Linear Statistical Modeling Methods with SAS

Introduction to Statisitical Learning and Model Validatation

Xuemao Zhang
East Stroudsburg University

April 1, 2024

Outline

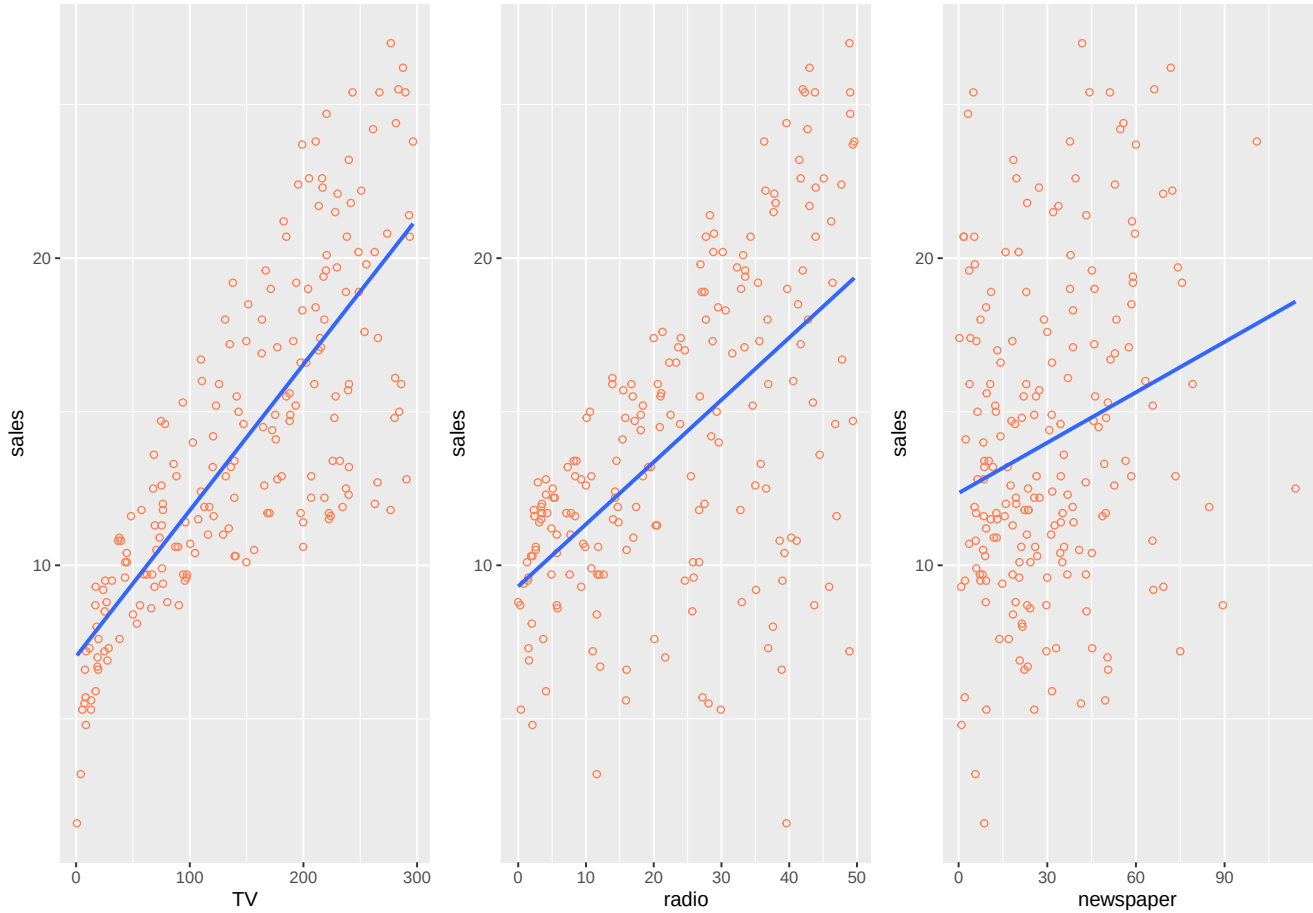
- Introduction to Statistical Learning
- Model Complexity
- Cross-validation
  - Validation Set Approach
  - Leave-one-out cross-validation (LOOCV).
  - K-fold cross-validation.

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

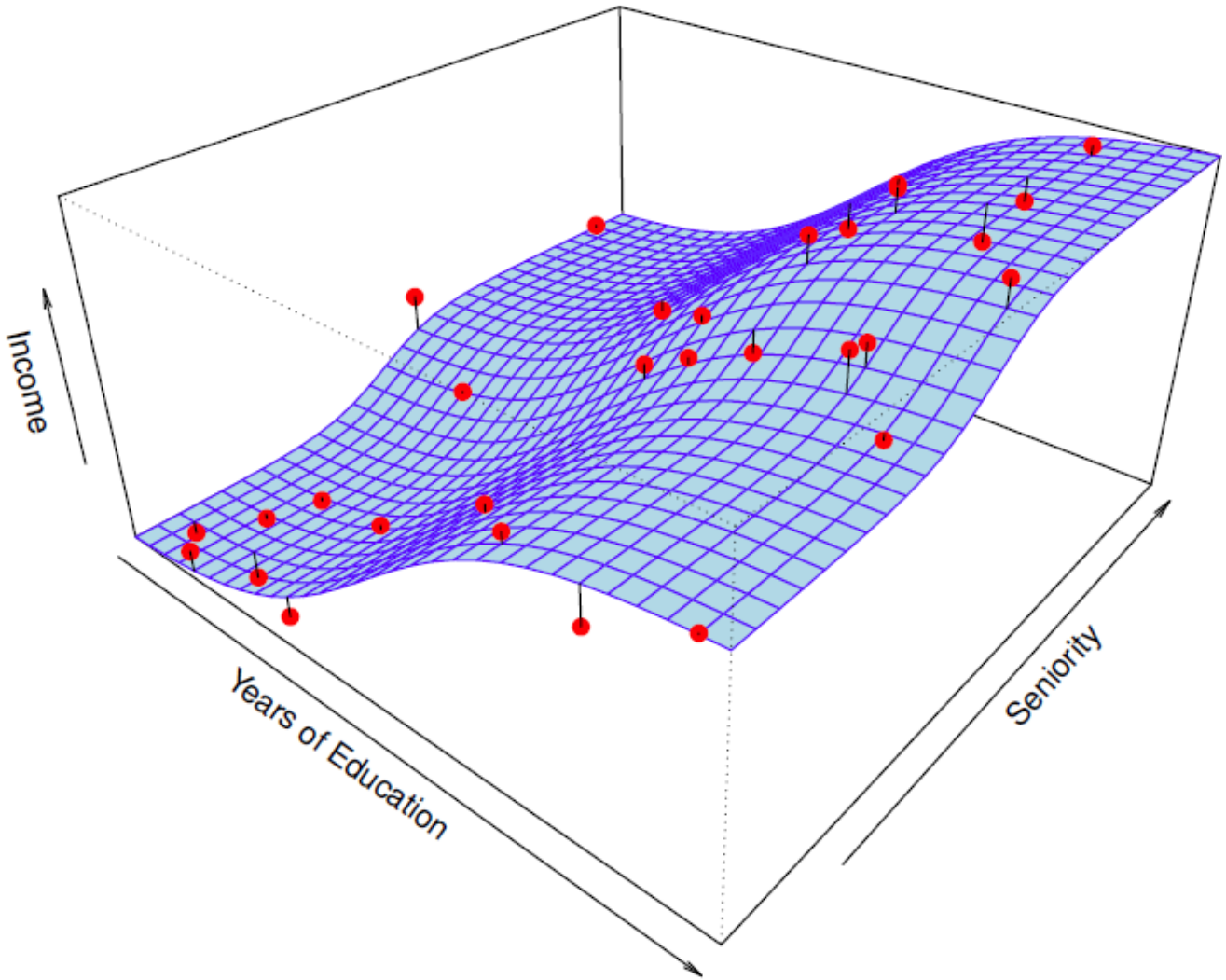
Statistical learning

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.

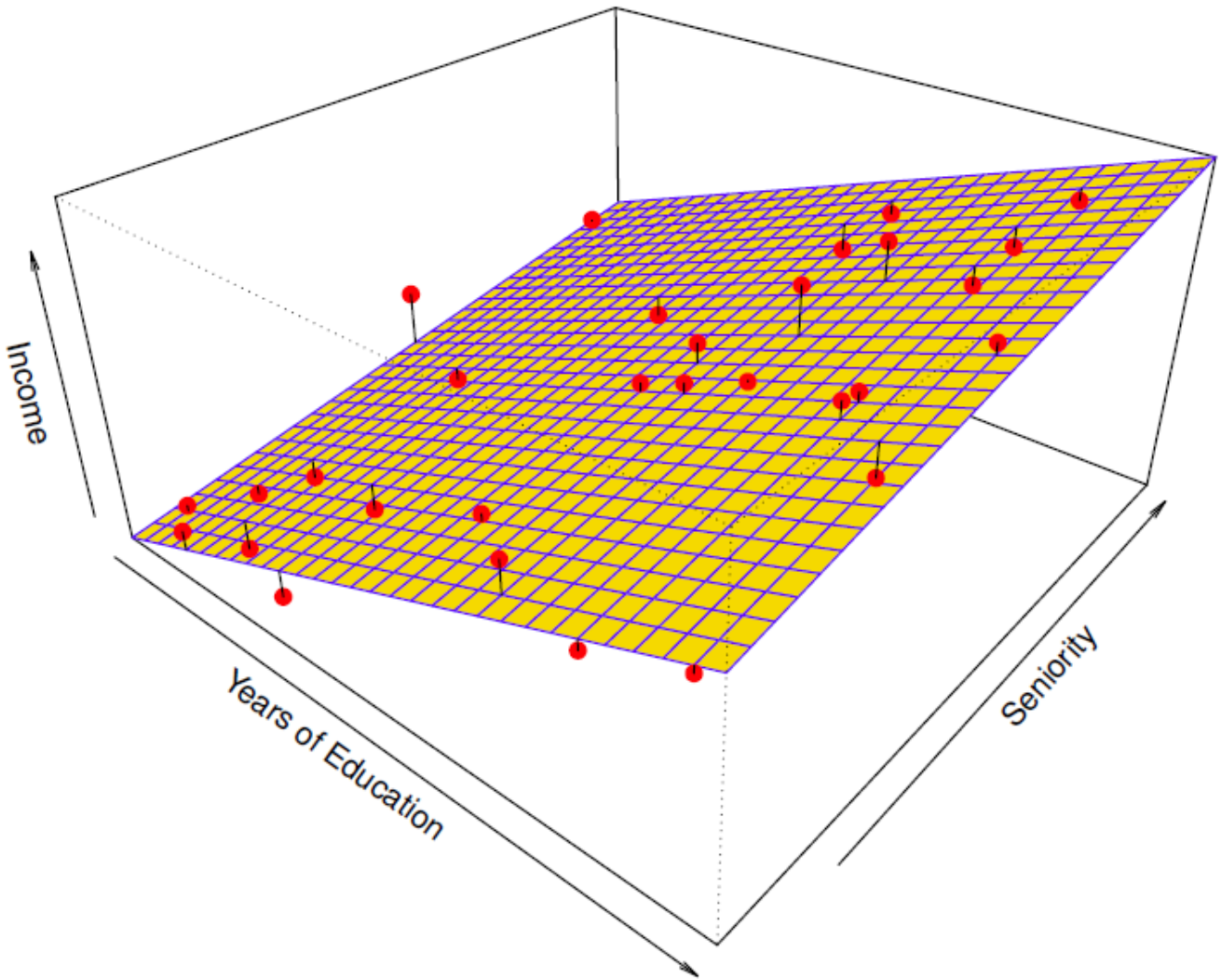
Statistical learning

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



Statistical learning

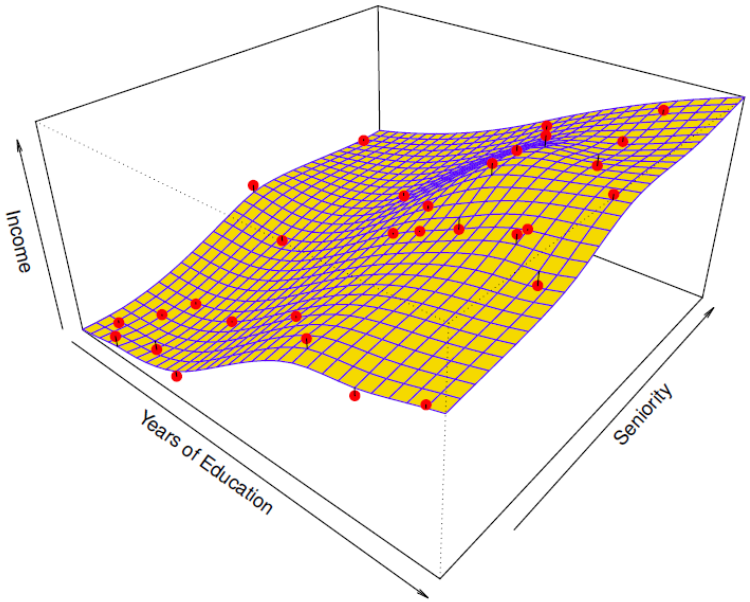
- Linear regression model fit to the simulated data.



Statistical learning

Non-parametric Methods

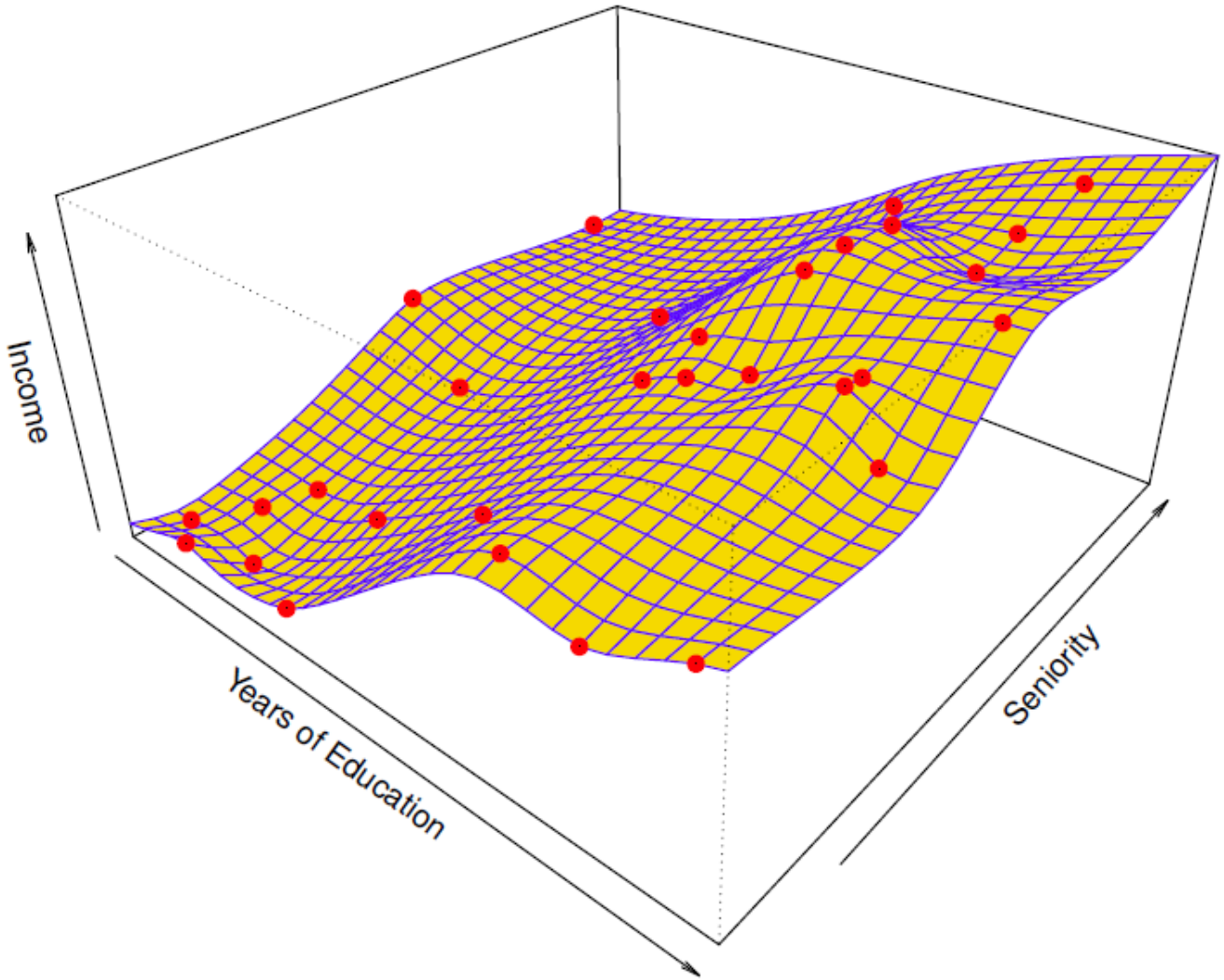
- They do not make explicit assumptions about the functional form of f.
- Advantages: They accurately fit a wider range of possible shapes of f.



- 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.

Model Complexity

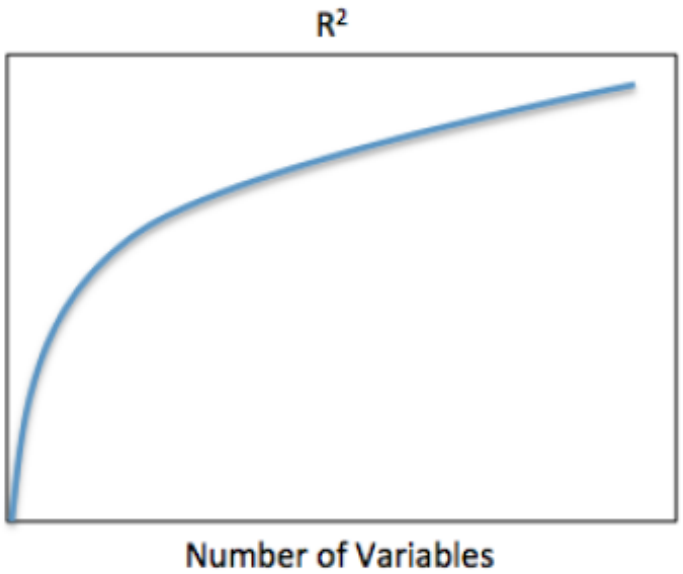
- When we fit a model, we use a **training set** of observations.
- We can evaluate the **training error**, i.e. the extent to which the model fits the observations used to train it. One way to quantify the training error is using the MSE (mean squared error)
- The training error is closely related to the R^2 for a linear model.
  - Big $R^2 \Leftrightarrow$ Small Training Error.
- Training error and R^2 are not good ways to evaluate a model's performance, because they will always improve as more variables are added into the model.

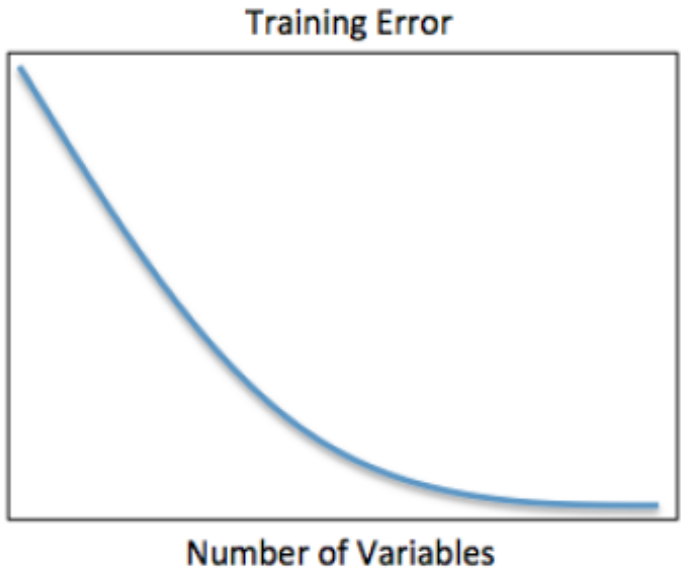
Model Complexity

- The problem? Training error and R^2 evaluate the model's performance on the training observations.
- If I had an unlimited number of features to use in developing a model, then I could surely come up with a regression model that fits the training data perfectly! Unfortunately, this model wouldn't capture the true signal in the data.
- We really care about the model's performance on **test observations** - observations not used to fit the model.
  - We split the data as **training data** and **test data**, and the **test data** is treated as future observations.

Model Complexity

- As we add more variables into the model. . .





- the training error decreases and the R^2 increases!

Model Complexity

- We really care about the model's performance on observations not used to fit the model!
  - Want to diagnose cancer for a patient not used in model training!
  - Want to predict risk of diabetes for a patient who wasn't used to fit the model!
- What we really care about:

$$(y_{test} - \hat{y}_{test})^2,$$

where

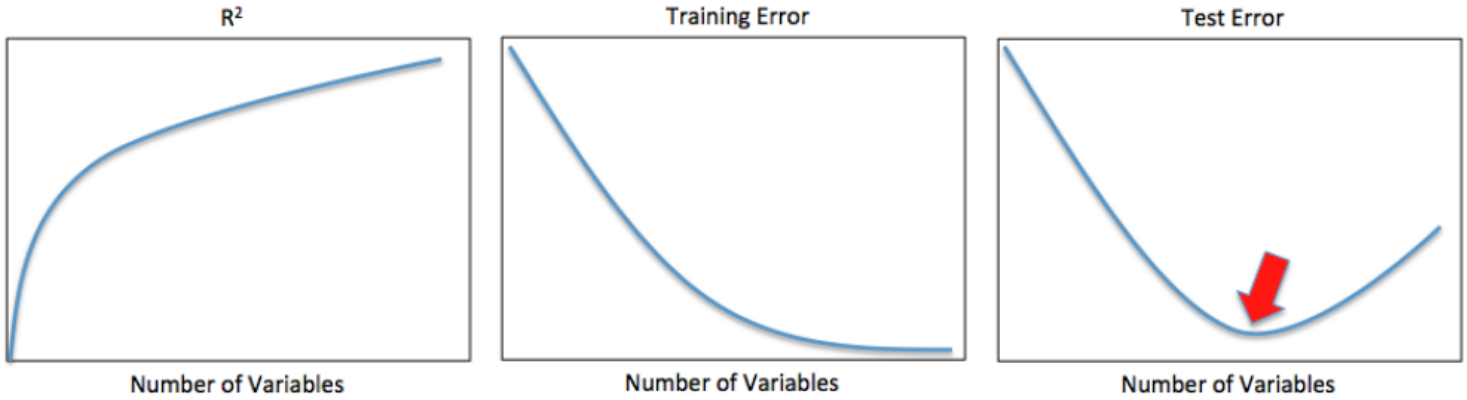
$$\hat{y}_{test} = \hat{\beta}_0 + \hat{\beta}_1 X_{test,1} + \cdots + \hat{\beta}_p X_{test,p},$$

and (X_{test}, y_{test}) was not used to train the model.

- The test error is the average of $(y_{test} - \hat{y}_{test})^2$ over a bunch of test observations.

Model Complexity

- As we add more variables into the model. . .



- the training error decreases and the R^2 increases!
- But the test error might not!

Model Complexity

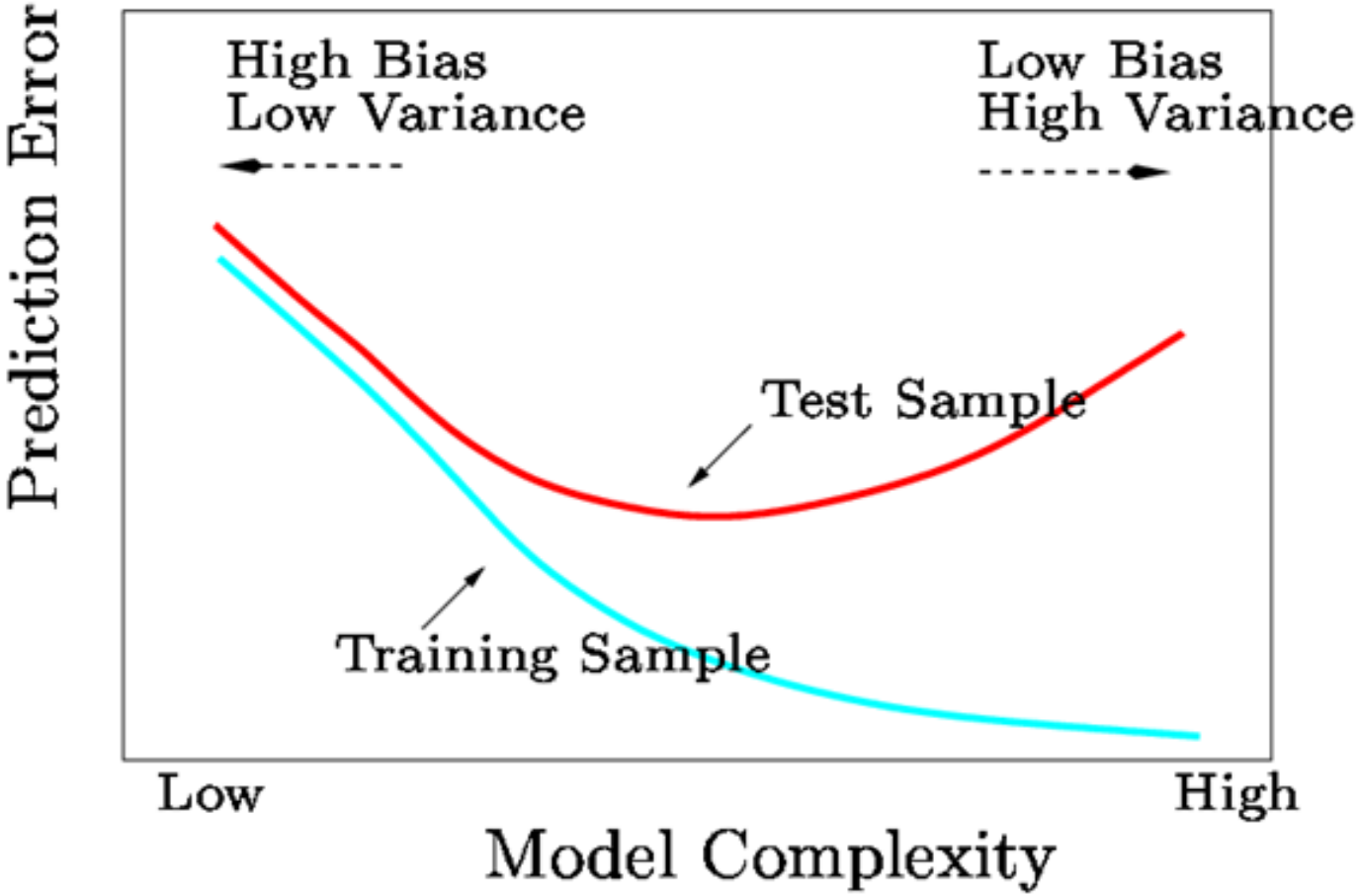
- As we fit more complex models - e.g. models with more variables - the training error will always decrease.
  - But the test error might not.
- Bias and Variance
  - As model complexity increases, if we were to repeat the experiment a huge number of times, the bias of $\hat{\beta}$ will decrease.
  - But as complexity increases, the variance of $\hat{\beta}$ will increase.
  - The test error depends on both the bias and variance:

$$\text{Test Error} = \text{Bias}^2 + \text{Variance}$$

  - There is a **bias-variance trade-off**. We want a model that is sufficiently complex as to have not too much bias, but not so complex that it has too much variance.
  - Fitting an overly complex model - a model that has too much variance - is known as overfitting.
    - ⋆ Avoid overfitting! Recall the figures about the true model and a overfitted model in last lecture.

Model Complexity

- A Really Fundamental Picture



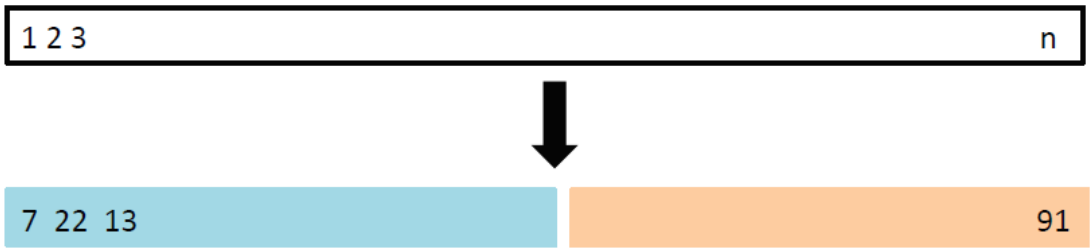
Model Complexity

- We must rely not on training error, but on test error, as a measure of model performance.

- We here consider a class of methods that estimate the test error by holding out a subset of the training observations from the fitting process, and then applying the statistical learning method to those held out observations.
- Three Ways to Estimate Test Error:
  - The validation set approach.
  - Leave-one-out cross-validation.
  - K-fold cross-validation.

Cross-validation - Validation Set Approach

- Split the n observations into two sets of approximately equal size. Train on one set, and evaluate performance on the other.



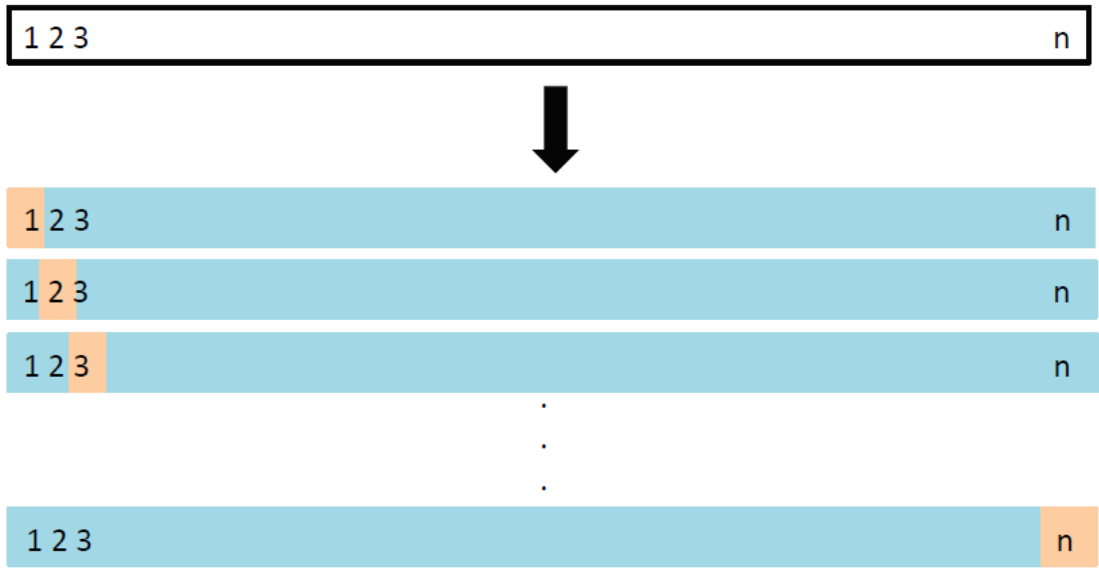
- The resulting validation-set error provides an estimate of the test error. This is typically assessed using MSE in the case of a quantitative response.

Cross-validation - Validation Set Approach

- 1. Split the observations into two sets of approximately equal size, a **training set** and a **validation set**.
  - a. Fit the model using the training observations. Let $\hat{\beta}_{(train)}$ denote the regression coefficient estimates.
  - b. For each observation in the validation set, compute $e_i = (y_i - \boldsymbol{x}_i{}' \hat{\beta}_{(train)})^2$
- 2. Calculate the total validation set error by summing the e_i's over all of the validation set observations.

Cross-validation - Leave-One-Out Cross-Validation

- LOOCV: Fit n models, each on $n-1$ of the observations. Evaluate each model on the left-out observation.



Cross-validation - Leave-One-Out Cross-Validation

- 1. For $i = 1, \ldots, n$:
  - a. Fit the model using observations $1, \ldots, i-1, i+1, \ldots, n$. Let $\hat{\beta}_{(i)}$ denote the regression coefficient estimates.
  - b. For each observation in the validation set, compute $e_i = (y_i - \boldsymbol{x}_i{}' \hat{\beta}_{(i)})^2$.
- 2. Calculate $\sum_{i=1}^{n} e_i^2$, the total CV (Cross-validation) error.
- Fortunately, the CV error can be calculated without requiring n separate regression runs.

$$\sum_{i=1}^{n} e_i^2 = PRESS_p = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}} \right)^2,$$

where $\hat{y}_i$ is the ith fitted value from the original least squares fit, and h_{ii} is the leverage which is the ith diagonal element of the Hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$.

Cross-validation - Leave-One-Out Cross-Validation

In the above,

$$PRESS_p = \sum_{i=1}^{n} (Y_i - \hat{Y}_{(i)})^2,$$

where $\hat{Y}_{(i)}$ is predicted value of Y_i based on the model fitted using $n-1$ data points only, omitting the ith observation.

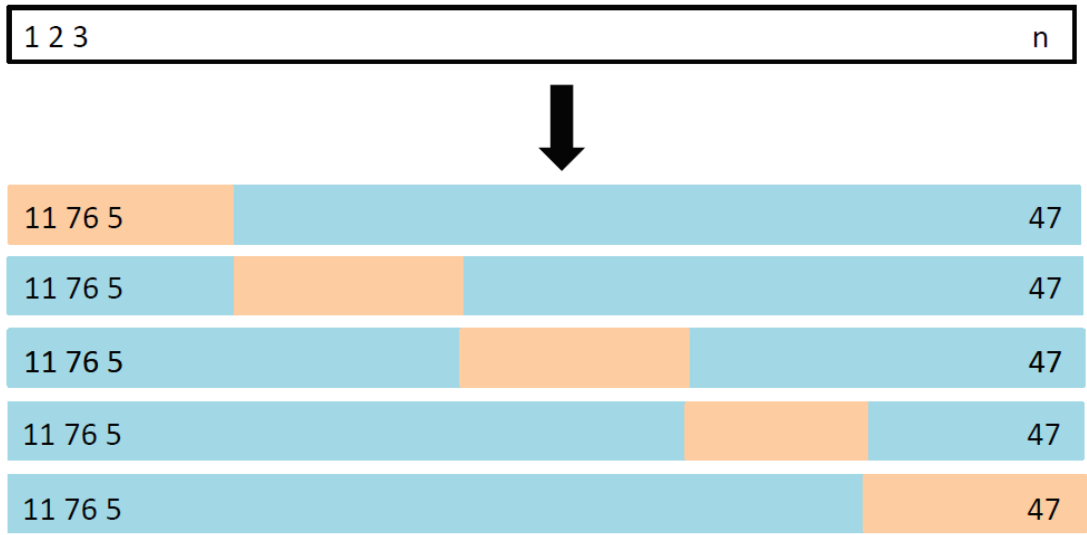
A $PRESS_p$ value reasonably close to SSE_p supports the validity of the fitted regression model and of MSE_p as an indicator of the predictive capability of the model.

Cross-validation - K-fold cross-validation

- Widely used approach for estimating test error.
- Estimates can be used to select best model, and to give an idea of the test error of the final chosen model.
- Idea is to randomly divide the data into K equal-sized parts. We leave out part k, fit the model to the other $K-1$ parts (combined), and then obtain predictions for the left-out kth part.
- This is done in turn for each part $k = 1, 2, \ldots, K$, and then the results are combined.

Cross-validation - K-fold cross-validation

- 5-Fold Cross-Validation: Split the observations into 5 sets. Repeatedly train the model on 4 sets and evaluate its performance on the 5th.



Cross-validation - K-fold cross-validation

A generalization of K-fold cross-validation:

- 1. Split the n observations into K equally-sized folds.
- 2. For $k = 1, \ldots, K$:
  - a. Fit the model using the observations not in the kth fold.
  - b. Let e_k denote the test error for the observations in the kth fold.
- 3. Calculate $\sum_{k=1}^{K} e_k$, the total CV error.
- Note. Setting $K = n$ yields n-fold or leave-one out cross-validation (LOOCV).
- Since each training set is only $(K-1)/K$ as big as the original training set, the estimates of prediction error will typically be biased upward. And LOOCV estimate has high variance, as noted earlier.
  - $K = 5$ or $K = 10$ provides a good compromise for this bias-variance tradeoff.

Cross-validation - K-fold cross-validation

- In SAS, the `GLMSELECT` Procedure can be used for model selection
- See The GLMSELECT Procedure
  https://support.sas.com/documentation/cdl/en/statug/68162/HTML/default/viewer.htm#statug_glmselect_details27.htm

Example 1: Validation Set Approach

- For big data, Validation Set Approach will be sufficient. We use a small data set to illustrate the method.

```
/* Read CSV file into SAS dataset */
PROC IMPORT
DATAFILE='/home/u5235839/my_shared_file_links/u5235839/Auto.csv'
    DBMS=CSV
    OUT=Auto;
    GETNAMES=YES;
RUN;

/* Add an ID variable */
data Auto;
    set Auto;
    id + 1;
run;
```

Example 1: Validation Set Approach

```
/* Randomly select a training set using PROC SURVEYSELECT */
proc surveyselect data=Auto method=srs n=196
seed=1953 out=Auto_slection outall;
run;

data train test;
set Auto_slection;
if selected =1 then output train;
else output test;
run;

proc print data=train;
run;

proc print data=test;
run;
```

Example 1: Validation Set Approach

```
/* Fit a linear regression model using PROC REG on the train data*/
/* Save the parameter estimates in a new dataset */
proc reg data=train outest=RegOut;
  model mpg=horsepower;
run;

proc print data=RegOut;
   run;
```

Example 1: Validation Set Approach

```
/* Apply the model to the test data using PROC SCORE */
proc score data=test score=RegOut out=test_fitted type=parms;
   var horsepower;
run;

proc print data=test_fitted;
   run;
```

Example 1: Validation Set Approach

```
/* Calculate square errors for the test set using PROC SQL */
proc sql;
    create table square_error as
    select mpg, horsepower, (MODEL1 - mpg) ** 2 as square_error
    from test_fitted;
quit;

proc print data=square_error;
run;

/* Calculate the mean square error for the test set using PROC MEANS
proc means data=square_error mean;
    var square_error;
run;
```

Example 2: k-fold CV

- Let's regard `mpg` as the response, and use `PROC GLMSELECT` to select a best multiple linear regression model

```
/*List variables in the data*/
proc contents data=Auto;
run;

/*check levels of the two categorical variables*/
proc freq data=Auto;
  tables cylinders origin;
run;
```

Example 2: k-fold CV

```
/*convert the two variables to categorical*/
data Auto1;
  set Auto;
  /* Create binary indicators */
  origin_1 = (origin = 1);
  origin_2 = (origin = 2);
  /*drop origin;*/
run;

data Auto2;
  set Auto1;
  /* Create binary indicators */
  cylinders_3 = (cylinders = 3);
  cylinders_4 = (cylinders = 4);
  cylinders_5 = (cylinders = 5);
  cylinders_6 = (cylinders = 6);
  /*drop cylinders;*/
run;
```

Example 2: k-fold CV

```
proc glmselect data=Auto2;
   model mpg=acceleration cylinders_3 cylinders_4 cylinders_5
cylinders_6 displacement horsepower origin_1 origin_2
weight/selection=forward(stop=CV) cvMethod=RANDOM;
/*default 5-fold with CVMETHOD=BLOCK, CVMETHOD=SPLIT, or CVMETHOD=RANDOM.*/
run;
```

License