Module 1: Nonparametric Preliminaries

LASSO cont’d

fMRI Prediction Subtask

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

Features of word

\[ \hat{\beta} = (X'X)^{-1} X'Y \]

\[ X \]

\[ X_i \]

\[ \cdots \]

\[ X_p \]

\[ \begin{bmatrix} X_i \\ \vdots \\ X_p \end{bmatrix} \rightarrow Y_{ij} \]

\[ j \text{th semantic feature} \]

\[ \text{# training examples} \]

\[ p \gg n \]

\[ \text{p > 0 voxels} \]

\[ \text{20,000} \]
Regularization in Linear Regression

- Overfitting usually leads to very large parameter choices, e.g.:
  \[-2.2 + 3.1 \, X - 0.30 \, X^2\]
  \[-1.1 + 4,700,910.7 \, X - 8,585,638.4 \, X^2 + \ldots\]

- Regularized or penalized regression aims to impose a "complexity" penalty by penalizing large weights
  - "Shrinkage" method

Ridge Regression

- Ameliorating issues with overfitting:

- New objective:
  \[
  \min_\beta \sum_{i=1}^n (y_i - (\beta_0 + \beta^T X_i))^2 + \lambda \lVert \beta \rVert_2^2
  \]

  - Don’t penalize intercept
  - Strength of penalty
  - Regularization of weights
  - Minimize \( RSS(\beta) \)
  - s.t. \( \lVert \beta \rVert_2^2 \leq S \)
Variable Selection

- 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

- Try new penalty: Penalize non-zero weights
  - Penalty: \[ L_1 \sum_j |\beta_j| \]
  - Leads to sparse solutions
  - Just like ridge regression, solution is indexed by a continuous param \( \lambda \)

LASSO Regression

- LASSO: least absolute shrinkage and selection operator

- New objective:
  \[ \min_{\beta} \left( \sum_i (y_i - \beta_0 - \beta_j x_i)^2 + \lambda \sum_j |\beta_j| \right) \]
  \[ \text{s.t. } \sum_j |\beta_j| \leq B \]
Geometric Intuition for Sparsity

\[ F(\beta) = \text{RSS}(\beta) + \lambda \| \beta \|_1 \]

Picture of Lasso and Ridge regression

\[ \beta \]

Soft Threshholding

- To see why LASSO results in sparse solutions, look at conditions that must hold at optimum
  - look at \( \beta_j \) ... do this for all \( j \) => set of simultaneous equations
  - \( L_1 \) penalty \( \| \beta \|_1 \) is not differentiable whenever \( \beta_j = 0 \)
  - Look at subgradient...
Subgradients of Convex Functions

- Gradients lower bound convex functions:
  \[ \frac{F(y) - F(x)}{y - x} \geq \nabla F(x) \]
  \[ \Rightarrow F(y) \geq F(x) + \nabla F(x) \cdot (y - x) \]

- Gradients are unique at \( x \) if function differentiable at \( x \)

- Subgradients: Generalize gradients to non-differentiable points:
  - Any plane that lower bounds function:
    \[ \text{For } |x_j|: \forall e \in [-1, 1] \]
    \[ \forall e \ n \text{d}F(x) \text{ subgrad. if } F(y) \geq F(x) + e(y - x) \]

Soft Thresholding

- Gradient of RSS term:
  \[ \frac{\partial}{\partial \beta_j} \text{RSS}(\beta) = a_j \beta_j - c_j \]

- Subgradient of full objective:
  \[ \partial \beta_j F(\beta) : (a_j \beta_j - c_j) + \lambda \beta_j \| \beta \|_1 \]
  \[ = \begin{cases} 
  a_j \beta_j - c_j - \lambda & \beta_j < 0 \\
  [c_j - \lambda, -c_j + \lambda] & \beta_j = 0 \\
  a_j \beta_j - c_j + \lambda & \beta_j > 0 
  \end{cases} \]
Soft Threshholding

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

  - If \( \beta_j < 0 \):
    \[ a_j \beta_j - c_j = 0 \Rightarrow \beta_j = \frac{c_j}{a_j} \leq 0 \Rightarrow c_j < \lambda \]
    If strong neg. corr., then \( \hat{\beta}_j < 0 \)

  - If \( \beta_j > 0 \):
    \[ a_j \beta_j - c_j + \lambda = 0 \Rightarrow \beta_j = \frac{c_j - \lambda}{a_j} > 0 \Rightarrow c_j > \lambda \]
    If strong pos. corr., then \( \hat{\beta}_j > 0 \)

  - If \( \beta_j = 0 \):
    \[ \lambda < c_j < \lambda \]
    if not strong corr., then \( \hat{\beta}_j = 0 \)

- The value of \( c_j = 2 \sum_{i=1}^{N} x_i^j (y_i - \hat{x}_{-j}^j \hat{x}_{-j}^j) \) constrains \( \beta_j \)

---

Soft Threshholding

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

- If \( \lambda^T X = 1 \):
  \[ \hat{\beta}_{\text{ridge}} = \frac{R_{\text{ridge}}}{1 + \lambda} \]

  \[ \hat{\beta}_{\text{lasso}} = \text{sign}(\hat{\beta}_{\text{ridge}}) \left( \frac{|c_j| - \lambda}{a_j} \right)_+ \]

- In lasso, all coeff. \( \hat{\beta}_{\text{lasso}} \) are shrunk relative to \( \hat{\beta}_{\text{ridge}} \)

From Kevin Murphy textbook
Coordinate Descent

- Given a function $F(\beta)$
  - Want to find minimum $\beta^* = \min_{\beta} F(\beta) \leftarrow F(\beta_1, \ldots, \beta_p)$
- Often, hard to find minimum for all coordinates, but easy for one coordinate
- Coordinate descent:
  - while not converged
  - pick coord. $j$
  - $\beta_j \leftarrow \min_{b} F(\beta_1, \ldots, \beta_{j-1}, b, \beta_{j+1}, \ldots, \beta_p)$
- How do we pick a coordinate?
  - Round robin, randomly, smartly, ...
- When does this converge to optimum?
  - e.g. strongly convex, separability

Stochastic Coordinate Descent for LASSO (aka Shooting Algorithm)

- Repeat until convergence
  - Pick a coordinate $j$ at random
  - Set: $\hat{\beta}_j = \begin{cases} 
  \frac{(c_j + \lambda)}{a_j} & c_j < -\lambda \\
  0 & c_j \in [-\lambda, \lambda] \\
  \frac{(c_j - \lambda)}{a_j} & c_j > \lambda 
  \end{cases}
  = \text{sign}(c_j) \frac{(|c_j| - \lambda)^+}{a_j}$
- Where: $c_j = 2 \sum_{i=1}^{N} (y_i - \hat{\beta}_j x_i)$
- For convergence rates, see Shalev-Shwartz and Tewari 2009
- Other common technique = LARS
  - Least angle regression and shrinkage, Efron et al. 2004
Recall: Ridge Coefficient Path

Typical approach: select $\lambda$ using cross validation

Now: LASSO Coefficient Path

Sols are sparse for any given $\lambda$
# LASSO Example

<table>
<thead>
<tr>
<th>Term</th>
<th>Least Squares</th>
<th>Ridge</th>
<th>Lasso</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_0$</td>
<td>2.465</td>
<td>2.452</td>
<td>2.468</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>0.680</td>
<td>0.420</td>
<td>0.533</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td>0.263</td>
<td>0.238</td>
<td>0.169</td>
</tr>
<tr>
<td>. age</td>
<td>-0.141</td>
<td>-0.046</td>
<td></td>
</tr>
<tr>
<td>. lbph</td>
<td>0.210</td>
<td>0.162</td>
<td>0.002</td>
</tr>
<tr>
<td>. svi</td>
<td>0.305</td>
<td>0.227</td>
<td>0.094</td>
</tr>
<tr>
<td>. lcp</td>
<td>-0.288</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>. gleason</td>
<td>-0.021</td>
<td>0.040</td>
<td></td>
</tr>
<tr>
<td>. pgg45</td>
<td>0.267</td>
<td>0.133</td>
<td></td>
</tr>
</tbody>
</table>

From Rob Tibshirani slides

---

# Sparsistency

- Typical Statistical Consistency Analysis:
  - Holding model size ($p$) fixed, as number of samples ($n$) goes to infinity, estimated parameter goes to true parameter:
    $\hat{\Theta} \to \Theta^*$ as $n \to \infty$
  - Here we want to examine $p >> n$ domains
  - Let both model size $p$ and sample size $n$ go to infinity!
    - Hard case: $n = k \log p$
      - $n$ 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
    \[
    \lambda_n > \frac{2}{n} \sqrt{2 \log p}
    \]
    Then for some $c_1 > 0$, the following holds with at least probability
    \[
    1 - 4 \exp \left( -c_1 n \lambda_n^2 \right) \rightarrow 1:
    \]
    - The LASSO problem has a unique solution with support contained within the true support
      $S(\hat{\beta}_{\text{LASSO}}) \subseteq S(\beta^*)$
    - If $\min_{j \in S(\beta^*)} |\beta_j^*| > c_2 \lambda_n$, for some $c_2 > 0$, then $S(\hat{\beta}) = S(\beta^*)$

Comments

- In general, can’t solve analytically for GLM (e.g., logistic reg.)
  - Gradually decrease $\lambda$ 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
    \[
    ||y - X\hat{\beta}||_2^2 + \lambda_1 \sum_j |\hat{\beta}_j| + \lambda_2 ||\hat{\beta}||_2^2
    \]
    (still some issues, but other solns)
Fused LASSO

- Might want coefficients of neighboring voxels to be similar
- How to modify LASSO penalty to account for this?

Graph-guided fused LASSO
- Assume a 2d lattice graph connecting neighboring pixels in the fMRI image
- Penalty:

\[ \|y - X\beta\|^2 + \lambda_1 \sum_{j \in E} |\beta_j| + \lambda_2 \sum_{(j, k) \in E} |\beta_j - \beta_k| \]

A Bayesian Formulation

- Consider a model with likelihood

\[ y_i | \beta \sim N(\beta_0 + x_i^T \beta; \sigma^2) \]

and prior

\[ \beta_j \sim \text{Lap}(\beta_j; \lambda) \]

where

\[ \text{Lap}(\beta_j; \lambda) = \frac{\lambda}{2} e^{-\lambda |\beta_j|} \]

- For large \( \lambda \)

more peaked around 0

- LASSO solution is equivalent to the mode of the posterior
- Note: posterior mode ≠ posterior mean in this case

any given posterior sample is not sparse, but it will be penalized like in ridge.

There is no closed-form for the posterior. Rely on approx. methods.
Reading

- Hastie, Tibshirani, Friedman: 3.4, 3.8.6

What you should know

- LASSO objective
- Geometric intuition for differences between ridge and LASSO solutions
- How LASSO performs soft thresholding
- Shooting algorithm
- Idea of sparsity
- Ways in which other L1 and L1-Lp objectives can be encoded
  - Elastic net
  - Fused LASSO