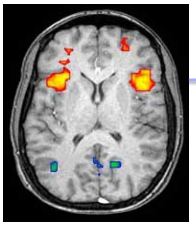
Module 1: Nonparametric Preliminaries

LASSO cont'd

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 9th, 2013

fMRI Prediction Subtask

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



Features of word

y_i

x_i

x_{i1} … x_{ip} → y_{ij} semantic feature

p = # voxels = 20,000

only small n

$p \gg n$

$\hat{\beta}^{LS} = (X^T X)^{-1} X^T y$

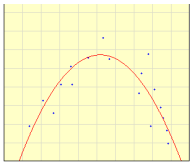
$X^T X$: $p \times p$, X^T: $p \times n$, y: n

rank deficient

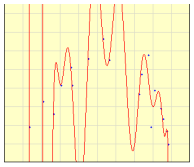
Regularization in Linear Regression

- 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 + …



even for $n \gg p$, p large

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

Ridge Regression

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

$$\min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta^T x_i))^2 + \lambda \|\beta\|_2^2$$

β_0: don't penalize the intercept

$\beta^T\beta$

λ: strength of the penalty

$$\Updownarrow$$

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

Variable Selection

- Ridge regression: Penalizes large weights
- What if we want to perform "feature selection"? (variable)
  - 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... can't do this
  - Stepwise procedures are sensitive to data perturbations and often include features with negligible improvement in fit ← greedy, ∄ backtracking alg.
- Try new penalty: Penalize non-zero weights
  - Penalty: $\|\beta\|_1 = \sum_j |\beta_j|$
  - Leads to sparse solutions
  - Just like ridge regression, solution is indexed by a continuous param λ

- not min this obj.
- coeff. sensitive to what's in c. in the model

discrete

LASSO Regression

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

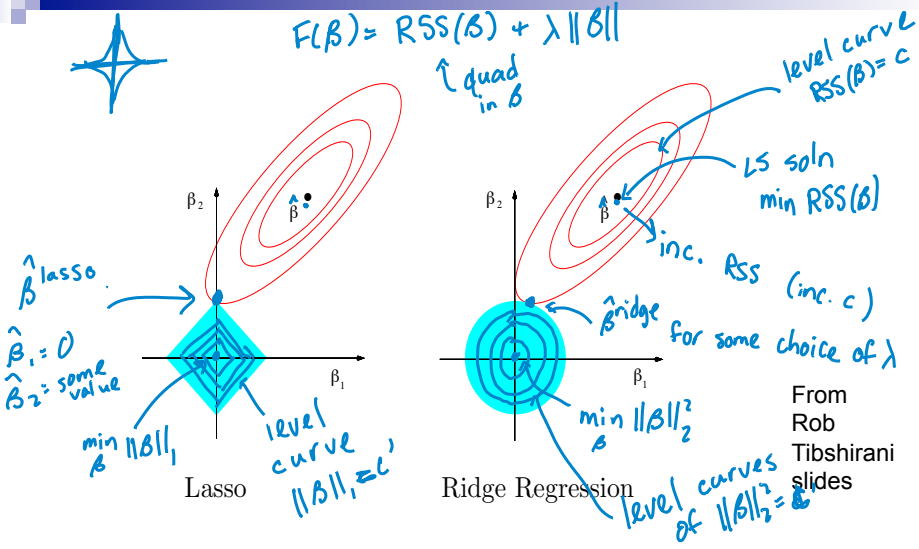
$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta^T x_i)\right)^2 + \lambda \|\beta\|_1$$

$$\sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta^T x_i)\right)^2 = RSS(\beta)$$

$$\Updownarrow$$

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

Geometric Intuition for Sparsity



From Rob Tibshirani slides

Soft Threshholding

- To see why LASSO results in sparse solutions, look at conditions that must hold at optimum
- L_1 penalty $||\beta||_1$ is not differentiable whenever $\beta_j = 0$
- Look at subgradient…

Subgradients of Convex Functions

- Gradients lower bound convex functions:

- Gradients are unique at **x** if function differentiable at **x**

- Subgradients: Generalize gradients to non-differentiable points:
  - Any plane that lower bounds function:

Soft Threshholding

- Gradient of RSS term:

- Subgradient of full objective:

Soft Threshholding

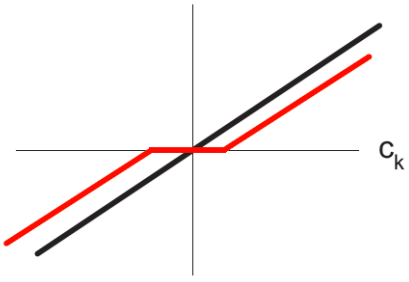
- Set subgradient = 0:

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

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

Soft Threshholding

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



From Kevin Murphy textbook

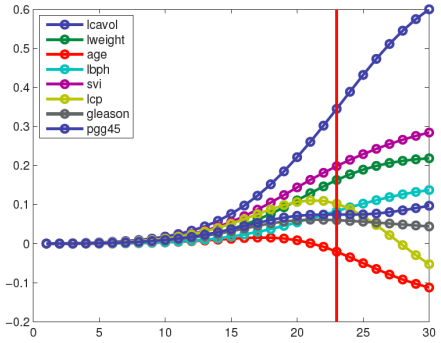
Coordinate Descent

- Given a function F
  - Want to find minimum
- Often, hard to find minimum for all coordinates, but easy for one coordinate
- Coordinate descent:
- How do we pick a coordinate?
- When does this converge to optimum?

Stochastic Coordinate Descent for LASSO (aka Shooting Algorithm)

- Repeat until convergence
  - Pick a coordinate *j* at random
    - Set:
$$\hat{\beta}_j = \begin{cases} (c_j + \lambda)/a_j & c_j < -\lambda \\ 0 & c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & c_j > \lambda \end{cases}$$
    - Where:
$$a_j = 2\sum_{i=1}^{N} (x_j^i)^2 \qquad c_j = 2\sum_{i=1}^{N} x_j^i (y^i - \beta_{-j}' x_{-j}^i)$$
  - For convergence rates, see Shalev-Shwartz and Tewari 2009
- Other common technique = LARS
  - Least angle regression and shrinkage, Efron et al. 2004

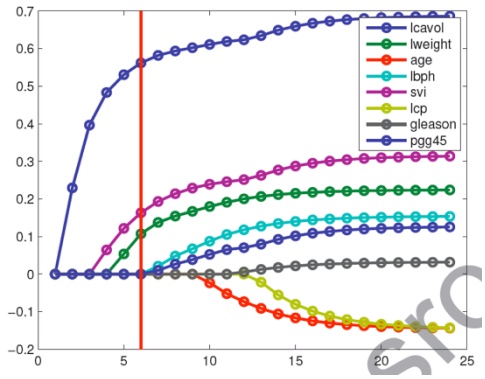
Recall: *Ridge Coefficient Path*



From
Kevin Murphy
textbook

- Typical approach: select λ using cross validation

Now: *LASSO Coefficient Path*



From
Kevin Murphy
textbook

LASSO Example

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	

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
- 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

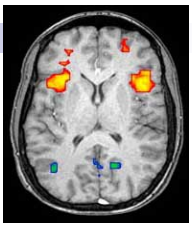
    - The LASSO problem has a unique solution with support contained within the true support
    - If $\min_{j \in S(\beta^*)} |\beta_j^*| > c_2 \lambda_n$ for some $c_2>0$, then $S(\hat{\beta}) = S(\beta^*)$

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

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:

A Bayesian Formulation

- Consider a model with likelihood

$$y_i \mid \beta \sim N(\beta_0 + x_i^T \beta, \sigma^2)$$

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 λ
- LASSO solution is equivalent to the ***mode*** of the posterior
- Note: posterior mode ≠ posterior mean in this case
- There is no closed-form for the posterior. Rely on approx. methods.

Module 1: Nonparametric Preliminaries

Selecting Smoothing Parameters

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 9th, 2013

Smoothing Parameter

- In both ridge and lasso regression, we saw that the parameter λ controlled the solution
  - Often, can straightforwardly equate with effective degrees of freedom
- Which λ ($\rightarrow$ estimator) should we choose???

Two Goals

- **Model Selection:** estimating the performance of models in order to select the best one
  - E.g., choosing λ
- **Model Assessment:** having chosen a final model, estimate its prediction error (generalization error) on new data
- Ideally, divide data into 3 parts

Focus on Model Selection

- Which estimator/smoothing parameter should we choose?

- Recall metrics for assessing the performance of an estimator…

Measuring Predictive Performance

- Assume estimate $\hat{f}_n(\cdot)$ based on training data $y_1, \ldots, y_n$

↑ fixed

- The ***generalization error*** provides a measure of predictive performance

$$GE(\hat{f}_n) = E_{Y,X}\left[L(Y, \hat{f}_n(X))\right]$$

↑ fixed

Measuring Predictive Performance

- Assume L_2 loss

$Y = f(X) + \epsilon$ ★ $E[\epsilon] = 0$ $\text{var}(\epsilon) = \sigma^2$

- Averaging over repeat training sets $\mathbf{Y_n} = Y_1, \ldots, Y_n$ we get the ***predictive risk*** at x^*

$$E_{Y^*,\mathbf{Y}_n}\left[(Y^* - \hat{f}_n(x^*))^2\right] = E_{Y^*,Y_n}\left[(Y^* - f(x^*) + f(x^*) - \hat{f}_n(x^*))^2\right]$$

test, training, ↑ fcn of training data Y_n

$$= E_{Y^*}\left[(Y^* - f(x^*))^2\right] + E_{Y_n}\left[(\hat{f}_n(x^*) - f(x^*))^2\right] + 2\,E_{Y^*}[Y^* - f(x^*)]\,E_{Y_n}[\hat{f}_n(x^*) - f(x^*)]$$

(the last term $= 0$)

$$= \sigma^2 + MSE(\hat{f}_n(x^*))$$

↑ "irreducible error" ↑ "risk" ✓

- Recall $MSE[\hat{f}_n(x)] = \text{bias}(\hat{f}_n(x))^2 + \text{var}(\hat{f}_n(x))$

Measuring Predictive Performance

- Finally, let's average over covariates *x*
  - ***Integrated MSE*** $\int MSE(\hat{f}_n(x))\, p(x)\, dx$ summary over all inputs
  - ***Average MSE*** $\frac{1}{n}\sum_{i=1}^{n} MSE(\hat{f}_n(x_i))$ Monte Carlo est. $x_i \sim p$
- Note: ***avg. pred. risk*** = σ^2 + ***avg. MSE***

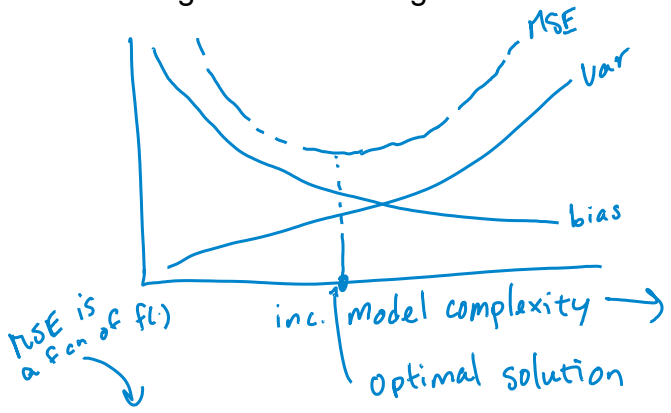
$$\uparrow \quad \frac{1}{n}\sum_{i=1}^{n} E_{Y_n, Y_n^*}\left[(Y_i^* - \hat{f}(x_i))^2\right]$$

Y_n: training; Y_n^*: new obs. $Y_n^* = Y_1^*, \ldots, Y_n^*$

Bias-Variance Tradeoff

recall polynomial reg. example

- Minimizing risk = balancing bias and variance



- Note: *f(x) is unknown, so cannot actually compute MSE*

Focus on Model Selection

- Which estimator/smoothing parameter should we choose?

- We saw that minimizing (average) prediction error can be equated with minimizing (average) MSE
- With a validation set, we can estimate the prediction error

Data Scarce Approximations

- Often, we do not have enough data to form suitably sized training and validation sets
  - What is a good training/test split? Sensitivity?
  - Typically want to use as much data for training as possible
- Rely on approximations

Approx 1: Training Data Only

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Use training error

Approx 2: Cross Validation

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Mimic heldout data using *training* data
- Leave-one-out (LOO) cross validation (CV) algorithm:
  - Estimate fit using all but i^{th} data point
  - Predict i^{th} observation
  - Repeat for all i
  - Repeat for all values of λ

Approx 2: Cross Validation

- Reasoning

- For linear smoothers

- Warning: Curves can be very flat…Don't just choose and use without thinking. Some rules of thumb (see Elements of Statistical Learning)

Approx 2: Cross Validation

- K-fold cross validation

- Algorithm
  1. Fit model using data with k^{th} fraction removed
  2. Using fitted model, compute

$$\mathrm{CV}_k = \frac{1}{n_k} \sum_{i \in J(k)} (y_i - \hat{f}^{\lambda}_{-k}(x_i))$$

  3. Store

$$\mathrm{CV} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{CV}_k$$

  4. Repeat for each value of λ using same split of the data

Approx 3: Generalized CV

- Recall LOO ordinary CV for linear smoothers

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - L_{ii}}\right)^2$$

- Instead of L_{ii}, use $\frac{1}{n}\sum_{i=1}^{n} L_{ii}$

- Often very close to OCV solution

Approx 3: Generalized CV

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2$$

- One motivation: Invariance to orthonormal transformations

Approx 3: Generalized CV

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^{\lambda}(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2$$

- Using $(1-x)^{-2} \approx 1 + 2x$

Approx 4: Mallows C_p Statistic

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Approximate directly

$$\text{avg. MSE} = \frac{1}{n}E\left[(f - \hat{f}_n^{\lambda})^T(f - \hat{f}_n^{\lambda})\right]$$

Approx 4: Mallows C_p Statistic

$$\text{avg. MSE} = \frac{1}{n} E\left[(Y - L^\lambda Y)^T (Y - L^\lambda Y)\right] - \sigma^2 + \frac{2}{n}\nu_\lambda \sigma^2$$

- Estimate as

- Note: Arises from considering L_2 loss. Log-likelihood loss leads to AIC. For BIC, consider Bayesian model selection

Bayesian Model Selection

- Assume some *M* possible models
  - Model M_m *m=1,…,M* has parameters θ_m and prior $p(\theta_m \mid M_m)$
  - Prior over models $p(M_m)$

- Model posterior

$$p(M_m \mid Z) \propto p(M_m) p(Z \mid M_m)$$
$$\propto p(M_m) \int p(Z \mid \theta_m, M_m) p(\theta_m \mid M_m) d\theta_m$$

- Compare models:

$$\frac{p(M_m \mid Z)}{p(M_\ell \mid Z)} = \frac{p(M_m) p(Z \mid M_m)}{p(M_\ell) p(Z \mid M_\ell)} \gtrless 1$$

Bayesian Model Selection

- For Bayes factor, approximate

$$\log p(Z \mid M_m) \approx \log p(Z \mid \hat{\theta}_m, M_m) - \frac{\nu_m}{2} \log n + O(1)$$

- If loss is $-2 \log p(Z \mid \hat{\theta}_m, M_m)$, then equivalent to BIC
  - Minimizing BIC = maximizing approximated posterior

- However, in addition to being able to select the best model, in Bayesian framework we also get the relative merit of each

$$\approx \frac{e^{-\frac{1}{2}\mathrm{BIC}_m}}{\sum_{\ell=1}^{M} e^{-\frac{1}{2}\mathrm{BIC}_\ell}}$$

- BIC is asymptotically consistent, but AIC is not
- For finite samples, BIC tends to choose too simple models