STA314, Lecture 5

October 03, 2019
Stanislav Volgushev

AIC vs BIC and consistent model selection I

Example illustrating why BIC can perform consistent model selection and AIC in general won't:

- true model $y_i = \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0,1)$. x_i fixed (non-random) one-dimensional predictors unrelated to y_i.
- Use AIC and BIC to select between model with no predictor and model with x_i as predictors.
- denote by $RSS(\mathcal{M}_0)$ residual sum of squares for model with just intercept and by $RSS(\mathcal{M}_1)$ residual sum of squares for model including x_i.
- Assume that we plug in the true value for $\sigma^2 = 1$ in AIC and BIC.

For simplicity assume $\overline{x} = n^{-1} \sum_i x_i = 0$. Then

$$\hat{b}_1 = \bar{y}, \quad \hat{b}_2 = \frac{\overline{xy}}{\overline{x^2}} = \frac{n^{-1} \sum_{i=1}^{n} x_i y_i}{n^{-1} \sum_{i=1}^{n} x_i^2}$$

This implies (see derivation in class) $RSS(\mathcal{M}_0) - RSS(\mathcal{M}_1) \sim \chi_1^2$ i.e. $RSS(\mathcal{M}_0) - RSS(\mathcal{M}_1)$ follows a chi-squared distribution with one degree of freedom.

AIC vs BIC and consistent model selection II

From previous slide: $RSS(\mathcal{M}_0) - RSS(\mathcal{M}_1) \sim \chi^2_1$.

AIC selects model correct $\mathcal{M}_0$ if and only if

$$RSS(\mathcal{M}_0) + 2\sigma^2 < RSS(\mathcal{M}_1) + 4\sigma^2 \Leftrightarrow \chi^2_1 < 2$$

The probability of this is roughly 0.84. Otherwise the 'wrong' model is selected.

BIC selects model correct $\mathcal{M}_0$ if and only if

$$RSS(\mathcal{M}_0) + \sigma^2 \log n < RSS(\mathcal{M}_1) + 2\sigma^2 \log n \Leftrightarrow \chi^2_1 < \log n.$$

The probability of this is depends on n but goes to 1 for $n \to \infty$ (why?).

Exercise: verify this by simulations in **R**.

Summary of results so far

We discussed best subset, forward stepwise and backward stepwise selection.

- Given p possible predictors, all three procedures produce candidate models $\mathcal{M}_0, ..., \mathcal{M}_p$ with $0, 1, ..., p$ predictors.
- Procedures differ in the way that candidate models are selected and in resulting computational cost.
- Best subset selection considers 2^p models, which is infeasible for large p. Forward and backward stepwise selection only considers $1 + p(p+1)/2$ models but is not guaranteed to find best model for a given number of predictors.

The last step of each procedure is to compare models $\mathcal{M}_0, ..., \mathcal{M}_p$ while taking into account their complexity (number of predictors). We discussed several 'analytic corrections' for this comparison.

- Basic idea of corrections: account for smaller training error in more flexible models by adding a term that increases with number of predictors (i.e. with model flexibility).

Next: cross-validation instead of analytic correction.

The right way to do cross-validation for model selection

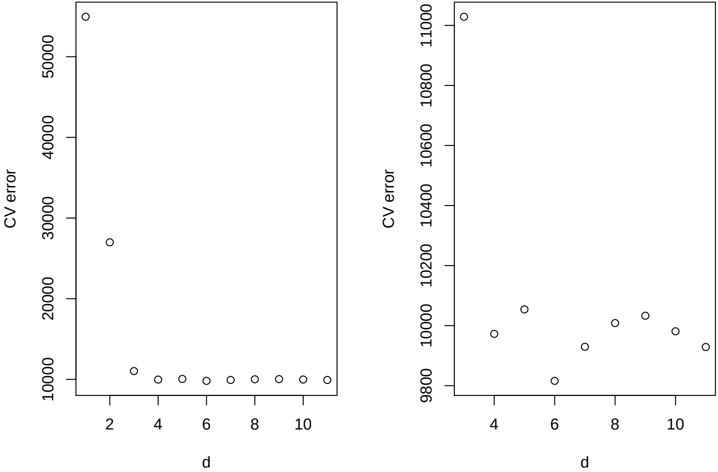
For performing K-fold cross-validation, split data into K roughly equal folds. Repeat the following steps for $k = 1, ..., K$

1. Run step 1 and step 2 of any of the selection procedures described earlier using the data not contained in fold d to obtain models $\mathcal{M}_0, ..., \mathcal{M}_p$.
2. Use each of the models $\mathcal{M}_0, ..., \mathcal{M}_p$ to compute prediction errors on the data in the k'th fold.

For each number of predictors $0, ..., p$ the above procedure gives a prediction error within each fold. Average those errors (over folds for each d) and select the number of predictors which gives the smallest error.

- Intuition: the number of predictors corresponds to a 'tuning parameter'.
- Models with same number of predictors but different sets of predictors have similar complexity and thus same tuning parameter.
- Advantage compared to AIC and BIC: no estimation of variance $\hat{\sigma}^2$ needed. Disadvantage: more computation.
- Wrong way: determining candidate models $\mathcal{M}_0, ..., \mathcal{M}_p$ on full data set and using cross-validation afterwards.

10-fold cross-validation for predicting `Balance` in Credit data set



Which number of predictors *d* would cross-validation choose?

The general one standard error rule

Assume we have collection of estimators $\widehat{f_\ell}$ where $\ell \in \mathcal{L}$ denotes a tuning parameter.

1. $\widehat{f_\ell}$ for k-nn: ℓ corresponds to k.
2. Linear models with different number of predictors: ℓ is the number of predictors.

Assume that for each $\ell \in \mathcal{L}$, K-fold cross-validation returns estimated errors on each of the K folds. Call those errors $e_1(\ell), ..., e_K(\ell)$.

---

1. Find ℓ_0 which minimizes $K^{-1}\sum_{k=1}^K e_k(\ell)$ (classical cross validation would output this as solution).

2. Compute $\widehat{sd}(\ell_0)$, the sample standard deviation of $e_1(\ell_0), ..., e_K(\ell_0)$.

3. Among all ℓ that satisfy

$$K^{-1}\sum_{k=1}^{K} e_k(\ell) \leq K^{-1}\sum_{k=1}^{K} e_k(\ell_0) + (K-1)^{-1/2}\widehat{sd}(\ell_0)$$

select the ℓ that corresponds to the least flexible model.

The general one standard error rule II

1. Find ℓ_0 which minimizes $K^{-1}\sum_{k=1}^{K} e_k(\ell)$ (classical cross validation would output this as solution).

2. Compute $\widehat{sd}(\ell_0)$, the sample standard deviation of $e_1(\ell_0), ..., e_K(\ell_0)$.

3. Among all ℓ that satisfy

$$K^{-1}\sum_{k=1}^{K} e_k(\ell) \leq K^{-1}\sum_{k=1}^{K} e_k(\ell_0) + (K-1)^{-1/2}\widehat{sd}(\ell_0)$$

   select the ℓ that corresponds to the least flexible model.

- k-nn: larger k correspond to less flexible models, so we select the largest k.
- Linear models with different number of predictors: smaller number of predictors correspond to less flexible models, so we select model with smallest number of predictors.
- This general rule can be used whenever we prefer less flexible models that still have a good predictive performance. Will see more examples later.

R examples part 2: file `STA314-Lecture04-CV-ModelSelection`

Alternative penalty methods: motivation

An equivalent formulation of the best subset selection procedure for AIC is (exercise: prove this!)

$$\hat{b}^{AIC} = \text{argmin}_{b \in \mathbb{R}^{p+1}} \left\{ RSS(b) + 2\hat{\sigma}^2 \#\{j = 2, .., p+1 : b_j \neq 0\} \right\}$$

Similarly, an equivalent formulation for BIC is (exercise: prove this!)

$$\hat{b}^{BIC} = \text{argmin}_{b \in \mathbb{R}^{p+1}} \left\{ RSS(b) + (\log n)\hat{\sigma}^2 \#\{j = 2, .., p+1 : b_j \neq 0\} \right\}$$

- The term $\#\{j = 2, .., p+1 : b_j \neq 0\}$ corresponds to the number of predictors used in the model. It can be viewed as measuring 'model complexity' (more predictors corresponds to more complex models).
- The two above minimization problems correspond to minimizing the sum of RSS (test error) and a penalty term that goes up with model complexity. This is called *regularization* or *penalized regression*.

There are other useful ways of measuring model complexity. The *lasso* and *ridge* penalties that follow are motivated by this idea.

The LASSO

Least **A**bsolute **S**hrinkage and **S**election **O**perator (Tibshirani 1996):

$$\hat{b}^L := \text{argmin}_{b=(b_1,\dots,b_{p+1})\in\mathbb{R}^{p+1}}\left\{RSS(b)+\lambda\sum_{k=1}^{p}|b_{k+1}|\right\}.$$

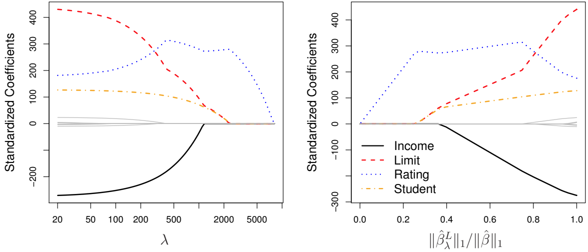
- The first part in above equation is the term we minimize when learning a linear model.
- λ is a tuning parameter which can be selected by cross-validation. In AIC this corresponded to $2\hat{\sigma}^2$, in BIC this corresponded to $(\log n)\hat{\sigma}^2$.
- The number of non-zero coefficients in AIC/BIC $\#\{k=2,..,p+1 : b_j \neq 0\}$ is replaced by the sum of their absolute values $\sum_{k=1}^{p}|b_{k+1}|$.
- Penalty is minimized for $b_2 = ... = b_{p+1} = 0$. $\lambda = 0$ is the usual linear model, very large λ lead to all coefficients set to 0 (see below for additional details).
- Note: no penalty on the coefficient b_1, usually we believe that there is a non-zero intercept.
- Not obvious, but one major advantage of this approach is computation.The problem is convex and can be solved with $p = 10000$ or even larger.

The lasso: example (Credit data set)

Least **A**bsolute **S**hrinkage and **S**election **O**perator (Tibshirani 1996):

$$\hat{b}^L := \text{argmin}_{b=(b_1,\ldots,b_{p+1})\in\mathbb{R}^{p+1}}\left\{RSS(b) + \lambda\sum_{k=1}^{p}|b_{k+1}|\right\}.$$

Example: plot of standardised regression coefficients against λ (left graph) and against relative norm of penalized and original (right graph).



Observation: for larger values of penalty, values of coefficients decrease. Closer look on next slide.

LASSO sets some coefficients to zero.

Example below: LASSO on a model with just `Age`, `Limit`, `Limit` as predictors and `Balance` as response for different values of λ. Some R output is given below.

```
> coef(lasso.mod)[,100] # lambda = 0.01
 (Intercept)          Age    Education        Limit
-198.7895479   -2.2928924    1.8728406    0.1734214
> coef(lasso.mod)[,80]  # lambda = 10
 (Intercept)          Age    Education        Limit
-187.4472377   -1.5992583    0.0000000    0.1681918
> coef(lasso.mod)[,70]  # lambda = 43
 (Intercept)          Age    Education        Limit
-203.8685458    0.0000000    0.0000000    0.1528599
> coef(lasso.mod)[,1]     # lambda = 10^10
(Intercept)           Age    Education        Limit
 520.015        0.000        0.000         0.000
```

As λ increases some coefficients are set to zero exactly. Thus LASSO can be used to perform model selection (just consider model with predictors that have non-zero coefficients)!

Why does the LASSO set coefficients to zero?

In general, there is no closed form solution for the LASSO optimization problem. To understand how it works, consider simple example (same as earlier in this lecture): one predictor with $\bar{x} = n^{-1} \sum_{i=1}^{n} x_i = 0$. Define

$$\overline{xy} := \frac{1}{n} \sum_{i=1}^{n} x_i y_i, \quad \overline{x^2} := \frac{1}{n} \sum_{i=1}^{n} x_i^2, \quad \overline{y} := \frac{1}{n} \sum_{i=1}^{n} y_i.$$

In that case

$$\hat{b}_1^L = \bar{y}$$

and

$$\hat{b}_2^L = \begin{cases} (\overline{xy} - .5\lambda/n)/\overline{x^2} & n\overline{xy} > \lambda/2 \\ (\overline{xy} + .5\lambda/n)/\overline{x^2} & n\overline{xy} < -\lambda/2 \\ 0 & n|\overline{xy}| \leq \lambda/2 \end{cases}$$

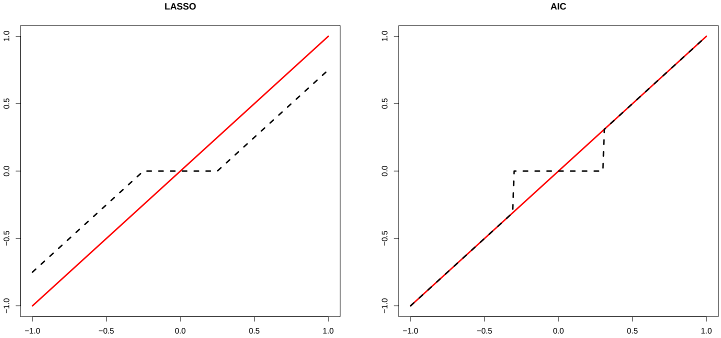
See detailed derivation in lectures.

Note: solution of least squares without penalty in this case is $\hat{b}_2 = \overline{xy}/\overline{x^2}$.

Comparison of LASSO, AIC, BIC in simple example

In setting of previous slide: $\hat{b}_1 = \hat{b}_1^L = \hat{b}_1^{AIC} = \hat{b}_1^{BIC} = \bar{y}$ and

$$\hat{b}_2^{AIC} = \begin{cases} \overline{xy}/\overline{x^2} & , |\overline{xy}|/\overline{x^2} > 2\hat{\sigma}^2 \\ 0 & , |\overline{xy}|/\overline{x^2} \leq 2\hat{\sigma}^2 \end{cases} \qquad \hat{b}_2^{BIC} = \begin{cases} \overline{xy}/\overline{x^2} & , |\overline{xy}|/\overline{x^2} > (\log n)\hat{\sigma}^2 \\ 0 & , |\overline{xy}|/\overline{x^2} \leq (\log n)\hat{\sigma}^2 \end{cases}$$



X-axis: $\overline{xy}/\overline{x^2}$. Y-Axis: estimators of b_2. Red line: least squares estimator. Dashed black: estimator based on LASSO (left) and AIC (right). See blackboard for more details.

Comments on LASSO

- In general, there is no simple closed form solution for LASSO.
- Under mild assumptions on the vectors X_i, the lasso problem is *strictly convex*. This implies the existence of a unique minimizer which is easy to compute using modern optimization procedures.
- Even with the R implementation, LASSO solutions can be computed with several *thousands* of predictors fairly efficiently.

LASSO vs best subset selection methods with AIC, BIC and cross-validation

- + LASSO has substantial advantages in terms of computational complexity. Can be computed even for very large numbers of predictors.
- = Similarly to BIC, LASSO can perform consistent model selection (under certain technical assumptions).
- +- Because of penalization, LASSO usually introduces bias. However, penalty can also help to reduce variance (bias-variance tradeoff).

- In practice, after running a LASSO to select variables, it is common to refit selected model after selecting variables.
- To make sure that all predictors get same treatment, it is advisable to standardise all predictors to sample mean zero and sample standard deviation 1.

Some useful functions in R

Library glmnet contains functions for LASSO and ridge regression.

Library plotmo contains useful plotting functions for output of lasso and ridge.

lasso.mod = glmnet(x,y,alpha = 1, lambda = grid) runs lasso regression.

- Inputs: x output of function model.matrix, y response values, alpha = 1 means do lasso, lambda = grid means use this grid of lambda values to compute lasso estimators.
- Output: object, can be used for plotting or extracting coefficients.
- coef(lasso.mod) provides array with estimated coefficients (each row corresponds to one lambda value from grid.)
- Plot result using function

  ```
  plot_glmnet(lasso.mod)
  ```

cv.la = cv.glmnet(x,y,alpha = 1, lambda = grid) can be used for cross-validation. Access to lambda selected by cross-validation and by one standard error rule through

```
cv.ri$lambda.min, cv.ri$lambda.1se
```

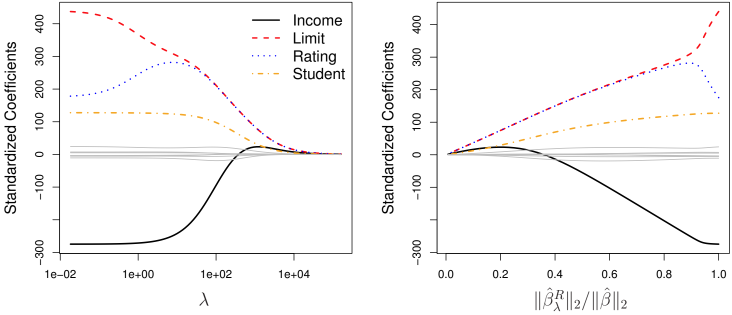
LASSO on 'Hitters' Data

See R code example

Ridge regression

$$\hat{b}^R := \text{argmin}_{b=(b_1,\ldots,b_{p+1})\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} (Y_i - X_i^\top b)^2 + \lambda \sum_{k=1}^{p} b_{k+1}^2.$$

- Instead of looking at the sum of absolute values of b_{k+1} we look at the sum of their squares.
- λ is a tuning parameter that can be selected by cross-validation.
- Ridge regression does not set coefficients to zero!
- Motivation for ridge regression: introduce some bias hoping to reduce variance.



Ridge regression does not set coefficients to zero

Same setting as before for LASSO.

```
> coef(lasso.mod)[,100] # lambda = 0.01
 (Intercept)          Age    Education        Limit
-198.8070412   -2.2934420    1.8760364    0.1734225
> coef(lasso.mod)[,80] # lambda = 10
 (Intercept)          Age    Education        Limit
-195.1052213   -2.2666645    1.8472106    0.1724079
> coef(lasso.mod)[,70] # lambda = 43
 (Intercept)          Age    Education        Limit
-142.6960719   -1.9079664    1.4656002    0.1582081
> coef(lasso.mod)[,1]  # lambda = 10^10
  (Intercept)           Age     Education         Limit
 5.200150e+02  2.245932e-09 -5.445753e-08  7.881306e-09
```

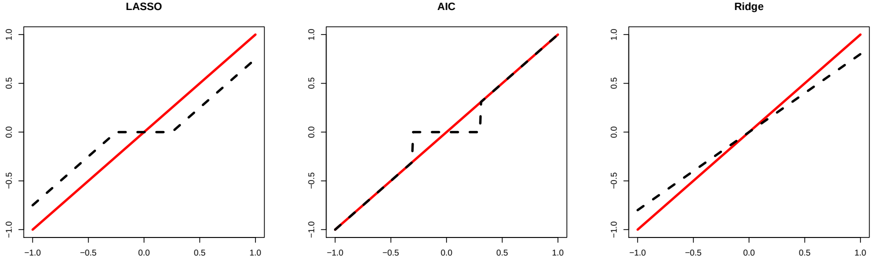
How does ridge regression work? A special case.

Simple example: one predictor with $\bar{x} = n^{-1} \sum_{i=1}^{n} x_i = 0$. Define

$$\overline{xy} := \frac{1}{n} \sum_{i=1}^{n} x_i y_i, \quad \overline{x^2} := \frac{1}{n} \sum_{i=1}^{n} x_i^2, \quad \overline{y} := \frac{1}{n} \sum_{i=1}^{n} y_i.$$

Then $\hat{b}_1^R = \overline{y}$ and

$$\hat{b}_2^R = \frac{n\overline{xy}}{n\overline{x^2} + \lambda} = \hat{b}_2 \frac{n\overline{x^2}}{n\overline{x^2} + \lambda}$$



- Ridge does not give values of exactly zero, so does not perform model selection.
- Larger values are pushed to zero. The larger the value the stronger the effect.

Ridge regression: general case.

The general case: assume predictors are p-dimensional and for $j = 1, ..., p$ we have $\sum_{i=1}^{n} x_{i,j} = 0$. Let $y_i = b_1 + \tilde{b}^\top X_i + \varepsilon_i$ and define

$$\tilde{\mathbf{X}} = \begin{pmatrix} x_{1,1}, ..., x_{1,p} \\ \vdots \\ x_{n,1}, ..., x_{n,p} \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

One can prove (see blackboard for detailed derivation): $\hat{b}_1^R = \overline{y}$ and

$$\check{b}^R := (\hat{b}_2^R, ..., \hat{b}_{p+1}^R)^\top = (\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} + \lambda \mathbf{I}_p)^{-1} \tilde{\mathbf{X}}^\top Y$$

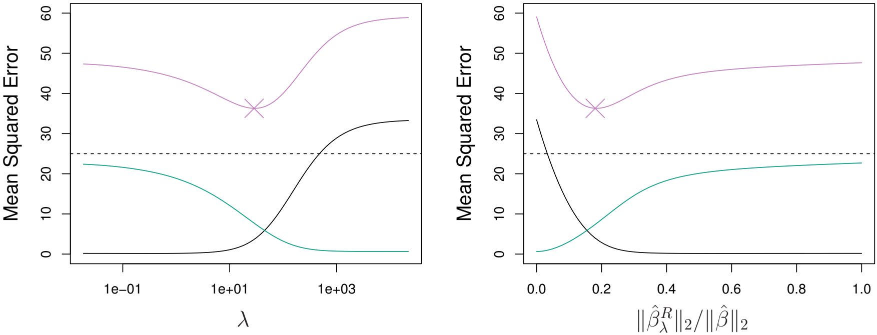
where $\mathbf{I}_p$ is $p \times p$ identity matrix.

- Even if the matrix $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$ is not invertible, the matrix $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} + \lambda \mathbf{I}_p$ is invertible.
- One can prove: if $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$ has full rank then $Var(w^\top \check{b}) < Var(w^\top \check{b}^R)$ for any $w \neq 0$ if $\lambda > 0$. So ridge regression reduces variance of predictions.
- For $\lambda \neq 0$ ridge introduces some bias:

$$\mathbb{E}[\check{b}^R] - \tilde{b} = [(\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} + \lambda \mathbf{I}_p)^{-1} \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} - \mathbf{I}_p] \tilde{b} \neq 0.$$

Ridge regression: why is it helpful?

Example of squared bias (black), variance (green) and test MSE (purple) of ridge regression in one simulated example. Dashed horizontal line is irreducible part of error.



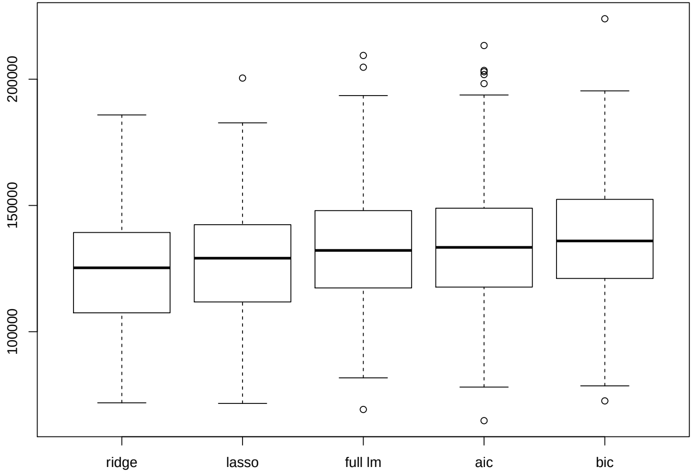
- $\lambda = 0$ or equivalently $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2 = 1$ corresponds to least squares solution.
- Best MSE for ridge is better compared to MSE of least squares, this corresponds to optimal bias-variance trade-off.

Ridge, LASSO, best subset selection: comments on comparison

- Ridge regression is helpful for prediction if there are many predictors that are important but weak, and if there is high correlation between some of the predictors.
- Ridge tends to give predictors which are highly correlated similar regression coefficients (example: `Limit` and `Rating` in `Credit` data).
- LASSO works best if only few predictors are important and many predictors have no influence on outcome ('sparse' case).
- LASSO, AIC, BIC can be used to perform model selection. Ridge regression does not perform model selection.
- If there are many predictors, LASSO and Ridge are computationally less expensive compared to best subset selection with AIC, BIC or cross-validation.

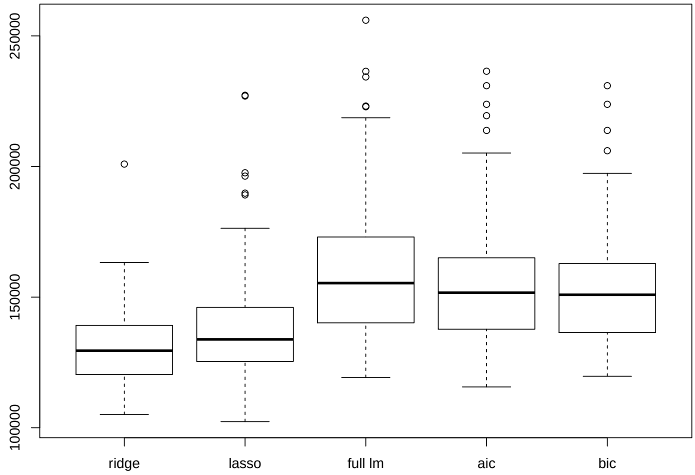
Ridge, LASSO, best subset selection: comparison for `Hitters` data

Test error for various methods based on 50-50 split of data set, corresponds to 131 training data. Ridge a bit better compared to lasso, lasso a bit better compared to full model. BIC a bit worse compared to full model.



Ridge, LASSO, best subset selection: comparison for `Hitters` data

Test error for various methods with roughly 70 training data. Ridge and lasso show more substantial advantage over linear model, AIC and BIC a bit better compared to linear model. Lesson: results depend on sample size. In smaller samples regularization is more helpful.



More comparison based on simulations

Some R examples.

STA314, Lecture 5

Some of the figures in this presentation are taken from ”An Introduction to Statistical Learning, with applications in R” (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani.