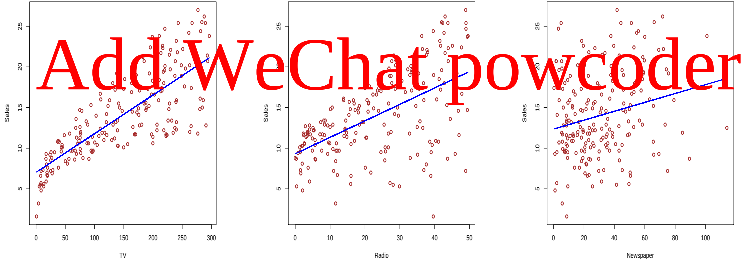
Introduction to Machine Learning

Sarat C. Dass
Department of Mathematical and Computer Sciences
Heriot-Watt University Malaysia Campus

Introduction to Machine Learning

What is Learning?

- Let's begin with a simple example. Suppose that you are a consultant hired to provide advice on how to improve sales of a particular product.
- The `Advertising` data set consists of the sales of a 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`.
- The data are displayed as below.



What is Statistical Learning? (cont.)

- What do you see from the plots? What do you conclude?
- It is not possible for our client to directly increase sales of the product. On the other hand, they can control the advertising expenditure in each of the three media.

- We determine if there is an association between advertising and sales, then we instruct our client to adjust advertising budgets, thereby indirectly increasing sales.

- In other words, our goal is to develop an accurate model that can be used to predict sales on the basis of the three media budgets.

Terms and Symbols Used for Statistical Learning

- In this setting, the advertising budgets are **input** variables while sales is an **output** variable.
- The inputs are typically denoted using the variable symbol X, with subscripts to distinguish them, e.g., X_1, X_2, X_3, etc.
- So X_1 might be the TV budget, X_2 the radio budget, and X_3 the newspaper budget.
- The output variable is usually denoted by the symbol Y. Here, $Y =$ sales.
- The inputs go by different names, such as **predictors**, **independent variables**, **features**, or sometimes just **variables**.
- The output variable is variable often called the **response** or **dependent variable**.
- Throughout the lectures and the textbook, we will use all of these terms interchangeably.

The Statistical Model

- More generally, suppose that we observe a quantitative response Y and p different predictors, $X_1, X_2, \cdots, X_p$.

- We assume that there is some relationship between Y and $X = (X_1, X_2, \cdots, X_p)$, which can be written in the very general form

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

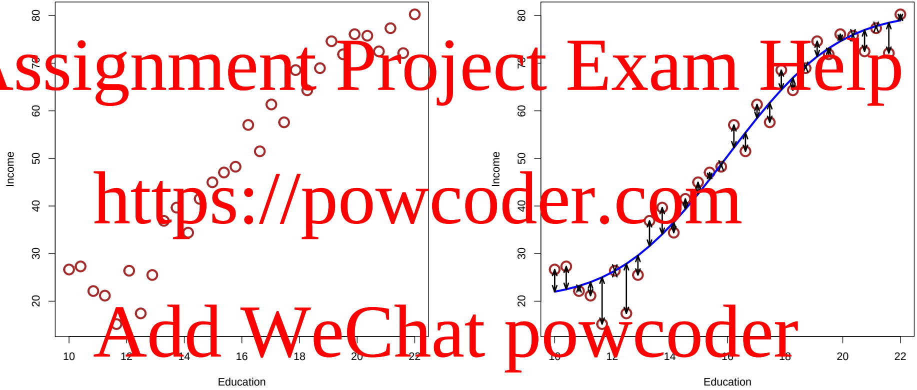
Here $f(X)$ is some fixed but unknown function of X and ϵ is a random error term, which is independent of X and has mean zero.

- In this formulation, $f(X)$ represents the **systematic** information that X provides about Y.
- Let's look at another example ...

Another example: Example 2

- As another example, consider Y = income and X = years of education for 30 individuals. The plot is given as below:
- The plot suggests that one might be able to predict income using years of education.
- However, the function $Y = f(X)$ that connects X to Y is gnerally unknown. In this situation, one must **estimate** f based on the observed points, call this $\hat{f}(X)$.
- To explain how this estimation will be performed, we use simulated data where the true $f(X) \equiv E(Y|X)$ is known. The true $f(X)$ is shown by the blue curve in the right-hand panel of the figure on the next slide.
- The vertical lines represent the error terms ϵ. We note that some of the 30 observations lie above the blue curve and some lie below it; overall, the errors have approximately mean zero.

Figures for Example 2



The vertical lines represent the error terms ϵ. We note that some of the 30 observations lie above the blue curve and some lie below it; overall, the errors have approximately mean zero.

More general fs

- In general, the function $f(X)$ may involve more than one input variable.
- In Figure 2.3 of the textbook, we plot income as a function of years of education and seniority: $f(X)$ is a $2D$ surface that must be estimated based on the observed data. But errors have same characteristics.

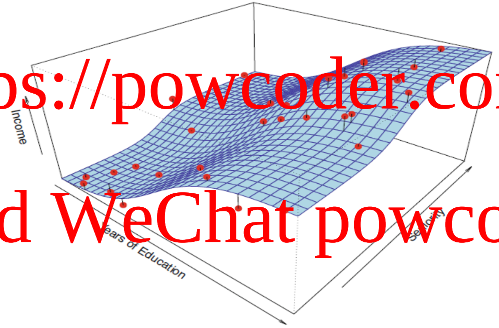


Figure: This figure is Figure 2.3 taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

The best regressor: formulation and derivation

- Suppose we wish to predict Y by $f(X)$.
- Let $(X, Y) \sim \pi(x, y)$ for which the marginal of X and the conditional of Y given X are denoted by $\pi(x)$ and $\pi(y|x)$, respectively.

- The mean square error (MSE) is given by

$$
\begin{aligned}
MSE &\equiv E_{\pi(x,y)}\left[Y - f(X)\right]^2 \\
&= E_{\pi(x,y)}\left[Y - E(Y|X) + E(Y|X) - f(X)\right]^2 \\
&= E_{\pi(x,y)}\left[Y - E(Y|X)\right]^2 + E_{\pi(x,y)}\left[E(Y|X) - f(X)\right]^2 \\
&= E_{\pi(x)} E_{\pi(y|x)}\left[Y - E(Y|X)\right]^2 + E_{\pi(x)}\left[E(Y|X) - f(X)\right]^2 \\
&= E_{\pi(x)}\left[Var(Y|X)\right] + E_{\pi(x)}\left[E(Y|X) - f(X)\right]^2
\end{aligned}
$$

- Thus, MSE is minimized when $f(X) = E(Y|X)$ which is the conditional expectation of Y given X calculated with respect to $\pi(y|x)$.
- $E(Y|X)$ is the best regressor or best predictor for Y based on X.

Steps of the proof

- To get the third equality from the second, note that

$$
\begin{aligned}
& E_{\pi(x,y)} \left[Y - E(Y|X) + E(Y|X) - f(X) \right]^2 \\
& = E_{\pi(x,y)} \left[(Y - E(Y|X))^2 - 2(Y - E(Y|X))(E(Y|X) - f(X)) \right. \\
& \qquad \left. + (E(Y|X) - f(X))^2 \right] \\
& = E_{\pi(x,y)} (Y - E(Y|X))^2 + E_{\pi(x,y)} (E(Y|X) - f(X))^2
\end{aligned}
$$

since the term $E_{\pi(x,y)} \left[(Y - E(Y|X))(E(Y|X) - f(X)) \right]$

$$
\begin{aligned}
& = E_{\pi(x)} E_{\pi(y|x)} \left[(Y - E(Y|X))(E(Y|X) - f(X)) \right] \\
& = E_{\pi(x)} \left[(E(Y|X) - f(X)) E_{\pi(y|x)} (Y - E(Y|X)) \right] \\
& \quad \text{(since } E(Y|X) - f(X) \text{ is a function of } X \text{ only)} \\
& = E_{\pi(x)} \left[(E(Y|X) - f(X))(E(Y|X) - E(Y|X)) \right] = 0
\end{aligned}
$$

Steps of the proof (cont.)

- To get the fourth and fifth equalities from the third, note that

$$
\begin{aligned}
& E_{\pi(x,y)}(Y - E(Y|X))^2 + E_{\pi(x,y)}(E(Y|X) - f(X))^2 \\
&= E_{\pi(x)} E_{\pi(y|x)}(Y - E(Y|X))^2 + E_{\pi(x)} E_{\pi(y|x)}(E(Y|X) - f(X))^2 \\
&= E_{\pi(x)} \mathrm{Var}(Y|X) + E_{\pi(x)}(E(Y|X) - f(X))^2
\end{aligned}
$$

since $E(Y|X) - f(X)$ does not depend on Y; it depends on X only, and hence

$$
\begin{aligned}
& E_{\pi(x)} E_{\pi(y|x)}(E(Y|X) - f(X))^2 \\
&= E_{\pi(x)} \left[(E(Y|X) - f(X))^2 E_{\pi(y|x)}(1) \right] \\
&= E_{\pi(x)}(E(Y|X) - f(X))^2,
\end{aligned}
$$

and $\mathrm{Var}(Y|X) \equiv E_{\pi(y|x)}(Y - E(Y|X))^2$ is the definition of the conditional variance of Y given X.

The learning target

- Recall the decomposition of the MSE as

$$MSE = E_{\pi(x)}\left[\text{Var}(Y|X)\right] + E_{\pi(x)}\left[E(Y|X) - f(X)\right]^2 \quad (1)$$

- It follows that

$$MSE \geq E_{\pi(x)}\left[\text{Var}(Y|X)\right]$$

with equality iff $E(Y|X) = f(X)$, the best predictor of Y given X under MSE criteria.

- $E(Y|X)$ is the target of our learning procedure. We want an $f(X)$ that is a good approximation of $E(Y|X)$ or even $f(X) = E(Y|X)$ exactly if possible.

The challenges involved

- The joint pdf $\pi(x, y)$ will usually be unknown and therefore $E(Y|X)$ is also unknown.
- The first error term on the RHS of (1) is the variance of ϵ which is inherent noise and hence irreducible.
- The second error term on the RHS of (1) is a measure of how well $f(X)$ approximates $E(Y|X)$.
- This error can be reduced by estimating f from a suitably selected class of functions which mimics the unknown form of $E(Y|X)$.
- This component is called the reducible error component - this is the focus for us.

Let's go back to the problem of estimation

- Recall the difficulties:
  - The joint pdf $\pi(x, y)$ is unknown and hence $E(Y|X)$ is unknown.
  - Need to choose a class of functions $\mathcal{C}$ which can mimic the form of the unknown $E(Y|X)$.
  - Need a error criteria to perform the estimation of f within $\mathcal{C}$.
- Here are the solutions:
  - Obtain a **training set** (x_i, y_i), $i = 1, 2, \cdots, n$ iid from $\pi(x, y)$.
  - Choose a class of functions $\mathcal{C}$ which can reasonably model $E(Y|X)$, i.e., the relationship between x and y in the training data set.
  - In the current context, choose the **empirical MSE** as the criteria for estimating $f \in \mathcal{C}$.

Measuring the quality of fit: The empirical MSE criteria

- In the regression setting, the most commonly-used measure is the (empirical) mean squared error (MSE), given by

$$MSE \equiv \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \qquad (2)$$

based on a training dataset (x_i, y_i), $i = 1, 2, \cdots, n$ of size n which are assumed to be iid from $\pi(x, y)$.

- Note that the above (empirical) MSE is an estimate of the population MSE given in (1) since

$$\text{pop. } MSE \equiv E_{\pi(x,y)}(Y - f(X))^2 \stackrel{\wedge}{=} E_{\hat{\pi}(x,y)}(Y - f(X))^2$$
$$= \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \equiv \text{emp. } MSE$$

where $\hat{\pi}(x, y)$ is the empirical distribution which puts mass $1/n$ on each point (x_i, y_i), $i = 1, 2, \cdots, n$.

Next, choose a class $\mathcal{C}$

- In order to estimate f, we have to choose a class of functions $\mathcal{C}$ that can reasonably model the relationship between X and Y that we observe in the training dataset.
- In Example 2, we can choose $\mathcal{C} = \mathcal{C}_1$ to be the class of linear functions: $f(X) = \beta_0 + \beta_1 X$, and β_0 and β_1 are unknown parameters that have to be estimated in order to obtain $\hat{f}$.
- The MSE criteria to obtain $\hat{f}$ becomes

$$\hat{f}(X) = \arg\min_{f \in \mathcal{C}_1} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 = \hat{\beta}_0 + \hat{\beta}_1 X$$

where

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min_{\beta_0, \beta_1} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Least squares regression

- Recall that

$$(\hat{\beta}_0, \beta_1) = \arg\min_{\beta_0, \beta_1} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

is precisely the least squares critria you learnt for simple linear regression previously.

- The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are the least squares estimates

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad \text{and}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

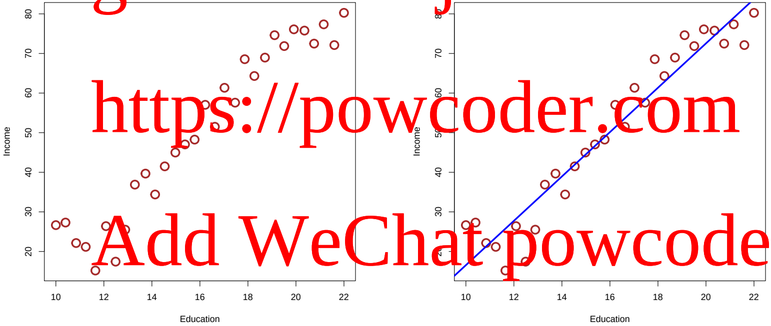
- Thus, the predictor of Y at given X is

$$\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 \, x$$

- In other words, the unknown $E(Y|X)$ is approximated by a linear function of X.

Least squares simple linear regression: Example 2

- Let's fit a least squares regression line to the plot of Y versus X in Example 2.



Ask yourself: Is the fit good? Your answer will determine whether $\mathcal{C}_1$ is reasonable class for the unknown $E(Y|X)$.

R codes

Here are the R codes used to generate the previous figures:


```
#Example 2: Income versus years of education
#Fit least squares regression line to scatter plot
library(splines)
#Need this library for residual diagnostics

#Read data
income1 <- read.csv("Income1.csv")
#Obtain scatter plot #Check out the trend
with(income1,
plot(Education, Income, type = "p", col = "brown",
xlab = "Education", ylab="Income", cex = 2, lwd=3)
)
```

R codes (cont.)


```
#Fit least squares regression line
lm_fit <- lm(Income ~ Education, data = income1)
#out is a lm object which will be used subsequently
#Summary of fit analysis
summary(lm_fit)
#Draw the regression line
abline(lm_fit, col = "blue", lwd=3)
```

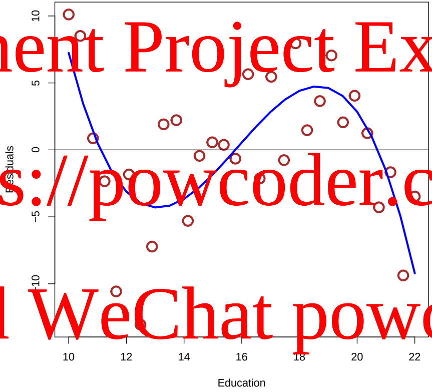
Output of lm fit

```
##
## Call:
## lm(formula = Income ~ Education, data = income1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.046  -2.293   0.472   3.288  10.110
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39.4463     4.7248  -8.349  4.4e-09 ***
## Education     5.5995     0.2882  19.431  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
##
## Residual standard error: 5.653 on 28 degrees of freedom
## Multiple R-squared:  0.931,Adjusted R-squared:  0.9285
```

Residual Diagnostics

```
#Check out the residual plot for diagnostics
resids <- residuals(lm_fit)
plot(income1$Education, resids,
type = "p", col = "brown",
xlab = "Education", ylab="Residuals", cex = 2, lwd=3)
abline(0,0)
#Obtain data driven trend of resids
resid_df <- data.frame(
Education = income1$Education, resids = resids)
#Same lm function works for fitting
resid_fit <- lm(resids ~ bs(Education, df=3), data = resid_df)
#Get predictions
xpoints <- with(resid_df,
seq(min(Education), max(Education),0.5))
ypoints = predict(resid_fit,
data.frame(Education=xpoints))
lines(xpoints, ypoints, col = "blue", lwd=3)
```

Residual Diagnostic Plot



So, conclude that the class $\mathcal{C}_1$ is not flexible enough. Need a more flexible class to model the unknown $E(Y|X)$ based on residual diagnostics.

A more flexible class: $\mathcal{C}_2$

- We can now consider a more flexible family compared to $\mathcal{C}_1$ which is $\mathcal{C}_2$, the class of polynomials in x of degree 2.
- The predictor $f \in \mathcal{C}_2$ has the representation

$$f(X) = \beta_0 + \beta_1 x + \beta_2 x^2$$

and clearly $\mathcal{C}_1 \subset \mathcal{C}_2$ where $\mathcal{C}_1 = \mathcal{C}_2 \Big|_{\beta_2 = 0}$

- We can now use the same MSE criteria and obtain $\hat{f} \in \mathcal{C}_2$ using

$$\hat{f}_2(X) = \arg\min_{f \in \mathcal{C}_2} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$$

where

$$(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2) = \arg\min_{\beta_0, \beta_1, \beta_2} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i - \beta_2 x_i^2)^2$$

- This is a special case of polynomial regression.

R codes to generate polynomial regression fit and plots

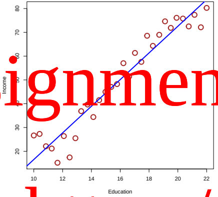
```
#Example 2: Polynomial regression with degree = 2
#Read data
income1 <- read.csv("Income1.csv")
#Obtain scatter plot #Check out the trend
with(income1,
plot(Education, Income, type = "p", col = "brown",
xlab = "Education", ylab="Income", cex = 2, lwd=3)
)

#Fit least squares polynomial regression with degree 2
poly_fit <- lm(Income ~ poly(Education, 2), data = income1)
#poly_fit is a lm object which will be used subsequently
#Summary of fit analysis
summary(poly_fit)
```

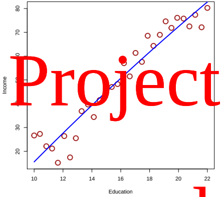
R codes to generate polynomial regression fit and plots (cont.)


```
#This part is to fit fhat to the scatter plot
xpoints = with(income1,
seq(min(Education), max(Education), 0.5))
#prediction using fhat at xpoints
ypoints <- predict(poly_fit,
data.frame(Education=xpoints))
#Plot the points on scatter plot
lines(xpoints, ypoints, col = "blue", lwd=3)
```

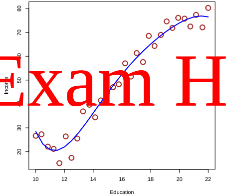
Figures of best fit for various $\mathcal{C}_p$



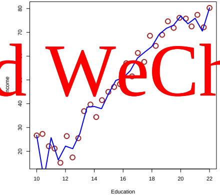
$p = 1$



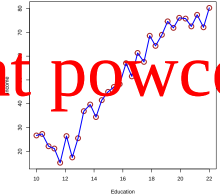
$p = 2$



$p = 5$



$p = 20$



$p = 29$

Points to note ...

- Define $MSE(\mathcal{C}_p)$ to be the smallest possible MSE based on the training set:

$$MSE(\mathcal{C}_p) = \arg\min_{f \in \mathcal{C}_p} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}_p(x_i))^2$$

where $\hat{f}_p \in \mathcal{C}_p$ is the best fit function for which the minimum is achieved.

- Note that as

$$\mathcal{C}_0 \subset \mathcal{C}_1 \subset \mathcal{C}_2 \subset \cdots \subset \mathcal{C}_P$$

the training set MSE satisfies

$$MSE(\mathcal{C}_0) \geq MSE(\mathcal{C}_1) \geq MSE(\mathcal{C}_2) \geq \cdots \geq MSE(\mathcal{C}_P) = 0$$

where $P = 29$ for our polynomial regression problem.

Training set MSE: Definition

- Definition: $MSE_{Train}(\mathcal{C})$ is called the training set MSE for class $\mathcal{C}$. Obtaining $\hat{f} \in \mathcal{C}$ is called training or learning from data. $\mathcal{C}$ is also called a class of learners.

- When $\mathcal{C}$ is a relatively small class, it is inflexible and is not able to mimic the behaviour of the true $E(Y|X)$.
- When $\mathcal{C}$ is a large class, it is very flexible and is able to mimic the behaviour of $E(Y|X)$ as well as other minute oscillations of y in the training set.

Overfitting: Definition

- The ability to over-represent minute oscillations (which are due to noise in Y) is called overfitting.
- Overfitting is bad and should be avoided since $\hat{f}$ obtained by overfitting only fits the training datset very well but does not generalize to similar "unseen" samples from $\pi(x, y)$
- Overfitted $\hat{f}$ does not give accurate prediction results on similar but "unseen" samples from $\pi(x, y)$
- Underfitting goes the other way. It is the inability of a class $\mathcal{C}$ to accurately approximate $E(Y|X)$, for example, in Example 2, the class $\mathcal{C}_1$ can only approximate linear trends well.

Overfitting and Underfitting for General Learning Classes

- Consider the ordering of $P+1$ classes as follows:

$$\mathcal{C}_0 \subset \mathcal{C}_1 \subset \mathcal{C}_2 \subset \cdots \subset \mathcal{C}_P$$

which implies

$$MSE_{Train}(\mathcal{C}_0) \geq MSE_{Train}(\mathcal{C}_1) \geq \cdots \geq MSE_{Train}(\mathcal{C}_{P-1}) \geq MSE_{Train}(\mathcal{C}_P)$$

$\leftarrow$ underfitting $\qquad\qquad$ $\rightarrow$ overfitting

- Somewhere in the middle is the optimal class p^* with optimal $\hat{f}_{p^*}$ that provides the best approximation to $E(Y|X)$ and can yield accurate and generalizable predictions.

Accurate and Generalizable Predictions: The Test Set

- Consider a new pair $(x_0, y_0) \sim \pi(x, y)$ and we wish to predict y_0 based on x_0
- We have obtained our estimated $\hat{f}$ from class $\mathcal{C}$ based on a training dataset.
- Important to note that (x_0, y_0) is NOT part of the training set since it is an unseen sample.
- The error in prediction is $\epsilon_0 \equiv y_0 - \hat{f}(x_0)$ and the MSE is

$$MSE = E_{\pi(x_0,y_0)}(y_0 - \hat{f}(x_0))^2$$

- Again, since $\pi(x, y)$ is unknown, we estimate the above MSE using iid samples from $\pi(x, y)$ NOT in the training dataset. This is called the test dataset denoted by $(x_{0,j}, y_{0,j}), j = 1, 2, \cdots, m$.
- Define the test set MSE by

$$MSE_{Test}(\mathcal{C}) = \frac{1}{m}\sum_{j=1}^{m}(y_{0,j} - \hat{f}(x_{0,j}))^2$$

$MSE_{Test}(\mathcal{C})$ guards against overfitting: An Intuitive Understanding

- Since $MSE_{Test}(\mathcal{C})$ is calculated based on unseen samples of (x_0, y_0) (unseen to the training of $\hat{f}$), the *random* fluctuations of y_0 around $\hat{f}(x_0)$ should ideally be ascertained by the class $\mathcal{C}$ as such: random, and not systematic.
- However, if $\mathcal{C}$ is too flexible and results in overfitting, $\hat{f}(x_0)$ will be close to the y value corresponding to x_0, $y(x_0)$ say, in the training dataset.
- As a result, fluctuations of y_0 from $\hat{f}(x_0)$ will be significantly larger because the deviations in this case are approximately $y_0 - y(x_0)$ and not $y_0 - E(Y|X = x_0)$.
- This will give rise to large values of $MSE_{Test}(\mathcal{C})$ if $\mathcal{C}$ is too flexible.

Cross-Validation Procedure: The Validation Set

- In practice, we are usually provided with a single database (DB) of samples $(x_i^{DB}, y_i^{DB}), i = 1, 2, \cdots, N$ and asked to perform machine learning tasks.

- We will need to determine the best learner $\hat{f}$ from a chosen collection of classes of learners

$$\{\mathcal{C}_p,\ p_0 \le p \le p_1\}$$

  indexed by the parameter p where a larger p represents a more flexible class.

- The way to approach this estimation problem to avoid underfitting and overfitting is to partition the original dataset into two: The Training and Test sets. The training dataset is used to learn f whereas the test dataset is used to guard against overfitting.

- This is the cross validation (CV) procedure, and the test dataset is also called the validation dataset.

Cross-Validation Procedure: Random Partitioning

- A random partition into training and validation sets can be done by randomly selecting n indices from 1 to N without replacement, say $\mathcal{T} \equiv \{i_1, i_2, \cdot, i_n\} \subset \{1, 2, \cdots, N\} \equiv \mathcal{N}$ and setting the training set as

$$\begin{aligned} \text{Training Set} &= \{(x_j^{DB}, y_j^{DB}), j \in \mathcal{T}\} \\ &\equiv \{(x_i, y_i), i = 1, 2, \cdots, n\} \end{aligned}$$

by a re-indexing of the indices in $\mathcal{T}$.

- The validation set is taken to consist of samples of all remaining indices:

$$\begin{aligned} \text{Validation Set} &= \{(x_j^{DB}, y_j^{DB}), j \in \mathcal{N} \setminus \mathcal{T}\} \\ &\equiv \{(x_{0,j}, y_{0,j}), j = 1, 2, \cdots, m\} \end{aligned}$$

by a re-indexing of indices in $\mathcal{V} \equiv \mathcal{N} \setminus \mathcal{T}$.

- $MSE_{\mathcal{T}}(\mathcal{C})$ and $MSE_{\mathcal{V}}(\mathcal{C})$ are defined as previously based on the partitioned training and test (validation) datasets, respectively.

Cross-Validation Procedure (cont.)

- For $k = 1, 2, \cdots, K$, do the following:
  - Randomly partition $\mathcal{N} = \mathcal{T}_k \cup \mathcal{V}_k$
  - For each $p = p_0, p_0 + 1, \cdots, p_1$:
  - Perform training. Obtain

$$\hat{f}_p = \arg\min_{f \in \mathcal{C}_p} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 \text{ and } MSE_{\mathcal{T}_k}(\mathcal{C}_p) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}_p(x_i))^2.$$

  - Perform validation. Obtain

$$MSE_{\mathcal{V}_k}(\mathcal{C}_p) = \frac{1}{m} \sum_{j=1}^{m} (y_{0,j} - \hat{f}_p(x_{0,j}))^2$$

Cross-Validation Procedure (cont.)

- Obtain the MSE averages for each p:

$$MSE_{Train}(\mathcal{C}_p) = \frac{1}{K}\sum_{k=1}^{K} MSE_{\mathcal{T}_k}(\mathcal{C}_p) \qquad \text{and}$$

$$MSE_{Validation}(\mathcal{C}_p) = \frac{1}{K}\sum_{k=1}^{K} MSE_{\mathcal{V}_k}(\mathcal{C}_p)$$

and plot these with respect to $p_0 \leq p \leq p_1$

- $MSE_{Train}(\mathcal{C}_p)$ should show a decreasing trend and $MSE_{Validation}(\mathcal{C}_p)$ should follow a U-shaped trend.

Illustration

We perform CV for the Income dataset in Example 2. Since $N = 30$, we take 70% of the samples for the training set and the rest are taken into the validation set.

The R codes for one attempt of CV is as follows


```
#Cross validation
#Example 2 Cross Validation procedure
library(dplyr)

#Example of one cross validation for one class of leaners
#Read data
income1 <- read.csv("Income1.csv")
#Training dataset data.frame
train <- income1 %>% sample_frac(0.7)
#Validation dataset data.frame
valid <- income1 %>% setdiff(train)
```

R codes (cont.)

```
#Determine class of learners (polynomial regression with degre
poly3_train_fit <- lm(Income ~ poly(Education,3),
data = train)
poly3_train_predcit <- predict(poly3_train_fit, train)
poly3_valid_predict <- predict(poly3_train_fit, valid)
MSE_train <- with(train,
 mean((Income - poly3_train_predcit)^2))
MSE_train

## [1] 16.47067

MSE_valid <- with(valid,
 mean((Income - poly3_valid_predict)^2))
MSE_valid

## [1] 14.2355
```

Now the full CV with $K = 50$ and $P = 6$

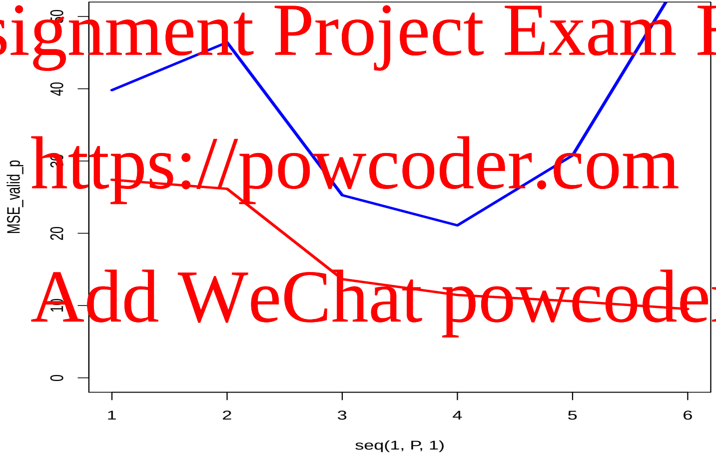

```r
#Now let's do full CV
K = 50;
P = 6;
MSE_train_mat <- vector("list", P)
MSE_valid_mat <- vector("list", P)

for (k in 1:K){
#Training dataset data.frame
train <- income1 %>% sample_frac(0.7)
#Validation dataset data.frame
valid <- income1 %>% setdiff(train)
```

Full CV with $K = 50$ and $P = 6$ (cont.)

```r
#Determine class of learners which are polynomials
#from degree 1 to 6
for (p in 1:P){
poly_train_fit <- lm(Income ~ poly(Education,p), data = train)
poly_train_predict <- predict(poly_train_fit, train)
poly_valid_predict <- predict(poly_train_fit, valid)
MSE_train_mat[[p]][k] <- with(train,
 mean((Income - poly_train_predict)^2))
MSE_valid_mat[[p]][k] <- with(valid,
 mean((Income - poly_valid_predict)^2))
}
}
MSE_train_p <- sapply(MSE_train_mat, mean)
MSE_valid_p <- sapply(MSE_valid_mat, mean)
plot(seq(1,P,1), MSE_valid_p, type="l",col="blue", ylim = c(0
lines(seq(1,P,1), MSE_train_p, type="l",col="red", lwd=3)
```

Plot of $MSE_{Train}(\mathcal{C}_p)$ (red) and $MSE_{Valid}(\mathcal{C}_p)$ (blue) versus p

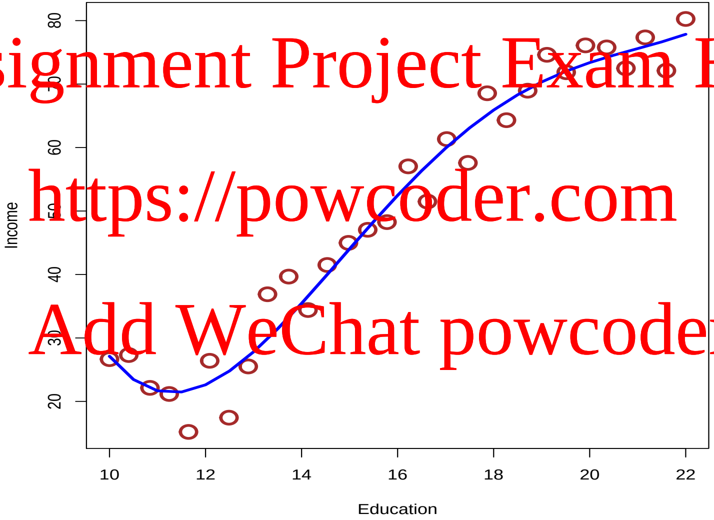


Note that best fit is at $p^* = 4$

Best fit plot with $p^* = 4$: R codes

```r
#Best fit is p*=4
poly_best_fit <- lm(Income ~ poly(Education,4), data = income1
with(income1,
plot(Education, Income, type = "p", col = "brown",
xlab = "Education", ylab="Income", cex = 2, lwd=3)
)
#This part is to fit that to the scatter plot
xpoints = with(income1,
 seq(min(Education), max(Education),0.5))
#prediction using that at xpoints
ypoints <- predict(poly_best_fit,
 data.frame(Education=xpoints))
#Plot the points on scatter plot
lines(xpoints, ypoints, col = "blue", lwd=3)
```

Best fit plot with $p^* = 4$



Choice of Class of Functions: Prediction Accuracy versus Model Interpretability

- Some classes are less flexible meaning that they produce a small range of shapes for f, e.g., linear regression.
- Other methods are considerably more flexible because they can generate a much wider range of possible shapes to estimate f.
- Why would we ever choose to use a more restrictive method instead of a very flexible approach?
- One answer that you have seen is to avoid overfitting.
- Another answer is that if you want to interpret the model parameters in a certain way which is related to the real problem, it is better to choose a less flexible method.
- In linear regression, we have an interpretation for the slope and the intercept but as we move to higher order polynomial functions, the interpretation of coefficients associated with higher powers of x is less clear.
- For example, if x is years of education, what does x^5 mean for the real problem?

Choice of Class of Functions: Prediction vs. Inference

- Why estimate f? Two reasons: Prediction and Inference.
- Prediction: We obtain $\hat{Y} = \hat{f}(X)$ and $\hat{f}$ is treated as a *black box*.
- We are only interested in how well $\hat{f}$ predicts future Ys. Here, we are not concerned with the form of f that results, and can select a more flexible class.
- On the other hand, inference means that we seek to understand how X affects Y.
- For example, we may want to know which independent variables are most associated with the response, are these relationships positive or negative relationships, or what effect will an increase in X have on Y?
- In these scenarios, $\hat{f}$ cannot be treated as a black box, and simpler model choices will help to answer such questions.

Bias versus Variance Trade Off: Introduction

- The CV procedure explained previously calculates $MSE_{Valid}(\mathcal{C})$ based on a validation dataset $\mathcal{V}$ after f has been trained on a training dataset $\mathcal{T}$.

- The sets $\mathcal{V}$ and $\mathcal{T}$ change at each cycle of the CV procedure.

Bias versus Variance Trade Off (cont.)

- The CV procedure thus tries to estimate the population version of $MSE_{Valid}(\mathcal{C})$ which is given by

$$E_{\pi(x_0,y_0)} E_{\pi^n(\underline{x},\underline{y})} \left(y_0 - \hat{f}(x_0\,;\,\underline{x},\underline{y}) \right)^2$$

  where

  - $\pi(x,y)$ is the joint pdf of (x,y) and $\pi^n(\underline{x},\underline{y})$ is the pdf of the iid training samples $(\underline{x},\underline{y}) \equiv \{(x_i,y_i), i = 1,2,\cdots,n\}$ under $\pi(x,y)$
  - $E_{\pi(x_0,y_0)}$ is the expectation with respect to a new unseen sample arising from $\pi(x,y)$

  - $\hat{f}(x_0\,;\,\underline{x},\underline{y})$ is the estimated f based on the training set $(\underline{x},\underline{y})$.

- We emphasize the dependence of $\hat{f}(x_0\,;\,\underline{x},\underline{y})$ on $(\underline{x},\underline{y})$ which can change when the training set changes.

Bias versus Variance Trade Off (cont.)

- Define $E(Y|X = x_0) \equiv f_0(x_0)$, E^n to be the expectation calculated with respect to $E_{\pi^n(\underline{x},\underline{y})}$, and $E^n(\hat{f}(x_0\,;\,\underline{x},\underline{y})) \equiv \bar{f}_0(x_0)$.
- Using same arguments as before, we can show that

$$E_{\pi(x_0,y_0)} E_{\pi^n(\underline{x},\underline{y})} \left(y_0 - \hat{f}(x_0\,;\,\underline{x},\underline{y}) \right)^2$$

$$= E_{\pi(x_0,y_0)} E^n \left(y_0 - f_0(x_0) + f_0(x_0) - \bar{f}_0(x_0) + \bar{f}_0(x_0) - \hat{f}(x_0\,;\,\underline{x},\underline{y}) \right)^2$$

$$= E_{\pi(x_0,y_0)} \left(y_0 - f_0(x_0) \right)^2 + E_{\pi(x_0,y_0)} \left(f_0(x_0) - \bar{f}_0(x_0) \right)^2 +$$

$$E_{\pi(x_0,y_0)} E^n \left(\bar{f}_0(x_0) - \hat{f}(x_0\,;\,\underline{x},\underline{y}) \right)^2$$

- Recall that the first term in the last equality,

$$E_{\pi(x_0,y_0)} \left(y_0 - f_0(x_0) \right)^2 = E_{\pi(x_0)} \mathrm{Var}(Y|X = x_0)$$

is the irreducible error.

Bias versus Variance Trade Off (cont.)

- The second term $E_{\pi(x_0,y_0)}\left(\bar{f}_0(x_0) - f_0(x_0)\right)^2$ is the expected square of the bias term $B \equiv \bar{f}_0(x_0) - f_0(x_0)$.
- This term measures how well, on the average, the trainings of f using class $\mathcal{C}$ is able to approximate the true $f_0(x_0)$.
- This term will become large if $\mathcal{C}$ is a more restrictive class which is not able to approximate $f_0(x_0)$ accurately.
- The third term $E_{\pi(x_0,y_0)} E^n \left(\bar{f}_0(x_0) - \hat{f}(x_0\,;\,\underline{x},\underline{y})\right)^2$ measures the variability of each training of f from its average over all trainings.
- This term will become large if $\mathcal{C}$ is a more flexible class which results in overfitting.

Bias versus Variance Trade Off (cont.)

- Thus, we have

$$MSE_{Valid}(\mathcal{C}) = \text{Irr. Error} + (\text{Training Bias})^2 + \text{Training Variance}$$

- For a general family of classes $\mathcal{C}_p$ indexed by p where large p indicates a more flexible class, we expect

$$(\text{Training Bias}(\mathcal{C}_p))^2 \quad \downarrow \quad \text{and} \quad \text{Training Variance}(\mathcal{C}_p) \quad \uparrow$$

as p increases.

Bias versus Variance Trade Off: Example 2

- This explains why we observed the U-shaped behaviour of $MSE_{Valid}(\mathcal{C}_p)$ in Example 2.
- When p is small, the bias term is large but the variance term is small. As p increases, the bias term becomes smaller but the variance term becomes larger.
- Thus, the sum of the two terms first decreases, achieves a minimum and then increases.
- Calculating the bias term requires the knowledge of $E(Y|X)$ which is unknown. So, the bias term cannot be calculated for real datasets. But this theoretical study is useful to understand the behaviour of $MSE_{Valid}(\mathcal{C}_p)$ in all situations.