7.91 Functional MRI of the Human Brain

Analyzing fMRI data: The General Linear Model

Finding the “language regions” in your brain

Question: which brain regions are engaged in high-level language processing? which regions increase their activity in response to sentences?

Finding the “language regions” in your brain

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

The structure of our data

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment:

- S
- N

Data: Time

What do these two conditions have in common? What do sentences have that nonwords don't?
The structure of our data

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment:

In MATLAB:

• Volume: a 3D matrix (a stack of slices, or 2D matrices)
• Voxel: a single entry \((x,y,z)\) in a volume
• Anatomical data: a 3D matrix of structural information
• Functional data: a 4D matrix of BOLD signals (a "movie" of 3D matrices)
  
The entry \((x,y,z,t)\) stores the signal from voxel \((x,y,z)\) at time \(t\).

The entry \((x,y,z,t)\) stores the signal from voxel \((x,y,z)\) at time \(t\).

FMRI DATA STRUCTURE

The data collected with fMRI are 3D pictures of the brain, called volumes. Each entry in such a 3D picture is called a voxel (similar to the entries of a 2D picture, called "pixels").

Each time we "run" the MRI scanner we collect a volume, or multiple volumes, of the brain. There are two common types of such runs:

• An anatomical run collects a single volume of the brain, while the subject is resting. We collect an anatomical volume in order to see the structure of the brain.
• A functional run collects a series of volumes, one after the other, while the subject is performing some tasks. We collect these functional volumes to see how the activity in the brain changes over time, according to the tasks being performed.

The brain slices we have studied so far are mainly used for visualization purposes. The actual analysis of functional MRI data is concerned with how brain activity changes throughout our experiment, and it therefore focuses on signal time-series: a voxel's signal from all functional volumes, collected across the entire run.

HOW DO YOU CONSTRUCT SIGNAL TIME-SERIES?
Finding the “language regions” in your brain

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment:

Data

Analysis: an intuitive approach
1. For each voxel, look at its signal time-series (activity across time)
2. Average the signal across volumes collected while reading sentences
3. Average the signal across volumes collected while reading nonwords
4. Compare the two averages: does the voxel show a S > N pattern?

Analysis: a somewhat better approach
- The difference in response to S and N might be due to chance (not real)
- To test how likely the difference is to be real, we take noise into account
- The simplest way is to compute a t-value:
  \[ t = \frac{\text{Mean}(S) - \text{Mean}(N)}{\text{noise}} \]
- more likely to be due to chance
- less likely to be due to chance

Mean(sentences) = 100  Mean(nonwords) = 98

BOLD signal = task-related activity changes + noise (other changes)
explained variations  unexplained variations

\[ t = \frac{\text{Mean}(S) - \text{Mean}(N)}{\text{noise}} \]
Finding the "language regions" in your brain

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment:

Data

The difference between the response to sentences and to nonwords might be due to chance. The more noisy our data are, the _______ this difference should be in order for us to believe that it is real.

A. Bigger
B. Smaller
C. I don't know

Finding the "language regions" in your brain

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment:

Data

Finding the "language regions" in your brain

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment:

Data

Findings

The BOLD signal prediction: When neurons are active, the fMRI BOLD signal will rise...

A. immediately
B. 1-3 seconds after the neural activity
C. 6-12 seconds after the neural activity
D. 25-30 seconds after the neural activity

What do we know about fMRI data?

Stimuli:

S
N

Neural activity:

BOLD signal prediction:

A voxel I took from outside the language system

A voxel I took from the language system

Finding the "language regions" in your brain

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment:

Data

Findings

The BOLD signal prediction: When neurons are active, the fMRI BOLD signal will rise...

A. immediately
B. 1-3 seconds after the neural activity
C. 6-12 seconds after the neural activity
D. 25-30 seconds after the neural activity

What do we know about fMRI data?

Stimuli:

S
N

Neural activity:

BOLD signal prediction:
What do we know about fMRI data?

Stimuli: S N

Neural activity:

BOLD signal prediction:

Finding the “language regions” in your brain

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment: S N

BOLD signal:

The time series of language voxels should look like a combination of the two signals above, with the upper (S) having more weight than the lower (N).

(de)Constructing BOLD signals

We can approximate the signal time-series of a voxel by combining these 3 signals:

Baseline signal:

Response to S:

Response to N:

MATLAB demonstration (classDemoLinearCombinations.m)

(de)Constructing BOLD signals

We can approximate the signal time-series of a voxel by combining these 3 signals:

$\beta_1 = 97.8$

$\beta_2 = 1.20$

$\beta_3 = 0.56$

• These signals we use to construct the approximation are called predictors.
• Each predictor is associated with a weight called a beta-weight.
• To create a linear combination of predictors, which approximates a true signal, we multiply each predictor by its beta-weight and then sum the results.

(de)Constructing BOLD signals

We can approximate the signal time-series of a voxel by combining these 3 signals:

$\beta_1 = 97.8$

$\beta_2 = 1.20$

$\beta_3 = 0.56$

In MATLAB:
• predictors are column vectors: $x_1, x_2, x_3$
• Beta-weights are scalars (numbers)
• An approximation is created by simple multiplication and addition

$\hat{y} = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$

$\beta_2$ estimates a voxel’s increase in activity in response to reading sentences
$\beta_1$ estimates a voxel’s increase in activity in response to reading nonwords
We can approximate the signal time-series of a voxel by combining these 3 signals:
\[
\beta_1 \times + \beta_2 \times + \beta_3 \times = \beta_1 \times + \beta_2 \times + \beta_3 \times
\]

For every point in time:
\[
\text{signal}(t) - \text{prediction}(t) = \text{error}(t)
\]

Example: \( \text{signal}(t) > \text{prediction}(t) \)
\( \text{signal}(t) - \text{prediction}(t) > 0 \)
\( \text{error}(t) > 0 \)

In the analysis we demonstrated, what is the input (things we know) and what is the output (things we are looking for)?

A. INPUT: a voxel's true BOLD signal
   OUTPUT: predictors and beta weights that best approximate the signal

B. INPUT: a voxel's true BOLD signal and our predictors
   OUTPUT: beta weights that combine with the predictors to give the best approximation of the true signal

C. INPUT: a voxel's true BOLD signal and beta weights
   OUTPUT: predictors that combine with the beta weights to give the best approximation of the true signal
(de)Constructing BOLD signals

We can approximate the signal time-series of a voxel by combining these 3 signals:

\[ \beta_1 \times + \beta_2 \times + \beta_3 \times = \]

Finding the “language regions” in your brain

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

Experiment:

\[ \text{S} \quad \text{N} \]

Analysis: the approach that works

- Find the beta weights that best approximate a voxel’s signal time-series. The best approximation is the one with the least errors.
- Compare the beta-weight for (estimated response to) sentences to the beta-weight for (estimated response to) nonwords.

BOLD signal = task-related activity changes + noise (other changes)

explained variations + unexplained variations

linear combination of predictors (approximation) + errors

Matrix notation

\[ \beta_1 \times + \beta_2 \times + \beta_3 \times = \]

\[ \begin{bmatrix} x \\ x \end{bmatrix} \times b \]

(de)Constructing BOLD signals: Exercises 10-11

EXERCISE 10: ERRORS

You will now explore errors between a linear combination of predictors and the actual signal that they try approximating. For this purpose, you will use the code gp10e. This code already has some variables that are given to you:

- \( x \): 12 rows of a data matrix, where each column contains a set of beta weights (this time you do not have to create this matrix, the code already has it).
- \( x \): 12 rows of the same predictors used in exercises 5-9.
- \( y \): a column vector with the true brain signal of voxel B from exercise 9.

Fill in the code (lines 18-20) to loop through the 3 sets of beta weights, constructing a linear combination \( y \) based on each set, and computing the errors between the resulting approximations and \( y \). When you run the code, it will generate figures of the errors. Inspect these figures and answer the question that follows.

EXERCISE 11: MATRIX NOTATION

This short exercise does not require any MATLAB code. Simply answer the questions below.

(de)Constructing BOLD signals

A voxel’s BOLD signal at rest is 100.
When the participant is reading sentences, the signal increases to 120.
When the participant is reading nonwords, the signal increases to 110.
When approximating the voxel’s BOLD signal using our three predictors, what should the beta weights for sentences and nonwords be?

A. Sentences: 120; Nonwords: 110
B. Sentences: 0.2; Nonwords: 0.1
C. Sentences: 20; Nonwords: 10
D. Sentences: 1.2; Nonwords: 1.2

(de)Constructing BOLD signals

At the beginning of the experiment, when the participant is not reading anything, the predictors that contribute to the approximation are ___.
Twelve seconds after the participant has started reading nonwords, the predictors that contribute to the approximation are ___.

A. baseline + sentences + nonwords; nonwords
B. baseline; nonwords
C. baseline + sentences + nonwords; baseline + nonwords
D. baseline; baseline + nonwords
E. baseline + sentences + nonwords; baseline + sentences + nonwords
F. baseline; baseline + sentences + nonwords
Finding the "language regions" in your brain

**Question:** which brain regions are engaged in high-level language processing? which regions respond more to *sentences* than to *nonwords*?

**Experiment:**

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Analysis:** the approach that works

- Find the beta weights that best approximate a voxel’s signal time-series. The best approximation is the one with the least errors.
- Compare the beta-weight for (estimated response to) *sentences* to the beta-weight for (estimated response to) *nonwords*.

Instead of trial and error: **The General Linear Model**

### The General Linear Model (GLM)

**Question:** which brain regions are engaged in high-level language processing? which regions respond more to *sentences* than to *nonwords*?

**GLM:**

$$\text{BOLD signal} = \mathbf{X} \times \mathbf{b} + \text{error}$$

- explained variations + unexplained variations
- task-related activity changes + noise (other changes)

In MATLAB:

```matlab
b = regress(BOLD_signal, X)
```

**What we know:**
- BOLD signal: we collect this from the brain (functional data).
- \( \mathbf{X} \): design matrix (each column is a predictor that we built ourselves).

**What we want to find:**
- \( \mathbf{b} \): vector of beta-weights (one weight per predictor in \( \mathbf{X} \)) that give the best approximation of the BOLD signal.

**How we find it:**
- By minimizing the sum of squared errors. In practice, the GLM has a formula, which guarantees to find these beta-weights.

**Example:**

- **Sentences**: 6 head-motion predictors: 3 translations and 3 rotations
- **Nonwords**: Time derivatives: slight time-shifts of the original \( S \) and \( N \) predictors

Instead of trial and error: **The General Linear Model**

**GLM:**

$$\text{BOLD signal} = \mathbf{X} \times \mathbf{b} + \text{error}$$

- explained variations + unexplained variations
- task-related activity changes + noise (other changes)

In MATLAB:

```matlab
b = regress(BOLD_signal, X)
```

**What we know:**
- BOLD signal: we collect this from the brain (functional data).
- \( \mathbf{X} \): design matrix (each column is a predictor that we built ourselves).

**What we want to find:**
- \( \mathbf{b} \): vector of beta-weights (one weight per predictor in \( \mathbf{X} \)) that give the best approximation of the BOLD signal.

**How we find it:**
- By minimizing the sum of squared errors. In practice, the GLM has a formula, which guarantees to find these beta-weights.

Instead of trial and error: **The General Linear Model**

**GLM:**

$$\text{BOLD signal} = \mathbf{X} \times \mathbf{b} + \text{error}$$

- explained variations + unexplained variations
- task-related activity changes + noise (other changes)

In MATLAB:

```matlab
b = regress(BOLD_signal, X)
```

**What we know:**
- BOLD signal: we collect this from the brain (functional data).
- \( \mathbf{X} \): design matrix (each column is a predictor that we built ourselves).

**What we want to find:**
- \( \mathbf{b} \): vector of beta-weights (one weight per predictor in \( \mathbf{X} \)) that give the best approximation of the BOLD signal.

**How we find it:**
- By minimizing the sum of squared errors. In practice, the GLM has a formula, which guarantees to find these beta-weights.
The General Linear Model (GLM)

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

GLM:

Any predictor that can help approximate the BOLD signal will decrease SSE. Therefore, we include additional predictors:

- 6 head-motion predictors: 3 translations and 3 rotations
- Time derivatives: slight time-shifts of the original S and N predictors

Why stop there?
Why not add a gazillion more predictors?

The General Linear Model (GLM): Exercises 12-14

Exercise 12: Do-it-yourself GLM

You will now implement a GLM in MATLAB, analyzing the signals from the six voxels you have worked with earlier (three of these are voxels A, B and C from exercise 9):

- voxel 1: [57, 63, 35]
- voxel 2: [54, 60, 24]
- voxel 3: [33, 60, 15]
- voxel 4: [12, 52, 28]
- voxel 5: [18, 45, 28]
- voxel 6: [28, 41, 24]

You will use the three predictors you have been working with (constant, sentences, nonwords) to find, for each voxel, beta weights that produce the best approximation of the voxel’s true signal.

Exercise 13: The SSE Measure

To study the behavior of errors and the SSE measure, let us focus on voxel number 2: [40, 40, 24]. The code q13a creates 5,001 linear combinations to approximate the signal from this voxel. One linear combination uses the beta weights found with a GLM (those you found in the exercise 12). The other 5,000 linear combinations use random beta weights.

The General Linear Model (GLM)

Question: which brain regions are engaged in high-level language processing? which regions respond more to sentences than to nonwords?

GLM: A comparison of beta-weights is called a contrast.
Formally, a contrast is a vector indicating which beta-weights we are testing:

\[
\begin{bmatrix}
\beta(S) - \beta(N) \\
\beta(S) > \beta(N)
\end{bmatrix}
\]

\[
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\beta_{11}
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.5 \\
0.5 \\
-0.5 \\
-0.5 \\
0
\end{bmatrix}
\]

\[
\begin{bmatrix}
\beta_1 + \beta_2 > 0 \\
\beta_1 + \beta_2 > 0 \\
\frac{1}{2} \beta_3 + \frac{1}{2} \beta_4 > 0 \\
\frac{1}{2} \beta_3 - \frac{1}{2} \beta_4 > 0
\end{bmatrix}
\]

A voxel I took from the language system
The General Linear Model (GLM)

Question: which brain regions are engaged in high-level language processing? which regions respond more to *sentences* than to *nonwords*?

GLM: A comparison of beta-weights is called a *contrast*.

Formally, a contrast is a vector indicating which beta-weights we are testing

- We can compare one set of beta-weights to another set of beta-weights
- Note: the sum of all positive entries should be 1; the sum of all negative entries should be -1.

• We can compare one set of beta-weights to another set of beta-weights
- Note: the sum of all positive entries should be 1; the sum of all negative entries should be -1.
- We can compare a single beta-weight (or a single set) to 0.

Example: some model with 5 predictors; is \( \beta_2 \) bigger than 0?

\[
\begin{bmatrix}
0 \\
1 \\
-1 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

Repeat for all voxels.
Significant voxels are those that have t-values that are highly unlikely to get by mere chance.

Answer:

The General Linear Model (GLM): exercise 15

EXERCISE 15: DEFINING CONTRASTS

When we locate someone’s language system using fMRI, we usually include two runs of the experiment: i.e., two runs of reading both sentences and nonwords. Therefore, our full design matrix usually includes:

1. A constant predictor for the first run
2. A predictor for the response to *sentences* in the first run
3. A predictor for the response to *nonwords* in the first run
4. A constant predictor for the second run
5. A predictor for the response to *sentences* in the second run
6. A predictor for the response to *nonwords* in the second run

You can also use the same data to ask other questions, using other contrasts:

Which regions respond more to *nonwords* than to *sentences*?

\( \beta(\text{N}) > \beta(\text{S}) \)

Which regions respond to *sentences*?

\( \beta(\text{S}) > 0 \)
Summary

- **Predictor**: a “mock signal” indicating the expected change in brain activity throughout the experiment caused by a hypothesized process (cognitive, physiological, etc.)
- **Beta-weight**: a weight indicating how much a predictor contributes to the true BOLD signal, i.e., how much the signal changes due to the hypothesized process, holding everything else in the model constant.
- **Error**: the difference between the true BOLD signal and the combination of predictors that best approximates it. These are changes (variations) in the true signal that we cannot explain with the predictors.
- **Contrast**: the difference between two (groups of) betas
- **Contrast t-value**: the number which results from dividing the contrast value by a measure of error (“taking the error into account”).

\[
t = \frac{\beta(S) - \beta(N)}{\text{error(ish)}}
\]

\[
\text{BOLD signal} = X \times b + e
\]
explained variations + unexplained variations

- **task-related activity changes + noise (other changes)**

Final Exercise

PUTTING IT ALL TOGETHER

You are now ready to find the language system in the brain!

It's time to use everything you've learned in this workshop and implement a GLM from start to finish.

This exercise will be run through the code FinalExercise.m. The following pages describe the code and give some instructions about the exercise. After you run the code, answer the questions on the last page.