Case Study 3: fMRI Prediction

LASSO Regression

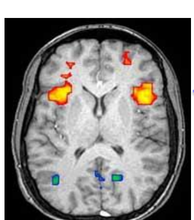
Machine Learning/Statistics for Big Data
CSE599C1/STAT592, University of Washington
Emily Fox

February 19th, 2013

fMRI Prediction Task

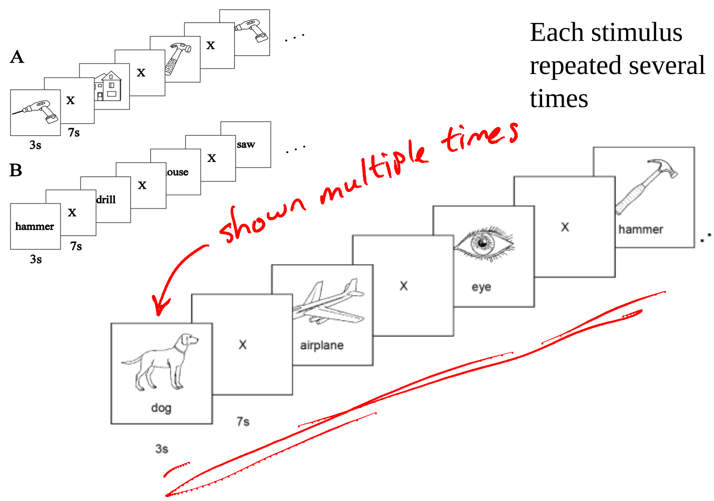
- **Goal:** Predict word stimulus from fMRI image

Can we read your brain?



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

Typical Stimuli

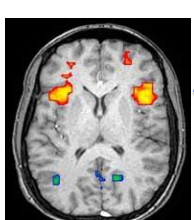


Each stimulus repeated several times

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 <u>HOUSE</u>

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

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

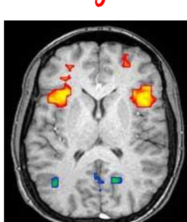
A = { [image] → "dog" } few

B = { [:] → "dog" } many

- Can use "cheap" co-occurrence data to help learn L (from B)

Training = { [image] → [:] ← "dog" } use both A + B

N examples … N small



Predict

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

new image → [:] → "giraffe" (using B)

S ← learned from training data

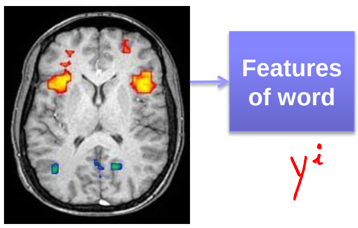
fMRI Prediction Subtask

challenge: $p >> N$ scenario

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

Learning S: images $\rightarrow$ semantic features

20,000 voxels



x^i $\quad$ y^i

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

let's consider each semantic feature separately $\in \mathbb{R}^{20,000}$

$d = \#$ of semantic features $\in \mathbb{R}^d$

Ridge Regression

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

same as in LS

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

don't penalize intercept term

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

- Reformulate:

$$F(\beta) = \underbrace{\tfrac{1}{2}\beta^T(X^TX)\beta - \beta^T(X^Ty) + \text{const.}}_{RSS(\beta)} + \tfrac{1}{2}\lambda\beta^T\beta$$

$$= \tfrac{1}{2}\beta^T(X^TX + \lambda I)\beta - \beta^T(X^Ty) + \text{const.}$$

- Set gradient = 0

$$\boxed{\hat{\beta}_{ridge} = (X^TX + \lambda I)^{-1}(X^Ty)}$$

Variable Selection

- Ridge regression: Penalizes large weights

min. this obj.; coeff. are very sensitive to what's inc. in model

- 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 .... can't do this

  - Stepwise procedures are sensitive to data perturbations and often include features with negligible improvement in fit ← greedy, but ∃ backtracking. –.

- Try new penalty: Penalize non-zero weights
  - Penalty:

$$\|\beta\|_1 = \sum_j |\beta_j| \qquad l_1\text{-reg.}$$

  - 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:

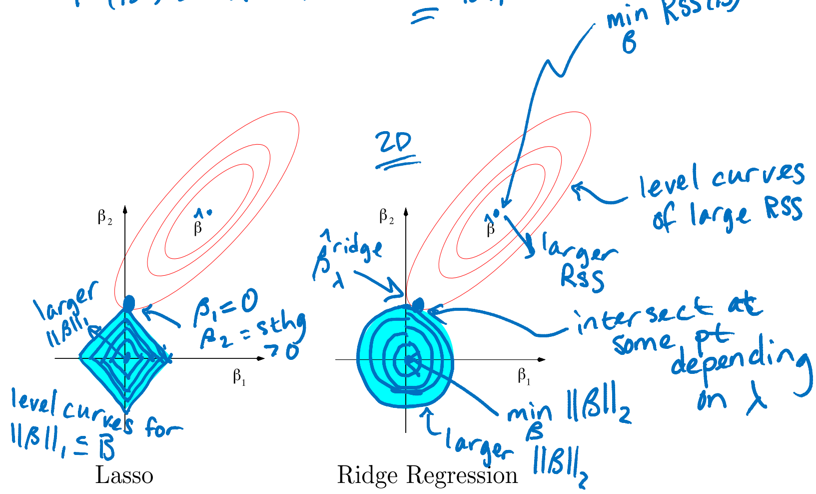
$$\min_\beta \underbrace{\sum_{i=1}^{N} (y^i - (\beta_0 + \beta^T x^i))^2}_{RSS(\beta)} + \lambda \|\beta\|_1$$

$$\Updownarrow$$

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

Geometric Intuition for Sparsity

$F(\beta) = RSS(\beta) + \lambda \|\beta\|$

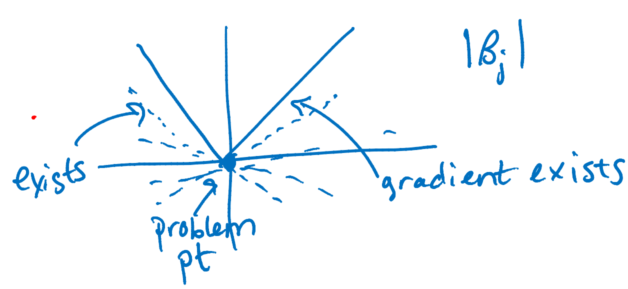


Soft Threshholding

- To see why LASSO results in sparse solutions, look at conditions that must hold at optimum

look at β_j fixing all others

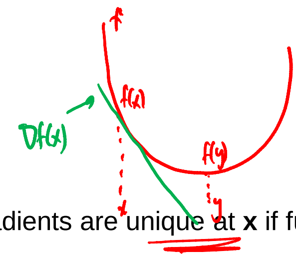
- L1 penalty $||\beta||_1$ is not differentiable whenever $\beta_j = 0$
- Look at subgradient…



From Case Study 1:

Subgradients of Convex Functions

- Gradients lower bound convex functions:



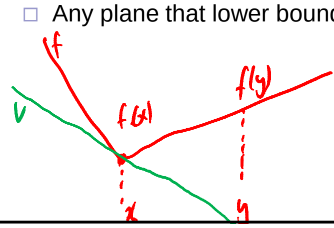
$$f(y) \geq f(x) + \nabla f(x)(y-x)$$

- Gradients are unique at **x** if function differentiable at **x**
- Subgradients: Generalize gradients to non-differentiable points:
  - Any plane that lower bounds function:

$v \in \partial f(x)$ subgradient

if

$$f(y) \geq f(x) + v \cdot (y-x)$$



Soft Threshholding

$F(\beta) = RSS(\beta) + \lambda\|\beta\|_1$

- Gradient of RSS term:

$$\frac{\partial}{\partial \beta_j} RSS(\beta) = a_j \beta_j - c_j$$

$$2\sum_{i=1}^{N} (x_j^i)^2$$

$$2\sum_{i=1}^{N} x_j^i \left(y^i - \beta_{-j}^T x_{-j}^i\right)$$

all but the jth coeff.

$c_j \propto corr(x_j, r_{-j})$

msr how relevant x_j is for pred y beyond what the others can

residual from model w/o jth cov.

- Subgradient of full objective:

$$\partial_{\beta_j} F(\beta) = (a_j \beta_j - c_j) + \lambda\, \partial_{\beta_j} \|\beta\|_1$$

$$= \begin{cases} a_j\beta_j - c_j - \lambda & \beta_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \beta_j = 0 \\ a_j \beta_j - c_j + \lambda & \beta_j > 0 \end{cases}$$

Soft Threshholding

- Set subgradient = 0:

$$\partial_{\beta_j} F(\beta) = \begin{cases} a_j\beta_j - c_j - \lambda & \beta_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \beta_j = 0 \\ a_j\beta_j - c_j + \lambda & \beta_j > 0 \end{cases}$$

$= 0$

If $\beta_j < 0$

$a_j\beta_j - c_j - \lambda = 0$

$\Rightarrow \beta_j = \frac{c_j + \lambda}{a_j} < 0 \Rightarrow c_j < -\lambda$ strong neg. corr., then $\beta_j < 0$

If $\beta_j > 0$

$a_j\beta_j - c_j + \lambda = 0 \Rightarrow \beta_j = \frac{c_j - \lambda}{a_j} > 0 \Rightarrow c_j > \lambda$ strong pos. corr., then $\beta_j > 0$

If $\beta_j = 0$ $\quad -\lambda < c_j < \lambda$ $\quad$ otherwise, $\beta_j = 0$

- The value of $c_j = 2\sum_{i=1}^{N} x_j^i (y^i - \beta'_{-j} x_{-j}^i)$ constrains β_j

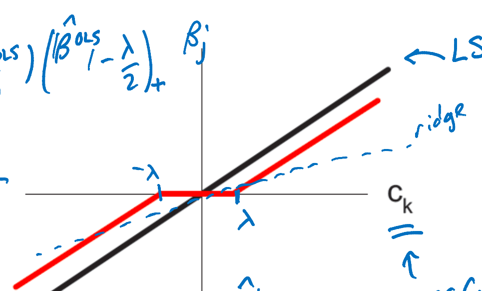
Soft Threshholding

$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases} = \text{sign}\left(\frac{c_j}{a_j}\right)\left(\frac{|c_j|}{a_j} - \frac{\lambda}{a_j}\right)_+$$

If $X^T X = I$

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

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



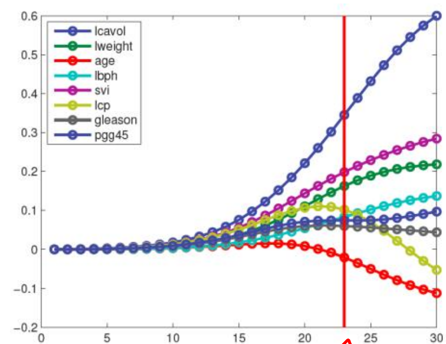
From Kevin Murphy textbook

$= \text{corr}(x_j, r_{-j})$

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

Recall: *Ridge Coefficient Path*

coeff.



From
Kevin Murphy
textbook

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

S

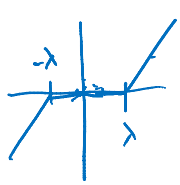
CV soln

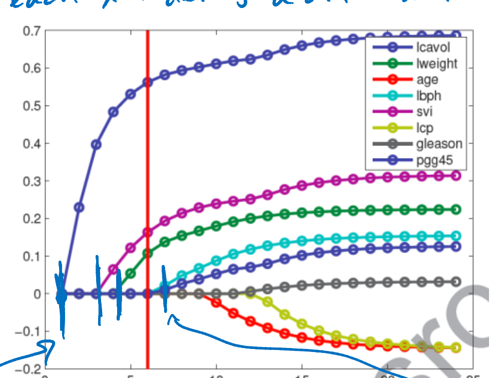
inc. λ

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

Now: *LASSO Coefficient Path*

Again, each λ indexes a diff. soln





From
Kevin Murphy
textbook

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

there are only a few critical values of λ where new cov. enter model

CV soln

B

inc. λ

solns are sparse for any given λ

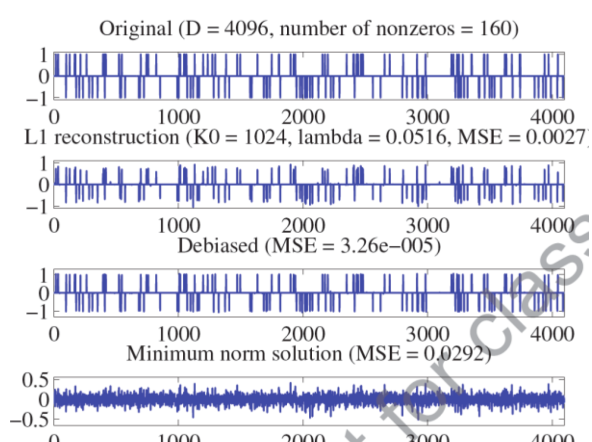
LASSO Example

red line (CV) solns

	Term	Least Squares	Ridge	Lasso
$\hat{\beta}_0$	Intercept	2.465	2.452	2.468
$\hat{\beta}_1$	lcavol	0.680	0.420	0.533
	lweight	0.263	0.238	0.169
$\vdots$	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	
$\hat{\beta}_p$	pgg45	0.267	0.133	

not in model

Debiasing



From Kevin Murphy textbook

LS est.

all coeff. shrunk
→ bias

Some people:

1. Use LASSO to find support
2. Run regression just w/ the selected cov.

⇒ removes bias for this 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

$$\hat{\theta} \rightarrow \theta^* \quad ?$$

- Here we want to examine $p >> N$ domains
- Let both model size p and sample size N go to infinity!
  - Hard case: $\boxed{N = k \log p}$

N grows slowly relative to p

Sparsistency

- Rescale LASSO objective by N:

$$\min_{\beta} \frac{1}{N} RSS(\beta) + \lambda_N \sum_j |\beta_j|$$

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

$$\lambda_N > \frac{2}{\gamma}\sqrt{\frac{2\sigma^2 \log p}{N}}$$

Then for some $c_1>0$, the following holds with at least probability

$$1 - 4\exp(-c_1 N \lambda_N^2) \rightarrow 1 :$$

- The LASSO problem has a unique solution with support contained within the true support $S(\hat{\beta}) \subseteq S(\beta^*)$
- If $\min_{j \in S(\beta^*)} |\beta_j^*| > c_2 \lambda_N$ for some $c_2>0$, then $\boxed{S(\hat{\beta}) = S(\beta^*)}$

Acknowledgements

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