SDSC3006 Lab
2-Linear Regression

Langming LIU langmiliu2-c@my.cityu.edu.hk

School of Data Science
City University of Hong Kong

Contents

- Simple Linear Regression (univariate)
- Multiple Linear Regression
- Interaction Term
- Non-linear Regression (Polynomial)
- Qualitative Predictors

Simple Linear Regression

Preliminary

- Boston data set in the MASS library
- Predict medv of a neighborhood based on various predictors such as rm, age, and lstat

    (Meaning of variable: ?Boston)
- n = 506 neighborhoods around Boston
- Fit a simple linear regression:

    Y = medv, X = lstat

Construct Linear Model

- Use function lm(Y~X,data=datasets_name) to construct model and function summary() to show information of model:

```
library(MASS)
attach(Boston)
lm.fit=lm(medv~lstat)
summary(lm.fit)
```

Result Analysis

- Analyze result:
  - Is the parameter estimate accurate?
  - what’s the type of their relationship?
- Plot the Fitting Line

```
plot(lstat,medv,pch=20,col="black")
abline(lm.fit,lwd=3,col="red")
```

Prediction and Interval

- Prediction of f(X) of given X:

```
predict(lm.fit,data.frame(lstat=c(5,10,15)))
```

- Interval
  - **Confidence interval**: intervals of prediction of f(x)

    ```
    predict(lm.fit,data.frame(lstat=c(5,10,15)),interval="confidence")
    ```
  - **Prediction interval**: intervals of prediction of y given x

    ```
    predict(lm.fit,data.frame(lstat=c(5,10,15)), interval="prediction")
    ```
  - Prediction intervals are **wider** than confidence intervals because of the **random error.**

Multiple Linear Regression

Multivariate-model

- Data set: Boston in MASS
- Bivariate model: Y = medv, X1 = lstat, X2=age
- Use function lm(Y~X1+X2,data=datasets_name):

```
lm.fit=lm(medv~lstat+age,data=Boston)
summary(lm.fit)
```

- Using all predictors:

```
lm.fit=lm(medv~.,data=Boston)
summary(lm.fit)
```

Interaction Terms

Interaction

- Data set: Boston in MASS
- Interaction model: Y = medv, X1 = lstat, X2=age, X3=lstat:age (interaction term in R)
- Use function lm(Y~X1+X2+X1:X2,data=datasets_name):

```
lm.fit=lm(medv~lstat+age+lstat:age,data=Boston)
#Or
lm.fit=lm(medv~lstat*age,data=Boston)
summary(lm.fit)
```

Non-linear Regression

Quadratic Model

- Data set: Boston in MASS
- Quadratic model: Y = medv, X1 = lstat, X2=lstat^2
- Use function lm(Y~X1+I(X1^2),data=datasets_name):

```
#Linear model
lm.fit=lm(medv~lstat,data=Boston)
summary(lm.fit)
#Add a quadratic term
lm.fit=lm(medv~lstat+I(lstat^2),data=Boston)
summary(lm.fit)
```

Polynomial Model

- Data set: Boston in MASS
- Polynomial model: Y = medv, X1 = lstat, X2=lstat^2,...,X5=lstat^5
- Use function for polynomial model lm(Y~poly(lstat,5),data=datasets_name):

```
lm.fit=lm(medv~poly(lstat,5),data=Boston)
summary(lm.fit)
```

Polynomial Model

- Is it good if order is very high?

  No, higher order means high computational cost and may cause over-fitting (What is it?)

- Test polynomial models of different orders and plot them:

```
plot(lstat,medv,pch=20,col="black")
test_data=seq(0,40,0.2)
lm.fit=lm(medv~poly(lstat,order),data=Boston)
data_predict=predict(lm.fit,data.frame(lstat=test_data))
points(test_data,data_predict,pch=20,col='color')
```

Notice you should set the order and color! Try 3,5,10,15

Qualitative Predictors

Qualitative Predictors

- Data set: Carseats in ISLR2
- Predict Sales based on predictors such as Price, Urban(No/Yes), US(No/Yes), ShelveLoc(Bad/Medium/Good)

```
library(ISLR2)
attach(Carseats)
head(Carseats)
```

- R generates dummy variables for qualitative predictors automatically.

  n classes lead to n-1 dummy variables

- Modeling directly using function lm():

```
lm.fit=lm(Sales~.,data=Carseats)
summary(lm.fit)
```