Cocktail Party

- microphone signals are mixed speech signals

\[ x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) \]
\[ x_2(t) = a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) \]
\[ x_3(t) = a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) \]

- **Input:** microphone signals \( x_1, x_2, x_3 \)
- **Goal:** recover the speech signals \( s_1, s_2, s_3 \)

http://research.ics.tkk.fi/ica/cocktail/cocktail_en.cgi
ICA vs. PCA

- **Similar to PCA**
  - Finds a new basis to represent the data

- **Different from PCA**
  - PCA removes only correlations, ICA removes correlations, and higher order dependence.
  - In PCA some components are more important than others (based on eigenvalues) in ICA components are equally important.
ICA vs. PCA

- PCA: principle components are orthogonal.
- ICA: independent components are not!
ICA vs. PCA

maximal variance directions

independent components
Model

- Assume data \( s \in \mathbb{R}^n \), generated by \( n \) independent sources.

- We assume:

\[
x = As,
\]

mixing matrix

\[
A \in \mathbb{R}^{n \times n} \quad \text{is unknown}
\]
Model

- Assume data $s \in \mathbb{R}^n$, generated by $n$ independent sources.

$s_{ij}$ signal from source $j$ at time $i$.

$$x_{ij} = \sum_{k=1}^{n} A_{jk} s_{ik}$$

sum over sources
Problem Definition

- We observe \( \{x_i; \ i = 1, \ldots, m\} \) (\( i \) denotes time)
- Goal: recover the sources \( s_j \), that generated the data \( (x = As) \).

- Let \( W = A^{-1} \) unmixing matrix
- Goal is to find \( W \), such that \( s_i = Wx_i \)
- Denote
  \[
  W = \begin{bmatrix}
  -w_1^T \\
  \vdots \\
  -w_n^T
  \end{bmatrix}
  \]
  then the \( j \)-th source can be recovered by \( s_{ij} = w_j^T x_i \)
ICA Intuition

\[ s_j \in \text{Uniform}[-1,1] \]
ICA Ambiguities

- If we have no prior knowledge about the mixing matrix, then there are inherent ambiguities in $A$ that are impossible to recover.
- The sources can be recovered up to
  - Permutation
  - Scaling
  - Sign
Permutation Ambiguity

Assume that $P$ is a $n \times n$ permutation matrix.

Examples:

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix};$$

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}; \quad P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad PW = \begin{bmatrix} w_{21} & w_{22} \\ w_{11} & w_{12} \end{bmatrix}$$

Given only the $x_i$'s, we cannot distinguish between $W$ and $PW$.

The permutation of the original sources is ambiguous.

Not important in most applications
Scaling Ambiguity

\[ x_i = A s_i \]

\[ A \rightarrow 2, \quad s_i \rightarrow (0.5 s_i) \quad \Rightarrow \quad x_i = 2A(0.5 s_i) \]

\[ A \rightarrow [a_1 \ldots \alpha a_j \ldots], \quad s_j \rightarrow 1/\alpha s_j \quad \Rightarrow \quad x_i = \begin{bmatrix} a_1 \ldots \alpha a_j \ldots \end{bmatrix} \begin{bmatrix} s_{i1} \\ \vdots \end{bmatrix} \]

We cannot recover the “correct” scaling of the sources.

Not important in most applications

Scaling a speaker's speech signal \( s_j \) by some positive factor affects only the volume of that speaker's speech.

Also, sign changes do not matter: \( s_j \) and \( -s_j \) sound identical when played on a speaker.
Gaussian sources are problematic

\[ n = 2, \ s \sim N(0, I), \ x = As \]

\[ x \sim N(0, AA^T) \]

\[ E[xx^T] = E[Ass^TA^T] = AA^T \]

Let \( R \) be an arbitrary orthogonal matrix, such that \( RR^T = R^TR = I \).

Let \( A' = AR \), then \( x' = A's \) \( \Rightarrow \) \( x' \sim N(0, AA^T) \)

\[ E[x'x'^T] = E[A'ss^TA'^T] = E[ARss^T(AR)^T] = ARR^TA = AA^T \]
Gaussian Sources are Problematic

- Whether the mixing matrix is $A$ or $A'$, we would observe data from a $N(0, AA^T)$ distribution.
- Thus, there is no way to tell if the sources were mixed using $A$ or $A'$.
- There is an arbitrary rotational component in the mixing matrix that cannot be determined from the data, and we cannot recover the original sources.
- Reason: The Gaussian distribution is spherically symmetric.
- For non-Gaussian data, it is possible, given enough data, to recover the $n$ independent sources.
Suppose $s$ is a r.v drawn according to $p_s(s)$.

Let $x \in R$ be a r.v. defined by $x = As$. The density of $x$ is given by:

$$p_x(x) = p_s(Wx) \cdot |W|$$

where $W = A^{-1}$ ($A$ is squared invertible matrix)

**Example:** $s \sim \text{Uniform}[0,1] : p_s(s) = 1 (0 \leq s \leq 1)$

Let $A = 2$, then $x = 2s$. Clearly, $x \sim \text{Uniform}[0,2]$

Thus, $p_x(x) = 0.5 (0 \leq x \leq 2)$. 
ICA algorithm

- Assume that the distribution of $s_i$ is $p_s(s_i)$.
- The joint distribution is
  \[ p(s) = \prod_{j=1}^{n} p_s(s_j) \]
  sources are independent

- Using the previous formulation, we can derive
  \[ p(x) = \prod_{j=1}^{n} p_s(w_j^T x)|W| \]
  \[ x = As = W^{-1}s \]
  \[ p(x) = p_s(Wx) \cdot |W| \]

- We must specify a density for the individual sources $p_s$. 
ICA algorithm

- A cumulative distribution of a real r.v. $z$ is defined by
  $$F(z_0) = P(z \leq z_0) = \int_{-\infty}^{z_0} p_z(z) dz$$

- The density of $z$ can be found by $p_z(z) = F'(z)$.

Specify a density for the $s_j$ specify its cdf.

If you have a prior knowledge that the sources' densities take a certain form, then use it here, otherwise make an assumption about cdf.
Density of $s$

cdf is has to be a monotonic function that increases from zero to one.

Gaussian CDF

sigmoid

\[
\begin{align*}
    g(s) &= \frac{1}{1 + e^{-s}} \\
p(s) &= g'(s)
\end{align*}
\]

We assume that the data $x_i$ has zero mean. This is necessary because our assumption that $p(s) = g'(s)$ implies $E(s) = 0$. Thus $E(x) = E(As) = 0$. 
ICA algorithm

- $W$ is a parameter of our model that we want to estimate.
- Given a training set $\{x_i; i = 1, \ldots, m\}$, the log likelihood is:

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

- Maximize $l(W)$ using gradient ascent:

$$W \leftarrow W + \eta \nabla l(W), \text{ where } \eta \text{ is the learning rate.}$$

Equivalently, $w_j \leftarrow w_j + \eta \frac{\partial}{\partial w_j} l(W)$.
ICA algorithm

- By taking the derivatives of $l(W)$ using:
  \[ g(x) = 1/(1 + e^{-x}); \quad g'(x) = g(x)(1 - g(x)) \]
  \[ \nabla_w |W| = |W|(W^{-1})^T \]

we obtain the update rule:

\[
W \leftarrow W + \eta \begin{bmatrix} 1 - 2g(w_1^T x_i) \\ 1 - 2g(w_2^T x_i) \\ \vdots \\ 1 - 2g(w_n^T x_i) \end{bmatrix} x_i^T + (W^T)^{-1}
\]

- When the algorithm converges, compute $s_i = Wx_i$. 
Remarks

- We assumed that \( \{x_i; i = 1, \ldots, m\} \) are independent of each other.
- This assumption is incorrect for time series where the \( x_i \)'s are dependent (e.g. speech data).
- It can be shown, that having correlated training examples will not hurt the performance of the algorithm if we have sufficient data.
- Tip: run stochastic gradient ascent on a randomly shuffled copy of the training set.
Application domains of ICA

- Blind source separation
- Image denoising
- Medical signal processing – fMRI, ECG, EEG
- Modelling of the hippocampus and visual cortex
- Feature extraction, face recognition
- Compression, redundancy reduction
- Watermarking
- Clustering
- Time series analysis (stock market, microarray data)
- Topic extraction
- Econometrics: Finding hidden factors in financial data
ICA Application, Removing Artifacts from EEG

- EEG $\sim$ *Neural cocktail party*
- Severe *contamination* of EEG activity:
  - eye movements
  - blinks
  - muscle
  - heart, ECG artifact
  - vessel pulse
  - electrode noise
  - line noise, alternating current (60 Hz)

- ICA can improve signal
  - effectively *detect, separate and remove* activity in EEG records from a wide variety of artifactual sources.  
  (Jung, Makeig, Bell, and Sejnowski)

- ICA weights help find *location* of sources
ICA decomposition

EEG Scalp Channels

VEOG
F3
FC5
Cz
Pz

unmixing (W)

Independent Comp

Fig. from Jung
Summed Projection of Selected Components

C1
C2
C3
C4

mixing $W^{-1}$

Artifact-corrected EEG

Fig from Jung
Original EEG

Corrected EEG

Fig from Jung
ICA basis vectors extracted from natural images

Gabor wavelets, edge detection, receptive fields of V1 cells...
Image denoising

Original image

Noisy image

Wiener filtering

ICA filtering