

AN INVESTIGATION OF THE USE  
OF AN ICA ALGORITHM IN THE  
ANALYSIS OF fMRI DATA

Cliona Golden

Program in Applied and Computational Mathematics  
Princeton University

# What is Independent Component Analysis (ICA)?

- ICA is a generalisation of Principal Component Analysis (PCA) and was developed to deal with the problem of blind source separation. ICA aims to separate statistically independent signals which have been mixed.

- Consider

$$\mathbf{x} = \mathbf{M}\mathbf{c},$$

where  $\mathbf{x} \in \mathbb{R}^n$  is the random vector of measured signals,  $\mathbf{c} \in \mathbb{R}^m$  is the random vector of source signals and  $\mathbf{M} \in \mathbb{R}^{n \times m}$  is a deterministic mixing matrix (unknown).

Assume the components of  $\mathbf{c}$  are statistically independent. Assume also that  $m = n$ , and  $\mathbf{M}$  is invertible.

Then, given a set of measurements  $\{\mathbf{x}_k\}$ , ICA aims to find  $\mathbf{W} \in \mathbb{R}^{n \times m}$ , such that

$$\mathbf{W}\mathbf{M} = \mathbf{A}\mathbf{P},$$

where  $\mathbf{A}$  is an invertible, diagonal scaling matrix and  $\mathbf{P}$  is a permutation matrix (i.e.  $\mathbf{W}$  is the inverse of  $\mathbf{M}$  up to scaling and permutation).

- Compare this with PCA: Given  $\{\mathbf{x}_k\}$ , find  $\mathbf{F}$  such that

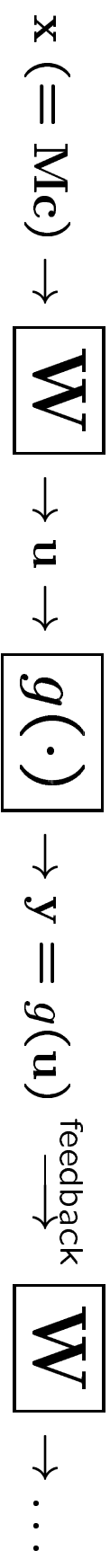
$$\mathbf{u} = \mathbf{F}^T \mathbf{x}$$

has unit variance decorrelated components.

In fact, if  $\mathbf{F}$  is a solution of PCA, then there is an orthogonal  $\mathbf{Q}$  such that  $\mathbf{W} = \mathbf{Q}\mathbf{F}$  solves the ICA problem.

# Bell-Sejnowski Approach to Solving ICA

This is an iterative neural network learning algorithm which is based on the **infomax principle**. The algorithm works as follows:



where  $g(\mathbf{u}) = (g(u_1), \dots, g(u_n))$ , and  $g$  is the sigmoidal function defined by

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

Note the relationship between entropy and mutual information:

$$H(\mathbf{y}) = \sum_{i=1}^n H(y_i) - I(\mathbf{y}).$$

This points towards the correspondence:

maximising joint entropy  $\leftrightarrow$  minimising mutual information.

In fact if  $g'_j(\mathbf{c}) = p_{c_j}(\mathbf{c})$ , then maximisation of joint entropy of  $\mathbf{y}$   
 $\Rightarrow$  minimisation of mutual information of  $\mathbf{y}$  (thus minimising the  
statistical dependence between components of  $\mathbf{y}$ ).

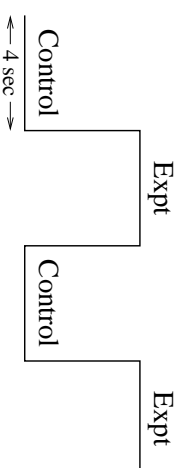
The algorithm iteratively updates  $\mathbf{W}$  according to

$$\mathbf{W}_{n+1} = \mathbf{W}_n + \epsilon(\mathbf{I} + (\mathbf{1} - 2g(\mathbf{u}_n))\mathbf{u}_n^T)\mathbf{W}_n,$$

where  $\epsilon$  is a learning rate and where  $\mathbf{W}_0 = \mathbf{I}$ . (This is a more  
efficient version of the algorithm due to Amari, Cichocki & Yang  
(1996).)

# Functional Magnetic Resonance Imaging (fMRI) in Neuroscience

Many fMRI experiments use a block design, where the subject is asked to perform experimental (e.g. finger tapping) and control tasks in an alternating sequence of (say) 4 second blocks.



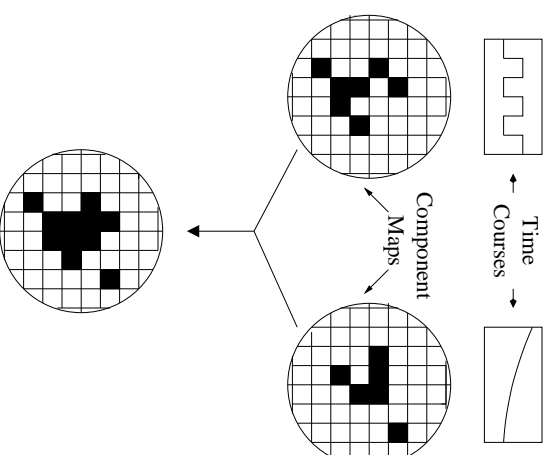
**Reference Function**

Signals from thousands of voxels in each of several brain slices are acquired every few seconds during such an experiment.

**Definition:** A **component** consists of

- a component map (a fixed 3D spatial distribution of brain voxel values),
- an associated time course of activation.

A component could, for example, correspond to the performance of a task or to physiological pulsations (respiratory and cardiac). The total brain activity at a given time is composed of the activity of several different components.



**The Problem:** To find a consistently (meaning throughout the experiment) task-related component whose time course correlates well with the reference function.

# Application of ICA to the analysis of fMRI signals

Consider the following model, where the signal is measured from  $N$  voxels at  $n$  time points.

$$\mathbf{X}_{n \times N} = \mathbf{M}_{n \times n} \mathbf{C}_{n \times N}$$

time pt  $i \rightarrow$       matrix of measured signals       $\left( \begin{array}{c} \text{matrix of} \\ \text{measured} \\ \text{signals} \end{array} \right)$       =      linear mixing matrix       $\left( \begin{array}{c} \text{linear} \\ \text{mixing} \\ \text{matrix} \end{array} \right)$       =      matrix of components       $\left( \begin{array}{c} \text{matrix} \\ \text{of compo-} \\ \text{nents} \end{array} \right)$        $\leftarrow$  cpt map  $k$

voxel  $\#j$        $\uparrow$       voxel  $\#l$        $\uparrow$

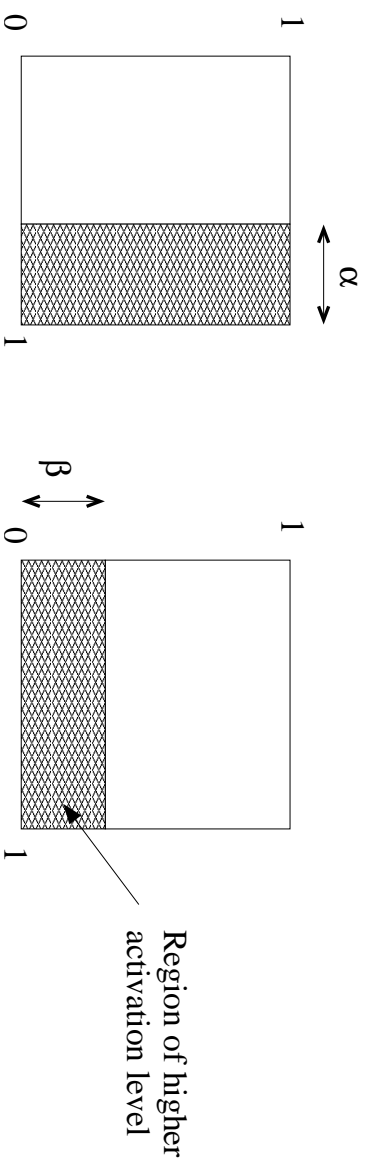
Assume the component maps are spatially independent, so the rows of  $\mathbf{C}$  are spatially independent. We use ICA to find an “unmixing matrix”  $\mathbf{W}$  such that  $\mathbf{C} = \mathbf{W}\mathbf{X}$ .

## **Advantages of ICA over other techniques used for the analysis of fMRI signals:**

- some techniques examine each voxel individually, and decide if a given voxel “qualifies” as being activated by the task. This ignores statistical relationships between voxels that ICA accounts for;
  - task-related fMRI changes make up only a small part of the variability of the signal (physiological pulsations, head movements and machine noise making up the bulk).
- Thus PCA may be poor at detecting task-related activations which ICA would capture.

## When does the algorithm work?

**Experiment:** Consider a square brain. We create the following two independent components:

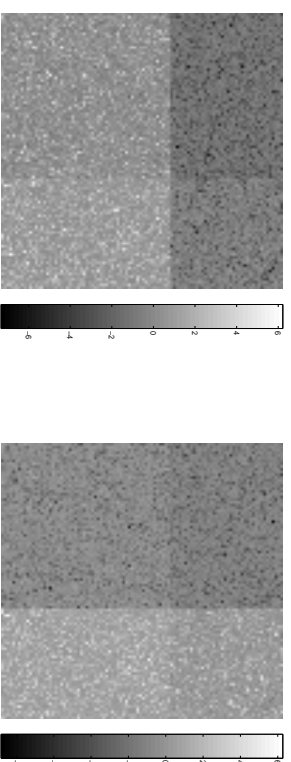


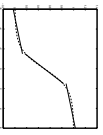
We then mix the components and input the mixture into the algorithm. Using this approach, we test the algorithm's performance.

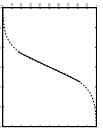
**Question:** How well does  $g$  have to “match” the cdf for the algorithm to be successful?

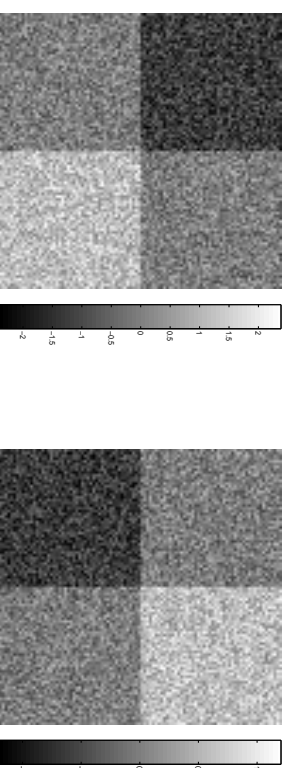
**Answers:** When you construct a r.v. whose cdf is:

- $g$ , the algorithm works quite well:



- , the algorithm works quite well;

- , the algorithm does not work well:



**Conclusion:** Having the tails in the distribution is important.

## Under investigation:

- The real data seems to have a cdf like:
  - How well does the algorithm work if the cumulative distribution function has a “gap”?
  - How wide does this “gap” have to be to affect the success of the algorithm?
  - Which locations of the “gap” have the most negative effect on the success of the algorithm?
  - Does changing the size of the activated portion of the component map affect the success of the algorithm?
- What happens if
  - # components  $\neq$  # time points?
- What is the effect, on the success of the algorithm, of the different time courses which the components may have?

- The assumption that the component maps are *mathematically* independent may be too strong.
  - What does the algorithm do if two statistically dependent components are mixed and inputted?
  - Is the method of testing for independence that the algorithm uses the most suitable? What happens if you input two identical components, but each very noisy? Does the algorithm find two components?

## **Future work:**

- How do you measure “how well the algorithm works”?
- Is it possible to make more use of the time information?
- What happens if you alter  $g$  to better fit the distribution of the actual data (also discussed in the Bell & Sejnowski paper)?
- What are the results of using a multiresolution model to determine the component maps?
- What are methods, other than ICA, for identifying the different components in the brain?