STAT302 Methods of Data Analysis 1
Module 10: Model Validation and Prediction

Austin Brown

University of Toronto, Department of Statistical Sciences

November 21, 2024

Regression model selection wrap-up and summary

Classical linear regression model selection strategy review

1. Choose a research hypothesis, determine variables of interest, and collect data
2. Exploratory data analysis and data cleaning
   - Data transformations
   - identify potential co/multicolinearity
   - Identify potential outliers and potential high leverage observations
3. Determine and fit a regression model, and assess the validity of the regression model (regression diagnostics)
4. Perform variable selection to determine a final model and assess the validity of the model assumptions (regression diagnostics)
5. Identify co/multicolinearity, outliers, high leverage, and influential observations. Take appropriate actions based on these findings.
6. Perform statistical inference, draw conclusions, assess goodness of fit, discuss properties of the final model

Which model selection method to use?

Which model selection method to use: hypothesis testing, AIC, or BIC?

Lecture 1

Learning objectives

- Predictive models
- Overfitting and the bias-variance trade-off
- Model validation and MSE
- Splitting off a training and test/validation set
- Potential issues with test/validation sets

Lecture 1 : Predictive models

Predictive models

- The goal of a **Predictive model** is to accurately predict on unseen data or generalize well.

Predictive model

For a predictive model, we do not necessarily require the modeling assumptions to hold true and we may include extra predictors that are not statistically significant to predict better.

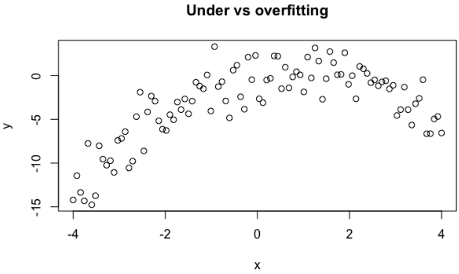
Lecture 1: Overfitting and the bias variance trade-off

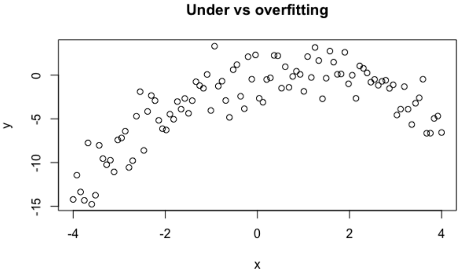
Underfitting and overfitting

Underfitting The model is too simple.

Overfitting the data means the model only provides good predictions on the data used to build it.

Underfitting and overfitting





Bias-variance tradeoff

The decomposition in general is:

$$\begin{aligned} E(MSE) &= E\left[(\hat{Y}^* - E(Y|X = \boldsymbol{x^*}))^2\right] \\ &= Bias(\hat{Y}^*)^2 + Var(\hat{Y}^*) \end{aligned}$$

or just MSE as it depends on how we define the MSE.

Bias-variance trade-off

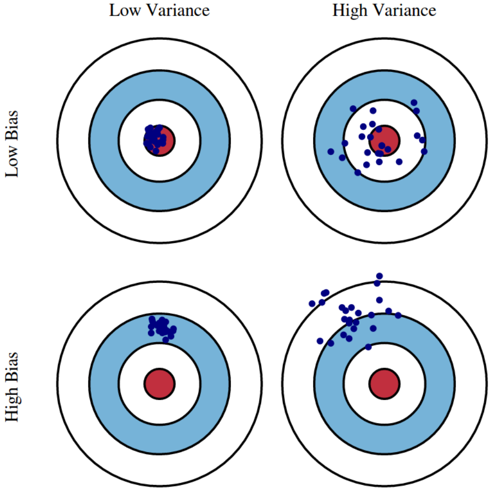


Image credit to Scott Fortmann-Roe

http://scott.fortmann-roe.com/docs/BiasVariance.html

Bias-variance tradeoff

Recall $\hat{Y}^* = \boldsymbol{x}^{*T}\hat{\boldsymbol{\beta}}$.

$$
\begin{aligned}
& E\left(\hat{Y}^* - \boldsymbol{\beta}^T\boldsymbol{x}^*\right)^2 \\
& = E\left(\hat{Y}^* - E(\hat{Y}^*) + E(\hat{Y}^*) - \boldsymbol{\beta}^T\boldsymbol{x}^*\right)^2 \\
& = E\left(\hat{Y}^* - E(\hat{Y}^*)\right)^2 + \left(E(\hat{Y}^*) - \boldsymbol{\beta}^T\boldsymbol{x}^*\right)^2 \\
& \qquad + 2\left(E(\hat{Y}^*) - \boldsymbol{\beta}^T\boldsymbol{x}^*\right) E\left(\hat{Y}^* - E(\hat{Y}^*)\right) \\
& = Var(\hat{Y}^*) + Bias(\hat{Y}^*)^2.
\end{aligned}
$$

Bias-variance tradeoff

We also have a bias-variance decomposition for MSE with respect to the response.

$$E\left[\left(\hat{Y}^* - Y^*\right)^2\right] = \sigma^2 + Var(\hat{Y}^*) + Bias(\hat{Y}^*)^2.$$

Variance with many predictors

$$Var(\hat{Y}^*) =$$

can become large with more predictors especially with multicollinearity such as polynomial and interaction terms.

Another motivation for model selection

The Bias variance trade-off

$$E(MSE) = Bias^2 + Var$$

gives another motivation for dropping variables in predictive models. For example, using BIC to choose a model can predict more accurately on new data than using AIC (important: this is not always true) since smaller models generally have less variance.

Lecture 1: Model validation

Using an independent validation data set

Consider new samples from the population $Y_1^*, \dots, Y_{n_{test}}^*$ and observed.

Model validation is the process of checking how your model performs on an independent dataset (new observations are from independent samples).

If we only check our predictions on the training set, we can **overfit**.

Using an independent validation data set

We have n_{train} samples from the population $\boldsymbol{Y}, \boldsymbol{X}$ and observations $\boldsymbol{y}, \boldsymbol{X}$ used to fit the model or **train** the model.

We have n_{test} independent samples from the population $\boldsymbol{Y}^*, \boldsymbol{X}^*$ and observions observations $\boldsymbol{y}^*, \boldsymbol{X}^*$ used to **validate** or **test** the fit.

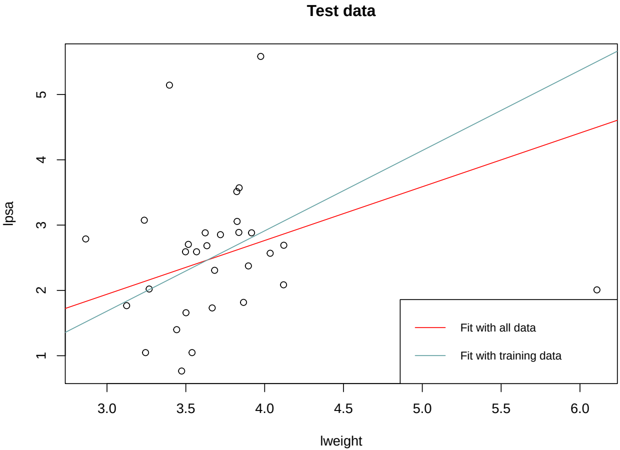
Train/validation split

We have a data set of n observations. Divide the data up into two sets: a training set and a test/validation set. So $n = n_{test} + n_{train}$.

- The training observations of size n_{train} are used to fit the linear regression model.
- The test/validation observations n_{test} of size are used to test the predictive power of the regression fit.

Important: The test set is never available to the methods used to fit the model. In other words, never train or use the test set to help fit the model on the training set.

Train/validation split in R



Question: Why are the two different fits different?

Train/validation split in R

Split the prostate data into a train/validation sets.

```
prostate_data = read.table("https://gattonweb.uky.edu/sheather/book/docs/
     datasets/prostateAlldata.txt",
                           header = T)
n = nrow(prostate_data)

set.seed(1)
n_train = floor(n/2)
n_test = n - n_train
train_indices = sample(1:n, # integers to sample
                       n_train, # sample size
                       replace = F) # without replacement

train_data = prostate_data[train_indices, ]
test_data = prostate_data[-train_indices, ]
```

Fitting on the training set

Least squares is fit on the training data set $\boldsymbol{Y}, \boldsymbol{X}$. We have a test/validation set $\boldsymbol{Y}^*, \boldsymbol{X}^*$ used to predict.

Predict the response of the new observation $y_i^*, \boldsymbol{x}_i^*$ (on average) using the new predictors with

$$\hat{Y}_i^* = \hat{\beta}_0 + \hat{\beta}_1 x_{i,1}^* + \cdots + \hat{\beta}_p x_{i,p}^*.$$

Each $\hat{\boldsymbol{\beta}}$ is fit/trained using only the training set here.

Fitting on the training set in R

[Sheather, 2009] has split the prostate data into a training and test set.

```
train_data = read.table("https://gattonweb.uky.edu/sheather/book/docs/datasets/
     prostateTraining.txt",
                           header = T)
test_data = read.table("https://gattonweb.uky.edu/sheather/book/docs/datasets/
     prostateTest.txt",
                       header = T)

fit = lm(lpsa ~ lcavol + lweight, data = train_data)
pred_test = predict(fit, test_data)
```

Training Mean-squared error

The (estimated) mean-squared training error is the average squared error of our predictions over the training set.

$$MSE_{train} = \frac{1}{n_{train}} \sum_{i=1}^{n_{train}} (Y_i - \hat{Y}_i)^2$$

where

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p.$$

Mean-squared prediction error

The (estimated) mean-squared prediction error is the average squared error of our predictions over the test set.

$$MSE^* = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (Y_i^* - \hat{Y}_i^*)^2$$

Using the training set, predict the response of the new observation $Y_i^*, \boldsymbol{x}_i^*$ (on average) using the new predictors.
Tries to estimate how the model will generalize to new data.

Mean-squared prediction error in R

Compute the mean squared prediction error on the test/validation set.

```
fit = lm(lpsa ~ lcavol + lweight, data = train_data)
pred_test = predict(fit, test_data)

y_test = test_data$lpsa
mse_test = mean( (y_test - pred_test)^2 )
```

A new model selection strategy

Assign each regression model a score based on prediction accuracy on the test/validation set. Instead of using AIC/BIC, we will assign each model a score with the **prediction MSE**.

For prediction, we want to pick the regression model with the smallest **prediction MSE**.

The final chosen model would then be fit to the **entire data set** and used.

Potential issues

- Smaller test/validation sets can create unreliable MSE prediction scores. In other words, the MSE prediction depends on the chosen test set.
- Highly influential observations may be left out of the training set
- We lose data when fitting the model from the training set for fitting the regression
- Determining the size of the validation set is subjective. Sometimes 10% is used for the test set of 50%. Some methodology exist.

Activity

For the prostate data set:

- Split off a 50% test set and select a predictive model with MSE.
- Using only the training set, select a model with backward selection using AIC/BIC. Compare the prediction MSE on the test set for both choices. Compare this with your model selected with MSE.

Lecture 2

Learning objectives

- Resampling methods for validation
- Cross validation

Repeating the train/validation splitting

One way is to attempt to reduce the variance in the test set is to repeat the train/validation split many times and average the MSE.

Repeating the train/validation splitting

- Randomly split the data into a train/validation split
- Fit the regression model and compute test MSE

$$MSE_r^*$$

- Repeat this R times and take the average over all of the computed MSE's

$$MSE_{resample} = \frac{1}{R}\sum_{r=1}^{R} MSE_r^*$$

- Choose the model with the lowest average MSE and then fit to the entire data.

Train/validation resampling example

```
set.seed(1)
n_resamples = 1000 # Number of resamples

resample_mse = c()
for (i in 1:n_resamples) {
  train_indices = sample(1:n, # integers to sample
                         floor(.5 * n), # sample size
                         replace = F) # without replacement

  train = prostate_data[train_indices, ]
  test = prostate_data[-train_indices, ]

  fit_train = lm("lpsa ~ lcavol + lweight + svi + lbph", data = train)
  pred_test = predict(fit_train, test)
  test_mse = mean( (test$lpsa - pred_test)^2 )
  resample_mse = c(resample_mse, test_mse)
}

print( mean(resample_mse) )
```

Lecture 2: LOO CV

Cross-validation

Instead of resampling split the data into K validation sets and use each one once as a validation set. The choice of K is subjective and has strengths and trade-offs. We will look at leave-one-out cross-validation which is $K = n$.

LOO cross-validation

1. Leave 1 observation out as your validation/test set and fit on the rest.
2. Compute the MSE.
3. Do this for each observation.
4. Take the average over all of the computed MSE's:

$$LOO_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_{(i)i}\right)^2$$

5. Choose the model with the lowest LOO_{MSE} and fit to the entire data.

LOO cross-validation

```
# Returns the MSE values from LOO CV with lpsa as response
prostate_data_loo_cv = function(model) {
  data = read.table("https://gattonweb.uky.edu/sheather/book/docs/datasets/
       prostateAlldata.txt",
                             header = T)
  loo_mse = c()

  for (i in 1:n) {
    train = data[-i, ]
    test = data[i, ]

    fit_train = lm(model, data = train)
    pred_test = predict(fit_train, test)
    mse = (test$lpsa - pred_test)^2
    loo_mse = c(loo_mse, mse)
  }

  return (loo_mse)
}


mean( loo_cv("lpsa ~ lcavol + lweight + svi + lbph", prostate_data) )
mean( loo_cv("lpsa ~ lcavol + lweight + svi", prostate_data) )
```

Lecture 2 : Activity

Activity

For the prostate data set:

- Choose a predictive model using LOO CV.
- Why might the prediction error be better for one model even though the coefficients are not statistically significant?

Module takeaways

- For what reason do we try to validate our model?
- Where does model validation occur in the flow of our analysis?
- How do we know if our model is validated?
- What do we do if we are not able to validate a model?

References I

Simon Sheather. *A modern approach to regression with R*. Springer Science & Business Media, 2009.