STA314, Lecture 6

October 11, 2019
Stanislav Volgushev

Recap from previous lecture

LASSO regression:

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

No explicit solution in general (computed through numerical optimization), but solution in a special case: one predictor, $\bar{x} = n^{-1}\sum_{i=1}^n x_i = 0$. Then

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

and

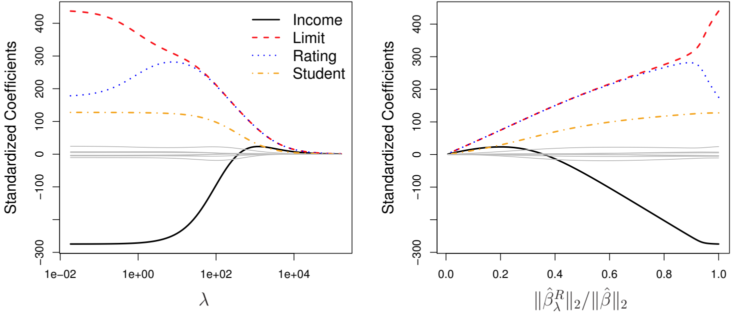
$$\hat{b}_2^L = \begin{cases} (\overline{xy} - .5\lambda/n)/\overline{x^2} & n\overline{xy} > \lambda/2 \\ (\overline{xy} + .5\lambda/n)/\overline{x^2} & n\overline{xy} < -\lambda/2 \\ 0 & n|\overline{xy}| \leq \lambda/2 \end{cases}$$

- Lasso can set coefficients to zero when λ is large and can be used for model selection.
- Lasso is computationally more efficient than best subset or forward/backward stepwise selection when number of predictors is large.
- Tuning parameter λ can be selected by cross-validation.

Ridge regression

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

- 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

Ridge in R: same as lasso, i.e. use function glmnet but set $\alpha = 1$. Example below: same setting as before for LASSO (see lecture 5).

```
grid = ...
ridge.mod = glmnet(x,y,alpha = 0, lambda = grid)

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

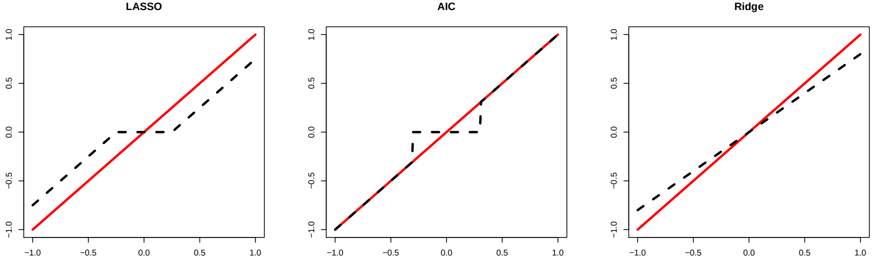
How does ridge regression work? A special case.

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.

Recap: eigenvalues and eigenvectors.

Throughout let M denote a symmetric $k \times k$ matrix with real entries.

A vector $v \in \mathbb{R}^k$ is *eigenvector* of M with *eigenvalue* α iff $Mv = \alpha v$.

Theorem For every symmetric matrix M we can find an orthonormal basis of $\mathbb{R}^k$ which is spanned by eigenvectors of M and all eigenvalues are real.

Theorem M is invertible if and only if all eigenvalues of M are non-zero.

If M invertible and v eigenvector of M with eigenvalue α then v is eigenvector of M^{-1} with eigenvalue $1/\alpha$. Proof: see lectures.

If $M = A^\top A$ for a matrix A the all eigenvalues of M are non-negative. Proof: see lectures.

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 lectures for detailed derivation): $\hat{b}_1^R = \overline{y}$ and

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

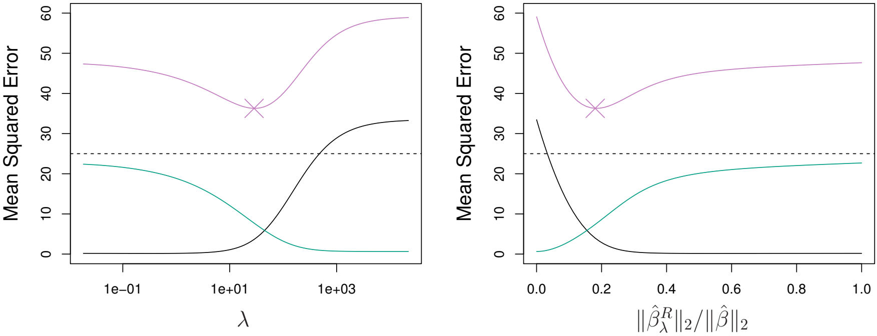
where $\mathbf{I}_p$ is $p \times p$ identity matrix.

- 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 $\lambda \neq 0, \tilde{b} \neq 0$ ridge introduces 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 $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2 = 1$ 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).

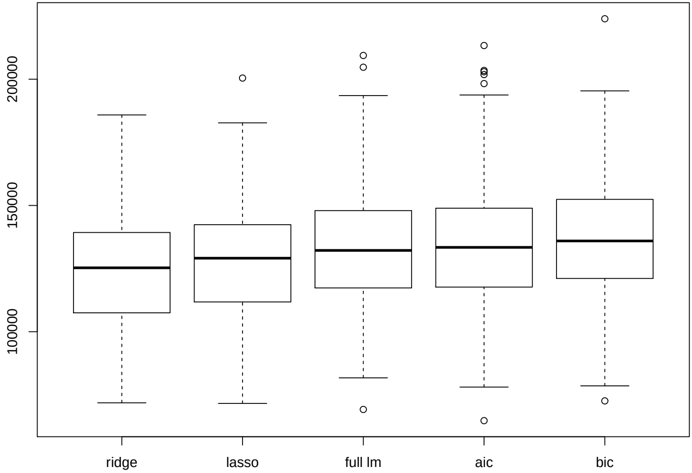
- LASSO works best if only few predictors are important and many predictors have no influence on outcome ('sparse' case).

- 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.

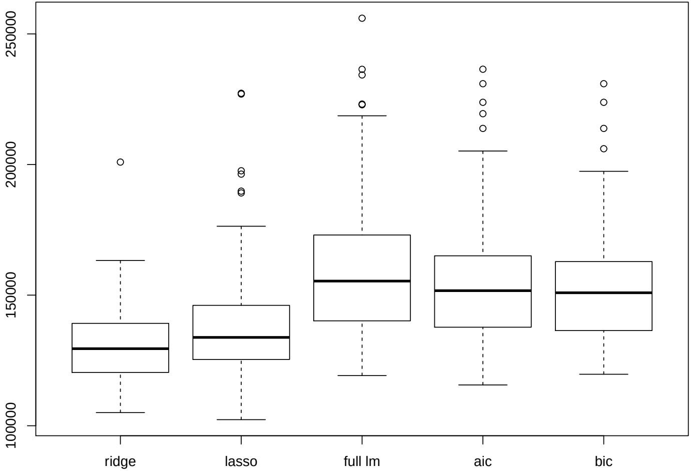
Ridge, LASSO, best subset selection: comparison for `Hitters` data

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.



Some R examples.

Ridge and other examples: `STA314-Lecture06-RExamples2`

Simulation-based comparisons: `STA314-Lecture-06-Simulations`

Dimension reduction methods

General idea of dimension reduction methods

Prediction perspective: all methods we discussed so far tried to balance bias vs variance.

- Model selection (AIC, BIC etc): reduce variance by discarding predictors that are not important.
- Ridge and Lasso: reduce variance by adding a penalty term. In Lasso: also perform model selection this way.

Next: alternative methods based on *dimension reduction*

- Model selection: only keep 'important' predictors.
- Dimension reduction: transform predictors to only keep their 'important properties'.

Dimension reduction methods: basic set-up

Data: (X_i, y_i) with predictors $X_i = (x_{i,1}, ..., x_{i,p})^\top \in \mathbb{R}^p$.

Step 1: define standardized predictors $\tilde{X}_i = (\tilde{x}_{i,1}, ..., \tilde{x}_{i,p}^\top)$ with $\tilde{x}_{i,k} = x_{i,k} - \bar{x}_{\cdot,k}$.

Step 2: form new predictors $Z_i = (z_{i,1}, ..., z_{i,M})^\top \in \mathbb{R}^M$ by taking linear combinations of original predictors, i.e.

$$z_{i,m} = \sum_{k=1}^{p} \phi_{k,m} \tilde{x}_{i,k}, \quad m = 1, ..., M$$

where $\phi_{k,m}$ are some weights determined later.

Step 3: run a linear regression using Z_i as predictors.

- If $M < p$, the model described above is less flexible compared to full linear model, thus variance can be reduced. Might introduce bias.
- There are several ways to select the weights $\phi_{k,m}$. Two popular approaches mentioned today: **P**rincipal **C**omponent **A**nalysis and **P**artial **L**east **S**quares.

Recap: projections on different directions in $\mathbb{R}^p$

Note: if $\sum_{k=1}^{p} \phi_{k,m}^2 = 1$, then $\Phi_m := (\phi_{1,m}, ..., \phi_{p,m})^\top$ describes direction in $\mathbb{R}^p$.

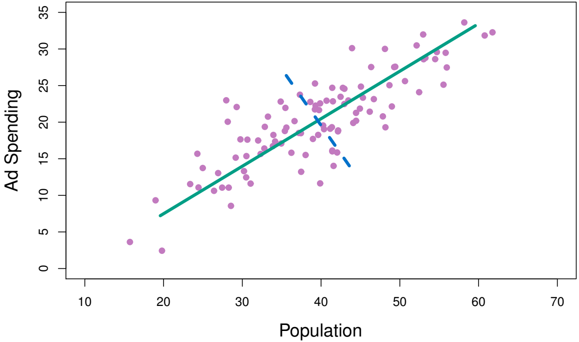
$\sum_{k=1}^{p} \phi_{k,m} \tilde{x}_{i,k} = \Phi_m^\top \tilde{X}_i$ corresponds to length of X_i when projected on that direction.

Some examples in $\mathbb{R}^2$: see blackboard.

Dimension reduction methods: Principal Component regression

PCR: form linear combinations that explain 'the most variation' in the predictors.

First 'principal component': $z_{i,m} = \sum_{k=1}^{p} \phi_{k,m} x_{i,k}$ describes the direction in $\mathbb{R}^p$ along that predictors show 'maximal variation'. Figure below: which line corresponds to first principal component?



Principal Component Regression

Mathematical definition of first principal component:

$$\begin{aligned}\Phi_1^{PCR} &= \arg \max_{V \in \mathbb{R}: \sum v_k^2 = 1} \frac{1}{n-1} \sum_{i=1}^{n} \Big(\sum_{k=1}^{p} v_k \tilde{X}_{i,k} \Big)^2 \\ &= \arg \max_{V \in \mathbb{R}: \sum v_k^2 = 1} \frac{1}{n-1} \sum_{i=1}^{n} (V^\top \tilde{X}_i)^2 .\end{aligned}$$

- For any given vector V the term $V^\top \tilde{X}_i$ is a candidate predictor.
- $\sum v_k^2 = 1$ ensures that V is a direction.
- Since $\sum_i V^\top \tilde{X}_i = 0$, the sum above is simply the sample variance of $V^\top \tilde{X}_1, \ldots, V^\top \tilde{X}_n$

The first principal component is the linear combination of predictors that has the largest sample variance.

Principal Component regression

After we explained as much variation as we can by picking one direction, there is still some variation in the other directions. That variation is explained the next principal components.

The second principal component corresponds to the direction with largest variance which is orthogonal to the first component.

$$\Phi_2^{PCR} = \arg \max_{V \in \mathbb{R}: \sum v_k^2 = 1, V^\top \Phi_1^{PCR} = 0} \frac{1}{n-1} \sum_{i=1}^{n} (V^\top \tilde{X}_i)^2.$$

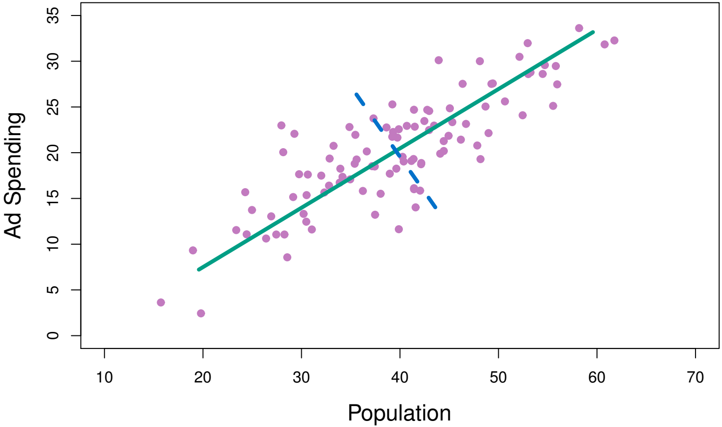
The third principal component corresponds to the direction with largest variance which is orthogonal to the first two principal components. Exercise: write down formula.

Subsequent principal components defined similarly.

How many principal components can we have at most if $X_i \in \mathbb{R}^p$? see lectures.

Exercise: describe principal directions through eigenvectors of $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$.

Where is the second principal component?



Comments on principal component regression

- Despite the fact PCR with $M < p$ leads to less flexible models, it does not lead to the selection of relevant predictors since each principal component can depend on all predictors.
- The number of components is usually determined by cross-validation.
- How well PCR works depends on how well the principal components can help to predict the response.
- When applying PCR, the predictors are usually standardised to have not only sample mean zero but also sample variance 1.

Note: the idea of using directions of maximal variation is sometimes also used to simply describe structure in a data set without a response. This is called *Principal Component Analysis* and is an example of *unsupervised learning*. More on this next semester...

Partial least squares

- Trouble with PCR: does not take into account any information about the response. It can happen that directions of high variation in X_i are not useful for predicting y_i.
- Partial Least Squares: determine directions in a way that maximises their power for predicting response.
- Details omitted here, see pp 80-82 and Algorithm 3.3 in 'Elements of Statistical Learning' for detailed description.
- We will see some `R` examples.

R examples: `STA314-Lecture06-DimensionReduction`

The high-dimensional setting

The high-dimensional setting

In Statistics, we talk about the 'high-dimensional' setting when the number of predictors p is of comparable order or even larger than the sample size n (sometimes the latter is called ultra high dimensional).

Examples:

1. Genetics: predict risk of certain type of cancer from DNA mutations. Usually number of objects n is in the hundreds, but roughly $p = 500000$ common DNA mutations.
2. Marketing: for each customer in an online shop, have 0-1 encoding of items customer decided to buy or not to buy. Predict future shopping behaviour based on past purchase.

Clearly k-nn won't work in this setting. Which of the methods for linear regression discussed so far could be applicable?

'Usual' linear regression when $p > n$

Usual linear regression is not applicable when $p > n$ because there will be no unique solution to OLS problem. Recall: there is a unique minimizer of

$$RSS(b) = \sum_{i=1}^{n} (y_i - X_i^\top b)^2$$

if and only if matrix $\mathbf{X}^\top \mathbf{X}$ is invertible. Matrix $\mathbf{X}^\top \mathbf{X}$ is ...-dimensional and has rank at most .... If $p > n$ it can not be invertible.

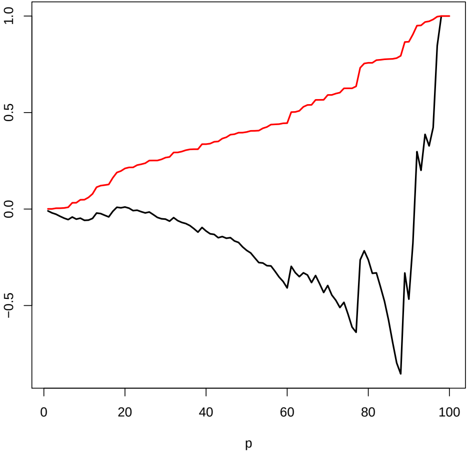
Even if p slightly smaller than n, linear regression will give a very small training error even if none of the predictors are relevant for response. Resulting test error will be very large.

Since usual least squares is not applicable, we need to look for *less flexible* models.

Examples: see discussion in lectures.

Adjusted R^2 fails for high dimensions

Setting: generate p predictors not related to the response, run linear regression. Sample size: $n = 100$. Plot below: R^2 and adjusted R^2 as a function of p. Which one is which? *R Exercise: generate this type of plot.*



Classical model selection methods

Backward stepwise selection: need to start with model that has all predictors, does not make sense when $p > n$.

Forward stepwise selection: can be used to select models as long as the largest model has less predictors than observations. So forward stepwise selection can be applied, but we need to stop before too many predictors selected.

Best subset: not feasible from a computational point of view. If there are p predictors, there are p choose k models with k predictors, that grows in p like p^k (can be very large!).

How do we compare models with different numbers of predictors?

- adjusted R^2 does not work for p close to n.
- Trouble with C_p, AIC, BIC: need $\hat{\sigma}^2$, difficult to obtain if $p > n$. The usual way to get this from 'full model' does not work, so not applicable when $p > n$.
- Cross-validation can be applied, but can also be expensive computationally.

Lasso, Ridge, PCR, PLS

Lasso: can often be applied even if $p > n$, provided λ is selected by cross-validation. In fact, one of the reasons for the great success of Lasso is the increased importance of data sets with $p > n$.

Ridge: will always lead to a unique solution if $\lambda > 0$, so can be applied when $p > n$. Disadvantage compared to Lasso: does set any coefficients to zero, so can not be used to perform model selection.

PCR, PLS: can be applied if $p > n$ but $M < n$ components are selected.

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.