Case Study 3: fMRI Prediction

Now:

$d \to p$

dim of features/predictors/covariates

fMRI Prediction Task

Machine Learning for Big Data

CSE547/STAT548, University of Washington

Emily Fox

April 23rd, 2015

"big data challenge":
large N, streaming N,
now big-p domain

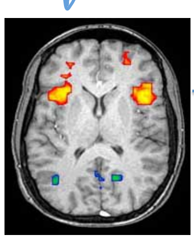
fMRI Prediction Task

- **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

~1 mm resolution

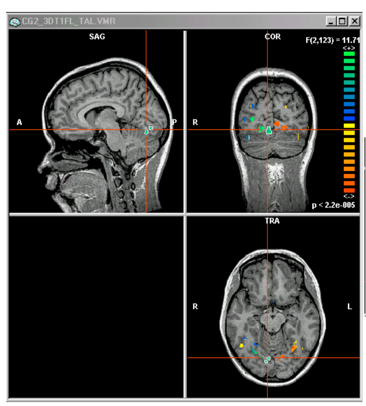
pretty slow

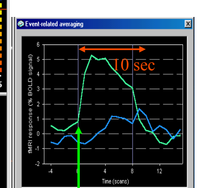
~1 image per sec.

20,000 voxels/image

safe, non-invasive

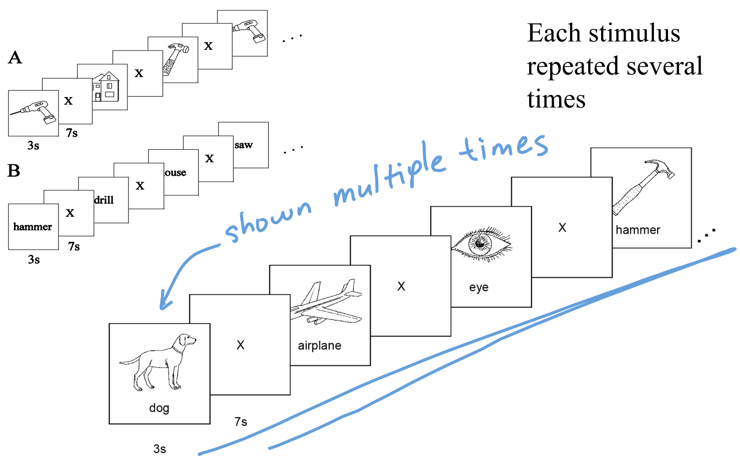
measures Blood Oxygen Level Dependent (BOLD) response





Typical fMRI response to impulse of neural activity

Typical Stimuli

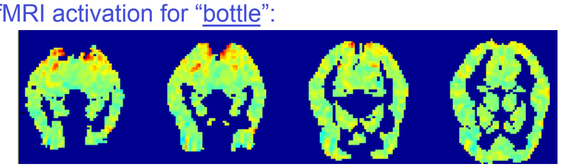


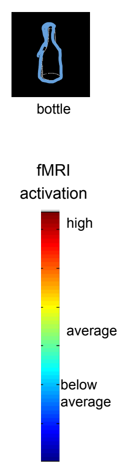
Each stimulus repeated several times

©Emily Fox 2015 5

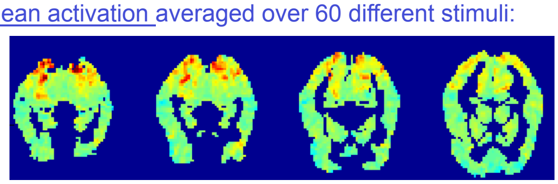
fMRI Activation

fMRI activation for "bottle":



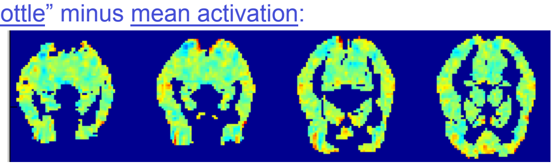


Mean activation averaged over 60 different stimuli:



"bottle" minus mean activation:

is this enough?



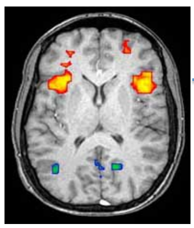
©Emily Fox 2015 6

fMRI Prediction Task

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

many more params than obs.

what can we do?



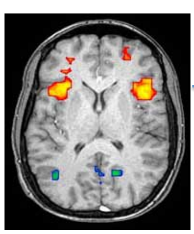
Classifier
(logistic regression, kNN, …)

~~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



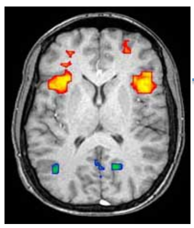
Classifier
(logistic regression, kNN, …)

~~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**"	Semantic feature values: "**airplane**"
0.8368, eat	0.8673, ride
0.3461, taste	0.2891, see
0.3153, fill	0.2851, say
0.2430, see	0.1689, near
0.1145, clean	0.1228, open
0.0600, open	0.0883, hear
0.0586, smell	0.0771, run
0.0286, touch	0.0749, lift
…	…
…	…
0.0000, drive	0.0049, smell
0.0000, wear	0.0010, wear
0.0000, lift	0.0000, taste
0.0000, break	0.0000, rub
0.0000, ride	0.0000, manipulate

co-occurence

Zero-Shot Classification

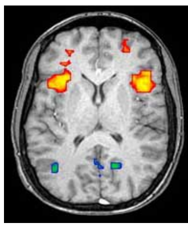
image
$A = \{ \text{[image]} \rightarrow \text{"dog"} \}$
few examples

semantic features
$B = \{ [\,\vdots\,] \rightarrow \text{"dog"} \}$
many examples

- From training data, learn two mappings:
  - S: input image → semantic features
  - L: semantic features → word
- Can use "cheap" co-occurrence data to help learn L

Training: $\{ \text{[image]} \rightarrow [\,\vdots\,] \xrightarrow{\text{from } B} \text{"dog"} \}$ N examples, N small

uses both A+B



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

new image

Predict: $\text{[image]} \xrightarrow{S} [\,\vdots\,] \rightarrow$ "giraffe"

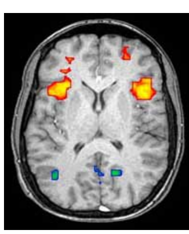
using B (e.g. w/ NN search) L

learned from training data

fMRI Prediction Subtask

- **Goal:** Predict semantic features from fMRI image

Learning S: images → semantic features



x^i

Features of word

y^i

$$x^i = \begin{bmatrix} x_1^i \\ \vdots \\ x_{20,000}^i \end{bmatrix} \xrightarrow{S} \begin{bmatrix} y_1^i \\ \vdots \\ y_d^i \end{bmatrix}$$

$\in \mathbb{R}^{20,000}$

d = # of semantic features

$y^i \in \mathbb{R}^d$

Simplification: Consider predicting each y_j^i (feature) separately

Case Study 3: fMRI Prediction

Ridge, LASSO Review

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
April 23rd, 2015

 13

Linear Regression

Note: previously w_p "weight", but β more common in stat

- Model: $y^i = \beta_0 + \beta_1 x_1^i + \dots + \beta_p x_p^i + \epsilon_i$

one feat. $\in \mathbb{R}$

$= \beta^T x^i + \epsilon^i$

$\epsilon_i \sim N(0, \sigma^2)$
$\Downarrow$
$y^i \sim N(\beta^T x^i, \sigma^2)$

All obs:

$$\underbrace{\begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} 1 & x_1^1 & x_2^1 & \cdots & x_p^1 \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & x_1^N & x_2^N & \cdots & x_p^N \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}}_{\beta} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

- MLE: $\hat{\theta} = \arg\max_{\theta} \log p(D \mid \theta)$ all obs. $\overset{iid}{\leftarrow} \sum_{i=1}^{N} \log p(y^i \mid x^i, \theta) = -\frac{1}{2\sigma^2} \underbrace{\sum_i (y^i - \beta^T x^i)^2}_{RSS(\beta)}$ + const.

$$\hat{\beta}^{ML} = \arg\min_{\beta} RSS(\beta) = (\underbrace{\overset{(p+1)\times N}{X^T} \overset{N\times(p+1)}{X}}_{(p+1)\times(p+1)})^{-1} X^T y$$

Here, $p \gg N$ so $X^T X$ is low rank + want its inverse

- Minimizing RSS= least squares regression

 14

Ridge Regression

- Ameliorating issues with overfitting: penalization of weights = "regularization"
- New objective:

$$\min_{\beta} \underbrace{\sum_{i=1}^{N} \left(y^i - (\beta_0 + \beta^T x^i)\right)^2}_{RSS(\beta)} + \lambda \|\beta\|_2^2$$

don't want to penalize intercept

$[\beta_1, \ldots, \beta_p]$

redefine X w/o 1's

$\beta^T\beta$

$$\Updownarrow$$

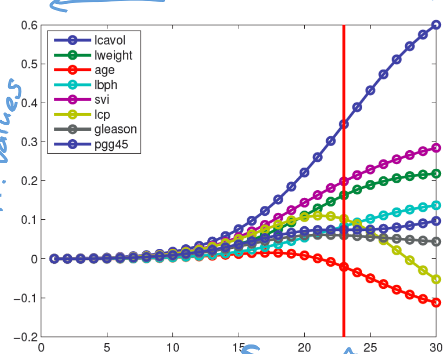
- Solution: $\min_{\beta} RSS(\beta) \quad \text{s.t.} \quad \|\beta\|_2^2 \leq S$

$$\hat{\beta}^{ridge} = (X^T X + \lambda I)^{-1} X^T y$$

Ridge Coefficient Path

$\hat{\beta}^{ridge} = 0$ ← → $\hat{\beta}^{ridge} = \hat{\beta}^{MLE=LS}$

Coeff. values



S

← inc. λ

CV soln

From Kevin Murphy textbook

$\|\beta\|_2^2 \leq S$

- Typical approach: select λ using cross validation (CV)

Case Study 3: fMRI Prediction

fMRI Prediction Results

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
April 23rd, 2015

fMRI Prediction Results

- Palatucci et al., "Zero-Shot Learning with Semantic Output Codes", NIPS 2009
- fMRI dataset:
  - 9 participants
  - 60 words (e.g., *bear, dog, cat, truck, car, train*, …)
  - 6 scans per word
  - Preprocess by creating 1 "time-average" image per word

·2 diff. sources of side info
→ compare perf. using each

- Knowledge bases
  - Corpus5000 – semantic co-occurrence features with 5000 most frequent words in Google Trillion Word Corpus
  - human218 – Mechanical Turk (Amazon.com)
    218 semantic features (*"is it manmade?", "can you hold it?"*,…)
    Scale of 1 to 5

fMRI Prediction Results

- **First stage:** Learn mapping from images to semantic features ($\underset{\wedge}{S}$)

- Ridge regression

fMRI images $X \in \mathbb{R}^{N \times p}$ ← matrix of x^i

obs $F \in \mathbb{R}^{N \times d}$ ← # of semantic features; ← matrix of f^i

from limited training data : pwd matrix: $\hat{\beta}^{ridge} = (X^T X + \lambda I)^{-1} X^T F$ — right soln assuming d ind. features .... d ind. subproblems stacked up

pred semantic features of new image: $\hat{f}^{new} = X^{new} \hat{\beta}^{ridge}$

- **Second stage:** 1-NN classification using knowledge base

©Emily Fox 2015 19

fMRI Prediction Results

- Leave-two-out-cross-validation ← out of 60
  - Learn ridge coefficients using 58 fMRI images
  - Predict semantic features of 1st heldout image
  - Compare whether semantic features of 1st or 2nd heldout image are closer

Table 1: Percent accuracies for leave-two-out-cross-validation for 9 fMRI participants (labeled P1-P9). The values represent classifier percentage accuracy over 3,540 trials when discriminating between two fMRI images, both of which were omitted from the training set.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	Mean
Google → corpus5000	79.6	67.0	69.5	56.2	77.7	65.5	71.2	72.9	67.9	**69.7**
Mech. Turk → human218	90.3	82.9	86.6	71.9	89.5	75.3	78.0	77.7	76.2	**80.9**

← stat. sig.

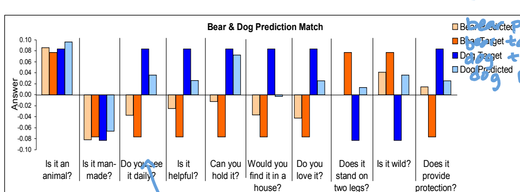


Figure 1: Ten semantic features from the human218 knowledge base for the words *bear* and *dog*. The true encoding is shown along with the predicted encoding when fMRI images for bear and dog were left out of the training set.

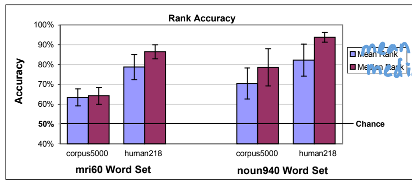
"yes" for dogs
"no" for bears
+ our pred agree

"do you see it every day"

©Emily Fox 2015 20

fMRI Prediction Results

- Leave-one-out-cross-validation
  - Learn ridge coefficients using 59 fMRI images
  - Predict semantic features of heldout image
  - Compare against very large set of possible other words



mean rank
median rank

Figure 2: The mean and median rank accuracies across nine participants for two different semantic feature sets. Both the original 60 fMRI words and a set of 940 nouns were considered.

Table 2: The top five predicted words for a novel fMRI image taken for the word in bold (all fMRI images taken from participant P1). The number in the parentheses contains the rank of the correct word selected from 941 concrete nouns in English.

Bear	**Foot**	**Screwdriver**	**Train**	**Truck**	**Celery**	**House**	**Pants**
(1)	(1)	(1)	(1)	(2)	(5)	(6)	(21)
bear	*foot*	*screwdriver*	*train*	jeep	beet	supermarket	clothing
fox	feet	pin	jet	*truck*	artichoke	hotel	vest
wolf	ankle	nail	jail	minivan	grape	theater	t-shirt
yak	knee	wrench	factory	bus	cabbage	school	clothes
gorilla	face	dagger	bus	sedan	*celery*	factory	panties

How high did true word fall on the list of ranked words from pred.

Case Study 3: fMRI Prediction

LASSO Review

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
April 23rd, 2015

Variable Selection

not min this obj coeff. are very sensitive to what was inc. in the model

- Ridge regression: Penalizes large weights
- What if we want to perform "feature selection"?
  - E.g., Which regions of the brain are important for word prediction?
  - Can't simply choose predictors with largest coefficients in ridge solution
  - Computationally impossible to perform "all subsets" regression

    2^p subsets of predictors ... clearly not feasible for large p

    discrete
  - Stepwise procedures are sensitive to data perturbations and often include features with negligible improvement in fit ← greedy, but ∃ backtracking approaches
- Try new penalty: Penalize non-zero weights
  - Penalty:

    $$\|\beta\|_1 = \sum_j |\beta_j| \quad \ell_1\text{-reg.}$$
  - Leads to sparse solutions
  - Just like ridge regression, solution is indexed by a continuous param λ

LASSO Regression

- **LASSO:** least absolute shrinkage and selection operator
- New objective:

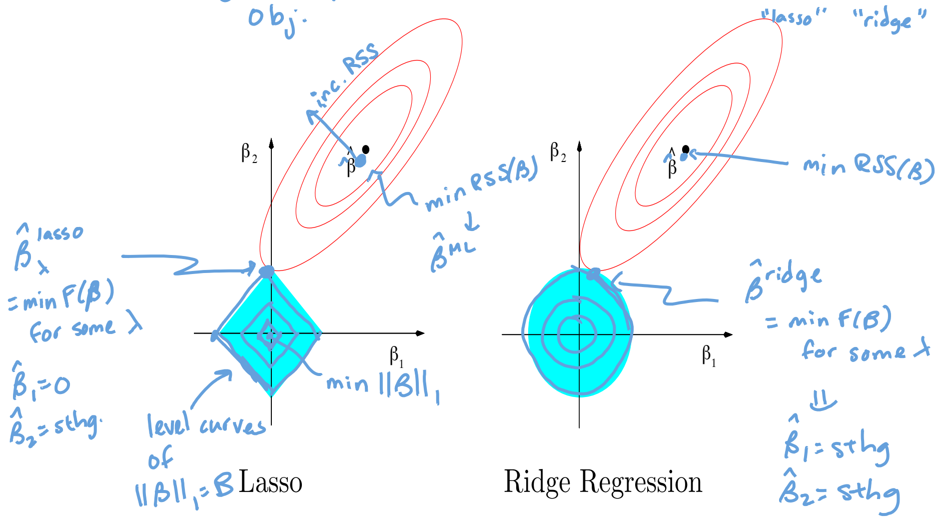
$$\min_\beta \underbrace{\sum_{i=1}^{N} (y^i - (\beta_0 + \beta^T x^i))^2}_{RSS(\beta)} + \lambda \|\beta\|_1$$

$$\Updownarrow$$

$$\min_\beta RSS(\beta) \quad \text{s.t. } \|\beta\|_1 \leq B$$

Geometric Intuition for Sparsity

overall obj.: $F(\beta) = RSS(\beta) + \lambda\|\beta\|$ ← 1 or 2 norm "lasso" "ridge"



Soft Threshholding

$c_j \propto \text{corr}(\underline{x}_j, r_{-j})$

all examples of feature j covariate

residual from model w/o using jth covariate

strength of penalty

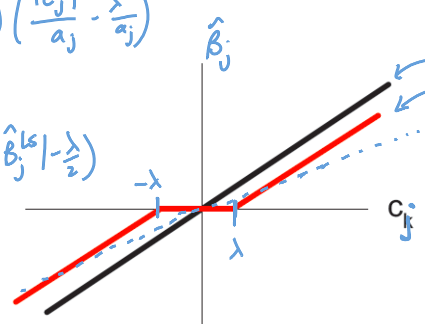
$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases}$$

$$= \text{sign}\left(\frac{c_j}{a_j}\right)\left(\frac{|c_j|}{a_j} - \frac{\lambda}{a_j}\right)$$

If $X^T X = I$

$$\hat\beta_j^{lasso} = \text{sign}(\hat\beta_j^{LS})\left(|\hat\beta_j^{LS}| - \frac{\lambda}{2}\right)$$

$$\hat\beta_j^{ridge} = \frac{\hat\beta^{LS}}{1+\lambda}$$



From
Kevin Murphy
textbook

In LASSO, all coeff. are shrunk relative to $\hat\beta^{LS}$

Acknowledgements

- Some material in this lecture was based on slides provided by:
  - Tom Mitchell – fMRI
  - Rob Tibshirani – LASSO

