosis is very sensitive to outliers when its value has to be estimated from a measured sample.

Non-Gaussianity Estimation

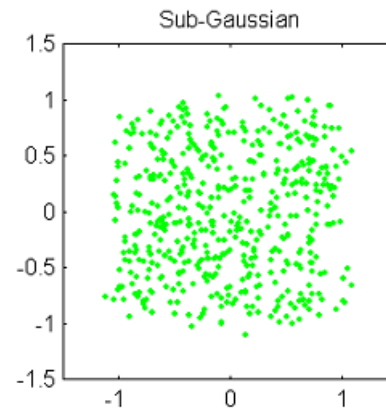
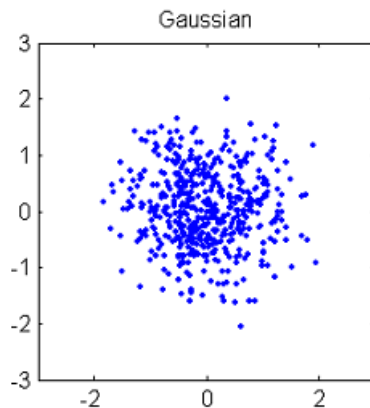
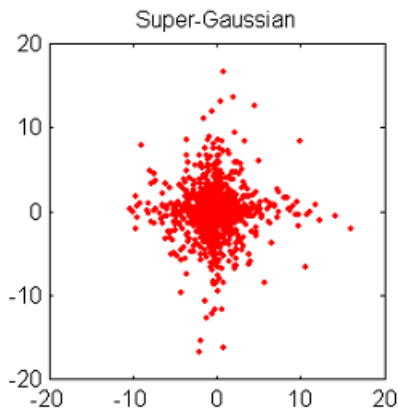
Measurement of non-Gaussianity

- **Kurtosis**

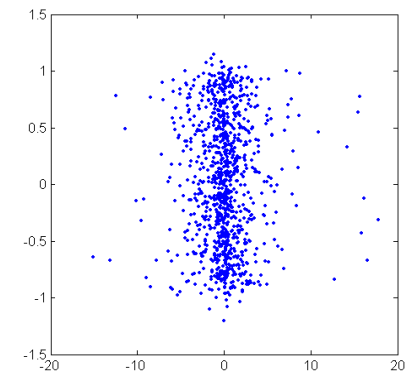
- Kurtosis can be positive or negative. Random variables that have negative kurtosis are called *subgaussian*, having a “flat” pdf and those with positive values for kurtosis are referred to as *supergaussian*, having a “spiky” pdf with heavy tails.



- **Super-Gaussian** = more peaked, than Gaussian, heavier tail
- **Sub-Gaussian** = flatter, more uniform, shorter tail than Gaussian



Sub- and Super-Gaussian



Non-Gaussianity Estimation

Measurement of non-Gaussianity

- **Negentropy**

- Based on the information-theoretic quantity of *entropy*.

- The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. The **more** unpredictable (**random**) and unstructured the variable is, the **larger** the **entropy** value.

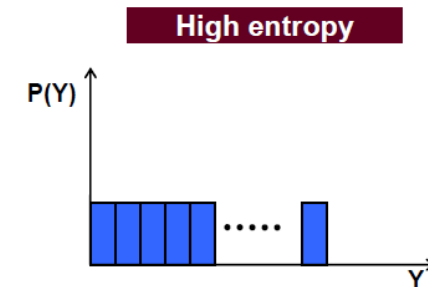
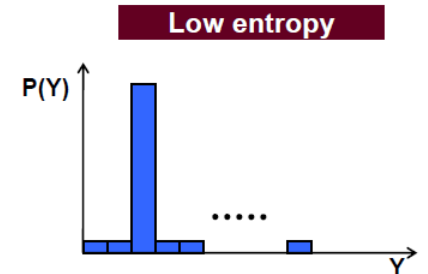
- For a discrete random variable Y , the entropy H is defined as:

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i)$$

where a_i are the possible values of Y .

- The entropy definition can also be generalized to the continuous case and is often called the *differential entropy*. The differential entropy H of a random variable y with density $f(y)$ is defined as:

$$H(y) = -\int f(y) \log f(y) dy$$



Non-Gaussianity Estimation

Measurement of non-Gaussianity

- **Negentropy**

- The **Gaussian** random variable has the **largest entropy** among all random variables of equal variance, which means that entropy can be used to measure nongaussianity.
- To obtain a measure of nongaussianity that is zero for Gaussian random variables and always nonnegative, a slightly modified version of differential entropy is employed, which is called *negentropy*.
- Negentropy J is defined as:

$$J(y) = H(y_{gauss}) - H(y)$$

- The use of negentropy as a measure for nongaussianity is well-justified in information theory but the problem with it lies in it being *computationally difficult to compute*. There are several approximations for entropy in the literature to alleviate this problem.

Non-Gaussianity Estimation

Measurement of non-Gaussianity

- **Approximations of Negentropy**

- The classical method of approximating negentropy is using higher-order moments:

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} kurt(y)^2$$

- The random variable y is assumed to be of zero mean and unit variance. However, the validity of such approximations may be rather limited.
- To avoid the problems encountered with the preceding approximation, new approximations were developed based on the maximum-entropy principle:

$$J(y) \approx \sum_{i=1}^p k_i [E\{G_i(y)\} - E\{G_i(v)\}]^2$$

- where k_i are some positive constants, and v is a Gaussian variable of zero mean and unit variance. The variable y is assumed to be of zero mean and unit variance, and the functions G_i are some nonquadratic functions
- In particular, choosing G that does not grow too fast, one obtains more robust estimators. The following choices of G have proved very useful:

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u \quad , \quad G_2(u) = -\exp(-u^2/2) \quad \text{where } 1 \leq a_1 \leq 2 \text{ is constant}$$

Principles of ICA Estimation

- Two popular methods in estimating the ICA model are,
 1. Minimization of Mutual Information
 2. Maximum Likelihood Estimation

1. Minimization of Mutual Information

- Using the concept of differential entropy, mutual information I between m random variables can be defined as follows,

$$I(y_1, y_2, \dots, y_m) = \sum_{i=1}^m H(y_i) - H(\mathbf{y})$$

- Mutual information is the natural measure of the dependence between random variables. Its value is always nonnegative, and zero if and only if the variables are statistically independent.
- When the original random vector \mathbf{x} undergoes an invertible linear transformation $\mathbf{y} = \mathbf{W}\mathbf{x}$, the mutual information for \mathbf{y} in terms of \mathbf{x} is

$$I(y_1, \dots, y_m) = \sum_i H(y_i) - H(\mathbf{x}) - \log |\det \mathbf{W}|$$

1. Minimization of Mutual Information

- Consider the scenario when y_i is constrained to be uncorrelated and of unit variance, which implies that $E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{W}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{W}^T = \mathbf{I}$. Applying the determinant on all sides of the equation leads to:

$$\det I = 1 = \det(\mathbf{W}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{W}^T) = (\det \mathbf{W})(\det E\{\mathbf{x}\mathbf{x}^T\})(\det \mathbf{W}^T)$$

- Hence $\det \mathbf{W}$ must be constant since $\det E\{\mathbf{x}\mathbf{x}^T\}$ does not depend on \mathbf{W} .
- For \mathbf{y} of unit variance, entropy and negentropy differ only by a constant and sign. Therefore, the fundamental relation between entropy and negentropy is:

$$I(y_1, \dots, y_n) = C - \sum_i J(y_i)$$

where C is a constant not dependent on \mathbf{W} .

- Thus finding an invertible transformation \mathbf{W} that minimizes the mutual information is roughly equivalent to finding directions in which negentropy (a concept related to nongaussianity) is maximized.

2. Maximum Likelihood Estimation

- To derive the likelihood of the noise-free ICA model, a well-known result on the density of a linear transform is used. According to the result, the density p_x of the mixture vector (the ICA model), $\mathbf{x} = \mathbf{A}\mathbf{s}$ is

$$f_x(x) = |\det W| f_s(s) = |\det W| \prod_{i=1}^n f_i(s_i)$$

where $\mathbf{W} = \mathbf{A}^{-1}$, and f_i denote the densities of the independent components \mathbf{s}_i .

- The density p_x can also be expressed as a function of \mathbf{x} and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2 \dots \mathbf{w}_n)^T$, that is,

$$f_x(x) = |\det W| \prod_{i=1}^n f_i(w_i^T x)$$

- Assuming that there are T observations of \mathbf{x} , denoted by $\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(T)$, and after some manipulations, the final equation for the log-likelihood is:

$$L = \sum_{t=1}^T \sum_{i=1}^n \log f_i(w_i^T x(t)) + T \log |\det W|$$

- **Problem:** Density functions f_i must be estimated correctly, otherwise ML estimation will give a wrong result.

Preprocessing for ICA

1. Centering

- The most basic and necessary preprocessing is to center the data matrix \mathbf{X} , that is, subtract the mean vector, $\boldsymbol{\mu} = E(\mathbf{X})$ to make the data a zero-mean variable.
- With this, \mathbf{s} can be considered to be zero-mean, as well.
- After estimating the mixing matrix \mathbf{A} , the mean vector of \mathbf{s} can be added back to the centered estimates of \mathbf{s} to complete the estimation.
- The mean vector of \mathbf{s} is given by $\mathbf{A}^{-1} \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ is the mean vector of the data matrix \mathbf{X} .

Preprocessing for ICA

2. Whitening

- Aside from centering, *whitening* the observed variables is a useful preprocessing step in ICA.
- The observed vector \mathbf{x} is linearly transformed to obtain a vector that is *white*, which means its components are uncorrelated and the variance is equal to unity.

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = I$$

- In terms of covariance, the covariance of the new vector $\tilde{\mathbf{x}}$ equals the identity matrix, i.e.
- One popular method for whitening is to use the eigen-value decomposition (EVD) of the covariance matrix

$$E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{V}\mathbf{D}\mathbf{V}^T$$

where \mathbf{V} is the orthogonal matrix of eigenvectors of $E\{\mathbf{x}\mathbf{x}^T\}$ and \mathbf{D} is the diagonal matrix of its eigenvalues, $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$.

Preprocessing for ICA

2. Whitening

- Whitening can now be done by:

$$\tilde{\mathbf{x}} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T \mathbf{x}$$

where the matrix $\mathbf{D}^{-1/2}$ is computed by a simple component-wise operation as $\mathbf{D}^{-1/2} = \text{diag}(d_1^{-1/2}, \dots, d_n^{-1/2})$.

- Whitening transforms the mixing matrix into a new one,

$$\tilde{\mathbf{x}} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T \mathbf{x} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T \mathbf{A}\mathbf{s} = \tilde{\mathbf{A}}\mathbf{s}$$

- Here we see that whitening reduces the number of parameters to be estimated.
- Instead of having to estimate the n^2 parameters that are the elements of the original matrix \mathbf{A} , we only need to estimate the new, orthogonal mixing matrix $\tilde{\mathbf{A}}$ which contains $n(n-1)/2$ degrees of freedom.
- Thus one can say that whitening solves half of the problem of ICA.
- For simplicity of notation, we denote the preprocessed data just by \mathbf{x} , and the transformed mixing matrix by \mathbf{A} , omitting the *tildes*.

Preprocessing for ICA

2. Whitening

- Because whitening is a very simple and standard procedure, much simpler than any ICA algorithms, it is a good idea to reduce the complexity of the problem this way.
- It may also be quite useful to reduce the dimension of the data at the same time as we do the whitening.
- Then we look at the eigen values d_j of $E\{\mathbf{xx}^T\}$ and discard those that are too small, as is often done in the statistical technique of principal component analysis (PCA).
- This has often the effect of reducing noise. Moreover, dimension reduction prevents over-learning, which can sometimes be observed in ICA.

Preprocessing for ICA

Centering + Whitening = Sphering

- Centering and whitening combined is referred to as *sphering*, and is necessary to speed up the ICA algorithm.
- *Sphering* removes the first and second-order statistics of the data; both the mean and covariance are set to zero and the variance are equalized.

ICA – Example of Algorithms

- In what follows, we will discuss two approaches for estimating independent components given the observed mixture \mathbf{x} :

- ICA gradient ascent:

This algorithm is based on maximizing the entropy of the estimated components, Matlab code will be provided as illustration.

- FastICA

This algorithm is based on minimizing mutual information, you can download the source code from

<http://www.cis.hut.fi/projects/ica/fastica/code/dlcode.html>

ICA Gradient Ascent

- Assume that we have n mixtures $\mathbf{x}_1, \dots, \mathbf{x}_n$ of n independent components/sources $\mathbf{s}_1, \dots, \mathbf{s}_n$:

$$\mathbf{x}_j = \mathbf{a}_{j1}\mathbf{s}_1 + \mathbf{a}_{j2}\mathbf{s}_2 + \dots + \mathbf{a}_{jn}\mathbf{s}_n \quad \text{for all } j$$

- Assume that the sources has a common cumulative density function (cdf) g and probability density function (pdf) p_s .
- Then given an unmixing matrix \mathbf{W} which extracts n components $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_n)^T$ from a set of observed mixtures \mathbf{x} , the entropy of the components $\mathbf{U} = g(\mathbf{u})$ will be, by definition:

$$H(U) = H(x) + E \left\{ \sum_{i=1}^n \ln p_s(u_i) \right\} + \ln |\mathbf{W}|$$

where $\mathbf{u}_i = \mathbf{w}_i^T \mathbf{x}$ is the i th component, which is extracted by the i th row of the unmixing matrix \mathbf{W} . This expected value will be computed using m sample values of the mixtures \mathbf{x} .

ICA Gradient Ascent

- By definition, the pdf p_s of a variable is the derivative of that variable's cdf g :

$$p_s(u_i) = \frac{d}{du_i} g(u_i)$$

Where this derivative is denoted by $g'(\mathbf{u}_i) = p_s(\mathbf{u}_i)$, so that we can write:

$$H(U) = H(x) + E \left\{ \sum_{i=1}^n \ln g'(u_i) \right\} + \ln |\mathbf{W}|$$

- We seek an unmixing \mathbf{W} that maximizes the entropy of \mathbf{U} .
- Since the entropy $H(\mathbf{x})$ of the mixtures \mathbf{x} is unaffected by \mathbf{W} , its contribution to $H(\mathbf{U})$ is constant, and can therefore be ignored.
- Thus we can proceed by finding that matrix \mathbf{W} that maximizes the function:

$$h(U) = E \left\{ \sum_{i=1}^n \ln g'(u_i) \right\} + \ln |\mathbf{W}|$$

Which is the change in entropy associated with the mapping from \mathbf{x} to \mathbf{U} .

ICA Gradient Ascent

$$h(U) = E \left\{ \sum_{i=1}^n \ln g'(u_i) \right\} + \ln |W|$$

- We can find the optimal \mathbf{W}^* using gradient ascent on h by iteratively adjusting \mathbf{W} in order to maximize the function h .
- In order to perform gradient ascent efficiently, we need an expression for the gradient of h with respect to the matrix \mathbf{W} .
- We proceed by finding the partial derivative of h with respect to one scalar element \mathbf{W}_{ij} of \mathbf{W} , where \mathbf{W}_{ij} is the element of the i th row and j th column of \mathbf{W} .
- The weight \mathbf{W}_{ij} determines the proportion of the j th mixture \mathbf{x}_j in the i th extracted component \mathbf{u}_i .

ICA Gradient Ascent

- Given that $\mathbf{u} = \mathbf{W}\mathbf{x}$, and that every component u_i has the same pdf g' .
- The partial derivative of h with respect to the ij th element in \mathbf{W} is:

$$\begin{aligned} \frac{\partial}{\partial W_{ij}} h(\mathbf{U}) &= E \left\{ \sum_{i=1}^n \frac{\partial \ln g'(u_i)}{\partial W_{ij}} \right\} + \frac{\partial \ln |\mathbf{W}|}{\partial W_{ij}} \\ &= E \left\{ \sum_{i=1}^n \frac{1}{g'(u_i)} \frac{\partial g'(u_i)}{\partial W_{ij}} \right\} + [\mathbf{W}^{-T}]_{ij} \quad \text{where } \mathbf{W}^{-T} = (\mathbf{W}^T)^{-1} \\ &\quad \text{and } [\mathbf{W}^{-T}]_{ij} \text{ is the } ij\text{th element of } (\mathbf{W}^T)^{-1} \\ &= E \left\{ \sum_{i=1}^n \frac{1}{g'(u_i)} \frac{\partial g'(u_i)}{\partial u_i} \frac{\partial u_i}{\partial W_{ij}} \right\} + [\mathbf{W}^{-T}]_{ij} \quad \text{using the chain rule} \\ &= E \left\{ \sum_{i=1}^n \frac{1}{g'(u_i)} g''(u_i) \frac{\partial u_i}{\partial W_{ij}} \right\} + [\mathbf{W}^{-T}]_{ij} \\ &\quad \text{where } g''(u_i) \text{ is the second derivative of } g \text{ with respect to } u_i \\ &= E \left\{ \sum_{i=1}^n \frac{g''(u_i)}{g'(u_i)} \frac{\partial W_{ij} x_j}{\partial W_{ij}} \right\} + [\mathbf{W}^{-T}]_{ij} \\ &= E \left\{ \sum_{i=1}^n \frac{g''(u_i)}{g'(u_i)} x_j \right\} + [\mathbf{W}^{-T}]_{ij} \\ &= E \left\{ \sum_{i=1}^n \psi(u_i) x_j \right\} + [\mathbf{W}^{-T}]_{ij} \end{aligned}$$

ICA Gradient Ascent

- If we consider all the element of \mathbf{W} , then we have:

$$\nabla h = \mathbf{W}^{-T} + E\{\psi(u)x^T\}$$

where ∇h is an $n \times n$ Jacobian matrix of derivatives in which the ij th element is $\frac{\partial h}{\partial W_{ij}}$

- Given a finite sample of N observed mixture values of \mathbf{x}^k for $k = 1, 2, \dots, N$ and a putative unmixing matrix \mathbf{W} , the expectation can be estimated as:

$$E\{\psi(u)x^T\} = \frac{1}{N} \sum_{k=1}^N \psi(u^k) [x^k]^T \quad \text{where} \quad u^k = \mathbf{W}x^k$$

- Thus the gradient ascent rule, in its most general form will be:

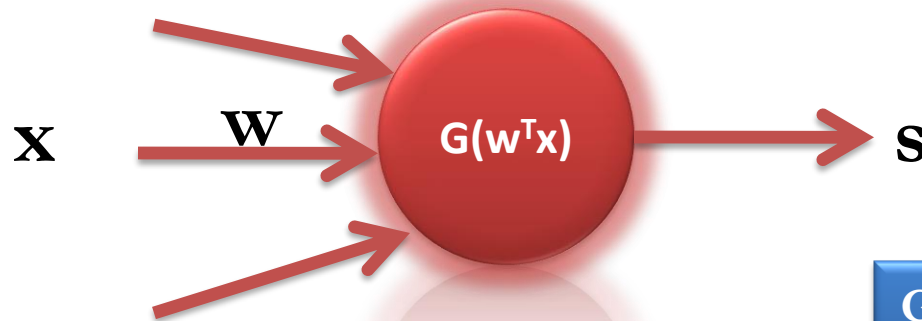
$$\mathbf{W}_{new} = \mathbf{W}_{old} + \eta \nabla h \quad \text{where} \quad \eta \text{ is a small constant}$$

- Thus the rule for updating \mathbf{W} in order to maximize the entropy of $\mathbf{U} = g(\mathbf{u})$ is therefore given by:

$$\mathbf{W}_{new} = \mathbf{W}_{old} + \eta \left(\mathbf{W}^{-T} - \frac{2}{N} \sum_{k=1}^N \tanh(u^k) [x^k]^T \right)$$

FastICA

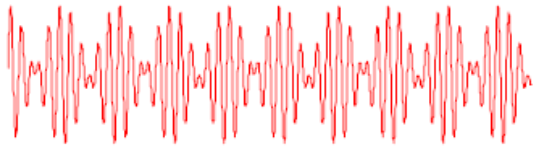
- FastICA is a very efficient method of maximizing the measures of non-Gaussianity mentioned earlier.
- In what follows, we will assume that the data has been centered and whitened.
- FastICA a version of the ICA algorithm that can also be described as a neural network
- Let's look at a single neuron in this network, i.e. we will first start with a single-unit problem, then generalize to multiple units.



Go to implementation

FastICA

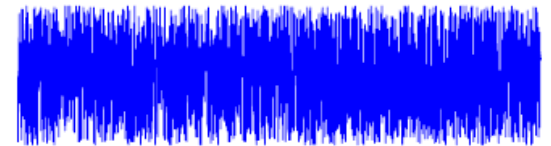
$$s_1(n) = \sin(100n)\cos(10n)$$



$$s_2 = \text{sign}(\sin(10n))$$



$$s_3 = \text{rand}(n)$$



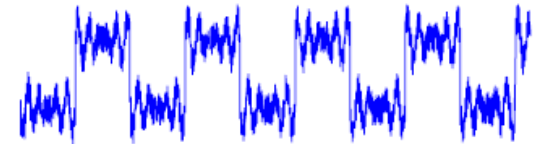
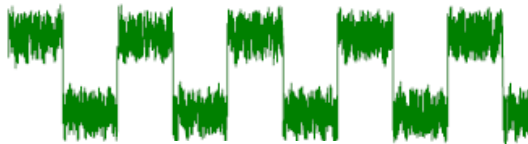
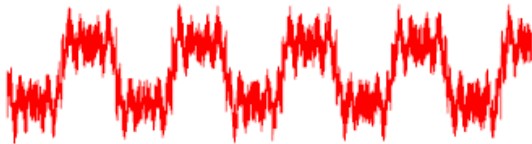
$$x = As$$



x_1

x_2

x_3



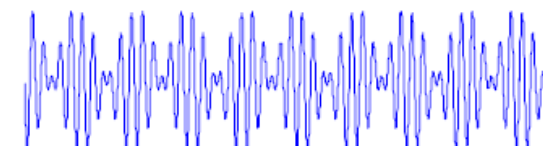
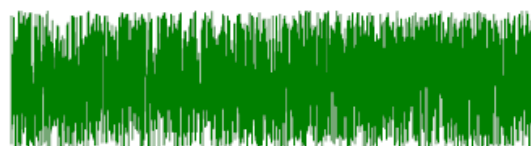
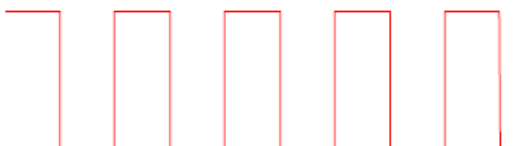
FastICA



y_1

y_2

y_3



FastICA – One Unit

- The goal is to find a weight vector \mathbf{w} that maximizes the *negentropy* estimate:

$$J(\mathbf{w}^T \mathbf{x}) \propto \left(E\{G(\mathbf{w}^T \mathbf{x})\} - E\{G(v)\} \right)^2$$

v is a Gaussian variable of zero mean and unit variance

- Note that the maxima of $J(\mathbf{w}^T \mathbf{x})$ occurs at a certain optima of $E\{G(\mathbf{w}^T \mathbf{x})\}$, since the second part of the estimate is independent of \mathbf{w} .
- According to the Kuhn-Tucker conditions, the optima of $E\{G(\mathbf{w}^T \mathbf{x})\}$ under the constraint $E\{(\mathbf{w}^T \mathbf{x})^2\} = \|\mathbf{w}\|^2 = 1$ occurs at points where:

$$F(\mathbf{w}) = E\{\mathbf{x} g(\mathbf{w}^T \mathbf{x})\} - \beta \mathbf{w} = \mathbf{0}$$


- where $g(u) = dG(u)/du$
- The constraint $E\{(\mathbf{w}^T \mathbf{x})^2\} = \|\mathbf{w}\|^2 = 1$ occurs because the variance of $\mathbf{w}^T \mathbf{x}$ must be equal to unity (by design): if the data is pre-whitened, then the norm of \mathbf{w} must be equal to one
- The problem can be solved as an approximation of *Newton's method*
 - To find a zero of $f(\mathbf{x})$, apply the iteration $\mathbf{x}_{n+1} = \mathbf{x}_n - f(\mathbf{x}_n)/f'(\mathbf{x}_n)$

FastICA – One Unit

- Computing the Jacobian of $F(w)$ yields:

$$JF(w) = \frac{\partial F(w)}{\partial w} = E\{xx^T g'(w^T x)\} - \beta I = 0$$

- To simplify inversion of this matrix, we approximate the first term of the expression by noting that the data is *sphered*;

$$E\{xx^T g'(w^T x)\} \approx E\{xx^T\} E\{g'(w^T x)\} = \underbrace{E\{g'(w^T x)\}}_{\text{scalar}} I$$


So the Jacobian is diagonal, which simplifies the inversion

- Thus, the (approximate) Newton's iteration becomes;

$$w^+ = w - \frac{E\{x g(w^T x)\} - \beta w}{E\{g'(w^T x)\} - \beta}$$

- This algorithm can be further simplified by multiplying both sides by $\beta - E\{g'(w^T x)\}$, which yields the FastICA iteration.

FastICA Iteration

(1) Choose an initial (e.g., random) weight vector \mathbf{w} .

(2) Let

$$\mathbf{w}^+ = E\{x g(\mathbf{w}^T x)\} - E\{g'(\mathbf{w}^T x)\}\mathbf{w}$$

(3) Let

$$\mathbf{w} = \frac{\mathbf{w}^+}{\|\mathbf{w}^+\|}$$

(4) If not converged, go back to (2)

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u \quad , \quad G_2(u) = -\exp(-u^2/2) \quad \text{Note}$$
$$g_1(u) = \frac{dG_1(u)}{du} = \tanh(a_1 u), \quad g_2(u) = \frac{dG_2(u)}{du} = u \exp(-u^2/2)$$

FastICA – Several Units

- To estimate several independent components, we run the one-unit FastICA with several units w_1, w_2, \dots, w_n
- To prevent several of these vectors from converging to the same solution, we decorrelate outputs $w_1^T \mathbf{x}, w_2^T \mathbf{x}, \dots, w_n^T \mathbf{x}$ at each iteration.
- This can be done using a *deflation* scheme based on Gram-Schmidt as follows.

FastICA – Several Units

- We estimate each independent component one by one
- With p estimated components $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_p$, we run the one-unit ICA iteration for \mathbf{w}_{p+1}
- After each iteration, we subtract from \mathbf{w}_{p+1} its projections $(\mathbf{w}_{p+1}^T \mathbf{w}_j) \mathbf{w}_j$ on the previous vectors \mathbf{w}_j
- Then, we renormalize \mathbf{w}_{p+1}

$$(1) \text{ Let } \mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \sum_{j=1}^p \mathbf{w}_{p+1}^T \mathbf{w}_j \mathbf{w}_j$$

$$(2) \text{ Let } \mathbf{w}_{p+1} = \frac{\mathbf{w}_{p+1}}{\sqrt{\mathbf{w}_{p+1}^T \mathbf{w}_{p+1}}}$$

- Or, if all components must be computed simultaneously (to avoid asymmetries), the following iteration proposed by Hyvarinen can be used:

$$(1) \text{ Let } W = \frac{W}{\sqrt{\|WW^T\|}}$$

$$(2) \text{ Repeat until convergence } W = \frac{3}{2}W - \frac{1}{2}WW^TW$$

Let's do it ...

ICA Gradient Ascent

- Dataset generation
- Preprocessing
- Finding the unmixing matrix \mathbf{W}
- Estimated independent components (sources) \mathbf{u} .



Let's do it ...

Dataset Generation



```
% Set random number seed.
seed=9; rand('seed',seed);  randn('seed',seed);

% n = number of source signals and signal mixtures.
n = 2;
% N = number of data points per signal.
N = 1e4;

% Load data, each of n=2 columns contains a different
  source signal.
% Each column has N rows (signal values).

% Load standard matlab sounds
  (from MatLab's datafun directory)
% Set variance of each source to unity.
load ('chirp.mat', 'y');  s1=y(1:N);  s1=s1/std(s1);
load ('gong.mat', 'y');   s2=y(1:N);  s2=s2/std(s2);

% Combine sources into vector variable s.
s=[s1,s2];

% Make new mixing matrix.
A = randn(n,n);

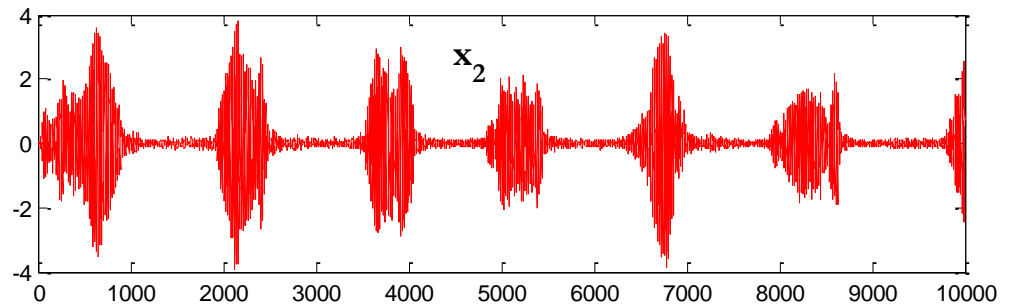
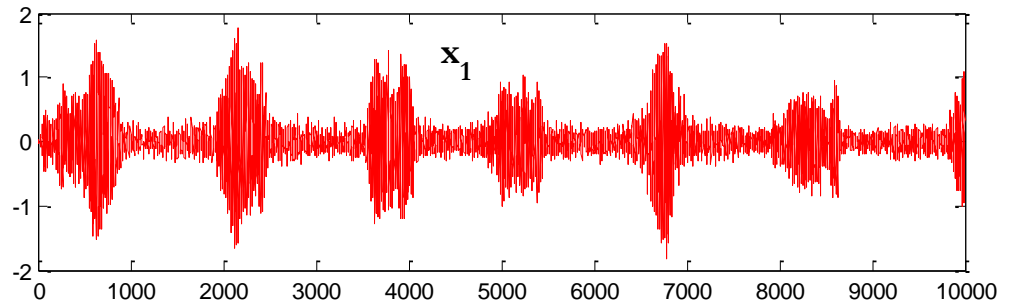
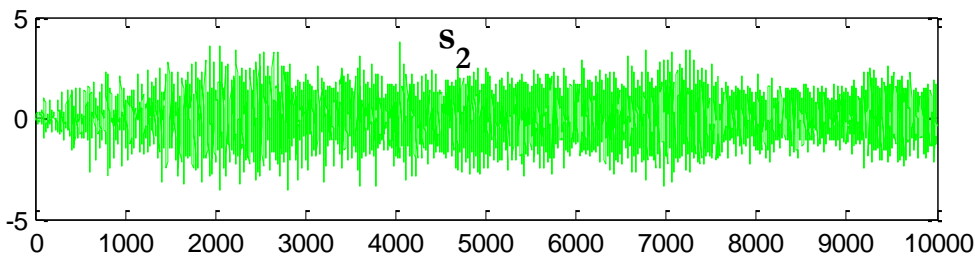
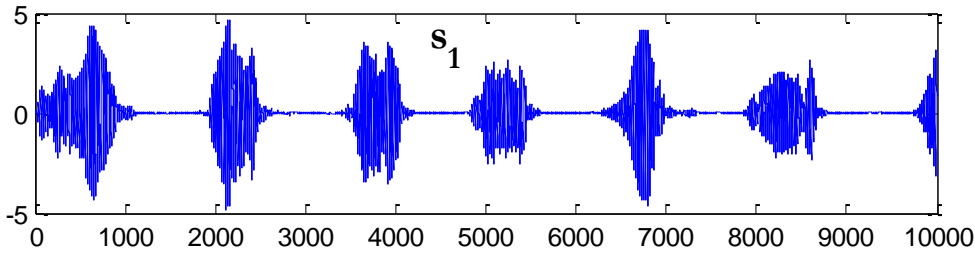
% Make n mixures x from n source signals s.
x = s*A;
```

Two audio signals will be used as sources, then mixed signals will be generated from them



Let's do it ...

Dataset Generation



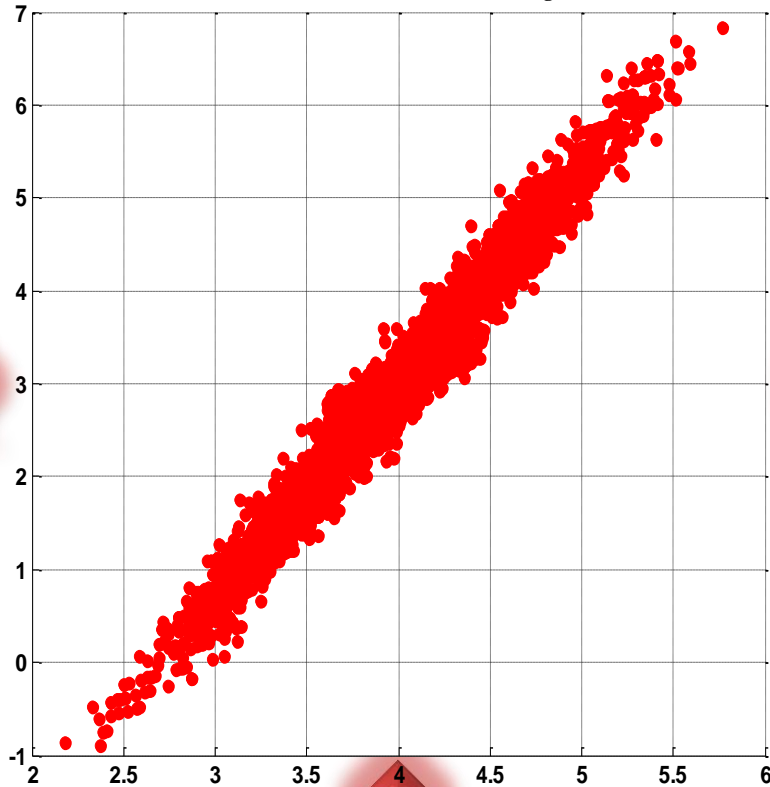
Preprocessing

Let's do it ...

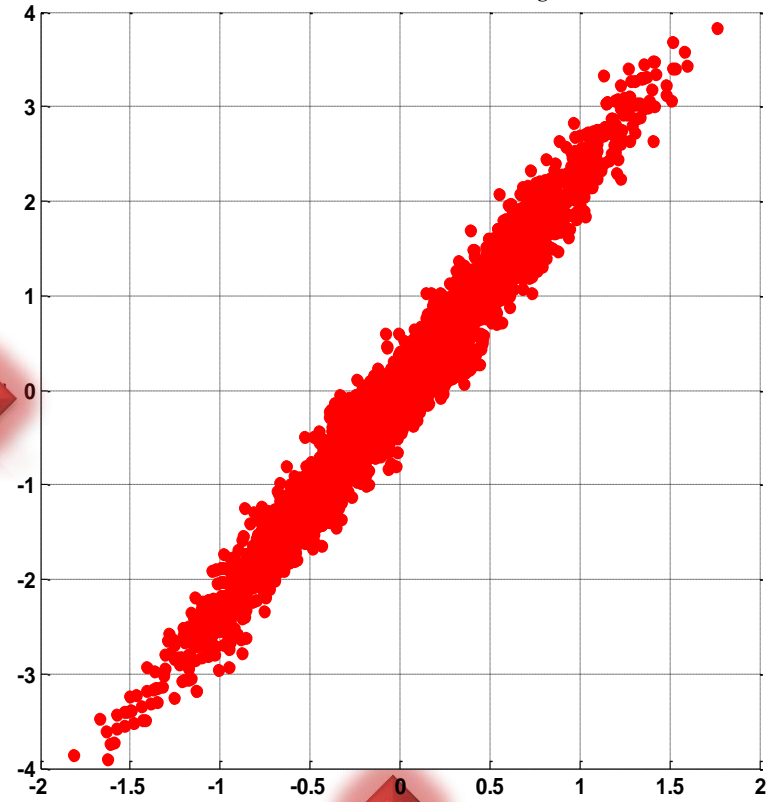


```
% Preprocessing
% 1. Centering
Mu = mean(x) ;
x = x - repmat (Mu , [N, 1]) ;
```

Observed mixture before centering



Observed mixture after centering



Preprocessing

Let's do it ...



```
% Preprocessing
```

```
% 2. Whitening
```

```
Sx = x' * x;
```

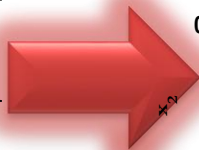
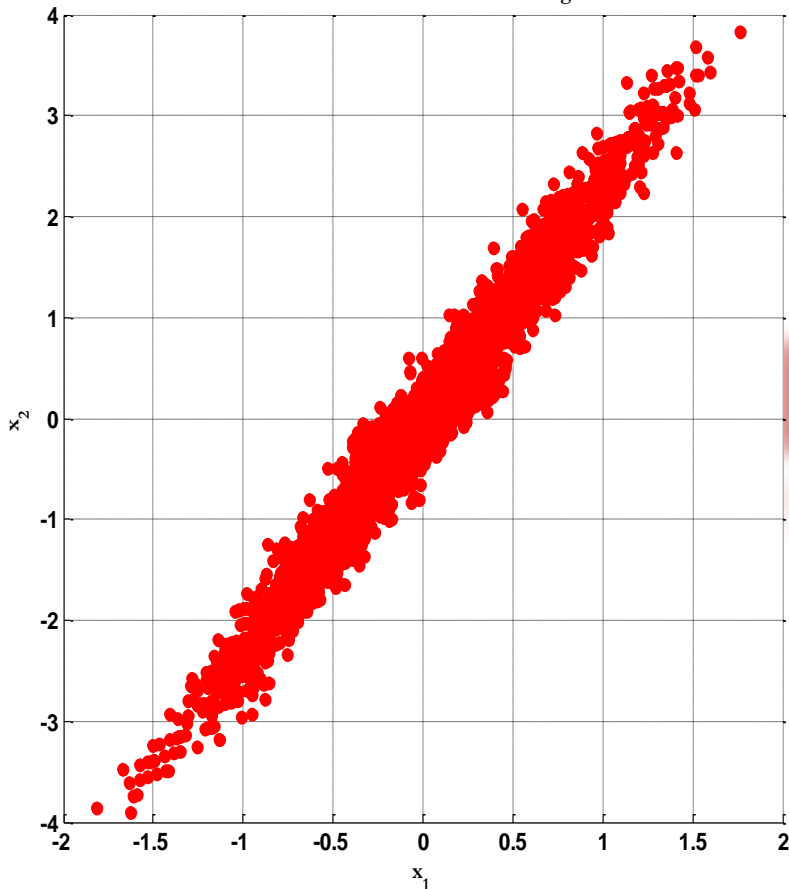
```
[V,D] = eig(Sx);
```

```
x = V*sqrt(inv(D)) * V' * x';
```

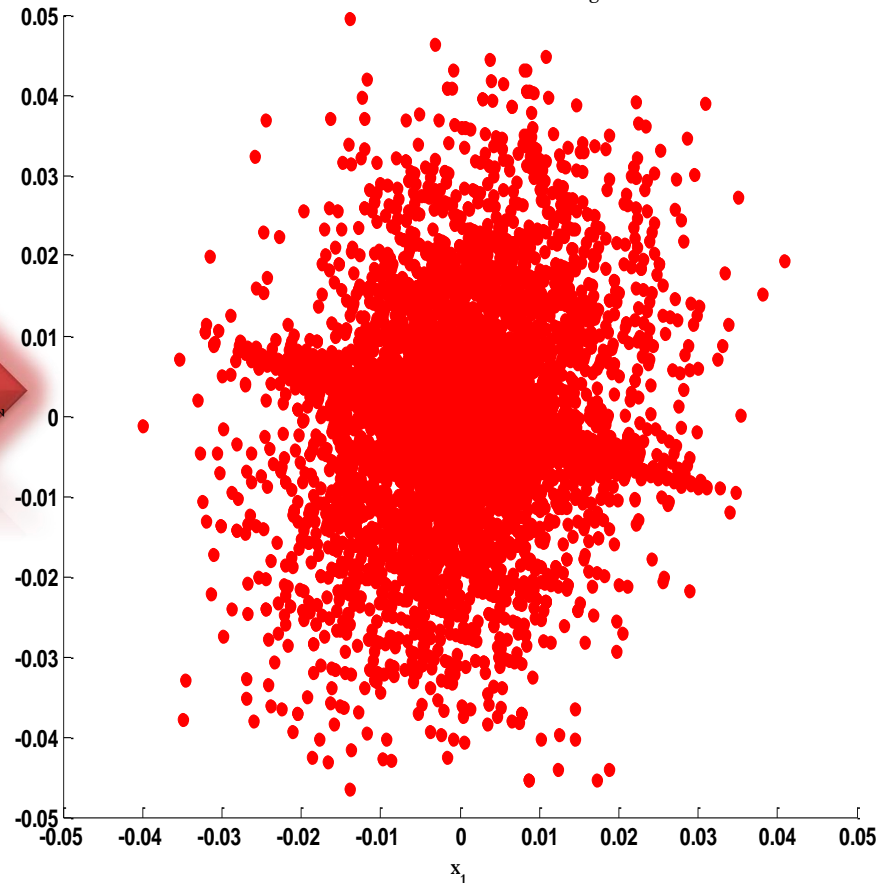
Recall

$$\tilde{x} = VD^{-1/2}V^T x$$

Observed mixture after centering

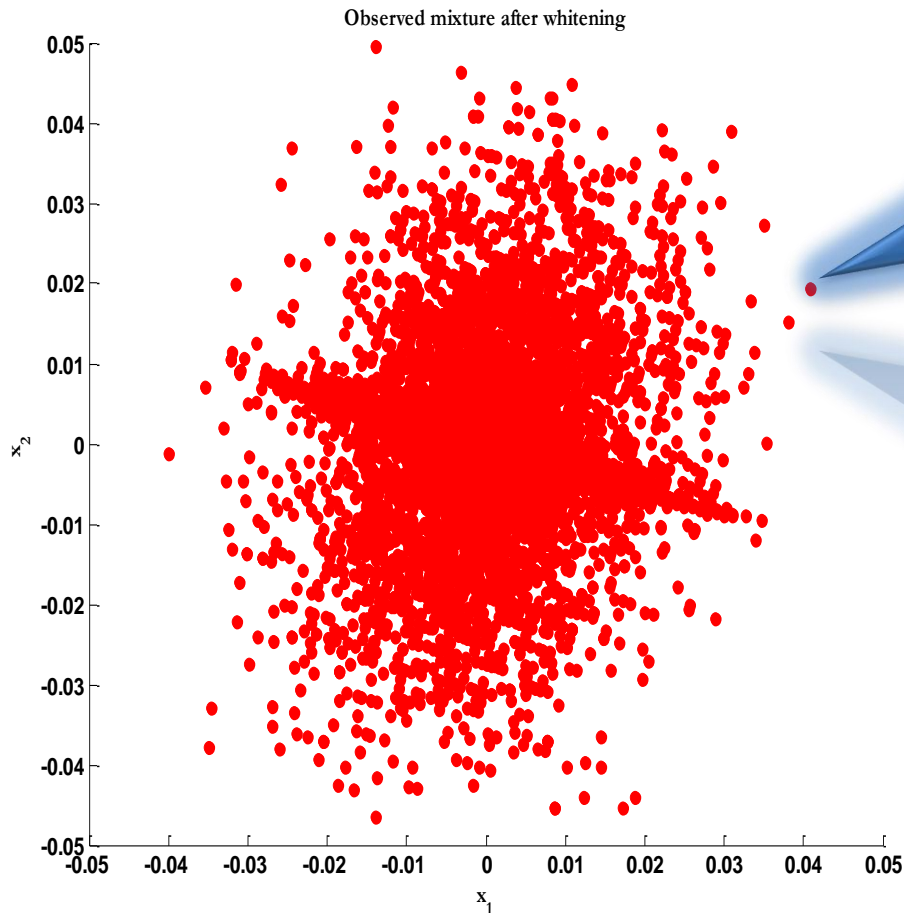


Observed mixture after whitening



Preprocessing

Let's do it ...



Something interesting !!!

The preprocessing step caused increased the gaussianity of the data at hand, hence ICA will **fail** to estimate the independent components.

Estimating W

Initialization

```
% Initialise unmixing matrix W to identity matrix.
W = eye(n,n);

% Initialise u, the estimated source signals.
u = x*W;

% Print out initial correlations between
% each estimated source u and every source signal s.
r = corrcoef([u s]);
fprintf('Initial correlations of source and extracted signals\n');
rinitial = abs(r(n+1:2*n,1:n))

maxiter=10000; % [100] Maximum number of iterations.
eta=1;        % [0.25] Step size for gradient ascent.

% Make array hs to store values of function and gradient magnitude.
hs = zeros(maxiter,1);
gs = zeros(maxiter,1);
```

Estimating W

Begin gradient ascent on $h \dots \text{☺}$

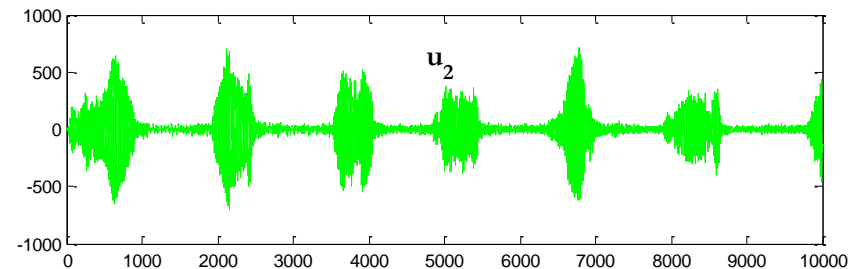
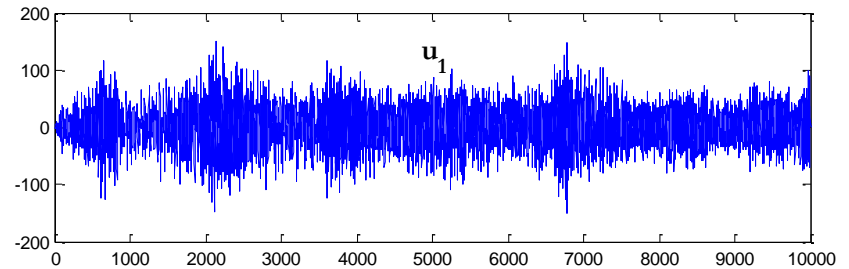
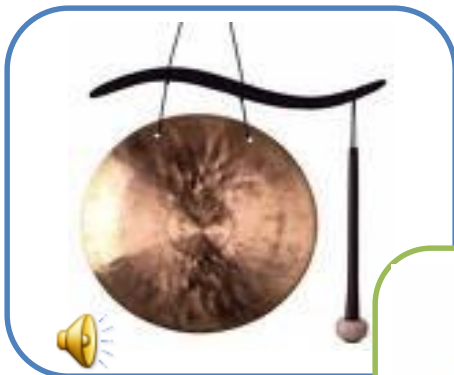
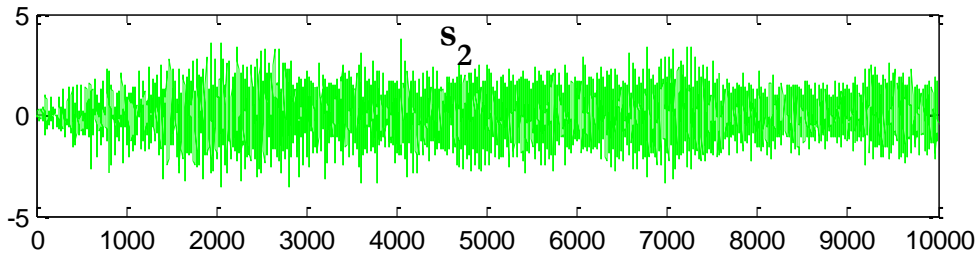
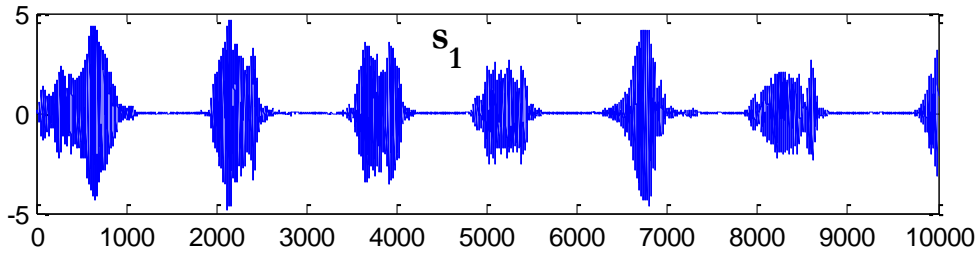
$$W_{new} = W_{old} + \eta \left(W^{-T} - \frac{2}{N} \sum_{t=1}^N \tanh(u^t) [x^t]^T \right)$$

```
for iter=1:maxiter
% Get estimated source signals, u.
u = x*W; % wt vec in col of W.
% Get estimated maximum entropy signals U = cdf(u).
U = tanh(u);
% Find value of function h.
% h = log(abs(det(W))) + sum( log(eps+1-U(:).^2) )/N;
detW = abs(det(W));
h = ( (1/N)*sum(sum(U)) + 0.5*log(detW) );
% Find matrix of gradients @h/@W_ji ...
g = inv(W') - (2/N)*x'*U;
% Update W to increase h ...
W = W + eta*g;
% Record h and magnitude of gradient ...
hs(iter) = h;
gs(iter) = norm(g(:));
end;
```

$$h(U) = E \left\{ \sum_{i=1}^n \ln g'(u_i) \right\} + \ln |W|$$

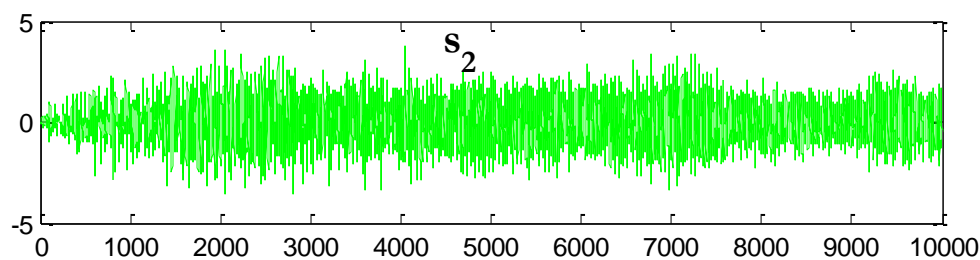
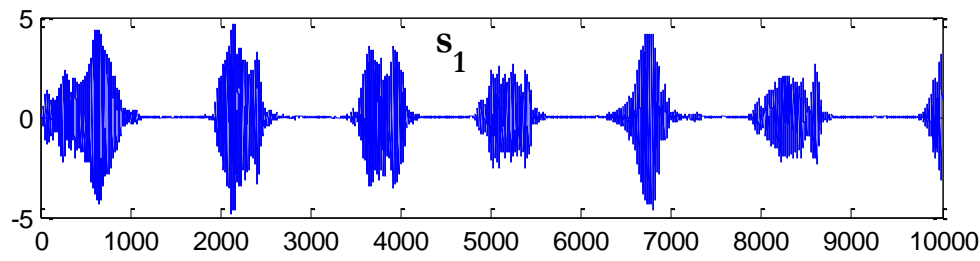
Let's do it ...

Estimated Sources – With Sphering



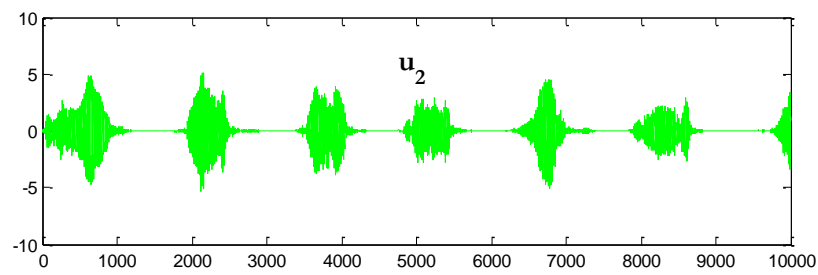
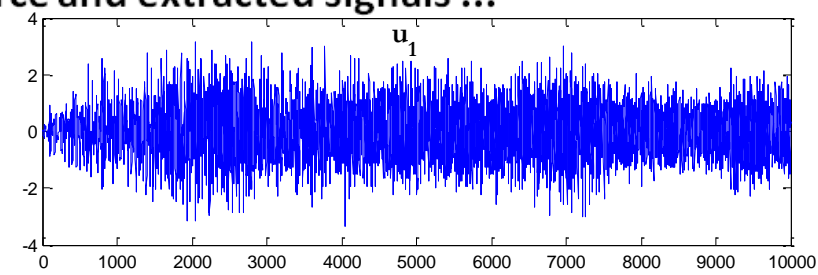
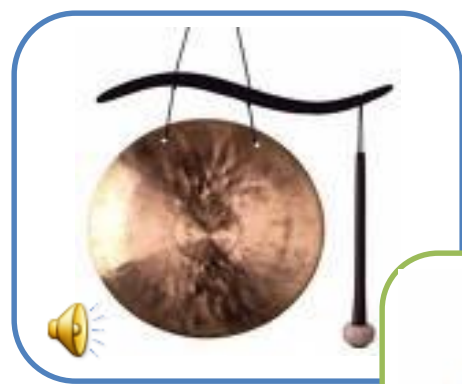


Estimated Sources – Without Sphering



Final correlations between source and extracted signals ...

1.0000	0.0072
0.0073	1.0000

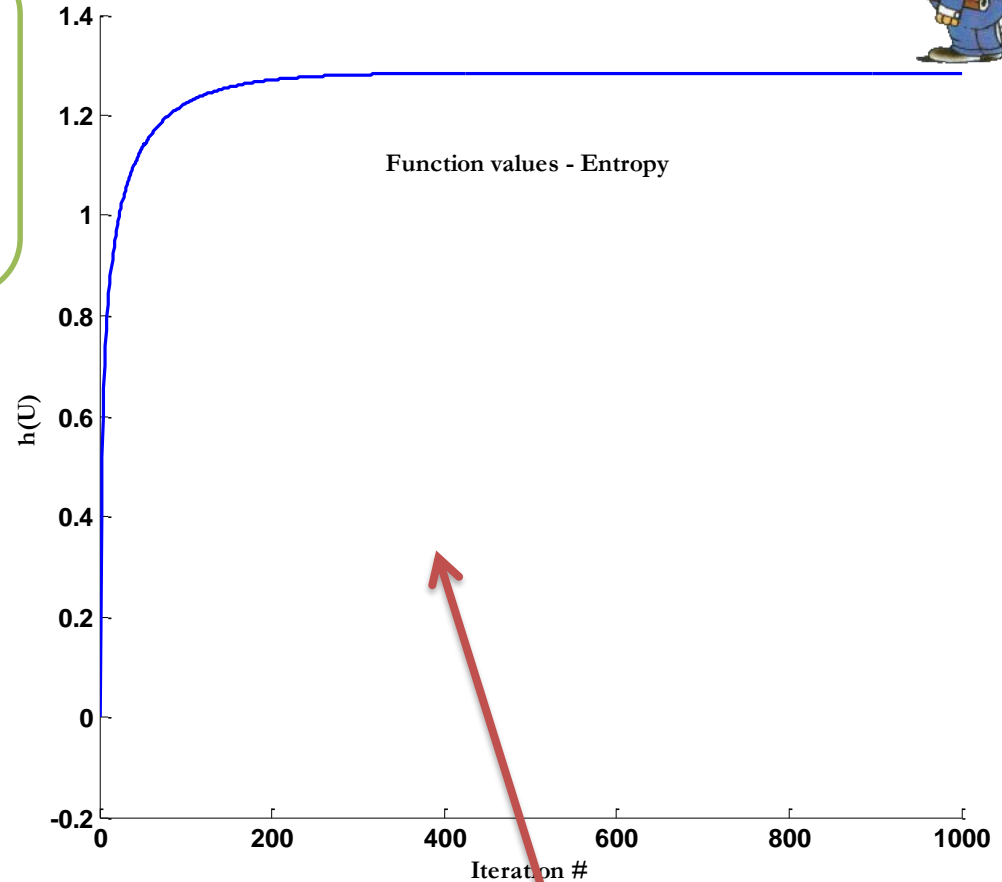
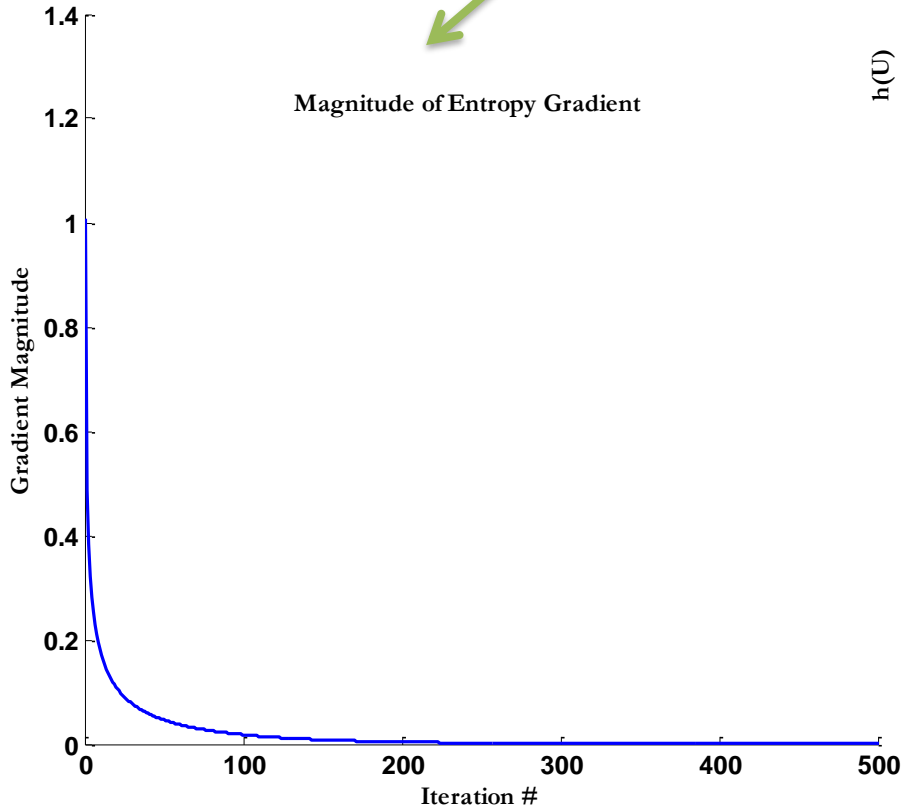


Estimated Sources – Without Sphering

Let's do it ...



Graph of magnitude of gradient of b during gradient ascent. At a maximum in b the gradient magnitude should be zero. As can be seen the gradient magnitude converges towards zero suggesting that a maximum has been reached.



Graph of b during the gradient ascent. This approximates the entropy of the signals $\mathbf{U} = g(\mathbf{u})$, where $\mathbf{u} = \mathbf{W}\mathbf{x}$

Take Home Messages ☺

- ICA relies on the assumption of
 - Statistically Independent underlying signals
 - That are non-Gaussian
 - zero mean and fixed variance
- The algorithm involves
 - minimizing mutual information between signals
 - which leads to maximizing non-gaussinaity
 - which leads to minimizing negentropy
 - which is approximated
 - which results in a NN-like update algorithm

Conclusion

- ICA is used to determine the most independent components in a mixed dataset
- Both mixing matrix and source signal are unknown in the ICA model
- Various estimation techniques are developed to evaluate the independent components in an ICA model, including Non-Gaussianity Estimation, Minimization of Mutual Information and Maximum Likelihood Estimation
- ICA can be used to extract and filter mixed dataset in numerous real life applications, such as separation of artifacts in MEG data and extraction of hidden driving mechanisms in economy.

References

- Independent Component Analysis: A Tutorial Introduction By James V. Stone
Published by MIT Press, 2004 ISBN 0262693151, 9780262693158
- Hyvarinen, A., "Fast and robust fixed-point algorithms for independent component analysis," Neural Networks, IEEE Transactions on , vol.10, no.3, pp.626-634, May 1999
- Aapo Hyvarinen and Erkki Oja, Independent component analysis: A tutorial, April 1999, <http://www.cis.hut.fi/projects/ica/>
- http://research.cs.tamu.edu/prism/lectures/pr/pr_l27.pdf
- [www.stanford.edu/~zhenwei/Presentation/Independent Component Analysis.ppt](http://www.stanford.edu/~zhenwei/Presentation/Independent%20Component%20Analysis.ppt)

Thank You