Case Study 3: fMRI Prediction

fMRI Prediction Task, Ridge, LASSO Review Fused LASSO, LARS

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

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

Emily Fox

January 30th, 2014

fMRI Prediction Task

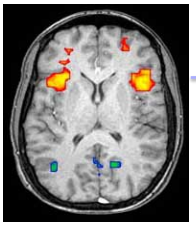
- **Goal:** Predict word stimulus from fMRI image
- **Challenges:**
  - p >> N (feature dimension >> sample size)
  - Cost of fMRI recordings is high
  - Only have a few training examples for each word

# of voxels = # params

many more params than obs.

What can we do?

20,000 voxels (big P)



Classifier (logistic regression, kNN, …)

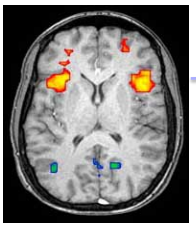
~~HAMMER~~ or HOUSE

few training example # = N

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

Semantic Features

side info =

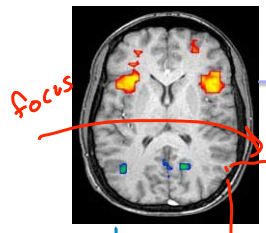
Google Trillion word corpus

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-occurrence

Zero-Shot Classification

- 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



Features of word

Classifier
(logistic regression, ~~kNN~~)

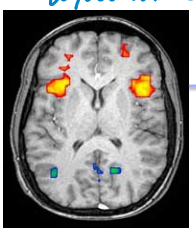
~~HAMMER~~
or
HOUSE

fMRI Prediction Subtask

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

Learning S : images → semantic features

20,000 voxels



Features of word

y^i

x^i

$x^i = \begin{bmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_{20,000}^i \end{bmatrix} \xrightarrow{S} \begin{bmatrix} y_1^i \\ y_2^i \\ \vdots \\ y_d^i \end{bmatrix}$

$x^i \in \mathbb{R}^{20,000}$, p

d = # of semantic features

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

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

Linear Regression

- Model: $y^i = \beta_0 + \beta_1 x_1^i + \dots + \beta_p x_p^i + \epsilon^i = \beta^T x^i + \epsilon^i$

  simplest model. Note: previously, wp β more common in stats. one feat. $\in \mathbb{R}$

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

  All obs: $\begin{bmatrix} y^1 \\ \vdots \\ y^N \end{bmatrix} = \begin{bmatrix} 1 & x_1^1 & \cdots & x_p^1 \\ \vdots & & & \vdots \\ 1 & x_1^N & \cdots & x_p^N \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon^1 \\ \vdots \\ \epsilon^N \end{bmatrix}$

  $y \quad X \quad \beta \quad \epsilon$

- MLE: $\hat{\theta} = \arg\max_\theta \log p(D \mid \theta) = \sum_{i=1}^N \log p(y^i \mid x^i, \theta) = \frac{-1}{2\sigma^2} \sum_i (y^i - \beta^T x^i)^2 + \text{const}$

  all obs. ind. obs. RSS(β)

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

  Here, p >> N so $X^T X$ low rank + we want its inverse...

- Minimizing RSS= least squares regression

Ridge Regression

- Ameliorating issues with overfitting:

  penalization of weights = "regularization"

- New objective: $\min_\beta \sum_{i=1}^N (y^i - (\beta_0 + \beta^T x^i))^2 + \lambda \|\beta\|_2^2$

  RSS(β); $\|\beta\|_2^2 \sim \beta^T \beta$ where $\beta = [\beta_1 \dots \beta_p]$

  don't want to penalize intercept; redefine X w/o 1's

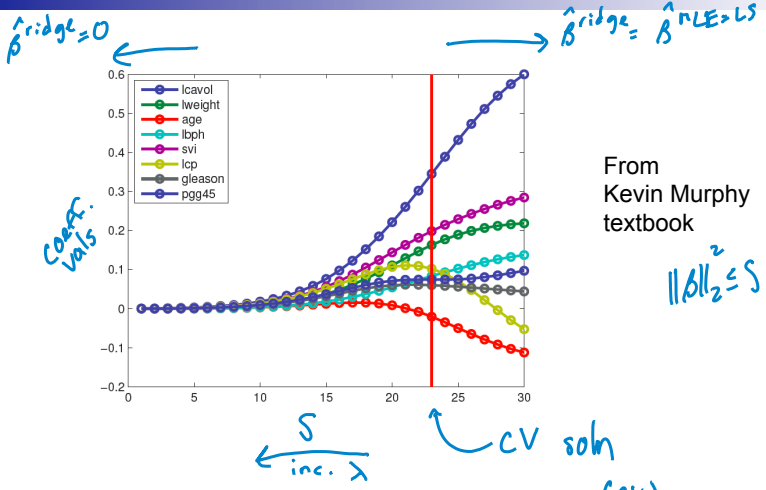
  $\Updownarrow$

  $\min_\beta RSS(\beta)$ s.t. $\|\beta\|_2^2 \le S$

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

    new term

Ridge Coefficient Path



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

January 30th, 2014

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
- 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

2 diff. sources of side info → compare performance using each

fMRI Prediction Results

- **First stage:** Learn mapping from images to semantic features
- Ridge regression

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

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

(From limited) training data: $\hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T F$ ← right soln if ind. assuming 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

look for word in B w/ f closest to $\hat{f}^{new}$ ← B

fMRI Prediction Results

- Leave-two-out-cross-validation
  - 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
corpus5000	79.6	67.0	69.5	56.2	77.7	65.5	71.2	72.9	67.9	**69.7**
human218	90.3	82.9	86.6	71.9	89.5	75.3	78.0	77.7	76.2	**80.9**

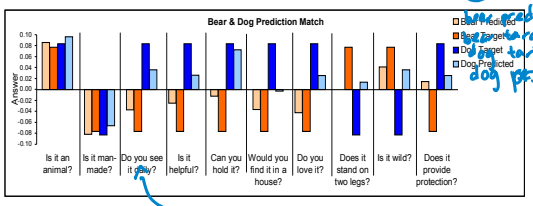


Figure 1: Ten semantic features from the human218 knowledge base for the words *bear* and *dog*. The true encoding is shown along with the predicted encoding when fMRI images for bear and dog were left out of the training set.

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

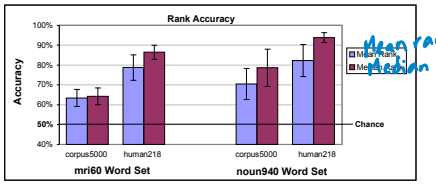


Figure 2: The mean and median rank accuracies across nine participants for two different semantic feature sets. Both the original 60 fMRI words and a set of 940 nouns were considered.

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

Case Study 3: fMRI Prediction

LASSO Review

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

Emily Fox

January 30th, 2014

Variable Selection

not min. this obj.; coeff are very sensitive to what was included in 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

discrete

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

  - Stepwise procedures are sensitive to data perturbations and often include features with negligible improvement in fit ← greed, but ∃ backtracking approaches
- Try new penalty: Penalize non-zero weights
  - Penalty: $\|\beta\|_1 = \sum_j |\beta_j|$ ℓ_1-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:

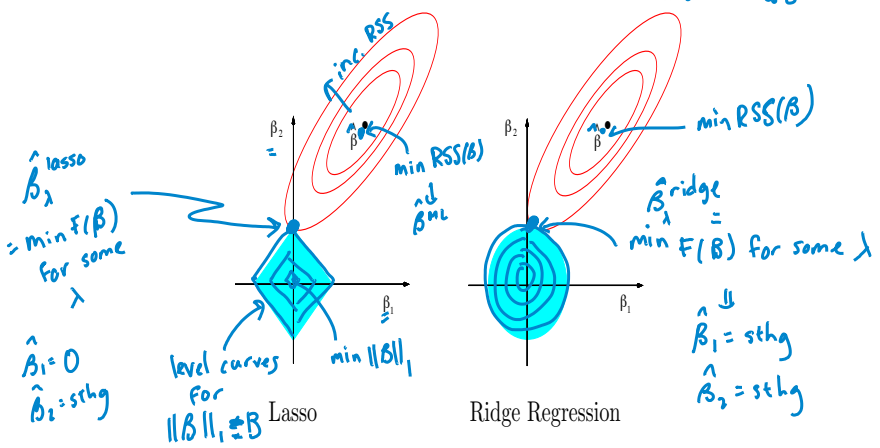
$$\min_{\beta} \sum_{i=1}^{N} \left(y^i - (\beta_0 + \beta^T x^i)\right)^2 + \lambda \|\beta\|_1$$

$$\Updownarrow$$

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

Geometric Intuition for Sparsity

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



Soft Threshholding

$c_j \propto \text{corr}(x_{\cdot j}, r_{-j})$ — all examples of feature j; residual from model w/o using j^{th} covariate

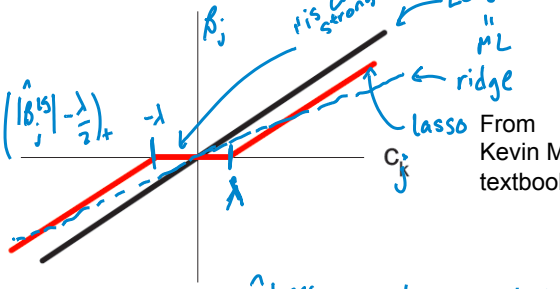
$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases}$$

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

If $X^T X = I$

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

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

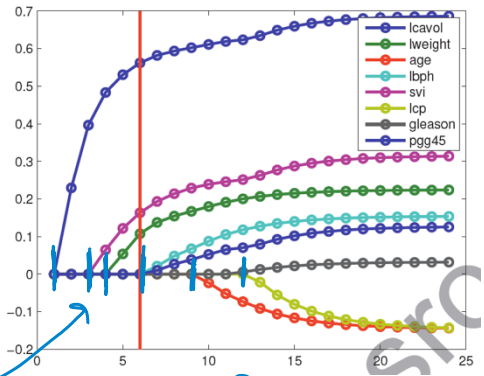


From Kevin Murphy textbook

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

LASSO Coefficient Path

Again, for each λ we have a diff. soln



From Kevin Murphy textbook

$\|\beta\|_1 \leq B$

a few critical points where coeff. enter model

CV soln

← inc. λ

LASSO Example

CV solns (red lines)

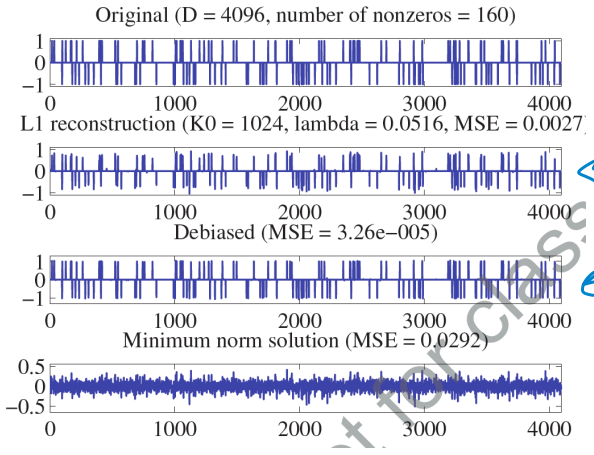
Term	Least Squares	Ridge	Lasso
Intercept	2.465	2.452	2.468
lcavol	0.680	0.420	0.533
lweight	0.263	0.238	0.169
age	−0.141	−0.046	
lbph	0.210	0.162	0.002
svi	0.305	0.227	0.094
lcp	−0.288	0.000	
gleason	−0.021	0.040	
pgg45	0.267	0.133	

$\hat{\beta}_0$, $\hat{\beta}_1$, ..., $\hat{\beta}_j$

not in the model

shrunk, but non-zero

Debiasing



From Kevin Murphy textbook

all coeff. shrunk → bias

Some people:

1. Use LASSO to find support

2. Run regression just w/ selected covariates

=) removes bias for this reduced model

Sparsistency

- Typical Statistical Consistency Analysis:
  - Holding model size (p) fixed, as number of samples (N) goes to infinity, estimated parameter goes to true parameter
- Here we want to examine $p >> N$ domains
- Let both model size p and sample size N go to infinity!
  - Hard case: $N = k \log p$

Sparsistency

- Rescale LASSO objective by N:
- Theorem (Wainwright 2008, Zhao and Yu 2006, …):
  - Under some constraints on the design matrix X, if we solve the LASSO regression using

    Then for some $c_1>0$, the following holds with at least probability
    - The LASSO problem has a unique solution with support contained within the true support
    - If $\min_{j \in S(\beta^*)} |\beta_j^*| > c_2 \lambda_n$ for some $c_2>0$, then $S(\hat{\beta}) = S(\beta^*)$

LASSO Algorithms

- Standard convex optimizer
- Least angle regression (LAR)
  - Efron et al. 2004
  - Computes entire path of solutions
  - State-of-the-art until 2008
- Pathwise coordinate descent – new
- More on these "shooting" algorithms next time…

Comments

- In general, can't solve analytically for GLM (e.g., logistic reg.)
  - Gradually decrease λ and use efficiency of computing $\hat{\beta}(\lambda_k)$ from $\hat{\beta}(\lambda_{k-1})$ = warm-start strategy
  - See Friedman et al. 2010 for coordinate ascent + warm-starting strategy
- If *N* > *p,* but variables are correlated, ridge regression tends to have better predictive performance than LASSO (Zou & Hastie 2005)
  - Elastic net is hybrid between LASSO and ridge regression

Acknowledgements

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