Independent Component Analysis (ICA)

2021. 11. 09.

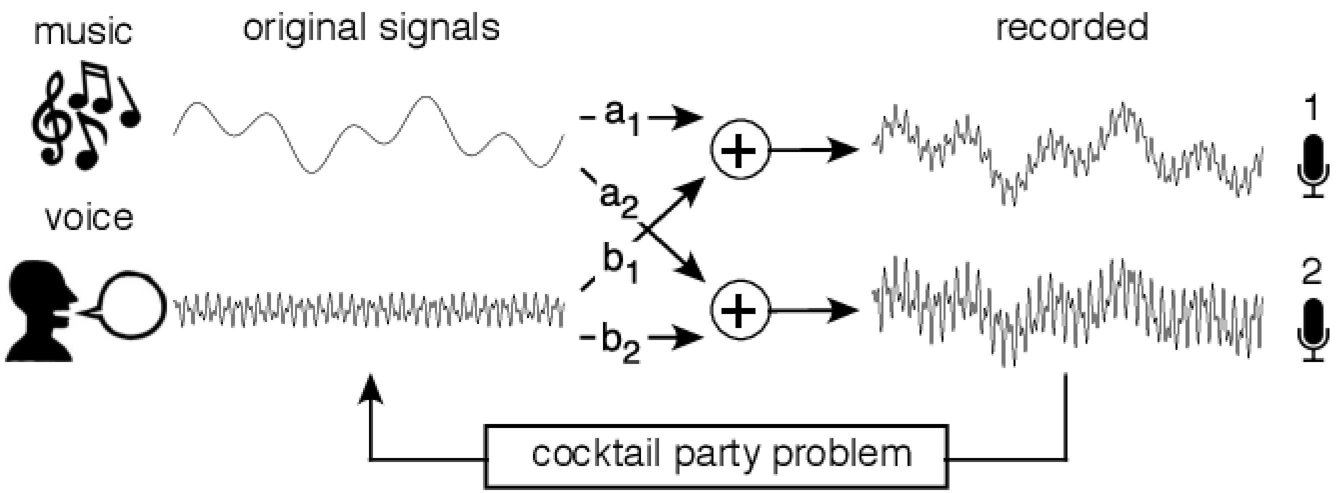
Hyunsoo, Yu

PROGRESS

- Overview
  - What is ICA
  - Compare with PCA
- Implementation
  - Central Limit Theorem
  - Densities and Linear transformation
  - ICA algorithm
    - Bell-Sejnowski algorithm
    - Natural gradient algorithm

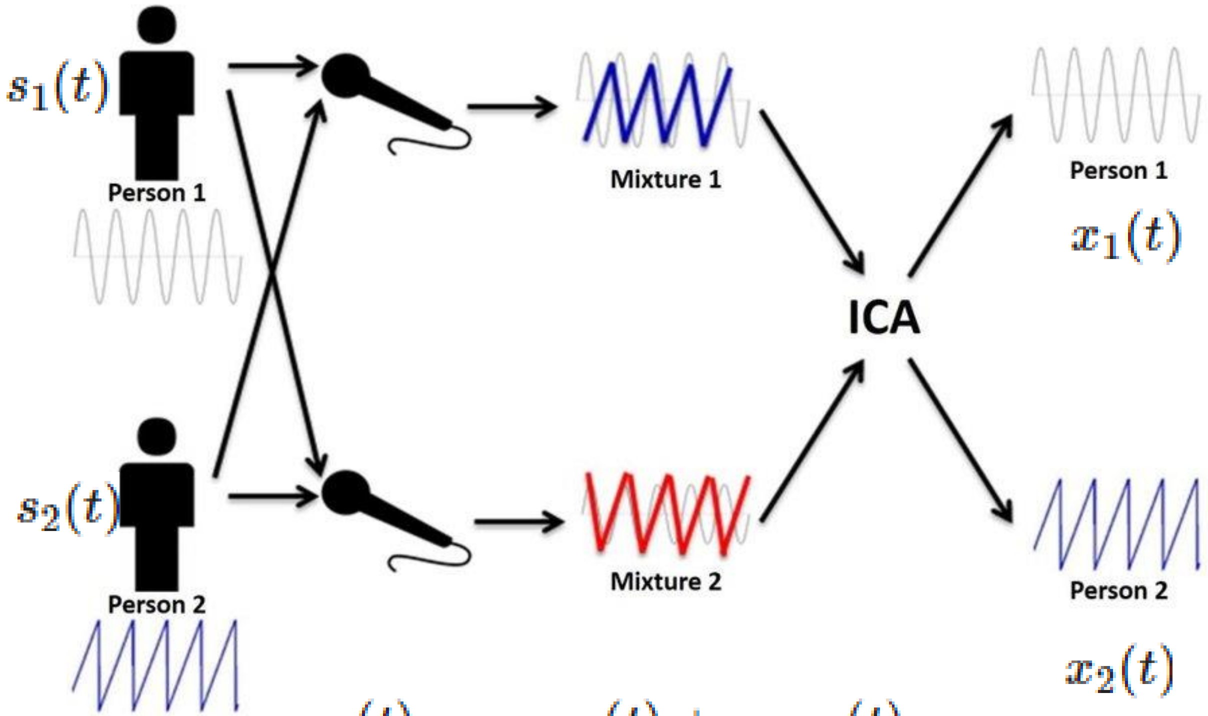
WHAT IS ICA (INDEPENDENT COMPONENT ANALYSIS) - PROBLEM

- Cocktail party problem
  - Two sources are mixed and observed from sensors
  - => Sensors can't identify original signals.



WHAT IS ICA (INDEPENDENT COMPONENT ANALYSIS)
- SOLUTION : ICA

- Independent Component Analysis
  - Transforms a set of vectors into a maximally independent set. (Convert to unmixed set.)
- Number of inputs and outputs are the same.
- Outputs are mutually independent.
- Assumption
  - Independent components are **statistically independent**
  - Independent components are **non-Gaussian**

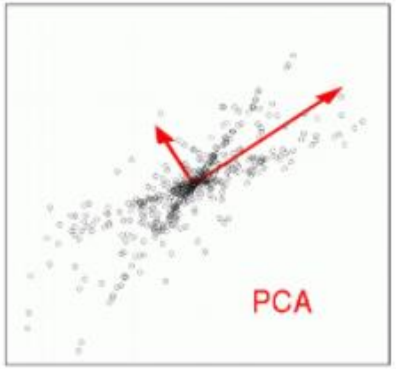


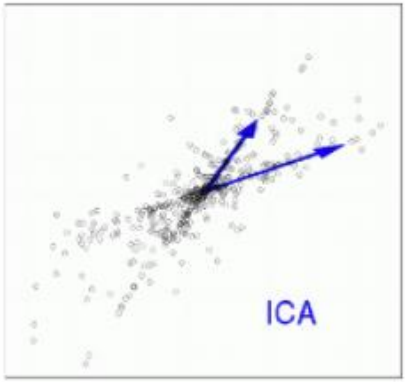
$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$

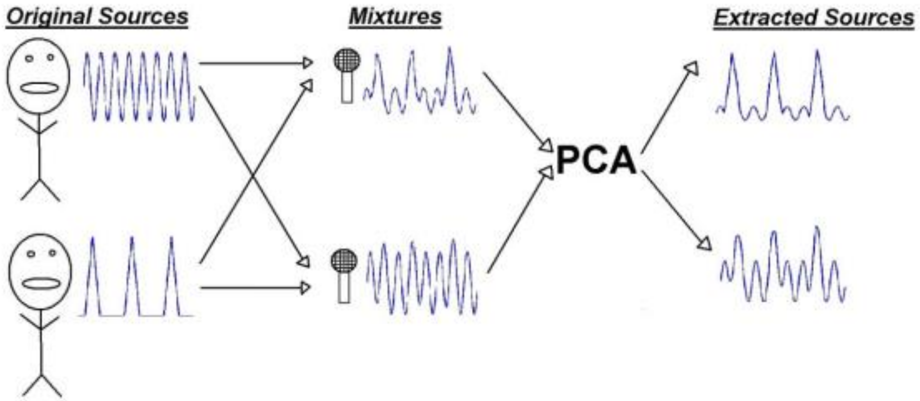
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t)$$

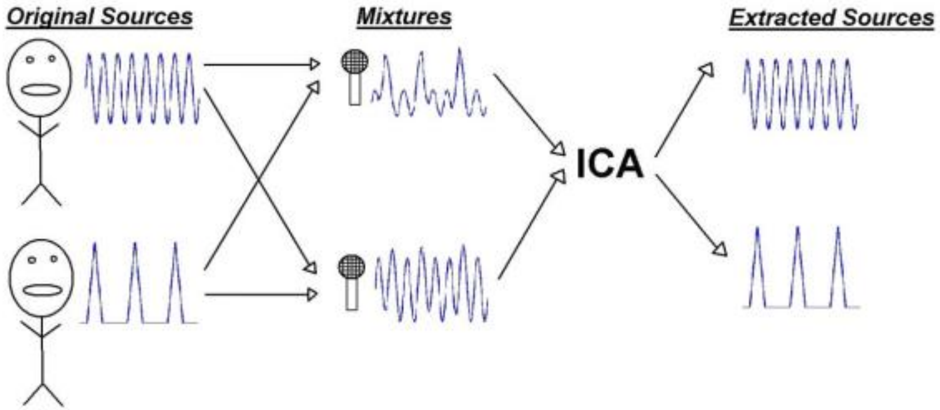
$$x = As$$

COMPARE WITH PCA









- PCA
  - Compress information (dimensionality reduction)
- ICA
  - Separate information
    - Transforming the input space into a maximally independent basis
- Commonality
  - Input data should be auto scaled $z = \frac{x - \mu}{\sigma}$

IMPLEMENTATION

- Goal : Get **W** by updating with **gradient ascent** after computing **likelihood** based on assumption of independence

$$s = A^{-1}x = Wx$$

Source Observation

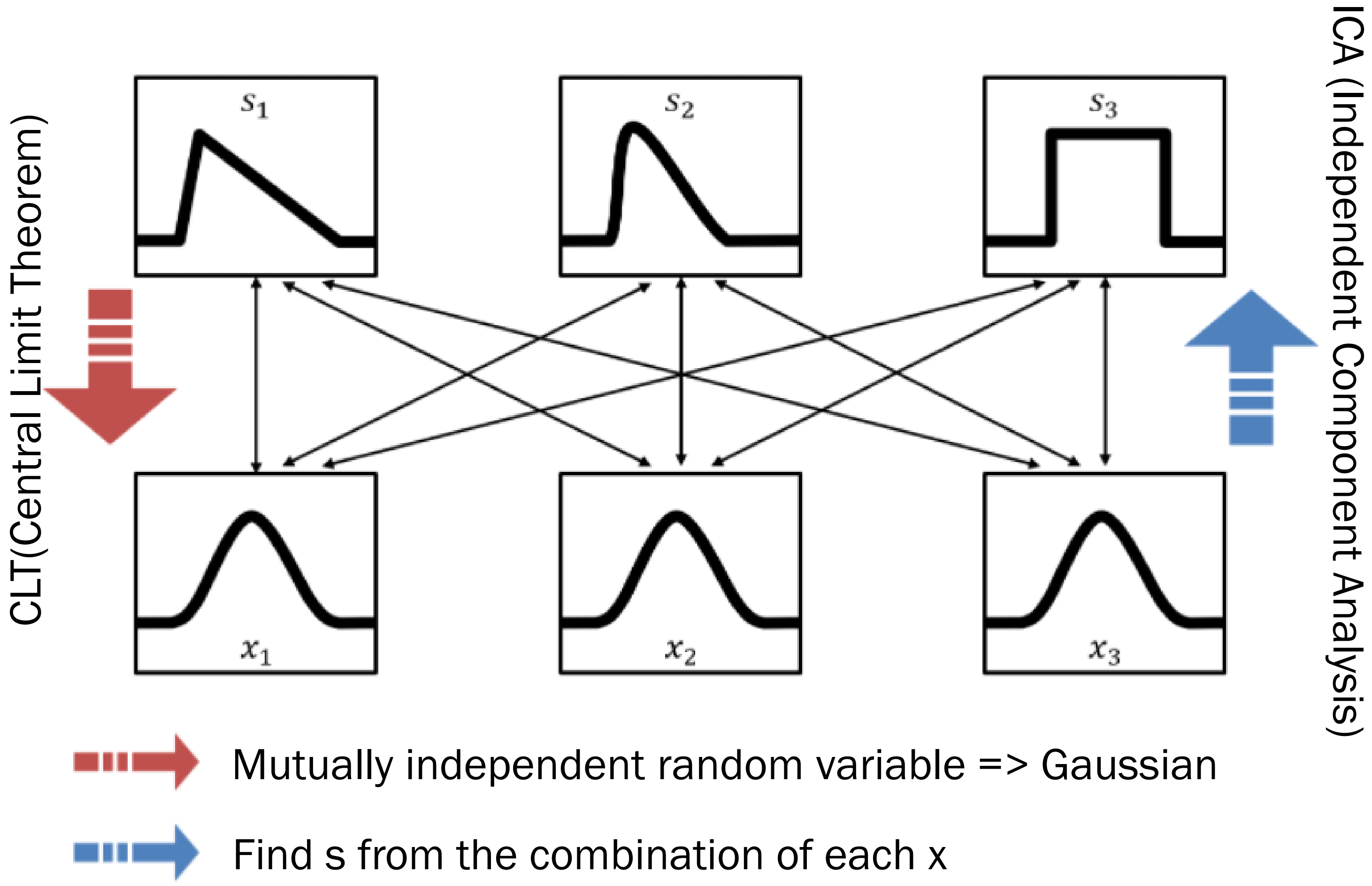
$$W := W + \alpha \frac{\partial l}{\partial W}$$

IMPLEMENTATION
- CENTRAL LIMIT THEOREM

- CLT
  - Linear combination of independent random variables

    => represent **Gaussian**
- ICA
  - Inverse calculation of CLT
    - Find original independent ‘s’ by multiplying ‘W’ to ‘x’
  - Goal : Find proper ‘W’

$$s = A^{-1}x = Wx$$



IMPLEMENTATION
- DENSITIES AND LINEAR TRANSFORMATION

- Probability Density Function of **linear-transformed** random variables
  - After **linear transformation**, the random variable should be adjusted with an inverse matrix.
  - Because the area of PDF is always 1.

About x (Observation)

$$p_x(x) = p_s(A^{-1}x)) \cdot |A^{-1}|$$

$$= p_s(s) \cdot |A^{-1}| = p_s(Wx) \cdot \boxed{|W|}$$

About s (Original source)

$$s \longrightarrow \boxed{A} \longrightarrow x$$

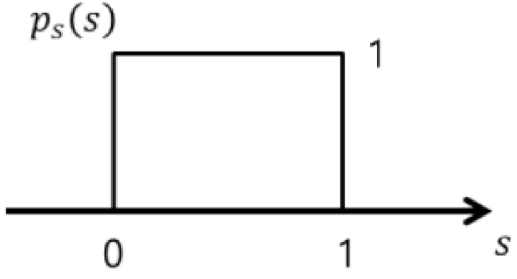
Ex) $A = 2$

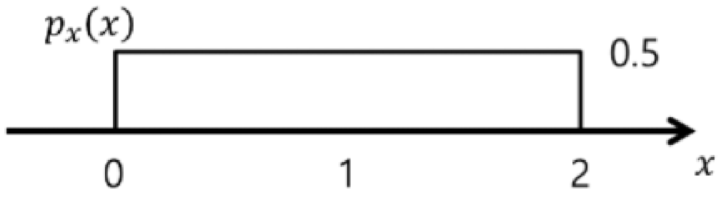
$s: [0, 1]$

$p_s(s) = 1\{0 \le s \le 1\}$

$x: [0, 2]$

$p_x(x) = \boxed{0.5}1\{0 \le x \le 2\}$





ICA ALGORITHM
- MAXIMUM LIKELIHOOD ESTIMATION

- p_s = probability density function of source 's_i'

$$p(x) = \prod_{j=1}^{n} p_s(w_j^T x) \cdot |W|$$

- Apply Maximum Likelihood Estimation
  - Compute Log likelihood about W
  - Find proper W which makes Likelihood maximum
  - **Compute $\frac{\partial l(W)}{\partial W}$ (l(w) : log likelihood)**

$$l(W) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \log p_s(w_j^T x^{(i)}) + log|W| \right)$$

n : # of source
m : # of training sample

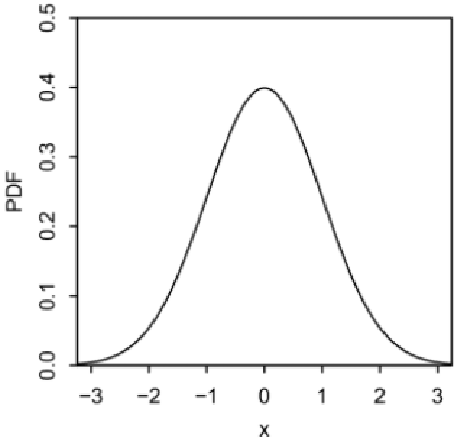
ICA ALGORITHM
- BELL-SEJNOWSKI ALGORITHM

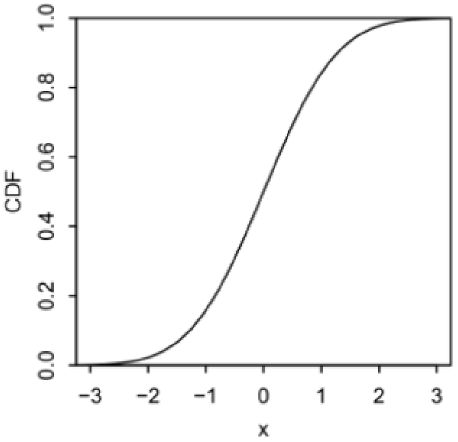
$$l(W) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \log p_s(w_j^T x^{(i)}) + log|W| \right)$$

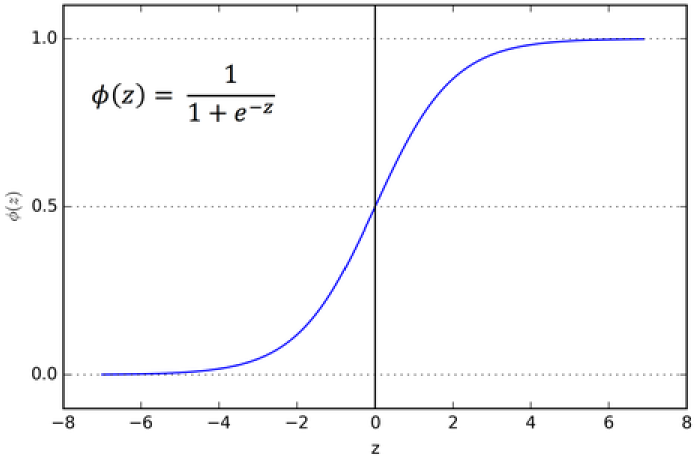
$$l(W) = \sum_{i=m}^{m} \left(\sum_{j=1}^{n} \log g'(w_j^T x^{(i)}) + log|W| \right)$$

- When it comes to p_s, specified CDF(Cumulative Density Function) is required.
  - p_s shouldn't be gaussian distribution (because of CLT)
    - **Sigmoid** function is chosen as a cdf.
    - If you have prior knowledge about the cdf of p_s, you can use that.

$$g(s) = \frac{1}{1+e^{-s}}$$







ICA ALGORITHM
- BELL-SEJNOWSKI ALGORITHM

$$l(W) = \sum_{i=m}^{m} \left(\sum_{j=1}^{n} \log g'(w_j^T x^{(i)}) + log|W| \right)$$

$$\left[w_j^T x^{(i)} = s_j \right]$$

$$\frac{\partial l}{\partial W} = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \frac{1}{g'(s_j)} \cdot g''(s_j) \cdot x^{(i)^T} + \frac{1}{|W|} |W| (W^{-1})^T \right)$$

$$\left[g'(x) = g(x)(1 - g(x)) \right]$$

$$\frac{\partial l}{\partial W} := \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \frac{1}{g(s_j)(1 - g(s_j))} \cdot g(s_j)(1 - g(s_j))(1 - 2g(s_j)) \cdot x^{(i)^T} + (W^{-1})^T \right)$$

$$\frac{\partial l}{\partial W} := \sum_{i=1}^{m} \left(\sum_{j=1}^{n} (1 - 2g(s_j)) \cdot x^{(i)^T} + (W^{-1})^T \right)$$

- Gradient Ascent

$$W := W + \alpha \frac{\partial l}{\partial W}$$

ICA ALGORITHM
- BELL-SEJNOWSKI ALGORITHM

- We can update W step by step

$$W := W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{bmatrix} x^{(i)^T} + (W^T)^{-1} \right)$$

ICA ALGORITHM
- NATURAL GRADIENT ALGORITHM

$$W := W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{bmatrix} x^{(i)^T} + \boxed{(W^T)^{-1}} \right)$$

- Computing an inverse matrix takes a lot of time
  - Multiply $W^T W$

$$W := W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{bmatrix} x^{(i)^T} W^T + I \right) W$$