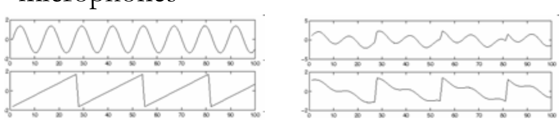
# Independent Component Analysis

Information Theory Lecture 8a

## And now for something completely different…

- Thus far, we've focused on taking a single signal, encoding it, and then decoding it
- Now we are going to concentrate on splitting apart two of more signals that have been combined
- We are going to do this while making some quite robust assumptions about the signals involved…

## Let's say we have two simultaneous signals that have been recorded by two microphones



The signals (we don't have these) The recordings we have

$$x_1(t) = a_{11}s_1 + a_{12}s_2$$
$$x_2(t) = a_{21}s_1 + a_{22}s_2$$
$$\mathbf{x}(t) = \mathbf{As}$$

## We call the *s* values "latent variables"

- We're going to assume that the latent variables are *statistically independent*
- It means that information about one of the *s* values gives you no information about another *s* value
- This is a stronger property than being uncorrelated

## Statistical Independence

$$P(X,Y) = P(X)P(Y)$$
$$E\{f_1(x)f_2(y)\} = \int\int f_1(x)f_2(y)p(x,y)dxdy$$
$$= \int\int f_1(x)p(x)f_2(y)p(y)dxdy$$
$$= \int f_1(x)p(x)dx \int f_2(y)p(y)dx$$
$$= E\{f_1(x)\}E\{f_2(y)\}$$

This means any statistics we gather about the joint variables we could have just gathered about the separate variables

Or, seen another way, statistics about *x* tell us nothing about *y*, and vice versa

## Uncorrelated does not mean independent

- We say two variables are *uncorrelated* if their *covariance* is zero

$$C(x,y) = E\{xy\} - E\{x\}E\{y\} = 0$$

- Consider the uncorrelated samples (0,1),(0,-1),(1,0),(1,-1)

$$E\{xy\} - E\{x\}E\{y\} = 0$$
$$E\{x^2y^2\} - E\{x^2\}E\{y^2\} = -\tfrac{1}{4}$$

- The variables are uncorrelated, but not statistically independent

## Limitations of ICA

- Since both *s* and **A** are unknown, we absolutely cannot determine the variances of the *s* values
- These are only defined up to a multiplier
- We'll assume the variances are all one
- We also can't determine which signal came from which microphone

## In the ICA algorithm

- We are going to assume that the variables are zero mean
- And we've assumed the variances are all one
- So, if our signals were gaussian, we'd have nothing to work with
- So, we assume that the *s* values are independent, and *non-gaussian*

## The opposite of what we usually do

- The ICA approach is based on *minimizing* the mutual information between the *s* values
- A fact that helps here
  - For a given variance, a Gaussian variable has the *maximum* entropy of all possible distributions
- So, our requirement here is like maximizing the sum of the departure of the *H(s)* values from Gaussinaity

$$I(\mathbf{s}) = I\left(\mathbf{A}^{-1}\mathbf{x}\right) = \sum_{j=1}^{N} H(s) - H(\mathbf{s})$$

$$= \sum_{j=1}^{N} H(s) - H(\mathbf{x}) - \log\left|\det \mathbf{A}\right|$$

$$= \sum_{j=1}^{N} H(s) - H(\mathbf{x})$$

## The *Negentropy*

- Is defined as

$$J(s) = H(z) - H(s)$$

- Where *z* is a Gaussian random variable with the same variance as *s*
- So, this is the quantity we want to maximize for ICA
- But we have to approximate it…

## A good statistical approximation

- Of negentropy

$$J(s) = \left[E\left\{G(s)\right\} - E\left\{G(z)\right\}\right]^2$$

- Where *G* is any non-quadratic
- A well-conditioned choice is

$$G(s) = \log\cosh(s)$$

## Preprocessing

- There are a few things we should do to the **x** data before we apply ICA
  - Centering: we subtract the mean from the data to give a new **x** that is zero mean
  - Whitening: we apply a *linear* transformation to give a new **x** that is uncorrelated and has variance of one
    - Whitening makes sure that **A** is orthogonal

## Whitening

- Our new data will have the property

$$E\left\{\hat{\mathbf{x}}\hat{\mathbf{x}}^{\mathbf{T}}\right\} = \mathbf{I}$$

- We can find the appropriate values through eigenvalue decomposition

$$E\left\{\mathbf{x}\mathbf{x}^{\mathbf{T}}\right\} = \mathbf{E}\Lambda\mathbf{E}^{\mathbf{T}}$$

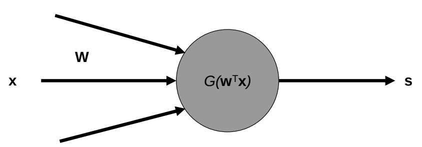
$$\hat{\mathbf{x}} = \mathbf{E}\Lambda^{-1/2}\mathbf{E}^{\mathbf{T}}\mathbf{x}$$

## A side benefit

- At the whitening stage, we could discard components of the whitened **x** that correspond to low eigenvalues
- This is very similar to what is done in *principle component analysis*, a data compression scheme

## FastICA

- Is 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




## As in neural networks

- We are going to update weights to take downhill steps in error
- In this case, the steps are in the (the negative of) negentropy (uphill is better)
- We need the derivative of our *G* function with respect to it's argument

$$G'(s) = \tanh(s)$$

## FastICA for one neuron

- Set the weight vector to random values
- Until convergence:

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

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

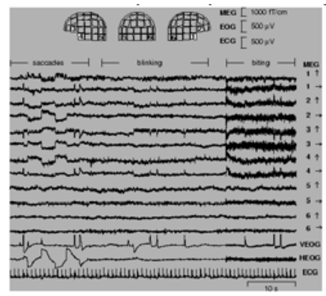
## For several neurons (several signals *s*)

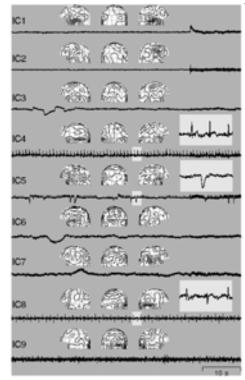
- We can do the same algorithm as before on each neuron
- We have to make sure that all the neurons don't go to the same weight vector (signal)
- We must de-correlate after each update
- One method: $\mathbf{W} = \mathbf{W} / \sqrt{\left\|\mathbf{W}\mathbf{W}^{\mathbf{T}}\right\|}$
- Repeat to convergence: $\mathbf{W} = \frac{3}{2}\mathbf{W} - \frac{1}{2}\mathbf{W}\mathbf{W}^{\mathbf{T}}\mathbf{W}$

## Example: Magnetoencephalography (MEG)

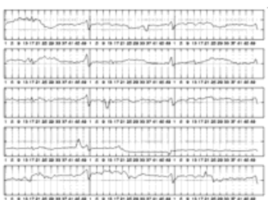
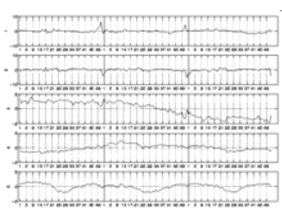
- A noninvasive technique for monitoring brain activity, via sensors on the scalp
- Problem: signals include muscle twitches, blinking, eye movement, heartbeat
- This was simulated by telling a patient to *saccade* eyes, then blink, then bite

## Results

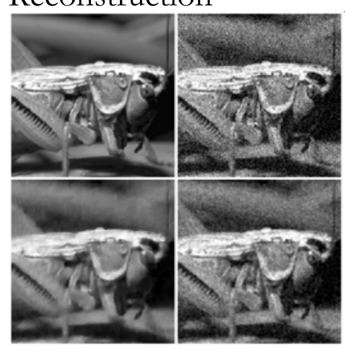




## Example: Cash Flow in Chain Stores

## Image Reconstruction




## 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