Analysis of Function Magnetic Resonance Images in R

John Myles White

April 6, 2010

What is fMRI?

Functional magnetic resonance imaging, usually referred to as fMRI, is a tool used to indirectly measure brain activity by measuring changes in the flow of oxygenated blood within small regions of the brain.

Why Use fMRI?

The goal of traditional fMRI research is the localization of functionality in the brain: for example, we may want to discover which regions of the brain are primarily responsible for processing visual images.

What Have We Learned from fMRI?

A canonical result from recent fMRI research is the discovery that one region of the brain is unusually sensitive to images of natural scenes, while another region of the brain is unusually sensitive to images of faces.
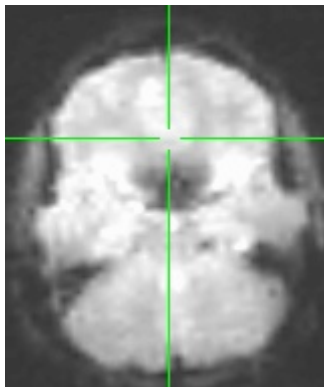
Types of Data

In fMRI, we generally work with two types of data:

- ▶ Anatomical data
- ▶ Blood volume data, which I will call functional data or BOLD (Blood Oxygenation Level Dependent) data
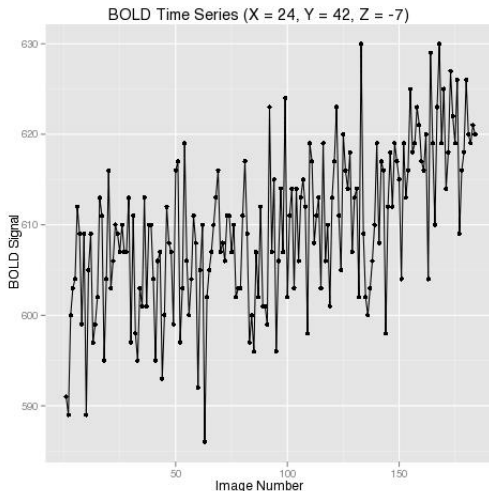
A Sample Anatomical Image
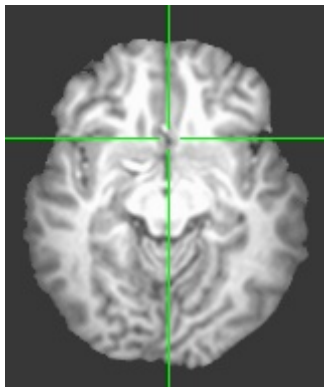
A Sample BOLD Image at Time $T = 0s$

A Sample BOLD Time Series from One Voxel

BOLD Time Series (X = 24, Y = 42, Z = -7)

Data Analytic Approach

To localize activity in the brain, we present a series of different stimuli over time and then regress the BOLD time series in every voxel against a predicted response derived from the stimulus time series using linear systems theory.
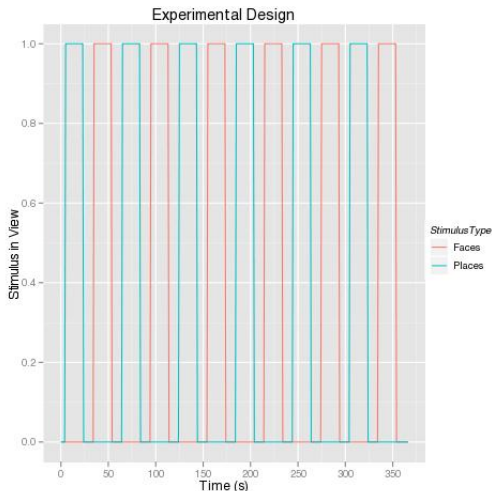
A Sample Face Stimulus

A Sample Place Stimulus

A Sample Stimulus Time Series

Experimental Design

Transforming Stimuli into Models of BOLD Signal

Before we can apply OLS regression to our problem, we need to
understand how the brain time series corresponds to the stimulus
time series. To do this, we assume that the brain is a linear signal
processing system.

Linear Systems Theory: The Big Ideas

▶ Well-Defined Unit Impulse Response: The system has a canonical response U to an input signal S

▶ Linear Scaling: The response to αS is αU

▶ Time Invariance: The response to a time-shifted version of S is a time-shifted version of U

Working with Linear Systems

The beauty of linear systems is a theorem that states that the response to any signal S is the convolution of S with the unit impulse response, U, where the convolution of two signals f and g is defined as:
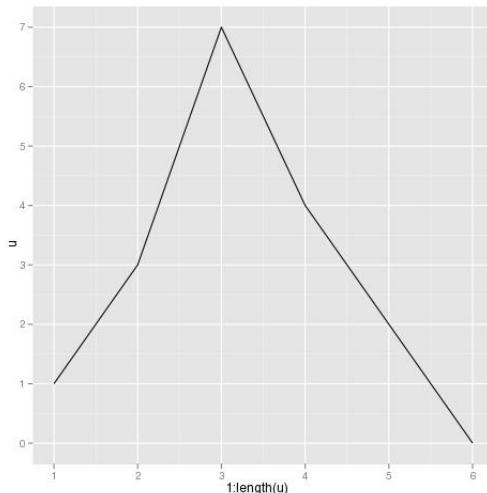
$$c(f, g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

Convolution in R: Encoding the Unit Impulse

Suppose that we have an unit impulse response function that looks like this:

```
u <- c(1, 3, 7, 4, 2, 0)
qplot(1:length(u), u, geom = 'line')
```
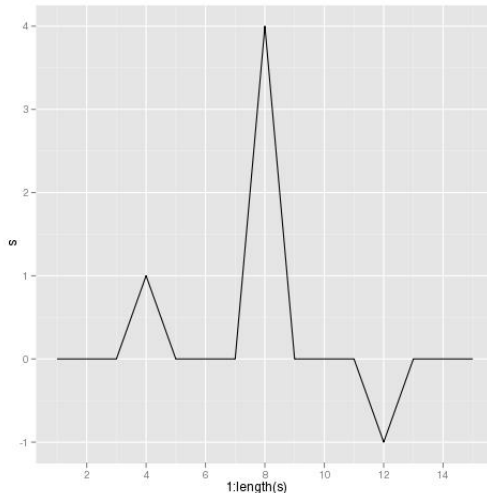
U

Convolution in R: Encoding the Input Signal

Suppose that we have an input signal that looks like this:

```
s <- c(0, 0, 0, 1, 0, 0, 0, 4, 0, 0, 0, -1, 0, 0, 0)
qplot(1:length(s), s, geom = 'line')
```
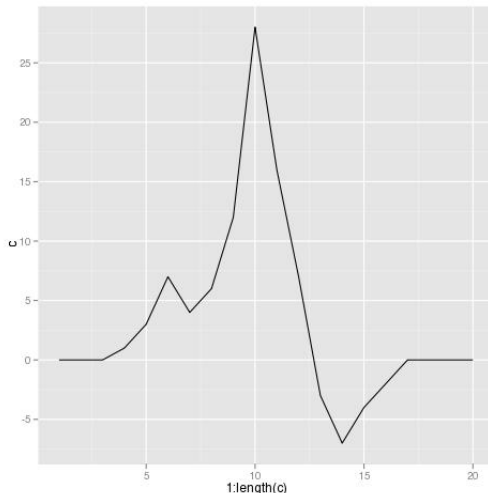
S

Convolution in R: Computing the Convolution

Given u and s, their convolution is easy to compute:

```
c <- convolve(s, rev(u), type = 'o')
qplot(1:length(c), c, geom = 'line')
```
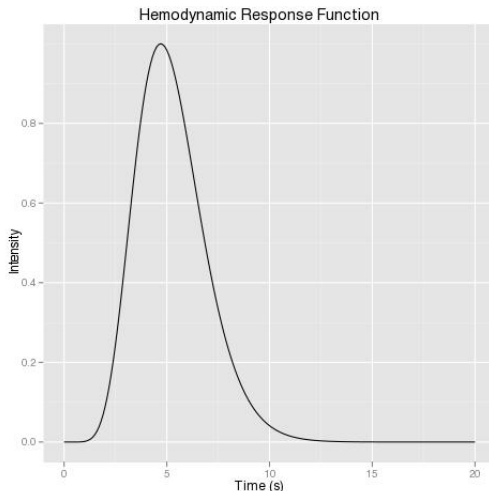
C

Modeling the Brain's Unit Impulse Response in R

The unit impulse response used by neuroscientists is called the hemodynamic response function. One possible model, which I tend to use for simplicity, is the gamma variate model:

$$(\frac{t}{pq})^p e^{\frac{p-t}{q}}$$

```
GammaHRF <- function(t, p = 8.6, q = 0.547)
{
 return((t / (p * q))^p * exp(p - t / q))
}
```
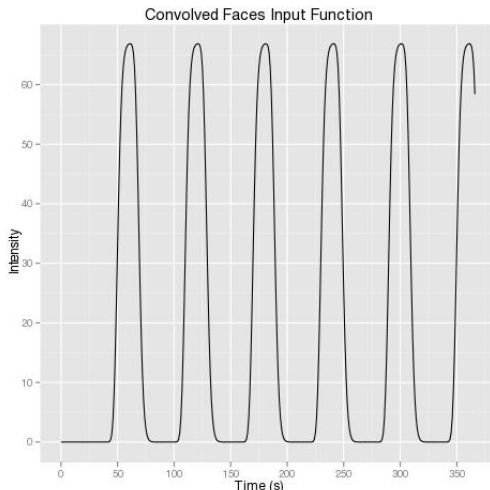
The Brain's Unit Impulse Response

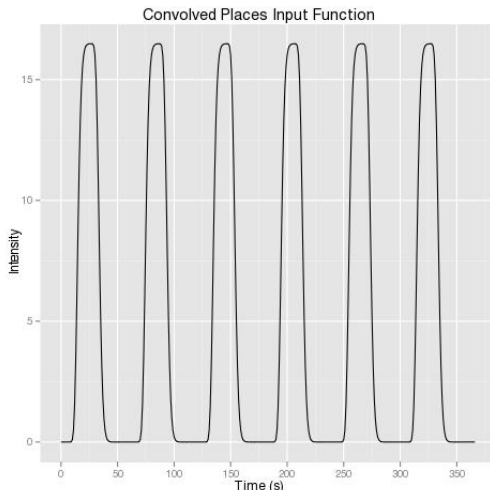
Hemodynamic Response Function

Convolving Our Stimuli

We can convolve our stimuli presentation time series with the HRF to get a prediction about the brain's response to our experimental manipulations:

Theoretical Faces Response

Convolved Faces Input Function

Theoretical Places Response

Convolved Places Input Function

The Mass Univariate Approach: Single Voxel OLS Regressions

We can then regress the time series of BOLD signal in each voxel against these two regressors:

```
lm(BOLD ~ ConvolvedPlaces + ConvolvedFaces)
```

Controlling for Artifacts

In practice, it proves useful to remove controllable artifacts from the time series when running these regressions. We usually do at least two things:

- ▶ Detrend data using a Legendre polynomial basis
- ▶ Correct for movement using rigid body transformations

Detrending

It is easy to add in a Legendre polynomial basis to any regression in R using poly():

```
lm(BOLD ~ poly(n) + ConvolvedPlaces + ConvolvedFaces)
```

We only need to agree upon a value for the polynomial order n. I generally will use a value between 2 and 6 depending upon the length of the EPI time series.

Rigid Body Transformations

Computing the optimal rigid body transformations to correct for head movement is more complex, so I'll skip this step. You would generally use something like optimize() to minimize the change in signal intensity between successive EPI images.

Final Linear Model

```
fit <- lm(BOLD ~ poly(n) + MovementCorrection
                 + ConvolvedPlaces + ConvolvedFaces)
```

Run Contrast Tests

Given the results from lm(), you can identify all of the voxels in which the contrast

```
coef(fit)[['ConvolvedPlaces']]
 - coef(fit)[['ConvolvedFaces']]
```

is significantly different from 0. All the voxels in which this occurs will be assumed to be more sensitive to places than to faces.

Visualize Results

The results from visualizing the t-stat for this contrast in every
voxel look something like this:
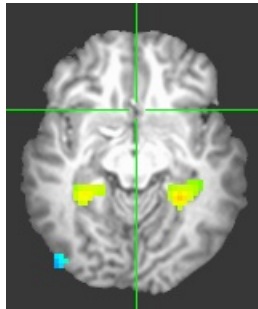
Other Possible Approaches

You can approach this problem in a variety of other ways as well:

▶ Run robust regressions instead of OLS regressions using rlm() from the MASS package.

▶ Classify inputs using the signal from many voxels with SVM's or another classifier using the caret package.

▶ Run autoregressive models to account for the autocorrelation in the BOLD signal using ar() or other time series functions.