Regularization: Ridge Regression and the LASSO

Statistics 305: Autumn Quarter 2006/2007

Wednesday, November 29, 2006

Agenda

1. The Bias-Variance Tradeoff
2. Ridge Regression
    - Solution to the ℓ_2 problem
    - Data Augmentation Approach
    - Bayesian Interpretation
    - The SVD and Ridge Regression
3. Cross Validation
    - K-Fold Cross Validation
    - Generalized CV
4. The LASSO
5. Model Selection, Oracles, and the Dantzig Selector
6. References

Part I

The Bias-Variance Tradeoff

Estimating β

- As usual, we assume the model:

$$y = f(\mathbf{z}) + \varepsilon, \quad \varepsilon \sim (0, \sigma^2)$$

- In regression analysis, our major goal is to come up with some good regression function $\hat{f}(\mathbf{z}) = \mathbf{z}^\top \hat{\beta}$
- So far, we've been dealing with $\hat{\beta}^{\text{ls}}$, or the least squares solution:
  - $\hat{\beta}^{\text{ls}}$ has well known properties (e.g., Gauss-Markov, ML)
- But can we do better?

Choosing a good regression function

- Suppose we have an estimator $\hat{f}(\mathbf{z}) = \mathbf{z}^\top \hat{\beta}$
- To see if $\hat{f}(\mathbf{z}) = \mathbf{z}^\top \hat{\beta}$ is a good candidate, we can ask ourselves two questions:
  1.) Is $\hat{\beta}$ close to the true β?
  2.) Will $\hat{f}(\mathbf{z})$ fit future observations well?

1.) Is $\hat{\beta}$ close to the true β?

- To answer this question, we might consider the **mean squared error** of our estimate $\hat{\beta}$:
  - i.e., consider squared distance of $\hat{\beta}$ to the true β:

$$MSE(\hat{\beta}) \;=\; \mathbb{E}[||\hat{\beta} - \beta||^2] \;=\; \mathbb{E}[(\hat{\beta} - \beta)^\top (\hat{\beta} - \beta)]$$

- **Example:** In least squares (LS), we now that:

$$\mathbb{E}[(\hat{\beta}^{\text{ls}} - \beta)^\top (\hat{\beta}^{\text{ls}} - \beta)] \;=\; \sigma^2 \text{tr}[(\mathbf{Z}^\top \mathbf{Z})^{-1}]$$

2.) Will $\hat{f}(\mathbf{z})$ fit future observations well?

- Just because $\hat{f}(\mathbf{z})$ fits our data well, this doesn't mean that it will be a good fit to new data
- In fact, suppose that we take new measurements y_i' at the same $\mathbf{z}_i$'s:

$$(\mathbf{z}_1, y_1'), (\mathbf{z}_2, y_2'), \ldots, (\mathbf{z}_n, y_n')$$

- So if $\hat{f}(\cdot)$ is a good model, then $\hat{f}(\mathbf{z}_i)$ should also be close to the new target y_i'
- This is the notion of **prediction error** (PE)

Prediction error and the bias-variance tradeoff

- So good estimators should, on average have, small prediction errors
- Let's consider the PE at a particular target point $\mathbf{z}_0$ (see the board for a derivation):

$$\begin{aligned} \mathrm{PE}(\mathbf{z}_0) &= \mathbb{E}_{Y|\mathbf{Z}=\mathbf{z}_0}\{(Y - \hat{f}(\mathbf{Z}))^2|\mathbf{Z} = \mathbf{z}_0\} \\ &= \sigma_\varepsilon^2 + \mathrm{Bias}^2(\hat{f}(\mathbf{z}_0)) + \mathrm{Var}(\hat{f}(\mathbf{z}_0)) \end{aligned}$$

- Such a decomposition is known as the **bias-variance tradeoff**
  - As model becomes more complex (more terms included), local structure/curvature can be picked up
  - But coefficient estimates suffer from high variance as more terms are included in the model
- So introducing a little bias in our estimate for β might lead to a substantial decrease in variance, and hence to a substantial decrease in PE

Depicting the bias-variance tradeoff

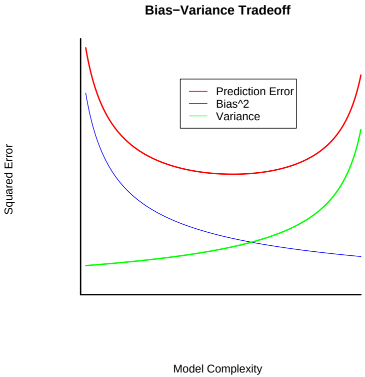

Figure: A graph depicting the bias-variance tradeoff.

Part II

Ridge Regression

Ridge regression as regularization

- If the β_j's are unconstrained...
  - They can explode
  - And hence are susceptible to very high variance
- To control variance, we might **regularize** the coefficients
  - i.e., Might control how large the coefficients grow
- Might impose the ridge constraint:

$$\text{minimize} \sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^\top \mathbf{z}_i)^2 \text{ s.t. } \sum_{j=1}^{p}\beta_j^2 \le t$$

$$\Leftrightarrow \text{ minimize } (y - \mathbf{Z}\beta)^\top(y - \mathbf{Z}\beta) \text{ s.t. } \sum_{j=1}^{p}\beta_j^2 \le t$$

- By convention (very important!):
  - $\mathbf{Z}$ is assumed to be standardized (mean 0, unit variance)
  - $\mathbf{y}$ is assumed to be centered

Ridge regression: ℓ_2-penalty

- Can write the ridge constraint as the following **penalized** residual sum of squares (PRSS):

$$\begin{aligned} PRSS(\beta)_{\ell_2} &= \sum_{i=1}^{n}(y_i - \mathbf{z}_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= (\mathbf{y} - \mathbf{Z}\beta)^\top (\mathbf{y} - \mathbf{Z}\beta) + \lambda ||\beta||_2^2 \end{aligned}$$

- Its solution may have smaller average PE than $\hat{\beta}^{\text{ls}}$
- $PRSS(\beta)_{\ell_2}$ is convex, and hence has a unique solution
- Taking derivatives, we obtain:

$$\frac{\partial PRSS(\beta)_{\ell_2}}{\partial \beta} = -2\mathbf{Z}^\top(y - \mathbf{Z}\beta) + 2\lambda\beta$$

The ridge solutions

- The solution to $PRSS(\hat{\beta})_{\ell_2}$ is now seen to be:

$$\hat{\beta}_{\lambda}^{\text{ridge}} = (\mathbf{Z}^{\top}\mathbf{Z} + \lambda \mathbf{I}_p)^{-1}\mathbf{Z}^{\top}\mathbf{y}$$

  - Remember that $\mathbf{Z}$ is standardized
  - $\mathbf{y}$ is centered
- Solution is indexed by the tuning parameter λ (more on this later)
- Inclusion of λ makes problem non-singular even if $\mathbf{Z}^{\top}\mathbf{Z}$ is not invertible
  - This was the original motivation for ridge regression (Hoerl and Kennard, 1970)

Tuning parameter λ

- Notice that the solution is indexed by the parameter λ
  - So for each λ, we have a solution
  - Hence, the λ's trace out a path of solutions (see next page)
- λ is the shrinkage parameter
  - λ controls the size of the coefficients
  - λ controls amount of **regularization**
  - As $\lambda \downarrow 0$, we obtain the least squares solutions
  - As $\lambda \uparrow \infty$, we have $\hat{\beta}^{\text{ridge}}_{\lambda=\infty} = 0$ (intercept-only model)

Ridge coefficient paths

- The λ's trace out a set of ridge solutions, as illustrated below

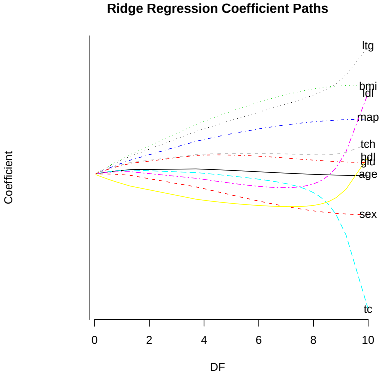

Figure: Ridge coefficient path for the `diabetes` data set found in the `lars` library in R.

Choosing λ

- Need disciplined way of selecting λ:
- That is, we need to "tune" the value of λ
- In their original paper, Hoerl and Kennard introduced **ridge traces**:
    - Plot the components of $\hat{\boldsymbol{\beta}}_\lambda^{\text{ridge}}$ against λ
    - Choose λ for which the coefficients are not rapidly changing and have "sensible" signs
    - No objective basis; heavily criticized by many
- Standard practice now is to use cross-validation (defer discussion until Part 3)

Proving that $\hat{\beta}_\lambda^{\text{ridge}}$ is biased

- Let $\mathbf{R} = \mathbf{Z}^\top \mathbf{Z}$
- Then:

$$
\begin{aligned}
\hat{\beta}_\lambda^{\text{ridge}} &= (\mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{I}_p)^{-1} \mathbf{Z}^\top \mathbf{y} \\
&= (\mathbf{R} + \lambda \mathbf{I}_p)^{-1} \mathbf{R} (\mathbf{R}^{-1} \mathbf{Z}^\top \mathbf{y}) \\
&= [\mathbf{R}(\mathbf{I}_p + \lambda \mathbf{R}^{-1})]^{-1} \mathbf{R} [(\mathbf{Z}^\top \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{y}] \\
&= (\mathbf{I}_p + \lambda \mathbf{R}^{-1})^{-1} \mathbf{R}^{-1} \mathbf{R} \hat{\beta}^{\text{ls}} \\
&= (\mathbf{I}_p + \lambda \mathbf{R}^{-1}) \hat{\beta}^{\text{ls}}
\end{aligned}
$$

- So:

$$
\begin{aligned}
\mathbb{E}(\hat{\beta}_\lambda^{\text{ridge}}) &= \mathbb{E}\{(\mathbf{I}_p + \lambda \mathbf{R}^{-1}) \hat{\beta}^{\text{ls}}\} \\
&= (\mathbf{I}_p + \lambda \mathbf{R}^{-1}) \beta \\
&\overset{(\text{if } \lambda \neq 0)}{\neq} \beta.
\end{aligned}
$$

Data augmentation approach

- The ℓ_2 PRSS can be written as:

$$\begin{aligned} PRSS(\boldsymbol{\beta})_{\ell_2} &= \sum_{i=1}^{n}(y_i - \mathbf{z}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= \sum_{i=1}^{n}(y_i - \mathbf{z}_i^\top \boldsymbol{\beta})^2 + \sum_{j=1}^{p}(0 - \sqrt{\lambda}\beta_j)^2 \end{aligned}$$

- Hence, the ℓ_2 criterion can be recast as another least squares problem for another data set

Data augmentation approach continued

- The ℓ_2 criterion is the RSS for the augmented data set:

$$\mathbf{Z}_\lambda = \begin{pmatrix} z_{1,1} & z_{1,2} & z_{1,3} & \cdots & z_{1,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ z_{n,1} & z_{n,2} & z_{n,3} & \cdots & z_{n,p} \\ \sqrt{\lambda} & 0 & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda} & 0 & \cdots & 0 \\ 0 & 0 & \sqrt{\lambda} & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \sqrt{\lambda} \end{pmatrix} ; \ \mathbf{y}_\lambda = \begin{pmatrix} y_1 \\ \vdots \\ y_n \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- So:

$$\mathbf{Z}_\lambda = \begin{pmatrix} \mathbf{Z} \\ \sqrt{\lambda}\mathbf{I}_p \end{pmatrix} \ \mathbf{y}_\lambda = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}$$

Solving the augmented data set

- So the "least squares" solution for the augmented data set is:

$$\begin{aligned}(\mathbf{Z}_\lambda^\top \mathbf{Z}_\lambda)^{-1}\mathbf{Z}_\lambda^\top y_\lambda &= \left((\mathbf{Z}^\top, \sqrt{\lambda}\mathbf{I}_p)\begin{pmatrix}\mathbf{Z}\\ \sqrt{\lambda}\mathbf{I}_p\end{pmatrix}\right)^{-1}(\mathbf{Z}^\top, \sqrt{\lambda}\mathbf{I}_p)\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix}\\ &= (\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^\top\mathbf{y},\end{aligned}$$

which is simply the ridge solution

Bayesian framework

- Suppose we imposed a multivariate Gaussian prior for β:

$$\beta \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{2p}\mathbf{I}_p\right)$$

- Then the posterior mean (and also posterior mode) of β is:

$$\beta_\lambda^{\text{ridge}} = (\mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{I}_p)^{-1} \mathbf{Z}^\top \mathbf{y}$$

Computing the ridge solutions via the SVD

- Recall $\hat{\beta}_\lambda^{\text{ridge}} \;=\; (\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^\top\mathbf{y}$
- When computing $\hat{\beta}_\lambda^{\text{ridge}}$ numerically, matrix inversion is avoided:
  - Inverting $\mathbf{Z}^\top\mathbf{Z}$ can be computationally expensive: $O(p^3)$
- Rather, the *singular value decomposition* is utilized; that is,

$$\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{V}^\top,$$

where:

- $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p)$ is an $n \times p$ orthogonal matrix
- $\mathbf{D} = \text{diag}(d_1, d_2, \ldots, \geq d_p)$ is a $p \times p$ diagonal matrix consisting of the singular values $d_1 \geq d_2 \geq \cdots d_p \geq 0$
- $\mathbf{V}^\top = (\mathbf{v}_1^\top, \mathbf{v}_2^\top, \ldots, \mathbf{v}_p^\top)$ is a $p \times p$ matrix orthogonal matrix

Numerical computation of $\hat{\beta}_\lambda^{\text{ridge}}$

- Will show on the board that:

$$\begin{aligned}\hat{\beta}_\lambda^{\text{ridge}} &= (\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^\top\mathbf{y} \\ &= \mathbf{V}\,\text{diag}\left(\frac{d_j}{d_j^2+\lambda}\right)\mathbf{U}^\top\mathbf{y}\end{aligned}$$

- Result uses the eigen (or spectral) decomposition of $\mathbf{Z}^\top\mathbf{Z}$:

$$\begin{aligned}\mathbf{Z}^\top\mathbf{Z} &= (\mathbf{UDV}^\top)^\top(\mathbf{UDV}^\top) \\ &= \mathbf{VD}^\top\mathbf{U}^\top\mathbf{UDV}^\top \\ &= \mathbf{VD}^\top\mathbf{DV}^\top \\ &= \mathbf{VD}^2\mathbf{V}^\top\end{aligned}$$

$\hat{\mathbf{y}}_\lambda^{\text{ridge}}$ and principal components

- A consequence is that:

$$\begin{aligned}\hat{\mathbf{y}}^{\text{ridge}} &= \mathbf{Z}\hat{\beta}_\lambda^{\text{ridge}} \\ &= \sum_{j=1}^{p}\left(\mathbf{u}_j \frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^\top\right)\mathbf{y}\end{aligned}$$

- Ridge regression has a relationship with principal components analysis (PCA):
  - **Fact:** The derived variable $\gamma_j = \mathbf{Z}\mathbf{v}_j = \mathbf{u}_j d_j$ is the jth principal component (PC) of $\mathbf{Z}$
  - Hence, ridge regression projects $\mathbf{y}$ onto these components with large d_j
  - Ridge regression shrinks the coefficients of low-variance components

Orthonormal **Z** in ridge regression

- If $\mathbf{Z}$ is orthonormal, then $\mathbf{Z}^\top\mathbf{Z} = \mathbf{I}_p$, then a couple of closed form properties exist
- Let $\hat{\boldsymbol{\beta}}^{\text{ls}}$ denote the LS solution for our orthonormal $\mathbf{Z}$; then

$$\hat{\boldsymbol{\beta}}_\lambda^{\text{ridge}} = \frac{1}{1+\lambda}\hat{\boldsymbol{\beta}}^{\text{ls}}$$

- The optimal choice of λ minimizing the expected prediction error is:

$$\lambda^* \;=\; \frac{p\sigma^2}{\sum_{j=1}^p \beta_j^2},$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ is the true coefficient vector

Smoother matrices and effective degrees of freedom

- A **smoother matrix S** is a linear operator satisfying:

$$\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$$

  - Smoothers put the "hats" on **y**
  - So the fits are a linear combination of the y_i's, $i = 1, \ldots, n$

- **Example:** In ordinary least squares, recall the hat matrix

$$\mathbf{H} = \mathbf{Z}(\mathbf{Z}^\top \mathbf{Z})^{-1}\mathbf{Z}^\top$$

  - For $\text{rank}(\mathbf{Z}) = p$, we know that $\text{tr}(\mathbf{H}) = p$, which is how many degrees of freedom are used in the model

- By analogy, define the **effective degrees of freedom** (or effective number of parameters) for a smoother to be:

$$\text{df}(\mathbf{S}) = \text{tr}(\mathbf{S})$$

Degrees of freedom for ridge regression

- In ridge regression, the fits are given by:

$$\hat{\mathbf{y}} \;=\; \mathbf{Z}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^\top\mathbf{y}$$

- So the smoother or “hat” matrix in ridge takes the form:

$$\mathbf{S}_\lambda = \mathbf{Z}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^\top$$

- So the *effective degrees of freedom* in ridge regression are given by:

$$\mathrm{df}(\lambda) \;= \mathrm{tr}(\mathbf{S}_\lambda) \;=\; \mathrm{tr}[\mathbf{Z}(\mathbf{Z}^\top\mathbf{Z} + \lambda\mathbf{I}_p)^{-1}\mathbf{Z}^\top] \;=\; \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

    - Note that $\mathrm{df}(\lambda)$ is monotone decreasing in λ
    - **Question:** What happens when $\lambda = 0$?

Part III

Cross Validation

How do we choose λ?

- We need a disciplined way of choosing λ
- Obviously want to choose λ that minimizes the mean squared error
- Issue is part of the bigger problem of **model selection**

Training sets versus test sets

- If we have a good model, it should predict well when we have new data
- In machine learning terms, we compute our statistical model $\hat{f}(\cdot)$ from the **training set**
- A good estimator $\hat{f}(\cdot)$ should then perform well on a new, independent set of data
- We "test" or assess how well $\hat{f}(\cdot)$ performs on the new data, which we call the **test set**

More on training and testing

- Ideally, we would separate our available data into both training and test sets
  - Of course, this is not always possible, especially if we have a few observations
- Hope to come up with the best-trained algorithm that will stand up to the test
  - Example: The Netflix contest (`http://www.netflixprize.com/`)
- How can we try to find the best-trained algorithm?

K-fold cross validation

- Most common approach is K-**fold cross validation**:
  (i) Partition the training data T into K separate sets of equal size
    - Suppose $T = (T_1, T_2, \ldots, T_K)$
    - Commonly chosen K's are $K = 5$ and $K = 10$

  (ii) For each $k = 1, 2, \ldots, K$, fit the model $\hat{f}_{-k}^{(\lambda)}(\mathbf{z})$ to the training set excluding the kth-fold T_k

  (iii) Compute the fitted values for the observations in T_k, based on the training data that excluded this fold

  (iv) Compute the cross-validation (CV) error for the k-th fold:

$$(\text{CV Error})_k^{(\lambda)} = |T_k|^{-1} \sum_{(\mathbf{z},y)\in T_k} (y - \hat{f}_{-k}^{(\lambda)}(\mathbf{z}))^2$$

K-fold cross validation (continued)

- The model then has overall cross-validation error:

$$(\text{CV Error})^{(\lambda)} = K^{-1}\sum_{k=1}^{K}(\text{CV Error})_k^{(\lambda)}$$

- Select λ^* as the one with minimum $(\text{CV Error})^{(\lambda)}$
- Compute the chosen model $\hat{f}(\mathbf{z})^{(\lambda^*)}$ on the entire training set $T = (T_1, T_2, \ldots, T_k)$
- Apply $\hat{f}(\mathbf{z})^{(\lambda^*)}$ to the test set to assess test error

Plot of CV errors and standard error bands

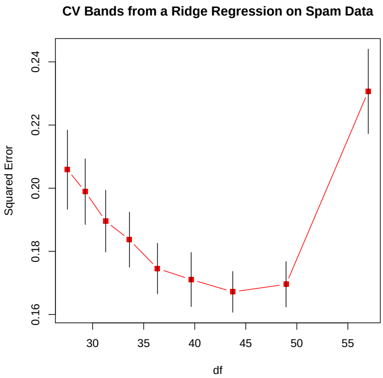

Figure: Cross validation errors from a ridge regression example on spam data.

Cross validation with few observations

- **Remark:** Our data set might be small, so we might not have enough observations to put aside a test set:
  - In this case, let all of the available data be our training set
  - Still apply K-fold cross validation
  - Still choose λ^* as the minimizer of CV error
  - Then refit the model with λ^* on the entire training set

Leave-one-out CV

- What happens when $K = 1$?
- This is called **leave-one-out cross validation**
- For squared error loss, there is a convenient approximation to CV(1), which is the leave one-out CV error

Generalized CV for smoother matrices

- Recall that a smoother matrix $\mathbf{S}$ satisfies:

$$\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$$

- In many linear fitting methods (as in LS), we have:

$$\mathrm{CV}(1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_{-i}(\mathbf{z}_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}(\mathbf{z}_i)}{1 - \mathbf{S}_{ii}}\right)^2$$

- A convenient approximation to CV(1) is called the **generalized cross validation**, or GCV error:

$$\mathrm{GCV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}(\mathbf{z}_i)}{1 - \frac{\mathrm{tr}(\mathbf{S})}{n}}\right)^2$$

  - Recall that tr($\mathbf{S}$) is the *effective degrees of freedom*, or *effective number of parameters*

Part IV

The LASSO

The LASSO: ℓ_1 penalty

- Tibshirani (*Journal of the Royal Statistical Society* 1996) introduced the **LASSO**: *least absolute shrinkage and selection operator*
- LASSO coefficients are the solutions to the ℓ_1 optimization problem:

$$\text{minimize } (\mathbf{y} - \mathbf{Z}\beta)^\top(\mathbf{y} - \mathbf{Z}\beta) \text{ s.t. } \sum_{j=1}^{p} |\beta_j| \leq t$$

- This is equivalent to loss function:

$$\begin{aligned} PRSS(\beta)_{\ell_1} &= \sum_{i=1}^{n}(y_i - \mathbf{z}_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= (\mathbf{y} - \mathbf{Z}\beta)^\top(\mathbf{y} - \mathbf{Z}\beta) + \lambda||\beta||_1 \end{aligned}$$

λ (or t) as a tuning parameter

- Again, we have a tuning parameter λ that controls the amount of regularization
- One-to-one correspondence with the threshhold t: recall the constraint:

$$\sum_{j=1}^{p} |\beta_j| \leq t$$

  - Hence, have a "path" of solutions indexed by t
  - If $t_0 = \sum_{j=1}^{p} |\hat{\beta}_j^{\text{ls}}|$ (equivalently, $\lambda = 0$), we obtain no shrinkage (and hence obtain the LS solutions as our solution)
  - Often, the path of solutions is indexed by a fraction of shrinkage factor of t_0

Sparsity and exact zeros

- Often, we believe that many of the β_j's should be 0
- Hence, we seek a set of **sparse solutions**
- Large enough λ (or small enough t) will set some coefficients exactly equal to 0!
    - So the LASSO will perform model selection for us!

Computing the LASSO solution

- Unlike ridge regression, $\hat{\beta}_{\lambda}^{\text{lasso}}$ has no closed form
- Original implementation involves quadratic programming techniques from convex optimization
- `lars` package in `R` implements the LASSO
- But Efron et al. (*Annals of Statistics* 2004) proposed LARS (**least angle regression**), which computes the LASSO path efficiently
  - Interesting modification called is called **forward stagewise**
  - In many cases it is the same as the LASSO solution
  - Forward stagewise is easy to implement: `http://www-stat.stanford.edu/~hastie/TALKS/nips2005.pdf`

Forward stagewise algorithm

- As usual, assume $\mathbf{Z}$ is standardized and $\mathbf{y}$ is centered
- Choose a small ε. The forward-stagewise algorithm then proceeds as follows:
  1. Start with initial residual $\mathbf{r} = \mathbf{y}$, and $\beta_1 = \beta_2 = \cdots = \beta_p = 0$.
  2. Find the predictor $\mathbf{Z}_j$ $(j = 1, \ldots, p)$ most correlated with $\mathbf{r}$
  3. Update $\beta_j \leftarrow \beta_j + \delta_j$, where $\delta_j = \varepsilon \cdot \text{sign}\langle \mathbf{r}, \mathbf{Z}_j \rangle = \varepsilon \cdot \text{sign}(\mathbf{Z}_j^\top \mathbf{r})$.
  4. Set $\mathbf{r} \leftarrow \mathbf{r} - \delta_j \mathbf{Z}_j$, and repeat Steps 2 and 3 many times.
- Try implementing forward stagewise yourself! It's easy!

Example: diabetes data

- Example taken from `lars` package documentation:

```
Call:
lars(x = x, y = y)
R-squared: 0.518
Sequence of LASSO moves:
     bmi ltg map hdl sex glu tc tch ldl age hdl hdl
Var    3   9   4   7   2  10  5   8   6   1  -7   7
Step   1   2   3   4   5   6  7   8   9  10  11  12
```

The LASSO, LARS, and Forward Stagewise paths

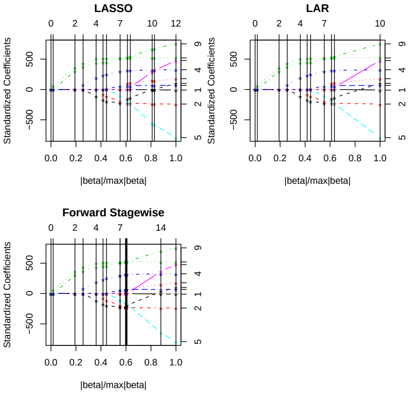

Figure: Comparison of the LASSO, LARS, and Forward Stagewise coefficient paths for the `diabetes` data set.

Part V

Model Selection, Oracles, and the Dantzig Selector

Comparing LS, Ridge, and the LASSO

- Even though $\mathbf{Z}^\top\mathbf{Z}$ may not be of full rank, both ridge regression and the LASSO admit solutions
- We have a problem when $p \gg n$ (more predictor variables than observations)
  - But both ridge regression and the LASSO have solutions
  - Regularization tends to reduce prediction error

Variable selection

- The ridge and LASSO solutions are indexed by the continuous parameter λ:
- Variable selection in least squares is "discrete":
    - Perhaps consider "best" subsets, which is of order $O(2^p)$ (combinatorial explosion – compare to ridge and LASSO)
    - Stepwise selection
        - In stepwise procedures, a new variable may be added into the model even with a miniscule improvement in R^2
        - When applying stepwise to a perturbation of the data, probably have different set of variables enter into the model at each stage
- Many model selection techniques based on Mallow's C_p, AIC, and BIC

More comments on variable selection

- Now suppose $p \gg n$
- Of course, we would like a parsimonious model (Occam's Razor)
- Ridge regression produces coefficient values for each of the p-variables
- But because of its ℓ_1 penalty, the LASSO will set many of the variables exactly equal to 0!
    - That is, the LASSO produces **sparse solutions**
- So LASSO takes care of model selection for us
    - And we can even see when variables jump into the model by looking at the LASSO path

Variants

- Zou and Hastie (2005) propose the **elastic net**, which is a convex combination of ridge and the LASSO
    - Paper asserts that the elastic net can improve error over LASSO
    - Still produces sparse solutions
- Frank and Friedman (1993) introduce **bridge regression**, which generalizes ℓ_q norms
- Regularization ideas extended to other contexts:
    - Park (Ph.D. Thesis, 2006) computes ℓ_1 regularized paths for generalized linear models

High-dimensional data and underdetermined systems

- In many modern data analysis problems, we have $p \gg n$
  - These comprise "high-dimensional" problems
- When fitting the model $y = \mathbf{z}^\top \boldsymbol{\beta}$, we can have many solutions
  - i.e., our system is *underdetermined*
- Reasonable to suppose that most of the coefficients are exactly equal to 0

S-sparsity and Oracles

- Suppose that only S elements of $\boldsymbol{\beta}$ are non-zero
  - Candès and Tao call this S-sparsity
- Now suppose we had an "Oracle" that told us which components of the $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)$ are truly non-zero
- Let $\boldsymbol{\beta}^\star$ be the least squares estimate of this "ideal" estimator;
  - So $\boldsymbol{\beta}^\star$ is 0 in every component that $\boldsymbol{\beta}$ is 0
  - The non-zero elements of $\boldsymbol{\beta}^\star$ are computed by regressing $\mathbf{y}$ on only the S important covariates

The Dantzig selector

- Candès and Tao developed the Dantzig selector $\hat{\beta}^{\text{Dantzig}}$:

$$\text{minimize} ||\beta||_{\ell_1} \text{ s.t. } ||\mathbf{Z}_j^\top \mathbf{r}||_{\ell_\infty} \leq (1 + t^{-1})\sqrt{2 \log p} \cdot \sigma$$

  - Here, $\mathbf{r}$ is the residual vector and $t > 0$ is a scalar

- They showed that with high probability,

$$||\hat{\beta}^{\text{Dantzig}} - \beta||^2 \ = \ O(\log p)\mathbb{E}(||\beta^* - \beta||^2)$$

- So the Dantzig selector does comparably well as someone who was told was S variables to regress on

Part VI

References

References

- Candès E. and Tao T. The Dantzig selector: statistical estimation when p is much larger than n. Available at `http://www.acm.caltech.edu/~emmanuel/papers/DantzigSelector.pdf`.
- Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, **32** (2): 409–499.
- Frank, I. and Friedman, J. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, **35**, 109–148.
- Hastie, T. and Efron, B. The `lars` package. Available from `http://cran.r-project.org/src/contrib/Descriptions/lars.html`.

References continued

- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Series in Statistics.
- Hoerl, A.E. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**: 55-67
- Seber, G. and Lee, A. (2003). Linear Regression Analysis, 2nd Edition. Wiley Series in Probability and Statistics.
- Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B.* **67**: pp. 301–320.