Variable Selection

Yang Wang

College of Charleston

Model misspecification

- Correctly specified model: the model contains all relevant predictors.
- Under specified model: the model does not include one or more important predictors.
- Over specified model: the model contains one or more redundant predictors.
- Model with extraneous variables: the model contains predictors that are neither related to the response nor to any other predictors.

Best subset selection using stepwise-type regression

- Evaluates only a small number of subset regression models by either adding or deleting regressors one at a time
- Three categories: (a) forward section, (b) backward elimination, (c) stepwise regression
- Set a significance level first
  - **alpha-to-enter** (α_{in}): significance level for deciding when to include a regressor into the model, typically greater than the usual 0.05 level so that it is not too difficult to enter regressors to the model
  - **alpha-to-out** (α_{out}): significance level for deciding when to drop a regressor from the model, typically greater than the usual 0.05 level so that it is not too easy to remove regressors from the model

Forward selection

- Begin with the assumption that there are no regressors in the model other than the intercept.
- Try each regressor one at a time and decide which regressor contributes most towards the mode.
- The regressor with the highest contribution value will be entered into the model if the corresponding p-value is less than α_{in}. Let's call this regressor x_1^*.
- In the next step,
  - for each of the remaining regressors calculate its contribution given that x_1^* is already present in the model.
  - the regressor with the highest contribution is included in the model if the corresponding p-value is less than α_{in}
- The process is repeated until (a) all regressors are added to the model, or (b) the p-value corresponding to the contribution at a particular step is greater than α_{in}

Backward elimination

- Begin with the model where all k regressors are present.
- For each regressors calculate its contribution given that the rest of the regressors are present in the model.
- The regressor with the smallest contribution will be dropped from the model if the corresponding p-value is greater than α_{out}
- The process is repeated until (a) all regressors are removed from the model, or (b) the p-value corresponding to the contribution at a particular step is less than α_{out}

Bidirection selection

- Begin with the assumption that there are no regressors in the model other than the intercept.
- Try each regressor one at a time and decide which regressor contributes most towards the mode.
- The regressor with the highest contribution value will be entered into the model if the corresponding p-value is less than α_{in}. Let's call this regressor x_1^*.
- In the next step,
  - for each of the remaining regressors calculate its contribution given that x_1^* is already present in the model.
  - the regressor with the highest contribution is included in the model if the corresponding p-value is less than α_{in}. Let's call this regressor x_2^*.
- In the next step,
  - calculate the contribution of x_1^* given that x_2^* is present in the model.
  - remove x_1^* from the model if the corresponding p-value is greater than α_{out}
- Repeat this process until no regressor can be added to the model

MLR full model for the autompg data

```
set.seed(123) ## set seed so that you get same partition each time
p2 <- partition.2(Dat, 0.7) ## creating 70:30 partition
training.data <- p2$data.train
test.data <- p2$data.test

########################
## Create full model ##
########################
mlr_full <- lm(mpg ~ ., data = training.data)
> summary(mlr_full)

Call:
lm(formula = mpg ~ ., data = training.data)
Residuals:
   Min     1Q Median     3Q    Max
-8.392 -2.396  0.142  2.236 14.430

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -15.027835   5.770755  -2.604  0.00973 **
cyl          -0.376568   0.397531  -0.947  0.34436
disp          0.017681   0.009369   1.887  0.06021 .
hp            0.006204   0.016982   0.365  0.71514
wt           -0.007902   0.000891  -8.869  < 2e-16 ***
acc           0.241237   0.123317   1.956  0.05148 .
year          0.739211   0.062903  11.752  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.414 on 266 degrees of freedom
Multiple R-squared:  0.8079,Adjusted R-squared:  0.8036
F-statistic: 186.5 on 6 and 266 DF,  p-value: < 2.2e-16
```

MLR full model for the autompg data

```
> vif(mlr_full)
      cyl      disp        hp        wt       acc      year
10.642017 22.047311  9.708623 13.550355  2.529953  1.200104

> cor(training.data)
            mpg        cyl       disp         hp         wt        acc       year
mpg   1.0000000 -0.7601165 -0.7945029 -0.7883175 -0.8321811  0.4494869  0.5507448
cyl  -0.7601165  1.0000000  0.9514141  0.8321914  0.9018888 -0.4813814 -0.2823076
disp -0.7945029  0.9514141  1.0000000  0.8877000  0.9486295 -0.5150932 -0.3050466
hp   -0.7883175  0.8321914  0.8877000  1.0000000  0.8805084 -0.6832878 -0.3805136
wt   -0.8321811  0.9018888  0.9486295  0.8805084  1.0000000 -0.4356627 -0.2787837
acc   0.4494869 -0.4813814 -0.5150932 -0.6832878 -0.4356627  1.0000000  0.2522982
year  0.5507448 -0.2823076 -0.3050466 -0.3805136 -0.2787837  0.2522982  1.0000000

# prediction on test data
yhat.full = predict(mlr_full, newdata=data.frame(test.data))
# RMSE for test data
error.test.full <- yhat.full - test.data$mpg
rmse.test.full <- sqrt(mean(error.test.full^2))
> rmse.test.full
[1] 3.568866
```

Stepwise selection on full model for the autompg data

- Full model:
  - *cyl* and *hp* are not significant
  - model suffers from severe multicollinearity.
- Use the *step*() function for subset selection.
- This function can be run with *direction* options: " *forward*" , " *backward*" and " *both*" .
- The function will return a model with best **Akaike information criterion** (AIC).
- $AIC = 2k - 2\log(\hat{\mathcal{L}})$ where k is the number of parameters in the model and $\hat{\mathcal{L}}$ is the maximum value of likelihood function for the model.
- The likelihood function measures the goodness of fit of a statistical model to a sample of data for given values of the unknown parameters.
- If there are several competing candidates, the model with the minimum AIC is considered as the best.
- The presence of k, the number of parameters, in the AIC formula serves as a penalty for overfitting.

Stepwise selection on full model for the autompg data

```
> step.model <- step(mlr_full)
Start:  AIC=677.39
mpg ~ cyl + disp + hp + wt + acc + year

       Df Sum of Sq    RSS    AIC
- hp    1      1.56 3102.6 675.53
- cyl   1     10.46 3111.5 676.31
<none>              3101.0 677.39
- disp  1     41.52 3142.5 679.02
- acc   1     44.61 3145.6 679.29
- wt    1    916.99 4018.0 746.12
- year  1   1609.95 4711.0 789.55

Step:  AIC=675.53
mpg ~ cyl + disp + wt + acc + year

       Df Sum of Sq    RSS    AIC
- cyl   1     11.23 3113.8 674.52
<none>              3102.6 675.53
- disp  1     45.05 3147.6 677.47
- acc   1     60.53 3163.1 678.80
- wt    1   1148.84 4251.4 759.53
- year  1   1699.39 4802.0 792.78

Step:  AIC=674.52
mpg ~ disp + wt + acc + year

       Df Sum of Sq    RSS    AIC
<none>              3113.8 674.52
- disp  1     36.60 3150.4 675.71
- acc   1     58.93 3172.7 677.64
- wt    1   1145.95 4259.8 758.07
- year  1   1693.84 4807.6 791.10
```

Stepwise selection on full model for the autompg data

```
> summary(step.model)

Call:
lm(formula = mpg ~ disp + wt + acc + year, data = training.data)

Residuals:
    Min      1Q  Median      3Q     Max
-8.7428 -2.5320  0.0946  2.1210 14.3700

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.476e+01  4.901e+00  -3.012  0.00284 **
disp         1.201e-02  6.766e-03   1.775  0.07704 .
wt          -7.735e-03  7.788e-04  -9.931  < 2e-16 ***
acc          2.087e-01  9.266e-02   2.252  0.02513 *
year         7.318e-01  6.061e-02  12.074  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.409 on 268 degrees of freedom
Multiple R-squared:  0.8072,Adjusted R-squared:  0.8043
F-statistic: 280.4 on 4 and 268 DF,  p-value: < 2.2e-16
```

Stepwise selection on full model for the autompg data

- The **Sum of Sq** associated to a variable refers to the increment in regression sum of square when that variable is added to the model given that the rest of the variables present in the model are already included.
- **RSS** column displays the residual sum of square of the model when the associated variable is dropped.
- The **AIC** column shows the AIC value of the model when the associated variable is removed.
- The $<$ *none* $>$ row has the RSS and AIC of the current model (*i.e.* when no variable is removed).
- At each step the variable having the lowest sum of square in the previous step is dropped if that results in a decrease in AIC value.
- The final model returned has four variables *disp*, *wt*, *acc* and *year*.
- The *step*() function also works on *glm* objects. That means, this function can be used for performing subset selection for logistic regression as well.
- Note that, it is possible to get a different subset of variables if the model runs on a different partition. A **cross validation** approach is thus preferred.

Stepwise selection on full model for the autompg data using cross validation approach

```
library(caret)

## K-fold Cross Validation
# value of K equal to 5
set.seed(0)
train_control <- trainControl(method = "cv",
                              number = 5)

# Fit K-fold CV model
step_kcv <- train(mpg ~ ., data = training.data,
                  method = "lmStepAIC", trControl = train_control)
> step_kcv$finalModel

Call:
lm(formula = .outcome ~ disp + wt + acc + year, data = dat)

Coefficients:
(Intercept)         disp           wt          acc         year
 -14.763222     0.012010    -0.007735     0.208670     0.731820

# prediction on test data
yhat.kcv = predict(step_kcv$finalModel, newdata=data.frame(test.data))
# RMSE for test data
error.test.kcv <- yhat.kcv - test.data$mpg
rmse.test.kcv <- sqrt(mean(error.test.kcv^2))
> rmse.test.kcv
[1] 3.558836
```

Cautions regarding stepwise-like procedures

- The stepwise-like methods yields a single final model, but it is possible to have multiple equally good models.
- The procedure doesn't allow the analyst to "think" and hence it may result in ignoring important information about the variable in the context of the business problem.
- It is possible to get a different subset of variables if the model runs on a different partition.
- It is also possible to get different models using backward and forward approach.
- Model suffers in the presence of multicollinearity.
- Greedy algorithm: often finds a local optimum instead of a global optimum.

Variable selection using regularized regression

- Linear regression uses the method of ordinary least square (OLS) to find the estimates of the regression coefficients.
- In OLS method, the estimates are obtained by minimizing the sum of square deviations $SSE = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_j i)^2$.
- OLS method works well when
  - there is little or no multicollinearity in data.
  - there are more observations than features (*i.e.* $n > p$).

Variable selection using regularized regression

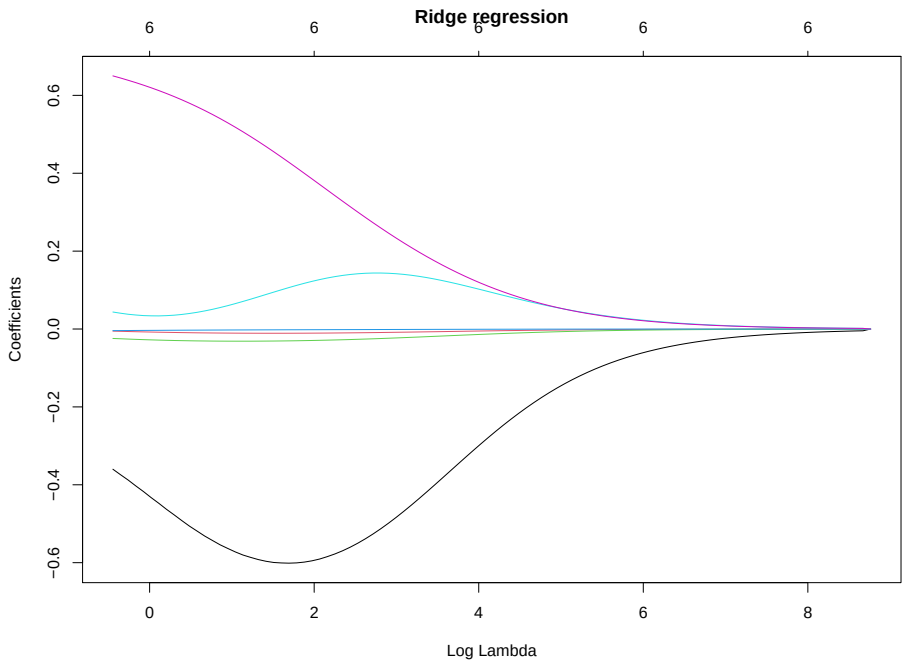
As the number of features increases, the following issues becomes more common.

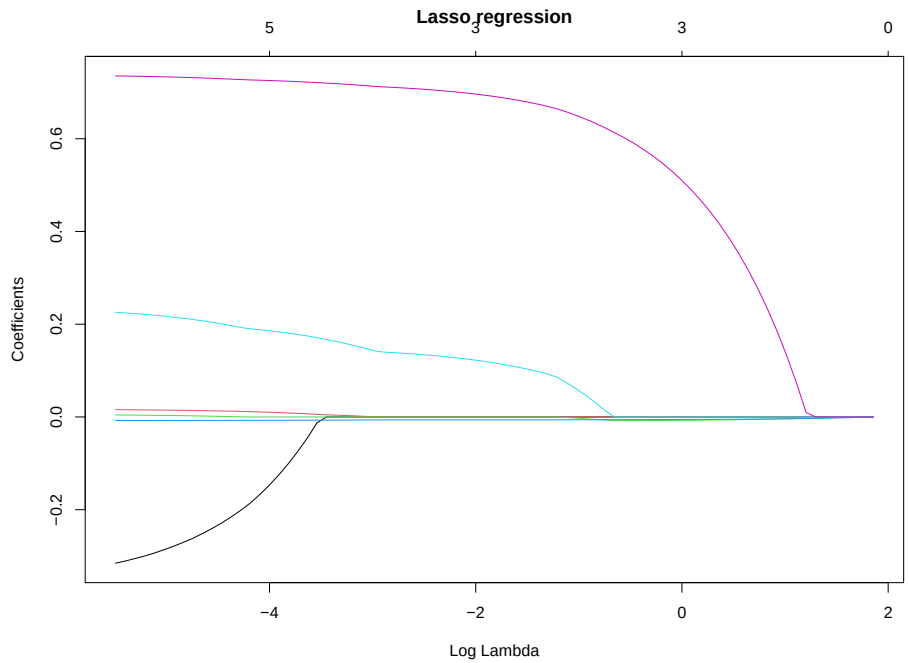
- Multicollinearity: If there are large number of features, it is more likely to have multiple features that are correlated.
- No solution: When the number of features are more than the number of observations, the OLS system of equation is not solvable i.e. it cannot produce estimates of the regression coefficients. (In particular, for MLR the least square estimate of the regression coefficients is given by $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$. When $p > n$, $\boldsymbol{X}$ is a $n \times p$ matrix resulting in $p \times p$ matrix $\boldsymbol{X}^\top \boldsymbol{X}$ which is not full rank and hence, not invertible.)
- Interpretability: With a large number of features, we often want to identify a smaller subset that are most important for the predictive model.

Variable selection using regularized regression

- In *regularized regression* or *shrinkage* method, the estimated effects are slowly shrunken towards zero.
- The main idea is to get the estimates by minimizing $SSE + P$ where P is the penalty factor.
- OLS vs. regularized regression:
  - OLS regression: minimize SSE
  - Ridge regression: minimize $SSE + \lambda \sum_{j=0}^{p} \beta_j^2$
  - Lasso regression: minimize $SSE + \lambda \sum_{j=0}^{p} |\beta_j|$
  - Elastic net regression: minimize $SSE + \lambda \sum_{j=0}^{p} (\alpha|\beta_j| + (1-\alpha)\beta_j^2)$
- $\lambda (\geq 0)$ is known as the *tuning parameter*.
- When $\lambda = 0$, ridge and lasso becomes OLS regression.
- As $\lambda \to \infty$, the penalty factor P becomes larger and that forces the regression coefficients to zero.

Regularized regression for the autompg data





Variable selection using regularized regression

The penalty imposed on the size of the coefficients is helpful for the following reasons (Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. "The elements of statistical learning." (2009)):

- When there are many correlated variables in a linear regression model, their coefficients can become poorly determined and exhibit high variance. A wildly large positive coefficient can be canceled by a similarly large negative coefficient on its correlated cousin. By imposing a size constraint on the coefficients, this problem is alleviated.
- The ridge regression solutions are given by $\hat{\beta} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$. The solution adds a positive constant to the diagonal of $\boldsymbol{X}^\top \boldsymbol{X}$ before inversion. This makes the problem nonsingular,even if $\boldsymbol{X}^\top \boldsymbol{X}$ is not of full rank.
- By imposing the penalty and forcing the coefficients to approach to zero, the variance of the model is reduced (because now estimates are less free to change), although this comes at the cost of adding bias. The OLS estimates are unbiased, but the regularization estimates are not.

Regularized regression for the autompg data

- Regularized regression requires that the features are standardized (*i.e.* zero mean and unit variance). *glmnet* has an inbuilt feature for standardization. So we don't need to do this.
- The *alpha* parameter in *glmnet* can be specified to use a ridge ($alpha = 0$) or lasso ($alpha = 1$) or elastic net ($0 < alpha < 1$) penalty.

```
library(glmnet)
library(caret)
set.seed(0)
train_control <- trainControl(method="cv", number=10)
glmnet.lasso <- train(mpg ~ ., data = training.data, method = "glmnet",
                      trControl = train_control,
                      tuneGrid = expand.grid(alpha = 1,lambda = seq(0.001,0.1,by = 0.001)))
> glmnet.lasso$bestTune
   alpha lambda
73     1  0.073
# best coefficient
lasso.model <- coef(glmnet.lasso$finalModel, glmnet.lasso$bestTune$lambda)
> lasso.model
7 x 1 sparse Matrix of class "dgCMatrix"
                            1
(Intercept) -13.533219157
cyl           .
disp          .
hp            .
wt           -0.006393455
acc           0.136054351
year          0.708378688
```

Comparison between shrinkage methods

- Although both ridge and lasso imposes a penalty term which forces the coefficients to approach to zero, lasso actually allow some of the coefficients to be zero, whereas ridge estimates are close to zero, but not exactly zero. Therefore, if a smaller subset of variables is preferred for the interpretability of the model, lasso does that job.
- If the data has highly correlated variables, ridge regression shrinks the two coefficients towards one another. Lasso, on the other hand, chooses one of them and makes the other close to zero. Elastic net is a compromise between the two. It selects variables like lasso, and shrinks together the coefficients of correlated predictors like ridge.
- If in doubt, in practice, one can implement both, observe the performance of the models on the test data and choose the best model.