Case Study 3: fMRI Prediction

fMRI Prediction Task, Ridge, LASSO Review

Machine Learning for Big Data
CSE547/STAT548, University of Washington

So far large/streaming N,
now big-p domain

Emily Fox
January 28th, 2014

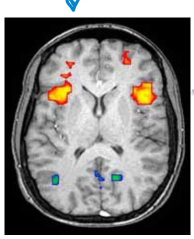
fMRI Prediction Task

- **Goal:** Predict word stimulus from fMRI image

Can we read your brain?

scan after seeing word

guess word



Classifier
(logistic regression, kNN, …)

~~HAMMER~~
or
HOUSE

fMRI

fMRI

high-res spatially

~1 mm resolution

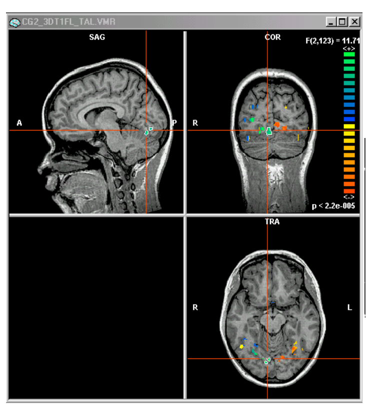
pretty slow

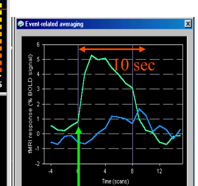
~1 image per sec.

20,000 voxels/image

safe, non-invasive

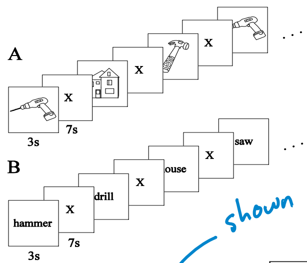
measures Blood Oxygen Level Dependent (BOLD) response





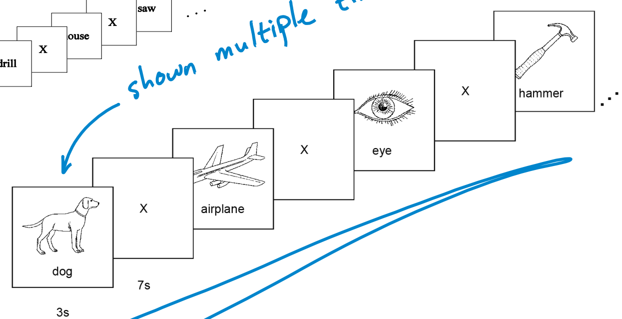
Typical fMRI response to impulse of neural activity

Typical Stimuli



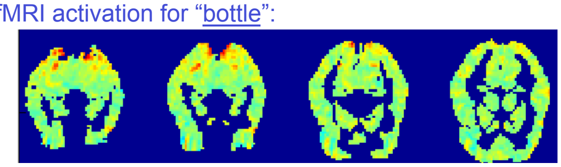
Each stimulus repeated several times

shown multiple times



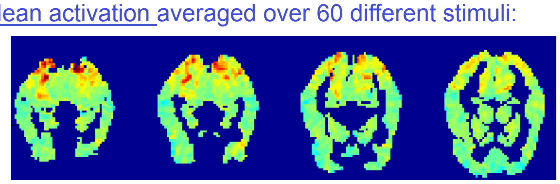
fMRI Activation

fMRI activation for "bottle":

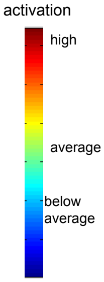


stimulus

Mean activation averaged over 60 different stimuli:

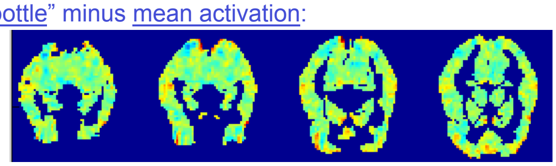


fMRI activation



"bottle" minus mean activation:

is this enough?

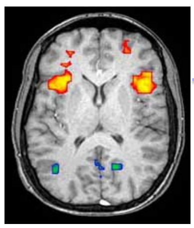


fMRI Prediction Task

- **Goal:** Predict word stimulus from fMRI image
- **Challenges:** # of voxels = # params
  - p >> N (feature dimension >> sample size)
  - Cost of fMRI recordings is high
  - Only have a few training examples for each word

many more params than obs.

What can we do?



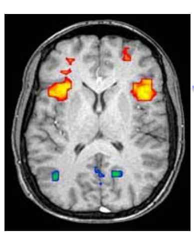
Classifier
(logistic regression, kNN, …)

~~HAMMER~~
or
HOUSE

Zero-Shot Classification

- **Goal:** Classify words not in the training set
- **Challenges:**
  - Cost of fMRI recordings is high
  - Can't get recordings for every word in the vocabulary

Never showed "giraffe" in scanner



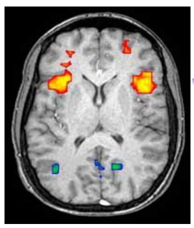
Classifier
(logistic regression, kNN, …)

~~HAMMER~~
or
HOUSE

Zero-Shot Classification

many docs that contain "giraffe" also contain "neck" "animal" "zoo" ⋮

- **Goal:** Classify words not in the training set
- **Challenges:**
  - Cost of fMRI recordings is high
  - Can't get recordings for every word in the vocabulary
- We don't have many brain images, but we have a lot of info about the words and how they relate (co-occurrence, etc.)
- How do we utilize this "cheap" information?



Classifier
(logistic regression, kNN, …)

~~HAMMER~~
or
HOUSE

Semantic Features

Google Trillion word corpus

Semantic feature values: "**celery**"
0.8368, eat
0.3461, taste
0.3153, fill
0.2430, see
0.1145, clean
0.0600, open
0.0586, smell
0.0286, touch
…
…
0.0000, drive
0.0000, wear
0.0000, lift
0.0000, break
0.0000, ride

co-occurrence

Semantic feature values: "**airplane**"
0.8673, ride
0.2891, see
0.2851, say
0.1689, near
0.1228, open
0.0883, hear
0.0771, run
0.0749, lift
…
…
0.0049, smell
0.0010, wear
0.0000, taste
0.0000, rub
0.0000, manipulate

Zero-Shot Classification

image

A= { → "dog" }

few examples

- From training data, learn two mappings:
  - S: input image → semantic features
  - L: semantic features → word

B= { [ ] → "dog" }

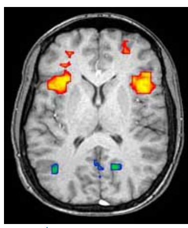
semantic feature vecs

many examples

- Can use "cheap" co-occurrence data to help learn L

Training = { → [ ] → "dog" } N examples... N small

use both A + B



Features of word → **Classifier** (logistic regression, kNN, …) → ~~HAMMER~~ or HOUSE

new image

using B

Predict : → [ ] → "giraffe"

S

learned on training data