CS 5785 – Applied Machine Learning – Lec. 7

Prof. Nathan Kallus, Cornell Tech

September 13, 2018

1 Changing Model Complexity in OLS: Subset Selection

How to select the best k features (out of p)?

$$\hat{\beta} = argmin||X - X_{\beta}||_2^2$$
$$\hat{\beta}_{bestk} = argmin||Y - X_{\beta}||_2^2$$

BSR and FSR are heuristic solutions that are computationally faster

BSR:

- Start with $S = 1....P$
- while $|S|$ is not equal to k:
  - Remove j with the smallest $|Z_j|$
    $Z_j = \hat{\beta}/(\sigma * \sqrt{V_j})$
    $V_j = (X^T * X)^{-1}_{jj}$
    $\sigma = 1/(N - k - 1)\sum(Y_i - \hat{Y}_i)$

FSR:

- Start with $S = emptyset$
- while $|S|$ is not equal to k:
  - Find $j^* = argmin||Y - X||_2^2$
  - Add j^* to S

At what k do we stop?
The AML Approach: use cross-validation!
CV is an approach to estimate R(A) Split the data into k folds:
1...n=S_1...S_k such that any two are disjoint
$||S_i| - |S_i|| \leq 1$
$\hat{f}^j = A((x_i, y_i)$: i is not equal to $S_j)$
$CV^j = 1/|S_j| * \sum_i(loss(Y_i, \hat{f}^{(j)}(x_i))$
$\hat{R}^{cv}(A)$=cross validation estimate of loss in algorithm $= 1/k * \sum_{j=1}^{k}(CV^j)$

Collection of algorithms $A_1 ... A_m$
How to choose?
Naive approach-choose algorithm with smallest estimated risk. The more principled approach known as the "one standard error" rule of thumb:

$$\text{Std Error } (\hat{R}_{cv}(A)) = 1/k * \sum_{k=1}^{k} * \sqrt{(\hat{R}^{cv}(A) - CV^i * A)^2}$$

Pick the "simplest" algorithm with $\hat{R}^{cv}$ within one standard error of the minimum one.
Simplest:

- least number of variables
- least complexity
- least higher order dependence
- least covariance
- least variable (knn with larger k)

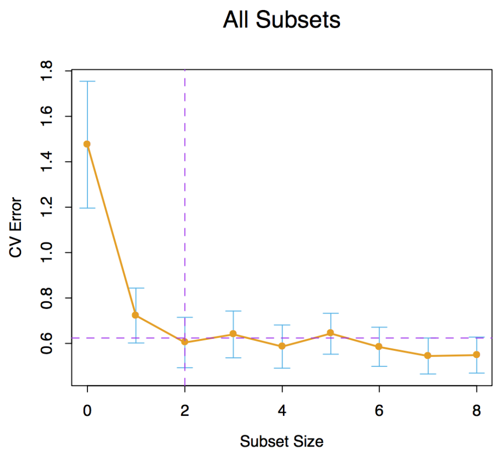

Figure 1: Prediction error curve using all subsets. Model complexity increases moving to the right.

2 Shrinkage

Subset Selection is very discrete. It's good for interpretation, but potentially (maybe not) bad for prediction.
Shrinkage is a more continuous way to trade off more bias for less variance.

$$\hat{\beta}^{ridge} = argmin||Y - X_\beta||_2^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Shrinks $\hat{\beta}^{OLS}$ (the most complex linear model) foward sample mean of X (simplest prediction we can have)

At one extreme $\lambda = 0$: $\hat{\beta}^{ridge} = \hat{\beta}^{OLS}$
At other extreme $\lambda = \infty$: $\hat{\beta}^{ridge}$ = intercept at sample mean of X

Shrinking to 0 prevents β from trying to reach far off outliers with extreme slopes.

Rewrite $\lambda \sum_{j=1}^{p} \beta_j^2 = \hat{\beta}^T \Lambda \beta$ where $\Lambda = \begin{pmatrix} 0 & \cdots & 0 \\ \lambda & & \\ & \ddots & \\ 0 & & \lambda \end{pmatrix}$ diagonal matrix

So that...
$\hat{\beta}^{ridge} = argmin(Y - X_\beta)^T(Y - X_\beta) + \beta^T \Lambda \beta$
$\Delta((Y - X_\beta)^T(Y - X_\beta) + \beta^T \Lambda \beta) = -2X^T(Y - X_\beta) + 2\Lambda = 0$
$\Rightarrow X^T Y = (X^T X + \Lambda)\beta$
$\Rightarrow \hat{\beta}^{ridge} = (X^T X + \Lambda)^{-1} X^T Y$

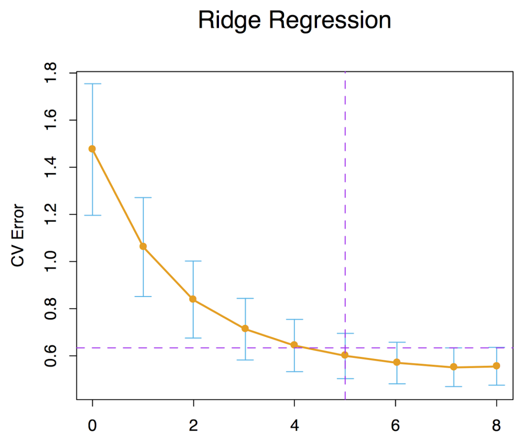

Figure 2: Prediction error curve using ridge regression.

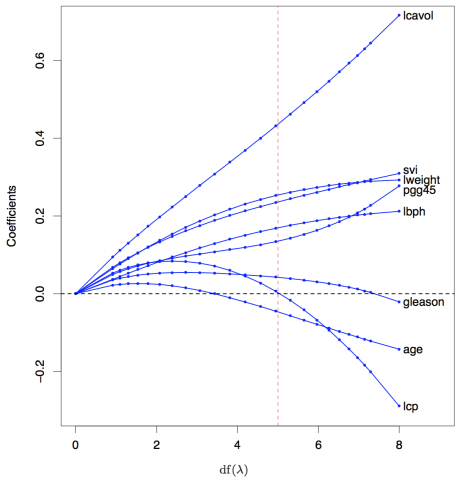

Figure 3: Ridge regression coefficients as determined by varying λ. Vertical line is chosen using cross validation

3 Lasso Regression

The idea behind Lasso Regression is to combine Shrinkage and Subset Selection

$$\hat{\beta}^{Lasso} = argmin||Y - X_{\beta}||_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Lasso, unike ridge and OLS, has no closed form solution. Fortunately, we can compute $\hat{\beta}^{Lasso}$ for all λ simultaneously.

In sklearn, we can use:

sklearn.linear_model.lasso_path

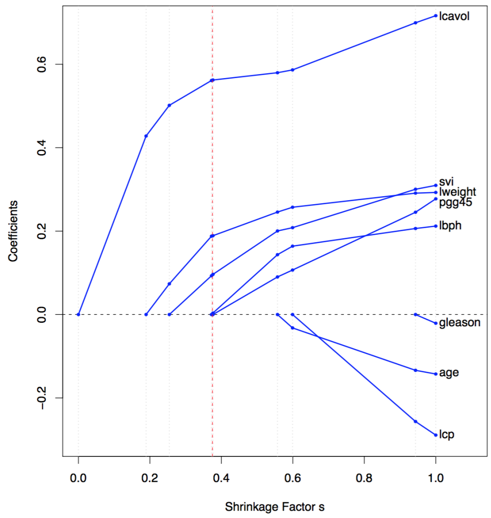

Figure 4: Lasso coefficients as determined by varying λ. Vertical line is chosen using cross validation

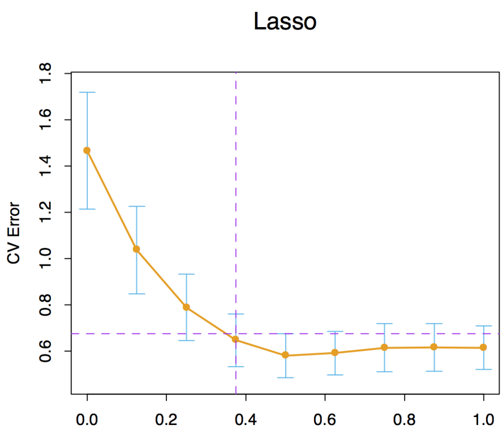

Figure 5: Prediction error curve using lasso.