DATA 300

Topic 6: Variable selection

Agenda

- Why variable selection?
- Naïve approach
  - an example on linear regression
  - different ways to determine the optimal model
- Complex approach
  - Ridge
  - Lasso

Bias-variance trade-off

MSE can be decomposed to

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$

Bias tends to be low as we have more predictors.

Variance tends to be low as we have less predictors.

- In other words, when bias is already *small enough,* having less predictors might be able to reduce variance A LOT without hurting bias too much.

Naïve approach

The naïve approach is to build multiple models, using different numbers of predictors, and see which one works the best.

- "Best" in terms of what?

Naïve approach

The naïve approach is to build multiple models, using different numbers of predictors, and see which one works the best.

- The key here is to determine what is ”best”. It could be smaller Mean Absolute Error for linear regression, could be better accuracy for classification models, could be having a specific number of significant predictors for generalized linear models.

Variable selection in Regression models

This naïve approach is long used in regression models. It has multiple formats:

- Best subset selection
- Stepwise selection
  - Backward selection
  - Forward selection
  - Hybrid

Best subset selection

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For $k = 1, 2, \ldots p$:

   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.

3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Objective function for linear regression

Therefore, the objective for a linear regression model becomes **minimize residuals.**

Specifically, it is to **minimize the sum of squared residuals (RSS)**, this is called the *Ordinary Least Squared (OLS) Method*.

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_p x_{ip})^2. \end{aligned}$$

Variable selection: forward selection

Step 0: start with a ***null*** model, a model that contains no predictors.

Step 1: For each of the D predictors, run a *simple* linear regression

$$y = b_0 + b_j\, x_j,$$

Find the regression with the lowest RSS.

Variable selection: forward selection

Step 0: start with a ***null*** model, a model that contains no predictors.

Step 1: For each of the D predictors, run a *simple* linear regression

$$y = b_0 + b_j\, x_j,$$

Find the regression with the lowest RSS.

Step 2: For each of the D-1 predictors, run linear regression:

$$y = b_0 + b_{j^*}\, x_{j^*} + b_j\, x_j,$$

where x_{j^*} is the variable picked from Step 2. Find the regression with the lowest RSS.

Variable selection: forward selection

Step 0: start with a ***null*** model, a model that contains no predictors.

Step 1: For each of the D predictors, run a *simple* linear regression

$$y = b_0 + b_j\, x_j,$$

Find the regression with the lowest RSS.

Step 2: For each of the D-1 predictors, run linear regression:

$$y = b_0 + b_{j^*}\, x_{j^*} + b_j\, x_j,$$

where x_{j^*} is the variable picked from Step 2. Find the regression with the lowest RSS.

Continue until the stopping rule is satisfied. A stopping rule could be you start to have variables with large p-values.

Variable selection: backward selection

Step 0: start with a ***full*** model, a model that contains all the predictors.

Step 1: Remove the variable with the largest p-value.

Variable selection: backward selection

Step 0: start with a ***full*** model, a model that contains all the predictors.

Step 1: Remove the variable with the largest p-value.

Step 2: Run a linear regression with the rest of the D-1 variables. Remove the variable with the largest p-value.

Continue until a stopping rule is reached. For example, we can stop if the model only contains significant variables.

Quick question:

Think about if forward would be better or backward for each of the following scenarios:

- We have 1,000,000 variables available, but only try to keep 10.
- We have 1,000,000 variables available, but only try to keep 900,000.
- We have 10 variables.

In R

All the naïve selection methods (best subset, forward, backward) requires a user-defined stopping rule. The easiest one to implement in R is to ***minimize the Akaike information criterion (AIC).***

- In short, AIC is an estimator of the prediction error.

- Use the lm() function to build a null model and a full model.
- Use the stepAIC function to do forward or backward selection.

In R

```
data_lm <- lm(Critic_rating~., data)
summary(data_lm)

###variable selection
data_null <- lm(Critic_rating~1., data)
variable_sel <- stepAIC(data_null, direction = "forward",
                        scope = list(upper = data_lm, lower = data_null))
summary(variable_sel)

variable_sel <- stepAIC(data_lm, direction = "backward",
                        scope = list(upper = data_lm, lower = data_null))
summary(variable_sel)
```

Full model

null model

Starting from null model for forward selection

Starting from full model for backward selection

Agenda

- Why variable selection?
- Naïve approach
  - an example on linear regression
  - **different ways to determine the optimal model**
- Complex approach
  - Ridge
  - Lasso

In R

All the naïve selection methods (best subset, forward, backward) requires a user-defined stopping rule. The easiest one to implement in R is to ***minimize the Akaike information criterion (AIC).***

- What is AIC?
- What are the other approaches?

Choosing the optimal combination of predictors

Which one (or more) would be a bad measure(s) to decide an optimal model with an optimal combination of variables?

- Training error
- Validation error
- R-squared

Choosing the optimal combination of predictors

Which one (or more) would be a bad measure(s) to decide an optimal model with an optimal combination of variables?

- Training error, because training error tends to decrease as the # of variables increase. By nature, training error is against the goal of reducing the # of variables.
- Validation error
- R-squared, same idea

Choosing the optimal combination of predictors: using validation error in the naïve approach

- Validation error from validation set

For example, in backward selection:

Step 0: start with a ***full*** model, a model that contains all the predictors.

Step 1: Remove the variable with the largest p-value.

Step 2: Run a linear regression with the rest of the D-1 variables. Remove the variable with the largest p-value.

Continue until a stopping rule is reached. For example, we can stop if the model only contains significant variables.

For each linear regression model built, document the validation error on the validation set. Stop the process when the validation error stopped decreasing.

Choosing the optimal combination of predictors: using validation error in the naïve approach

- Validation error from the validation set approach

- Validation error from cross-validation

Similar idea, if we decide to do cross-validation, then for each linear regression model built, document the cross-validation error. Stop the process when the cross-validation error stopped decreasing.

Choosing the optimal combination of predictors: adjustments based on training errors

We can’t use training error/R-squared to decide the optimal combination of predictors, because chances are training error/R-squared reaches optimal when we have ALL predictors.

Choosing the optimal combination of predictors: adjustments based on training errors

We can’t use training error/R-squared to decide the optimal combination of predictors, because chances are training error/R-squared reaches optimal when we have ALL predictors.

In response, we could, e.g., adjust the formula for R-squared, so that it factors in the number of predictors.

Choosing the optimal combination of predictors: common training-error based measures

Measures defined on the training set	Corresponding adjusted measure
1. Mean squared error (MSE)	$C_p = \text{MSE} + \frac{2d\hat{\sigma}^2}{\text{n}}$
2. Some sort of training error (e.g., MSE for regression, likelihood for classification)	AIC
3. Some sort of training error (e.g., MSE for regression, likelihood for classification)	BIC
4. R-squared	Adjusted R-squared

Choosing the optimal combination of predictors: common training-error based measures

1. Instead of looking for a model with the smallest MSE, we look for a model with the smallest C_p:

$$C_p = \mathrm{MSE}_p + \frac{2d\hat{\sigma}^2}{\mathrm{n}},$$

where

- MSE_p is the mean squared error with the p predictors,
- d is the number of predictors,
- $\hat{\sigma}^2$ is the estimated variance of the error, which is usually approximated by MSE_D (the MSE of the complete model).

Choosing the optimal combination of predictors: common training-error based measures

2. Instead of looking for a model with the smallest MSE, we could also define AIC (Akaike Information Criterion):

$$AIC = 2d + some\ training\ error$$

Specifically, if the objective function is to minimize MSE (regression):

$$AIC = \frac{MSE}{\hat{\sigma}^2} + 2\frac{d}{n},$$

if the objective function is to maximize likelihood (logistic regression):

Review: objective function of logistic regression

Therefore, in logistic regression, the goal is to find a parameter set that maximizes:

$$L(b_0, b_1, \dots, b_D) = \prod_{i:y_i=1} Prob(y_i = 1) \prod_{i:y_i=0} (1 - Prob(y_i = 1))$$

For the sample units that indeed have a response variable equal to 1, we want $Prob(y_i = 1)$ to be large

For the sample units that indeed have a response variable equal to 0, we want $Prob(y_i = 1)$ to be small, and thus $1 - Prob(y_i = 1)$ to be large

This is called **Maximize Likelihood Estimation** (MLE), the objective function $L(b_0, b_1, \dots, b_D)$ is called the likelihood function.

MLE is one of the most common approaches to fit non-linear models. However, it often does not have an analytical solution.

Choosing the optimal combination of predictors: common training-error based measures

2. Instead of looking for a model with the smallest MSE, we could also define AIC (Akaike Information Criterion):

$$AIC = 2d + some\ training\ error$$

Specifically, if the objective function is to minimize MSE (regression):

$$AIC = \frac{MSE}{\hat{\sigma}^2} + 2\frac{d}{n},$$

if the objective function is to maximize likelihood (logistic regression):

$$AIC = 2d\ - 2\log\left(\hat{L}\right),$$

where $\hat{L}$ is the likelihood function shown in the review.

Choosing the optimal combination of predictors: common training-error based measures

3. Instead of looking for a model with the smallest MSE, we could also define BIC (Bayesian Information Criterion):

$$BIC = 2d + some\ training\ error$$

Specifically, if the objective function is to minimize MSE (regression):

$$BIC = \frac{MSE}{\hat{\sigma}^2} + \frac{d*log(n)}{n},$$

if the objective function is to maximize likelihood (logistic regression):

$$BIC = d * \log(n) \ - 2\log\left(\hat{L}\right),$$

where $\hat{L}$ is the likelihood function shown in the review.

Quick math checking:

Using linear regression as an example, which one of the measures prefers smaller set of variables?

$$AIC = \frac{MSE}{\hat{\sigma}^2} + 2\frac{d}{n}$$

$$BIC = \frac{MSE}{\hat{\sigma}^2} + \frac{d * log(n)}{n}$$

Quick math checking:

Using linear regression as an example, which one of the measures prefers a smaller set of variables?

$$AIC = \frac{MSE}{\hat{\sigma}^2} + 2\frac{d}{n}$$

$$BIC = \frac{MSE}{\hat{\sigma}^2} + \frac{d * log(n)}{n}$$

- BIC, because log(n) > 2 for any n > 7, with n being the number of sample units. In other words, having more variables makes BIC increase faster than AIC.

Choosing the optimal combination of predictors: common training-error based measures

4. Instead of looking for a model with the largest R-squared, we define:

$$Adjusted\ R^2 = 1 - \frac{\frac{RSS}{n-d-1}}{\frac{TSS}{n-1}}.$$

Recall that $R^2 = 1 - \frac{RSS}{TSS}$, where RSS is the sum of the squared error, and TSS is the sum of variance in response.

While RSS tends to decrease (similar to MSE) as the number of predictors increases, $\frac{RSS}{n-d-1}$ is impacted by the number of predictors d, and thus might not decrease as the number of predictors decrease.

Choosing the optimal combination of predictors: common training-error based measures

Measures defined on the training set	Corresponding adjusted measure
1. Mean squared error (MSE)	$C_p = \text{MSE} + \frac{2d\hat{\sigma}^2}{\text{n}}$
2. Some sort of training error (e.g., MSE for regression, likelihood for classification)	AIC
3. Some sort of training error (e.g., MSE for regression, likelihood for classification)	BIC
4. R-squared	Adjusted R-squared

To recap

The easiest way to decide on **the best combination of predictors** is to try different combinations. In a linear regression setting, they are called *best subset selection, forward selection, backward selection,* etc.

All the naïve selection methods (best subset, forward, backward) requires a user-defined stopping rule. The easiest one to implement in R is to ***minimize the Akaike information criterion (AIC).***

- In short, AIC is an estimator of the prediction error.

- Use the lm() function to build a null model and a full model.
- Use the stepAIC function to do forward or backward selection.

To recap

The easiest way to decide on **the best combination of predictors** is to try different combinations. In a linear regression setting, they are called *best subset selection, forward selection, backward selection,* etc.

All the naïve selection methods (best subset, forward, backward) requires a user-defined stopping rule. In other words, **we can stop searching when we have a good enough**

- validation error, ***or***
- an adjusted training error
  - C_p, AIC, BIC, Adjusted R-squared

Agenda

- Why variable selection?
- Naïve approach
  - an example on linear regression
  - different ways to determine the optimal model
- Complex approach
  - Ridge
  - Lasso

Shrinkage methods

The objective function of a linear regression is to find a set of parameters that could minimize the squared error:

$$\min (y - f(x))^2,$$

where $f(x)$ is the linear regression model. Recall that each parameter is the coefficient for one of the predictors.

How do I control the number of predictors?

Shrinkage methods

The objective function of a linear regression is to find a set of parameters that could minimize the squared error:

$$\min (y - f(x))^2,$$

where $f(x)$ is the linear regression model. Recall that each parameter is the coefficient for one of the predictors.

How do I control the number of predictors?

- If a parameter is estimated at 0, then the corresponding predictor gets eliminated.

Shrinkage methods: ridge regression

Instead of purely minimizing the squared error, *ridge regression* minimizes

$$\min\left(y - (\beta_0 + \beta_1 x_1 + \ \ldots + \beta_D x_D)\right)^2 + \ \lambda\left({\beta_1}^2 + \cdots + {\beta_D}^2\right),$$

In other words, in ridge regression, while we are still trying to find parameter estimates that could minimize MSE, a larger parameter estimate β_j gets ***penalized***.

Shrinkage methods: ridge regression

Instead of purely minimizing the squared error, *ridge regression* specifically refers to a regression with the following objective function:

$$\min\left(y - (\beta_0 + \beta_1 x_1 + \ \ldots + \beta_D x_D)\right)^2 + \ \lambda\left({\beta_1}^2 + \cdots + {\beta_D}^2\right),$$

In other words, in ridge regression, while we are still trying to find parameter estimates that could minimize MSE, a larger parameter estimate β_j gets ***penalized***.

- This type of objective function is called *loss + penalty* objective function, and the objective function of almost *any* model could be formulated in this way.

Shrinkage methods: ridge regression

Instead of purely minimizing the squared error, *ridge regression* minimizes

$$\min\left(y - (\beta_0 + \beta_1 x_1 + \ \ldots + \beta_D x_D)\right)^2 + \ \lambda\left({\beta_1}^2 + \cdots + {\beta_D}^2\right),$$

In other words, in ridge regression, while we are still trying to find parameter estimates that could minimize MSE, a larger parameter estimate β_j gets ***penalized***.

- Which scenario corresponds to a larger penalty: $\lambda = 0, \lambda = 0.5, \lambda = 50$?

Shrinkage methods: ridge regression

Instead of purely minimizing the squared error, *ridge regression* minimizes

$$\min (y - (\beta_0 + \beta_1 x_1 + \ \ldots + \beta_D x_D))^2 + \ \lambda\left({\beta_1}^2 + \cdots + {\beta_D}^2\right),$$

In other words, in ridge regression, while we are still trying to find parameter estimates that could minimize MSE, a larger parameter estimate β_j gets ***penalized***.

- λ is called a tuning parameter/hyperparameter. A larger λ signals a heavier penalty on large parameter estimations.
- How to find the best λ?

Shrinkage methods: ridge regression

Instead of purely minimizing the squared error, *ridge regression* minimizes

$$\min (y - (\beta_0 + \beta_1 x_1 + \ \ldots + \beta_D x_D))^2 + \ \lambda\left({\beta_1}^2 + \cdots + {\beta_D}^2\right),$$

In other words, in ridge regression, while we are still trying to find parameter estimates that could minimize MSE, a larger parameter estimate β_j gets ***penalized***.

- λ is called a tuning parameter/hyperparameter. A larger λ signals a heavier penalty on large parameter estimations.
- How to find the best λ?
  - **Cross-validation! Define a list of λ, and see which one works the best.**

Shrinkage methods: Lasso

Similar to Ridge regression, the Lasso has the following *loss + penalty* objective function:

$$\min\ (y - (\beta_0 + \beta_1 x_1 + \ \ldots + \beta_D x_D))^2 + \ \lambda(|\beta_1| + \cdots + |\beta_D|)\,.$$

We are still trying to penalize large parameter estimates, but what is the difference?

Shrinkage methods: Lasso

Similar to Ridge regression, the Lasso has the following *loss + penalty* objective function:

$$\min\ (y - (\beta_0 + \beta_1 x_1 + \ \ldots + \beta_D x_D))^2 + \ \lambda(|\beta_1| + \cdots + |\beta_D|)\,.$$

We are still trying to penalize large parameter estimates, but what is the difference?

- The penalty in *ridge regression* are squared coefficients (aka ℓ_2 penalty)
- The penalty in Lasso is the absolute value of coefficients (aka ℓ_1 penalty)

But practically, what is the difference between *Ridge* and *Lasso*?

Let's rewrite the objective function for both:

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s \tag{6.8}$$

and

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s, \tag{6.9}$$

But practically, what is the difference between *Ridge* and *Lasso*?

Let's rewrite the objective function for both:

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s \tag{6.8}$$

and

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s, \tag{6.9}$$

Think about a two-variable setting, where we are only trying to calculate β_1 and β_2. Can you plot the contour of all four functions on the coordinates?

Comparing Ridge regression and the Lasso: a two-variable setting

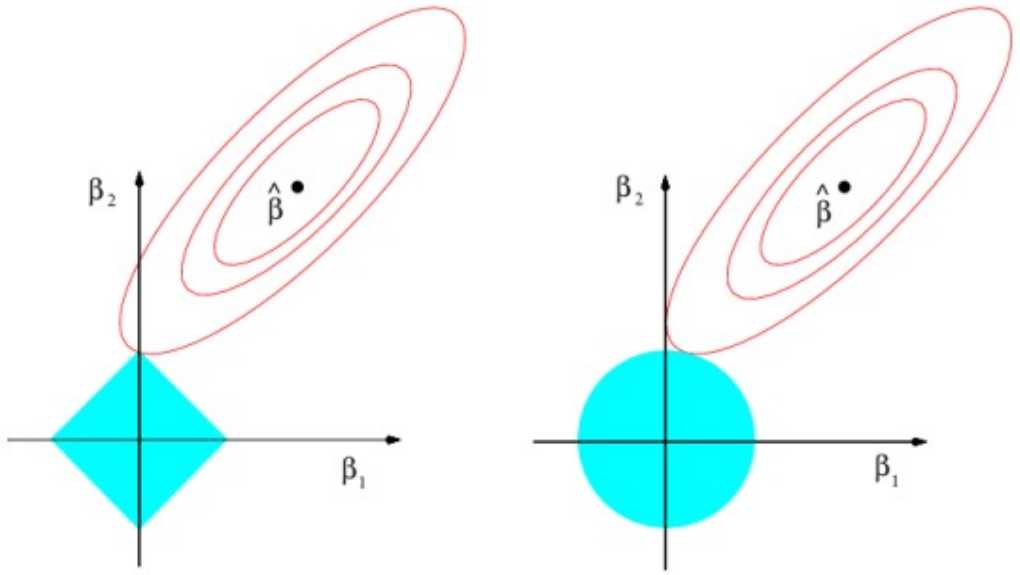

FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

This suggests that, in a ridge regression setting, the contour of the loss function will NEVER touch the axes.

However, we need the contour to touch y-axis to make β_1 = 0 and to then eliminate x_1 as a coefficient.

But practically, what is the difference between *Ridge* and *Lasso*?

Let's rewrite the objective function for both:

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s \tag{6.8}$$

and

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s, \tag{6.9}$$

Therefore, practically, the parameter estimates could be really small in *Ridge*, but they will NEVER be 0. *(why is it annoying?)*

Whereas in Lasso, some parameters could obtain an estimate of 0, and the corresponding variables could thus get eliminated from the model.