EE Department, Sharif University of Technology

# Blind Separation of Nonlinear Mixtures of Stochastic Processes

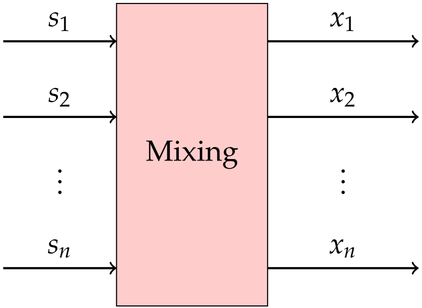
Behrad Moniri
bemoniri@ee.sharif.edu

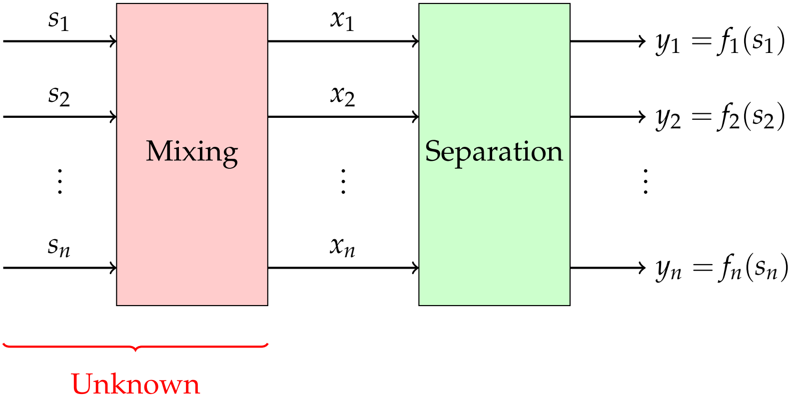
**Advisor**:
Prof. Massoud Babaie-Zadeh

# Outline

1. Blind Source Separation (BSS)
2. Noise-Contrastive Learning
3. Gaussanity-based Methods
4. Another Idea!

# Blind Source Separation (BSS)


$s_1$
$s_2$
$s_n$
Mixing
$x_1$
$x_2$
$x_n$


$s_1$
$s_2$
$s_n$
Mixing
$x_1$
$x_2$
$x_n$
Separation
$y_1 = f_1(s_1)$
$y_2 = f_2(s_2)$
$y_n = f_n(s_n)$
Unknown

**Darmois-Skitovic Theorem** [Darmois-Skitovich 1950]

In the linear setting, the model is identifiable if the sources are **non-Gaussian** random variables.

## Darmois-Skitovic Theorem [Darmois-Skitovich 1950]

In the linear setting, the model is identifiable if the sources are **non-Gaussian** random variables.

## Non-Linear Mixtures

Non-linear mixtures are harder!

$$\begin{cases} S_1 = \text{Rayleigh}(\sigma) \\ S_2 = \text{Uniform}[0, 2\pi] \end{cases} \implies X_1 = S_1 \cos(S_2) \perp\!\!\!\perp X_2 = S_1 \sin(S_2)$$

# Separation of Stochastic Processes (SP)

## Conjecture

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be an invertible smooth mapping and $\mathbf{x}(t) \in \mathbb{R}^n$ be a vector of independent SPs. If $\mathbf{y}(t) = f(\mathbf{x}(t))$ is a vector of independent SPs, then $f$ is Affine.

## Conjecture

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be an invertible smooth mapping and $\mathbf{x}(t) \in \mathbb{R}^n$ be a vector of independent SPs. If $\mathbf{y}(t) = f(\mathbf{x}(t))$ is a vector of independent SPs, then $f$ is Affine.

## Counterexample

- Functions:

$$f\left([s_1, s_2]^\top\right) = \begin{bmatrix} s_1 \\ \mathrm{sign}(s_1 s_2) \end{bmatrix}$$

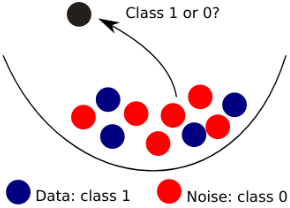
- Stochastic Processes:

$$\begin{cases} s_1[i] = s_1[i-1] + \mathcal{N}(0,1) \\ s_2[i] = s_2[i-1] + \mathcal{N}(0,1) \end{cases}$$

# Noise-Contrastive Estimation (NCE)

Let $\{P_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ be a parametric family of distributions.

- Data: $X_1, X_2, \ldots, X_n \sim P(, ; \boldsymbol{\theta}^*)$
- Noise: $Y_1, Y_2, \ldots, Y_n \sim P_n$




$$\Big\{ \overbrace{(\mathbf{x}_1, 0), (\mathbf{x}_2, 0), \ldots, (\mathbf{x}_n, 0)}^{P(.,\boldsymbol{\theta}^*)}, \overbrace{(\mathbf{y}_1, 1), (\mathbf{y}_2, 1), \ldots, (\mathbf{y}_n, 1)}^{P_n} \Big\}$$

- **Model**: $P(C=1|\mathbf{u},\boldsymbol{\theta}) = \frac{1}{1+G(\mathbf{u},\boldsymbol{\theta})}, \quad G(\mathbf{u},\boldsymbol{\theta}) \geq 0$

- **Loss Function**:

$$J_n^{\text{NCE}}(\boldsymbol{\theta}) = \frac{1}{n}\Big(\sum_{i=1}^{n} \log P(C=1|\mathbf{x}_i;\boldsymbol{\theta}) + \sum_{i=1}^{n} \log P(C=0|\mathbf{y}_i;\boldsymbol{\theta})\Big)$$

- **Learning Algorithm**: $\hat{\boldsymbol{\theta}}_n = \text{argmax}\, J_n^{\text{NCE}}(\boldsymbol{\theta})$

**Consistency** [Gutmann & Hyvarinen, JMLR 2010]

Asymptotically as $n \to \infty$: $G(\mathbf{u}, \hat{\boldsymbol{\theta}}_n) \overset{\text{a.s.}}{\to} \frac{P_n(\mathbf{u},\boldsymbol{\theta}^*)}{P(\mathbf{u})}$

- **Model**: $P(C=1|\mathbf{u},\boldsymbol{\theta}) = \frac{1}{1+G(\mathbf{u},\boldsymbol{\theta})}, \quad G(\mathbf{u},\boldsymbol{\theta}) \geq 0$
- **Loss Function**:

$$J_n^{\mathrm{NCE}}(\boldsymbol{\theta}) = \frac{1}{n}\Big(\sum_{i=1}^{n} \log P(C=1|\mathbf{x}_i;\boldsymbol{\theta}) + \sum_{i=1}^{n} \log P(C=0|\mathbf{y}_i;\boldsymbol{\theta})\Big)$$

- **Learning Algorithm**: $\hat{\boldsymbol{\theta}}_n = \operatorname{argmax} J_n^{\mathrm{NCE}}(\boldsymbol{\theta})$

## Consistency [Gutmann & Hyvarinen, JMLR 2010]

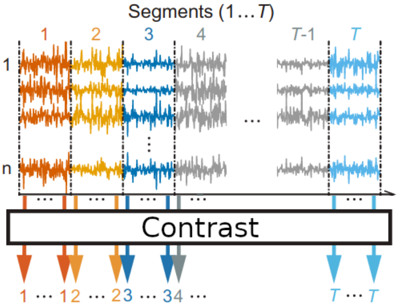
Asymptotically as $n \to \infty$: $G(\mathbf{u},\hat{\boldsymbol{\theta}}_n) \overset{\text{a.s.}}{\to} \frac{P_n(\mathbf{u},\boldsymbol{\theta}^*)}{P(\mathbf{u})}$

## Interesting question

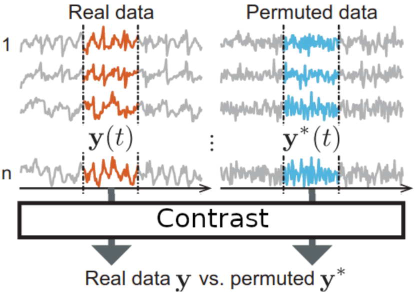
The non-asymptotic behavior of this estimator from a high-dimensional statistics point of view.

Two ideas:

**Time-Contrastive Learning**
[Hyvarinen et al., NIPS 2016]




**Permutation-Contrastive Learning**
[Hyvarinen et al., AISTAT 2017]



## Time Contrastive Learning [Hyvarinen et al., NIPS 2016]

The smooth mixture $\mathbf{x}(t) = \mathbf{f}\big(\mathbf{s}(t)\big)$ is separable if

$$\log p_\tau(s_i) = \lambda_i(\tau)q(s_i) + C$$

plus some technical conditions on $\lambda_i$.

## Generalization

We have generalized the theorem above for

$$\log p_\tau(s_i) = \sum_{v=1}^{V} \lambda_{i,v}(\tau) q_{i,v}(s_i) + C.$$

# Permutation Contrastive Learning

## Permutation Contrastive Separation [Hyvarinen et al., NIPS 2016]

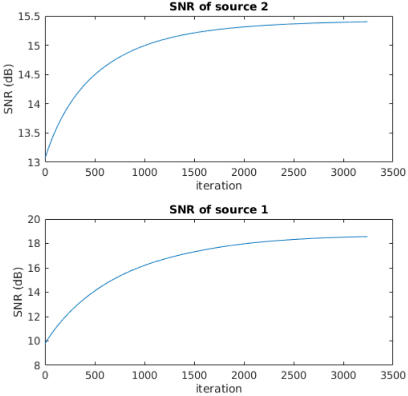
The mixture $\mathbf{x}(t) = \mathbf{f}\big(\mathbf{s}(t)\big)$ is separable if:

- $\mathbf{f}$ is invertible and smooth!
- $s_i(t)$: **stationary**, **ergodic** and ***uniformly dependent***.
- $s_i(t)$ are not ***quasi-Gaussian***.

## Contribution

The proof presented in [Hyvarinen et al., NIPS 2016] is flawed and assumes that the time shuffled SP is independent in time. This error can be fixed by a re-sampling trick.

- Mutual information minimization similar to the method proposed in [Babaie-Zadeh et al., SP 2005].



# Gaussanity-based Methods

**A fundamental question:**
How can we separate Gaussian sources?

**A fundamental question:**
How can we separate Gaussian sources?

- Can nonlinear functions preserve Normality of random variables?

**A fundamental question:**
How can we separate Gaussian sources?

- Can nonlinear functions preserve Normality of random variables? YES!

## Example

Define the function $h$ as follows

$$h(x) = \begin{cases} -x & a \leq |x| < b \\ x & \text{otherwise} \end{cases}$$

If $X$ is a Normal Random variable, then $h(X)$ is also a Normal random variable.

There are also many other examples!

**A fundamental question:**
How can we separate Gaussian sources?

- Can nonlinear functions preserve Normality of random variables? YES!

### Example

Define the function $h$ as follows

$$h(x) = \begin{cases} -x & a \leq |x| < b \\ x & \text{otherwise} \end{cases}$$

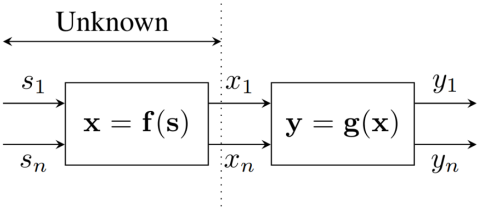
If $X$ is a Normal Random variable, then $h(X)$ is also a Normal random variable.

There are also many other examples!

- How about specific classes of functions?

# Polynomial Mixing Theorem

Only linear polynomials can transform a Gaussian vector to a Gaussian vector.



# Separation Algorithm

- A parametric model for the separating polynomial:

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \dots & \theta_{1s} \\ \theta_{21} & \theta_{22} & \dots & \theta_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1} & \theta_{n2} & \dots & \theta_{ns} \end{bmatrix} \begin{bmatrix} x_1 \\ x_1 x_2 \\ x_1 x_2 x_3 \\ \vdots \\ x_k^p \end{bmatrix} = \boldsymbol{\Theta}\mathbf{k}(\mathbf{x})$$

- Measures of non-Gaussanity:
  - Negative Entropy: $\mathcal{J}_1(y_i) = \boldsymbol{H}(\tilde{y}_i) - \boldsymbol{H}(y_i)$
  - Kolmogrov Distance: $\mathcal{J}_2(x_i) = \sup_x |\Phi(x) - \hat{F}(x)|$
  - Kurtosis: $\mathcal{J}_3(x_i) = \left[\hat{\mathbb{E}}[X^4] - 3(\hat{\mathbb{E}}[X^2])^2\right]^2$
- Optimization problem:

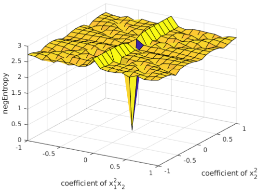
$$\min_{\boldsymbol{\Theta}} \|\mathcal{J}(\boldsymbol{\Theta}\mathbf{k}(\mathbf{x}))\|_2^2$$

# Simulations

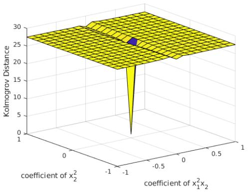
Let $s_1, s_2 \sim \mathcal{N}(0,1)$ and $s_1 \perp\!\!\!\perp s_2$.

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} s_1 + (s_1 + s_2)^3 \\ s_2 - (s_1 + s_2)^3 \end{bmatrix}$$

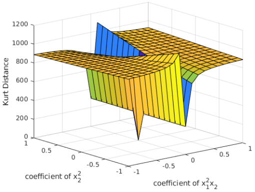
Negative Entropy

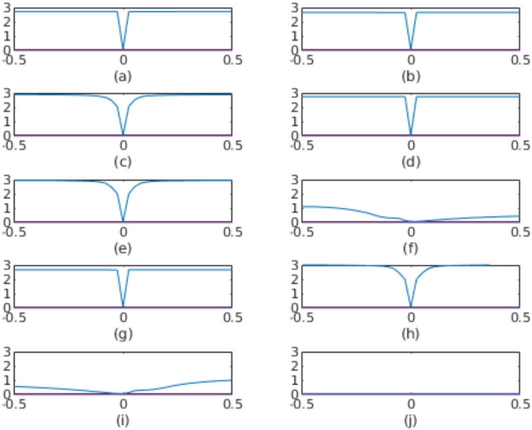


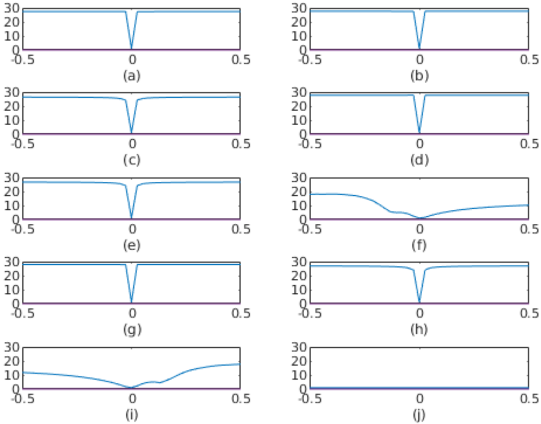

Kolmogrov Distance

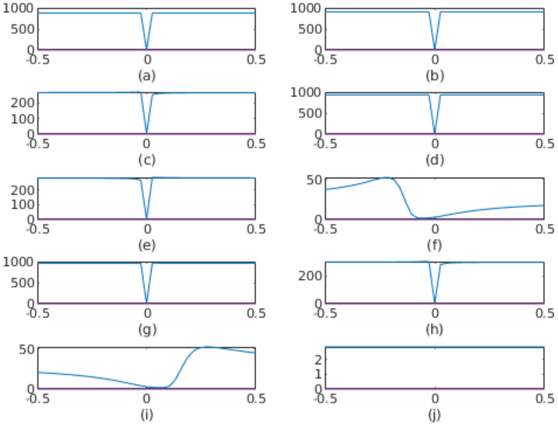



Kurtosis




3
2
1
0
-0.5
0
0.5
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)


30
20
10
0
-0.5
0
0.5
(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)


1000
500
0
-0.5
0
0.5
(a)
1000
500
0
-0.5
0
0.5
(b)
200
100
0
-0.5
0
0.5
(c)
1000
500
0
-0.5
0
0.5
(d)
200
100
0
-0.5
0
0.5
(e)
50
0
-0.5
0
0.5
(f)
1000
500
0
-0.5
0
0.5
(g)
200
0
-0.5
0
0.5
(h)
50
0
-0.5
0
0.5
(i)
2
1
0
-0.5
0
0.5
(j)

# Another Idea!

We have proved the following theorem:

**Monotone functions do not preserve Gaussanity!**

Let $\mathbf{f} = (f_1, f_2, \ldots, f_n)^\top : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous and invertible mixing system and all $f_i$s be monotone functions with respect to all of their inputs. If $\mathbf{f}$ preserves Gaussanity, then $\mathbf{f}$ is Affine.

We have proved the following theorem:

**Monotone functions do not preserve Gaussanity!**

Let $\mathbf{f} = (f_1, f_2, \ldots, f_n)^\top : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous and invertible mixing system and all $f_i$s be monotone functions with respect to all of their inputs. If $\mathbf{f}$ preserves Gaussanity, then $\mathbf{f}$ is Affine.

Connections to BSS:

- Not that obvious. Mixing and Demixing?

We have proved the following theorem:

**Monotone functions do not preserve Gaussanity!**

Let $\mathbf{f} = (f_1, f_2, \ldots, f_n)^\top : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous and invertible mixing system and all $f_i$s be monotone functions with respect to all of their inputs. If $\mathbf{f}$ preserves Gaussanity, then $\mathbf{f}$ is Affine.

Connections to BSS:

- Not that obvious. Mixing and Demixing?
- How about subsets of monotone functions?

# Thank You!