Lecture 29: Review

Reading: All chapters in ISLR

STATS 202: Data mining and analysis

Sergio Bacallado
December 4, 2018

Stepwise selection methods

Use AIC / BIC to score a model $\mathcal{M}$ – how to optimize over $\mathcal{M}$?

Best subset selection has 2 problems:

Stepwise selection methods

Use AIC / BIC to score a model $\mathcal{M}$ – how to optimize over $\mathcal{M}$?

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!

Stepwise selection methods

Use AIC / BIC to score a model $\mathcal{M}$ – how to optimize over $\mathcal{M}$?

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!
2. If for a fixed k, there are too many possibilities, we increase our chances of overfitting.

Stepwise selection methods

Use AIC / BIC to score a model $\mathcal{M}$ – how to optimize over $\mathcal{M}$?

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!
2. If for a fixed k, there are too many possibilities, we increase our chances of overfitting. The model selected has *high variance*.

Stepwise selection methods

Use AIC / BIC to score a model $\mathcal{M}$ – how to optimize over $\mathcal{M}$?

Best subset selection has 2 problems:

1. It is often very expensive computationally. We have to fit 2^p models!
2. If for a fixed k, there are too many possibilities, we increase our chances of overfitting. The model selected has *high variance*.

In order to mitigate these problems, we can restrict our search space for the best model.

This reduces the variance of the selected model at the expense of an increase in bias.

Ridge regression

Ridge regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

In blue, we have the RSS of the model.

In red, we have the squared ℓ_2 norm of β, or $\|\beta\|_2^2$.

The parameter λ is a tuning parameter. **Use cross-validation**.

The Lasso

Lasso regression solves the following optimization:

$$\min_{\beta} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

In blue, we have the RSS of the model.

In red, we have the ℓ_1 norm of β, or $\|\beta\|_1$.

The parameter λ is a tuning parameter. **Use cross-validation**.

Ridge / LASSO / Best subset

- **Ridge:** for every λ, there is an s such that $\hat{\beta}_\lambda^R$ solves:

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 < s.$$

Ridge / LASSO / Best subset

- **Ridge:** for every λ, there is an s such that $\hat{\beta}_\lambda^R$ solves:

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \beta_j^2 < s.$$

- **Lasso:** for every λ, there is an s such that $\hat{\beta}_\lambda^L$ solves:

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| < s.$$

Ridge / LASSO / Best subset

- **Ridge:** for every λ, there is an s such that $\hat{\beta}^R_\lambda$ solves:

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \beta_j^2 < s.$$

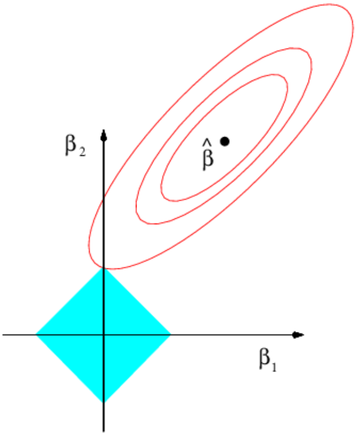
- **Lasso:** for every λ, there is an s such that $\hat{\beta}^L_\lambda$ solves:

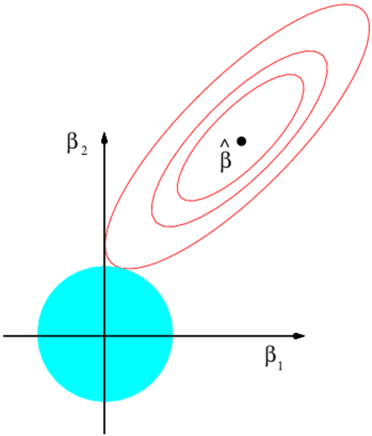
$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| < s.$$

- **Best subset:**

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{i,j} \right)^2 \right\} \text{ s.t. } \sum_{j=1}^{p} \mathbf{1}(\beta_j \neq 0) < s.$$

Ridge vs. LASSO





The Lasso

◆ : $\sum_{j=1}^{p} |\beta_j| < s$

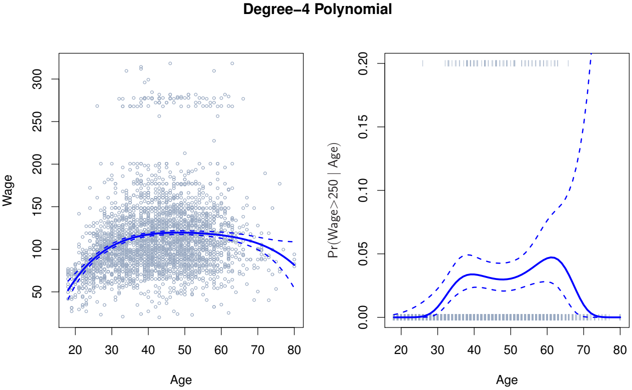
Ridge Regression

● : $\sum_{j=1}^{p} \beta_j^2 < s$

Best subset with $s = 1$ is union of the axes...

Non-linear regression

Problem: How do we model a non-linear relationship?



Left: Regression of `wage` onto `age`.
Right: Logistic regression for classes `wage`> 250 and `wage`≤ 250

Basis functions

Strategy:

- Define a model:

$$Y = \beta_0 + \beta_1 f_1(X) + \beta_2 f_2(X) + \cdots + \beta_d f_d(X) + \epsilon.$$

Basis functions

Strategy:

- Define a model:

$$Y = \beta_0 + \beta_1 f_1(X) + \beta_2 f_2(X) + \cdots + \beta_d f_d(X) + \epsilon.$$

- Fit this model through least-squares regression: f_j's are nonlinear, model is linear!

Basis functions

Strategy:

- Define a model:

$$Y = \beta_0 + \beta_1 f_1(X) + \beta_2 f_2(X) + \cdots + \beta_d f_d(X) + \epsilon.$$

- Fit this model through least-squares regression: f_j's are nonlinear, model is linear!
- Options for $f_1, \ldots, f_d$:

Basis functions

Strategy:

- Define a model:

$$Y = \beta_0 + \beta_1 f_1(X) + \beta_2 f_2(X) + \cdots + \beta_d f_d(X) + \epsilon.$$

- Fit this model through least-squares regression: f_j's are nonlinear, model is linear!
- Options for $f_1, \ldots, f_d$:
  1. Polynomials, $f_i(x) = x^i$.

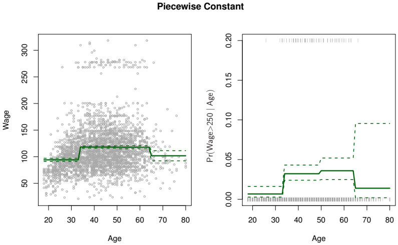
Basis functions

Strategy:

- Define a model:

$$Y = \beta_0 + \beta_1 f_1(X) + \beta_2 f_2(X) + \cdots + \beta_d f_d(X) + \epsilon.$$

- Fit this model through least-squares regression: f_j's are nonlinear, model is linear!
- Options for $f_1, \ldots, f_d$:
  1. Polynomials, $f_i(x) = x^i$.
  2. Indicator functions, $f_i(x) = \mathbf{1}(c_i \leq x < c_{i+1})$.



Smoothing splines

Find the function f which minimizes

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

- The RSS of the model.
- A penalty for the roughness of the function.

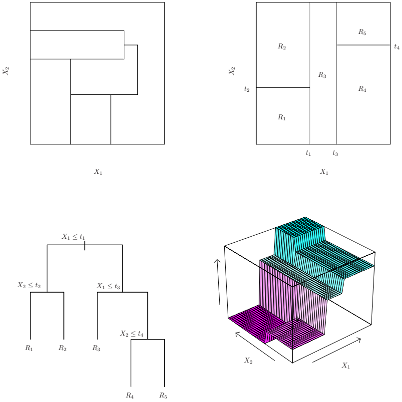
GAM – generalized additive model

Fit a model with regression function of the form

$$f(X_1, \ldots, X_p) = \sum_{j=1}^{p} f_j(X_j)$$

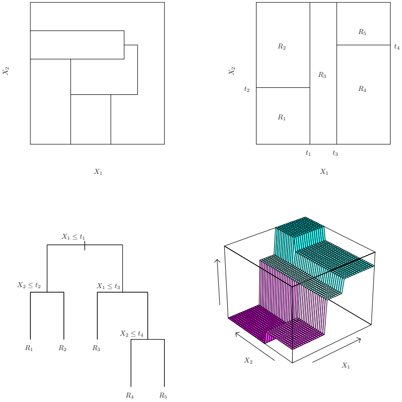
- Each term can be a local regression or smoothing spline

Decision trees, 10,000 foot view



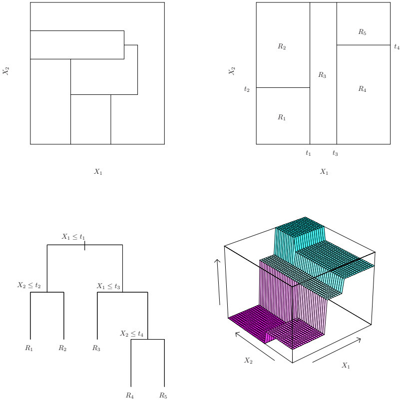
1. Find a partition of the space of predictors.
2. Predict a constant in each set of the partition.

Decision trees, 10,000 foot view



1. Find a partition of the space of predictors.
2. Predict a constant in each set of the partition.
3. The partition is defined by splitting the range of one predictor at a time.

Decision trees, 10,000 foot view



1. Find a partition of the space of predictors.
2. Predict a constant in each set of the partition.
3. The partition is defined by splitting the range of one predictor at a time.

   $\rightarrow$ Not all partitions are possible.

Bagging

- In **Bagging** we average the predictions of a model fit to many Bootstrap samples.

  Example. Bagging the Lasso

Bagging

- In **Bagging** we average the predictions of a model fit to many Bootstrap samples.

  Example. Bagging the Lasso

  - Let $\hat{y}^{L,b}$ be the prediction of the Lasso applied to the bth bootstrap sample.

Bagging

- In **Bagging** we average the predictions of a model fit to many Bootstrap samples.

  Example. Bagging the Lasso

  - Let $\hat{y}^{L,b}$ be the prediction of the Lasso applied to the bth bootstrap sample.
  - Bagging prediction:

$$\hat{y}^{\text{boot}} = \frac{1}{B} \sum_{b=1}^{B} \hat{y}^{L,b}.$$

Boosting

Boosting

1. Set $\hat{f}(x) = 0$, and $r_i = y_i$ for $i = 1, \ldots, n$.

Boosting

1. Set $\hat{f}(x) = 0$, and $r_i = y_i$ for $i = 1, \ldots, n$.
2. For $b = 1, \ldots, B$, iterate:

Boosting

1. Set $\hat{f}(x) = 0$, and $r_i = y_i$ for $i = 1, \ldots, n$.
2. For $b = 1, \ldots, B$, iterate:
   2.1 Fit a decision tree $\hat{f}^b$ with d splits to the response $r_1, \ldots, r_n$.

Boosting

1. Set $\hat{f}(x) = 0$, and $r_i = y_i$ for $i = 1, \ldots, n$.
2. For $b = 1, \ldots, B$, iterate:
   2.1 Fit a decision tree $\hat{f}^b$ with d splits to the response $r_1, \ldots, r_n$.
   2.2 Update the prediction to:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$

Boosting

1. Set $\hat{f}(x) = 0$, and $r_i = y_i$ for $i = 1, \ldots, n$.
2. For $b = 1, \ldots, B$, iterate:
   2.1 Fit a decision tree $\hat{f}^b$ with d splits to the response $r_1, \ldots, r_n$.
   2.2 Update the prediction to:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$
   2.3 Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$

Boosting

1. Set $\hat{f}(x) = 0$, and $r_i = y_i$ for $i = 1, \ldots, n$.
2. For $b = 1, \ldots, B$, iterate:
   2.1 Fit a decision tree $\hat{f}^b$ with d splits to the response $r_1, \ldots, r_n$.
   2.2 Update the prediction to:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$
   2.3 Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$
3. Output the final model:
$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$

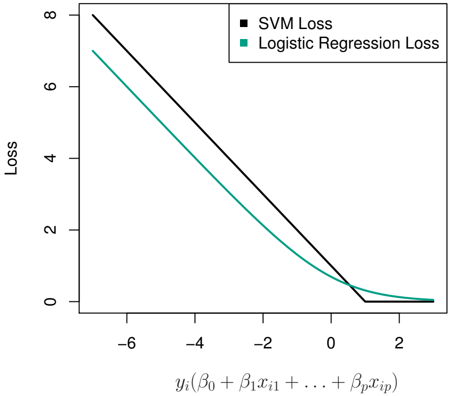
Boosting vs. bagging (in a nutshell)

1. Bagging averages over a random collection of trees.
2. Boosting is "gradient descent" in a space of "trees".

Support vector machine

1. Relaxation of maximum margin classifier.
2. Dual problem involves only the kernel matrix $K(x_i, x_j)$.
3. Can be replaced with a “kernel”: radial basis function, polynomial, etc.

Support vector machine



1. Similar to logistic regression but uses *hinge loss*.

Regression methods

- Nearest neighbors regression
- Multiple linear regression
- Stepwise selection methods
- Ridge regression and the Lasso
- Principal Components Regression
- Partial Least Squares
- Non-linear methods:
  - Polynomial regression
  - Cubic splines
  - Smoothing splines
  - Local regression
  - GAMs: Combining the above methods with multiple predictors
- Decision trees, Bagging, Random Forests, and Boosting

Classification methods

- Nearest neighbors classification
- Logistic regression
- LDA and QDA
- Stepwise selection methods (for logistic)
- Decision trees, Bagging, Random Forests, and Boosting
- Support vector classifier and support vector machines

Self testing questions

For each of the regression and classification methods:

1. What are we trying to optimize?
2. What does the fitting algorithm consist of, roughly?
3. What are the tuning parameters, if any?
4. How is the method related to other methods, mathematically and in terms of bias, variance?
5. How does rescaling or transforming the variables affect the method?
6. In what situations does this method work well? What are its limitations?