Applied Statistical Methods

Statisitical Learning Using Regression Models

Xuemao Zhang
East Stroudsburg University

March 24, 2023

Outline

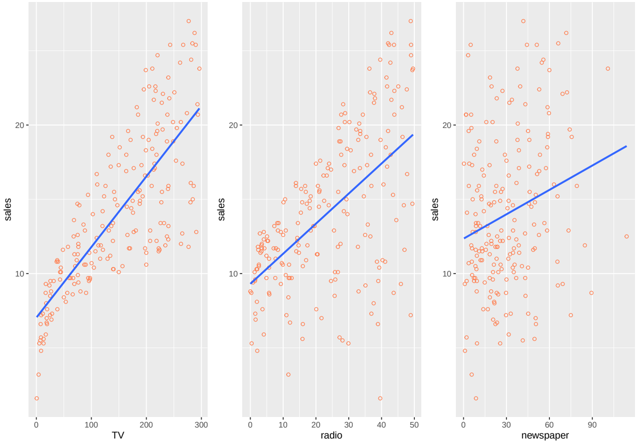
- Introduction to Statistical Learning
- Interactions in Regression Models
- Non-linear effects of predictors

Statistical learning

- Statistical learning arose as a subfield of Statistics.
- Statistical learning can be classified as supervised learning and unsupervised learning
- Supervised learning: Use a data set X to predict or detect association with a response y.
  - Regression
  - Classification
  - Hypothesis Testing
- Unsupervised learning: Discover the signal in X, or detect associations within X.
  - Dimension Reduction
  - Clustering

Statistical learning

- Example: Suppose that we are statistical consultants hired by a client to provide advice on how to improve sales of a particular product. The Advertising data set consists of the sales of that product in 200 different markets, along with advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper.



Statistical learning

- Shown are Sales vs TV, Radio and Newspaper, with a blue linear-regression line fit separately to each. Can we predict Sales using these three? Perhaps we can do better using a model

$$\text{Sales} \approx f(\text{TV, Radio, Newspaper})$$

- Here `Sales` is a response or target that we wish to predict. We generically refer to the response as Y .
- The variable `TV` is a `feature`, or `input`, or `predictor`; we name it X_1.
- Likewise name Radio as X_2, and so on.

Statistical learning

- We can refer to the input vector collectively as

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

- Now we can write our model as

$$Y = f(X) + \varepsilon$$

where ε captures measurement errors and other discrepancies.

Statistical learning

What is $f(X)$ good for?

- With a good f we can make predictions of Y at new points $X = x$.
- We can understand which components of $X = (X_1, X_2, \ldots, X_p)$ are important in explaining Y, and which are irrelevant. Here p is the number of features/predictors.
  - For example `Seniority` and `Years of Education` have a big impact on `Income`, but `Marital Status` typically does not.
- Depending on the complexity of f, we may be able to understand how each component X_j of X affects Y.
- Is there an ideal $f(X)$? In particular, what is a good value for $f(X)$ at any selected value of X, say $X = 4$? There can be many Y values at $X = 4$. A good value based on our knowledge in regression is the regression function

$$E(f(X)|X = 4)$$

  which means expected value (average) of Y given $X = 4$.

Statistical learning

- Given any x, $\varepsilon = Y - f(x)$ is the irreducible error - i.e. even if we knew $f(x)$, we would still make errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values. There are many possible estimates of $f(x)$.
- The ideal or optimal predictor of Y with regard to mean-squared prediction error: $f(x) = E(Y|X = x)$ is the function that minimizes $E[(Y - g(X))^2|X = x]$ over all functions g at all points $X = x$.
- For any estimate $\hat{f}(x)$ of $f(x)$, we have

$$E[(Y - \hat{f}(X))^2|X = x] = [f(x) - \hat{f}(x)]^2 + Var(\varepsilon).$$

Statistical learning

Methods to estimate f

- We will assume we have observed a set of training data

$$(x_1, y_1), \ldots, (x_n, y_n).$$

  We must then use the training data and a statistical method to estimate f.

- Statistical Learning Methods:
  - Parametric Methods
  - Non-parametric Methods

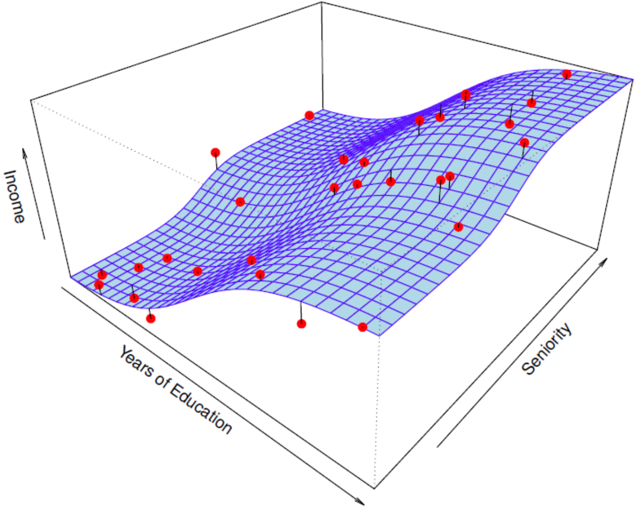
Statistical learning

Parametric Methods:

- It reduces the problem of estimating f down to one of estimating a set of parameters.
- They involve a two-step model based approach
  - STEP 1: Make some assumption about the functional form of f. For example, we propose a linear regression model.
  - STEP 2: Use the training data to fit the model i.e. estimate the unknown parameters in the proposed model.
- Even if the standard deviation is low we could get a bad answer if we use the wrong model. See the graphs about the true model and a fitted model using linear regression model.

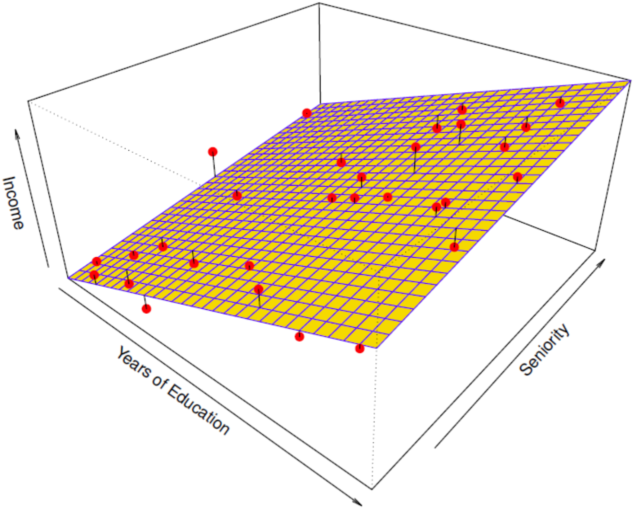
Statistical learning

- True model between Income and the two variables Seniority and Years of Education.



Statistical learning

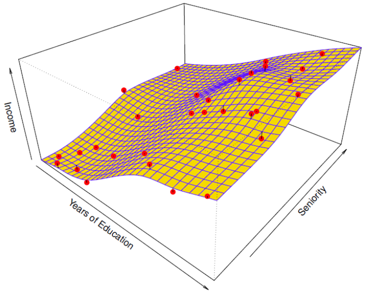
- Linear regression model fit to the simulated data.



Statistical learning

Non-parametric Methods

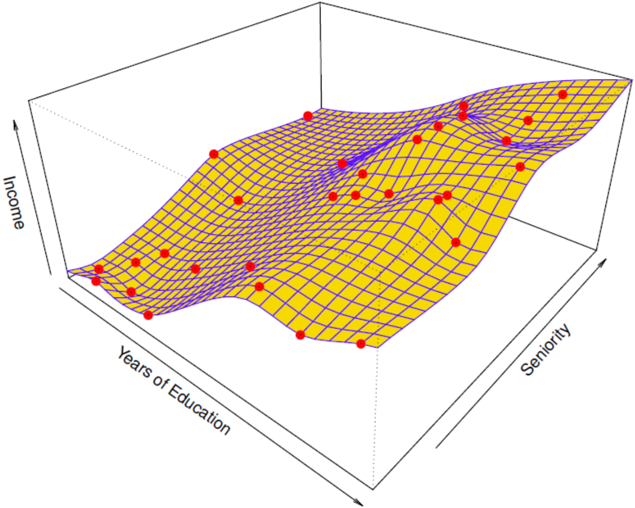
- They do not make explicit assumptions about the functional form of f.
- Advantages: They accurately fit a wider range of possible shapes of f.



- Non-parametric methods can also be too flexible and produce poor estimates for f when overfitting occurs.
- Disadvantages:
  - A very large number of observations is required to obtain an accurate estimate of f.

Statistical learning

- A fitted model makes no errors on the training data! Also known as overfitting.



Statistical learning

Some trade-offs

- Prediction accuracy versus interpretability.
  - A simple method such as linear regression produces a model which is much easier to interpret (the Inference part is better).
  - Even if you are only interested in prediction, it is often possible to get more accurate predictions with a simple, instead of a complicated model.
- Good fit versus over-fit or under-fit.
  - How do we know when the fit is just right?
- Parsimony versus black-box.
  - We often prefer a simpler model involving fewer variables over a black-box predictor involving them all.

Supervised vs. Unsupervised Learning

- We can divide all statistical learning problems into Supervised and Unsupervised situations
- Supervised Learning:
  - Supervised Learning is where both the predictors, $X_1, \ldots, X_p$, and the response, Y, are observed.
  - Most of this course will deal with supervised learning.
- Unsupervised Learning:
  - In this situation only the X_i's are observed.
  - We need to use the X_i's to guess what Y would have been and build a model from there.
  - A common example is market segmentation where we try to divide potential customers into groups based on their characteristics.
  - We will consider unsupervised learning at the end of this course.

Interactions in Regression Models

- Consider the advertising data with a MLR fit

```
import pandas as pd
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.formula.api import ols
Advertising=pd.read_csv("../data/Advertising.csv")
Advertising.columns
```

```
## Index(['ID', 'TV', 'radio', 'newspaper', 'sales'], dtype='object')
```

```
fit1=ols('sales~TV+radio+newspaper', data=Advertising).fit()
```

Interactions in Regression Models

- Note that the average effect on sales of a one-unit increase in TV is always 0.045765, regardless of the amount spent on radio.

```
print(fit1.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                  sales   R-squared:                       0.897
## Model:                            OLS   Adj. R-squared:                  0.896
## Method:                 Least Squares   F-statistic:                     570.3
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           1.58e-96
## Time:                        22:02:10   Log-Likelihood:                -386.18
## No. Observations:                 200   AIC:                             780.4
## Df Residuals:                     196   BIC:                             793.6
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept      2.9389      0.312      9.422      0.000       2.324       3.554
## TV             0.0458      0.001     32.809      0.000       0.043       0.049
## radio          0.1885      0.009     21.893      0.000       0.172       0.206
## newspaper     -0.0010      0.006     -0.177      0.860      -0.013       0.011
## ==============================================================================
## Omnibus:                       60.414   Durbin-Watson:                   2.084
```

Interactions in Regression Models

- update the model by removing newspaper

```
fit1=ols('sales~TV+radio', data=Advertising).fit()
```

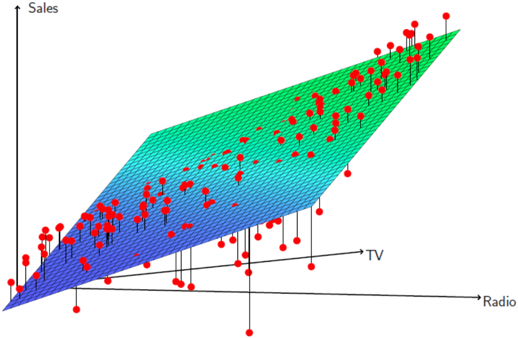
- R code

```
library(readr)
Advertising = read_csv("../data/Advertising.csv")
fit1=lm(sales~TV+radio+newspaper, data=Advertising)
fit1
fit1=lm(sales~TV+radio, data=Advertising);
fit1
```

Interactions in Regression Models

- But suppose that spending money on radio advertising actually increases the effectiveness of `TV` advertising, so that the slope term for `TV` should increase as radio increases.
- In this situation, given a fixed budget of $100,000, spending half on `radio` and half on `TV` may increase sales more than allocating the entire amount to either `TV` or to `radio`.
- In marketing, this is known as a *synergy effect, and in statistics it is referred to as an interaction effect.*

Interactions in Regression Models



- When levels of either `TV` or `radio` are low, then the true sales are lower than predicted by the linear model. But when advertising is split between the two media, then the model tends to underestimate sales.

Interactions in Regression Models

- Model takes the form

$$\begin{aligned} sales =& \beta_0 + \beta_1 \times TV + \beta_2 \times radio + \beta_3(radio \times TV) + \varepsilon \\ =& \beta_0 + (\beta_1 + \beta_3 \times radio) \times TV + \beta_2 \times radio + \varepsilon \end{aligned}$$

Interactions in Regression Models

```
fit2=ols('sales~TV+radio+TV*radio', data=Advertising).fit()
print(fit2.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                  sales   R-squared:                       0.968
## Model:                            OLS   Adj. R-squared:                  0.967
## Method:                 Least Squares   F-statistic:                     1963.
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):          6.68e-146
## Time:                        22:02:14   Log-Likelihood:                -270.14
## No. Observations:                 200   AIC:                             548.3
## Df Residuals:                     196   BIC:                             561.5
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept      6.7502      0.248     27.233      0.000       6.261       7.239
## TV             0.0191      0.002     12.699      0.000       0.016       0.022
## radio          0.0289      0.009      3.241      0.001       0.011       0.046
## TV:radio       0.0011   5.24e-05     20.727      0.000       0.001       0.001
## ==============================================================================
## Omnibus:                      128.132   Durbin-Watson:                   2.224
## Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1183.719
## Skew:                          -2.323   Prob(JB):                    9.09e-258
## Kurtosis:                      13.975   Cond. No.                     1.80e+04
```

Interactions in Regression Models

Interpretation

- The results suggests that interactions are important.
- The p-value for the interaction term $TV \times radio$ is extremely low, indicating that there is strong evidence for $H_a : \beta_3 \neq 0$.
- The R^2 for the interaction model is 96.8%, compared to only 89.7% for the model that predicts sales using `TV` and `radio` without an interaction term.

```
fit2.rsquared
```

```
## 0.9677905498482523
```

```
fit1.rsquared
```

```
## 0.8971942610828957
```

```
fit2.rsquared_adj
```

```
## 0.9672975480602154
```

```
fit1.rsquared_adj
```

```
## 0.8961505479974429
```

Hierarchy

- Sometimes it is the case that an interaction term has a very small p-value, but the associated main effects (in this case, `TV` and `radio`) do not.
- The **hierarchy principle**: If we include an interaction in a model, we should also include the main effects, even if the p-values associated with their coefficients are not significant.
  - The rationale for this principle is that interactions are hard to interpret in a model without main effects - their meaning is changed.

Interactions between qualitative and quantitative variables

- Consider the Credit data set, and suppose that we wish to predict balance using income (quantitative) and student (qualitative).

```
Credit= pd.read_csv("../data/Credit.csv")
fit3=ols('Balance~Income+Student', data=Credit).fit()
print(fit3.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                Balance   R-squared:                       0.277
## Model:                            OLS   Adj. R-squared:                  0.274
## Method:                 Least Squares   F-statistic:                     76.22
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           9.64e-29
## Time:                        22:02:19   Log-Likelihood:                -2954.4
## No. Observations:                 400   AIC:                             5915.
## Df Residuals:                     397   BIC:                             5927.
## Df Model:                           2
## Covariance Type:            nonrobust
## ===============================================================================
##                       coef    std err          t      P>|t|      [0.025      0.97
## -------------------------------------------------------------------------------
## Intercept         211.1430     32.457      6.505      0.000     147.333     274.9
## Student[T.Yes]    382.6705     65.311      5.859      0.000     254.272     511.0
## Income              5.9843      0.557     10.751      0.000       4.890       7.0
```

Interactions between qualitative and quantitative variables

- With interactions

```
fit4=ols('Balance~Income*Student', data=Credit).fit()
print(fit4.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                Balance   R-squared:                       0.280
## Model:                            OLS   Adj. R-squared:                  0.274
## Method:                 Least Squares   F-statistic:                     51.30
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           4.94e-28
## Time:                        22:02:21   Log-Likelihood:                -2953.7
## No. Observations:                 400   AIC:                             5915.
## Df Residuals:                     396   BIC:                             5931.
## Df Model:                           3
## Covariance Type:            nonrobust
## ================================================================================
##                             coef    std err          t      P>|t|      [0.025
## --------------------------------------------------------------------------------
## Intercept               200.6232     33.698      5.953      0.000     134.373
## Student[T.Yes]          476.6758    104.351      4.568      0.000     271.524
## Income                    6.2182      0.592     10.502      0.000       5.054
## Income:Student[T.Yes]    -1.9992      1.731     -1.155      0.249      -5.403
## ==============================================================================
```

Interactions between qualitative and quantitative variables

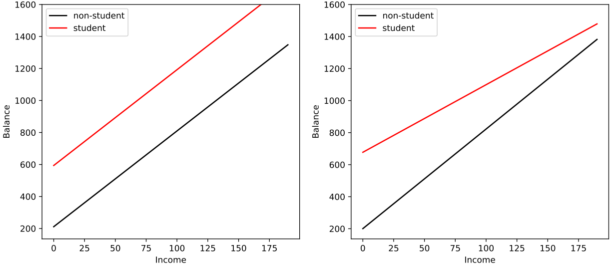
- Plot group regression line for no-interaction model

```
import matplotlib.pyplot as plt
import numpy as np
#min(Credit['Income']) and max(Credit['Income'])
income = np.linspace(0,190)

b0=fit3.params['Intercept']
b1=fit3.params['Student[T.Yes]']
b2=fit3.params['Income']

c0=fit4.params['Intercept']
c1=fit4.params['Student[T.Yes]']
c2=fit4.params['Income']
c3=fit4.params['Income:Student[T.Yes]']

fig, (ax1,ax2) = plt.subplots(1,2, figsize=(12,5))
ax1.plot(income, b0+b2*income, 'k')
ax1.plot(income, b0+b2*income+b1, 'r')
ax2.plot(income, c0+c2*income, 'k')
ax2.plot(income, c0+c1+c2*income+c3*income, 'r')
for ax in fig.axes:
    ax.legend(['non-student', 'student'], loc=2)
    ax.set_xlabel('Income')
    ax.set_ylabel('Balance')
    ax.set_ylim(ymax=1600)
```

```
## <matplotlib.legend.Legend object at 0x00000218402FE140>
```

Interactions between qualitative and quantitative variables

```
## <matplotlib.legend.Legend object at 0x000002183DC6A170>
## Text(0.5, 0, 'Income')
## Text(0, 0.5, 'Balance')
## (135.15824958790256, 1600.0)
## <matplotlib.legend.Legend object at 0x000002183DC6A260>
## Text(0.5, 0, 'Income')
## Text(0, 0.5, 'Balance')
## (136.70869106777747, 1600.0)
```



Interactions between qualitative and quantitative variables

- R code

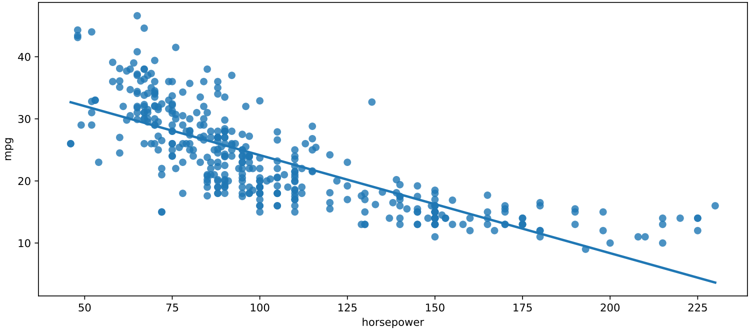
```
library(readr);
Credit= read_csv("../data/Credit.csv");
fit3=lm(Balance~Income+Student, data=Credit);
fit3;
fit4=lm(Balance~Income*Student, data=Credit);
fit4;
```

```
library(sjPlot);
plot_model(fit3, type = "pred", terms = c("Income","Student"),
           se=FALSE)
plot_model(fit4, type = "pred", terms = c("Income","Student"),
           se=FALSE)
```

Non-linear effects of predictors

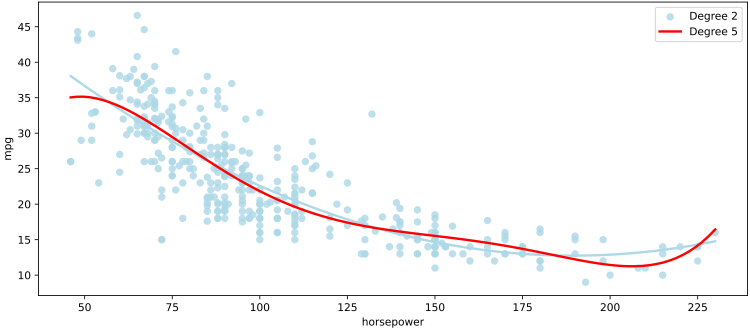
- There is a nonlinear relationship between mpg and horsepower

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
Auto= pd.read_csv("../data/Auto.csv")
sns.regplot(data=Auto,x='horsepower',y='mpg',ci=None)
plt.show()
```



Non-linear effects of predictors

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.regplot(data=Auto,x='horsepower',y='mpg',ci=None,
label='Degree 2', order=2,  color='lightblue')
sns.regplot(data=Auto,x='horsepower',y='mpg',ci=None,
label='Degree 5', order=5, scatter=False, color='red')
plt.legend()
plt.show()
```



Non-linear effects of predictors

- The figure suggests that the polynomial of degree may provide a better fit

```
fit = ols('mpg~horsepower + I(horsepower**2)', data = Auto).fit()
print(fit.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    mpg   R-squared:                       0.688
## Model:                            OLS   Adj. R-squared:                  0.686
## Method:                 Least Squares   F-statistic:                     428.0
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           5.40e-99
## Time:                        22:02:35   Log-Likelihood:                -1133.2
## No. Observations:                 392   AIC:                             2272.
## Df Residuals:                     389   BIC:                             2284.
## Df Model:                           2
## Covariance Type:            nonrobust
## ==========================================================================
##                           coef    std err          t      P>|t|      [0.025      0
## --------------------------------------------------------------------------
## Intercept              56.9001      1.800     31.604      0.000      53.360      6
## horsepower             -0.4662      0.031    -14.978      0.000      -0.527      -
## I(horsepower ** 2)      0.0012      0.000     10.080      0.000       0.001
## ==============================================================================
## Omnibus:                       16.158   Durbin-Watson:                   1.078
## Prob(Omnibus):                  0.000   Jarque-Bera (JB):               30.662
## Skew:                           0.218   Prob(JB):                     2.20e-07
```

License