Statistical Learning

Mohd Aftar bin Abu Bakar

[1]Pusat Pengajian Sains Matematik
Fakulti Sains dan Teknologi
UKM

[2]DELTA
UKM

2017

Section 1

Statistical Learning

Advertising Data

The advertising data consists of product sales from 200 markets and their associated tv, radio, and newspaper advertising budgets.

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
5	8.7	48.9	75.0	7.2
6	57.5	32.8	23.5	11.8
7	120.2	19.6	11.6	13.2
8	8.6	2.1	1.0	4.8
9	199.8	2.6	21.2	10.6

What kind of relationship can be seen between advertising budget and sales?

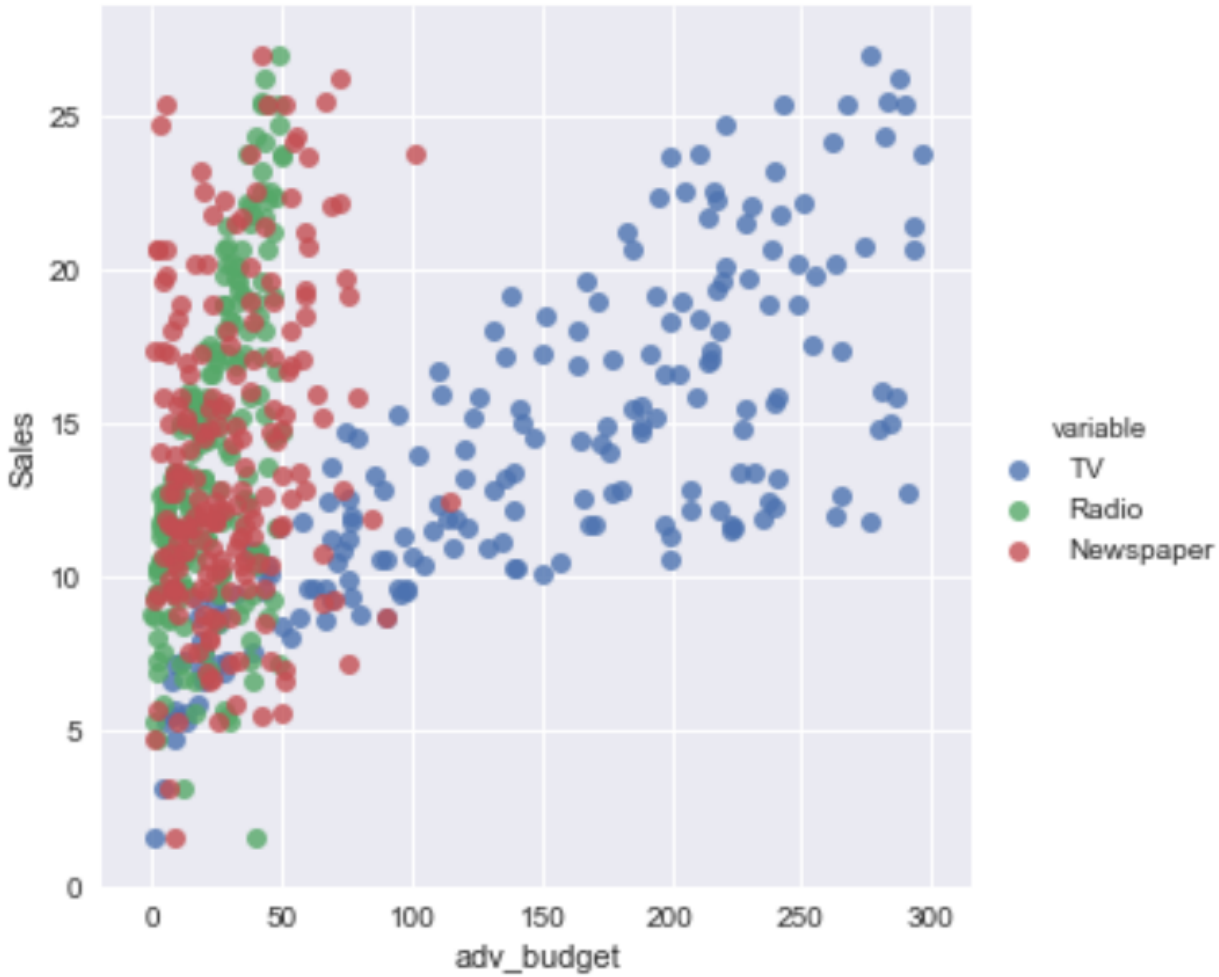
- **input variables / predictors / independent vars / features, X** - advertising budget
- **output variable / response / dependent vars / target, Y** - sales

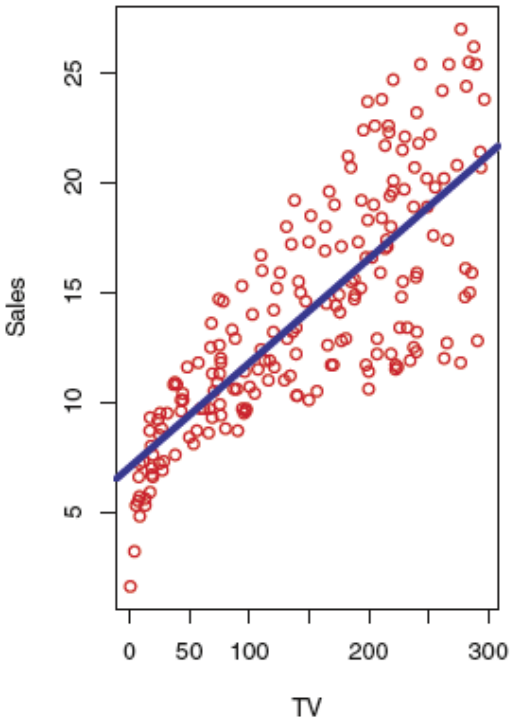
General form of linear model

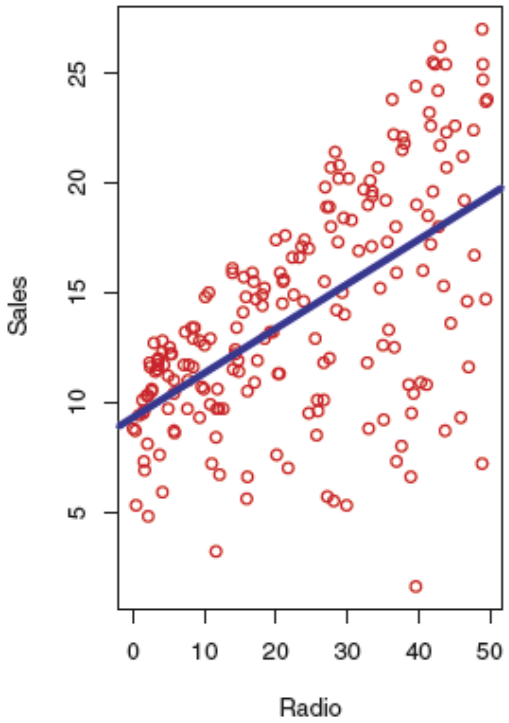
$$\mathbf{Y} = f(\mathbf{X}) + \epsilon$$

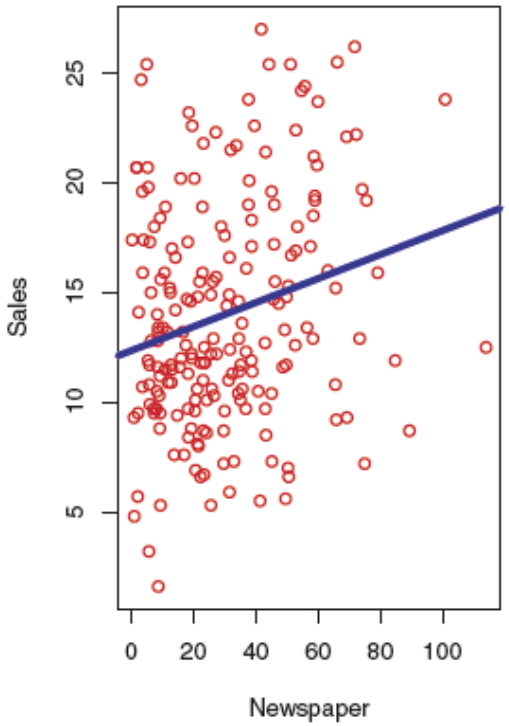
where ϵ is some random errors.

- f is some fixed but unknown systematic relation between $\mathbf{X}$ and $\mathbf{Y}$.
- Statistical learning is trying to approximate this function f.

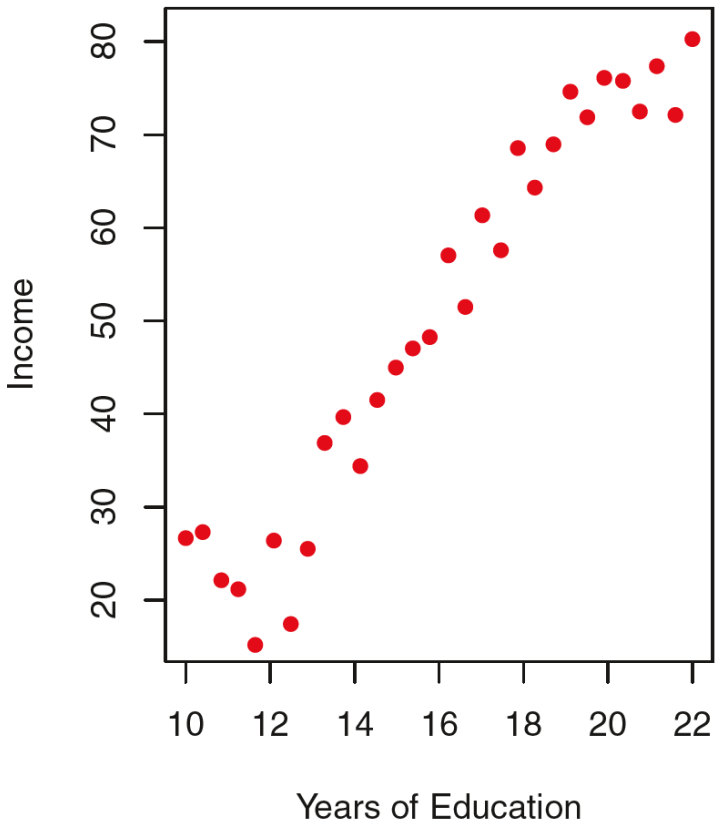

Sales
adv_budget
0
5
10
15
20
25
0
50
100
150
200
250
300
variable
TV
Radio
Newspaper

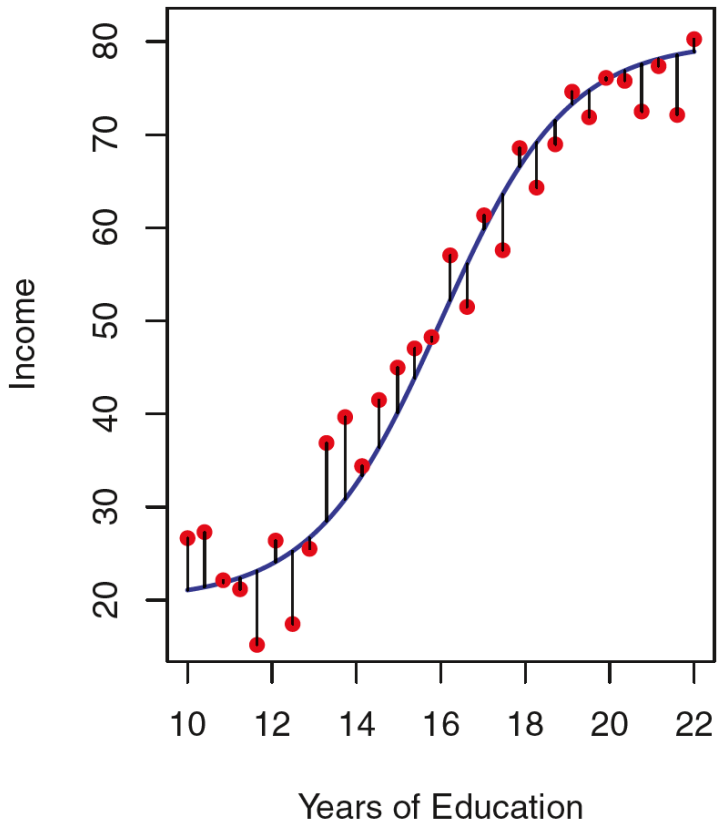

Sales
5
10
15
20
25
0
50
100
200
300
TV


Sales
5
10
15
20
25
0
10
20
30
40
50
Radio

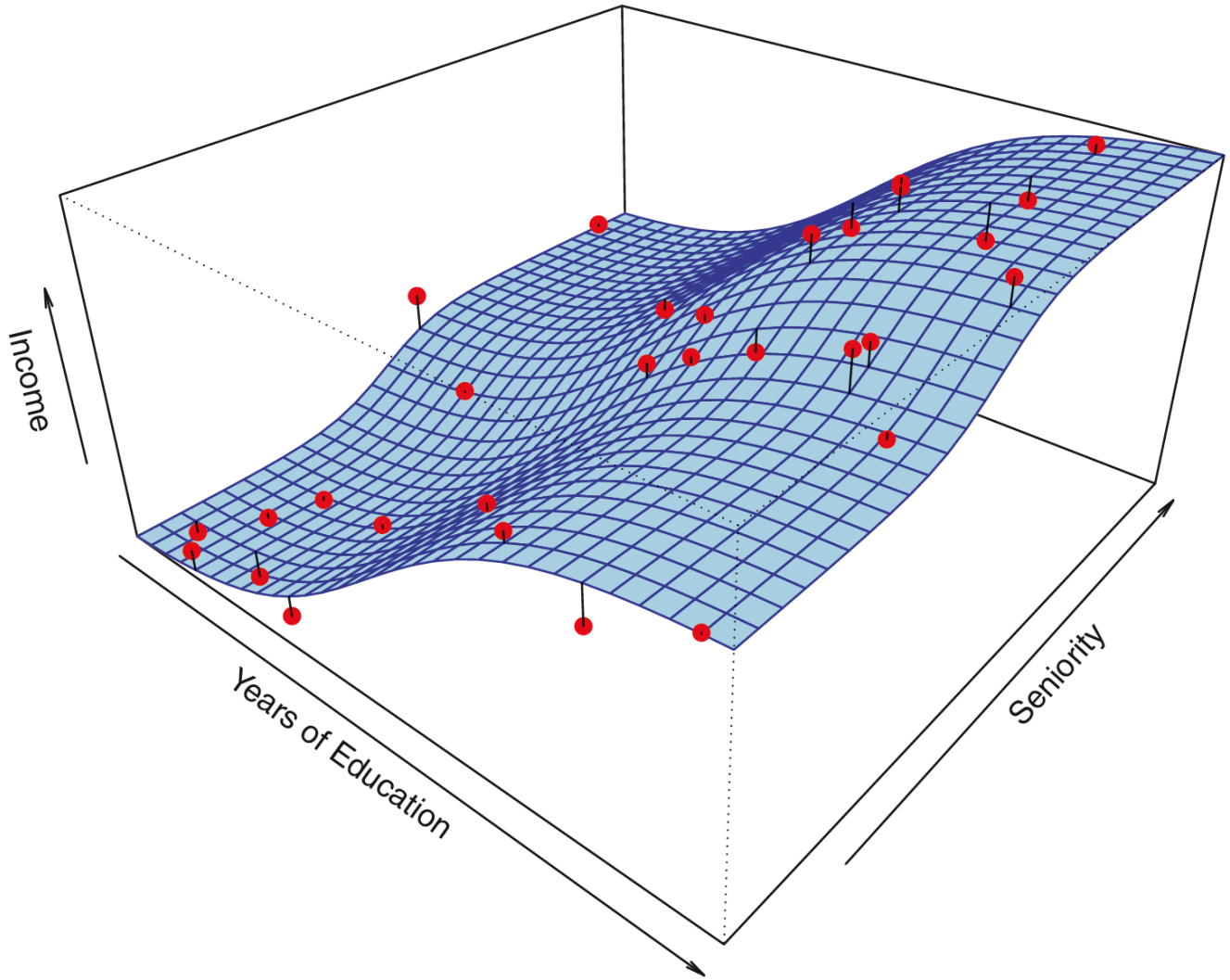

Sales
5
10
15
20
25
0
20
40
60
80
100
Newspaper

Income Data


Income
20
30
40
50
60
70
80
10
12
14
16
18
20
22
Years of Education


Income
20
30
40
50
60
70
80
10
12
14
16
18
20
22
Years of Education

Income Data



Prediction:

- Try to find a function $\hat{f}$ that closely matches f.
- Not that concerned about the shape or decipherability of f.
- Just want a good prediction for each input.

Inference:

- Want to understand the relationship between $\mathbf{X}$ and $\mathbf{Y}$.
- How does changing one parameter change the output.
- Finding only the important predictors.
- What type of model to use (linear vs non-linear)?

Reducible error

- This is the difference between f and $\hat{f}$.
- In practice f is never known so we will never be able to accurately measure this but this error can still be reduced by finding the best ML method to do the learning.

Irreducible error

- Even if you were to perfectly estimate f you would still have error in your prediction as f has inherent randomness in it.

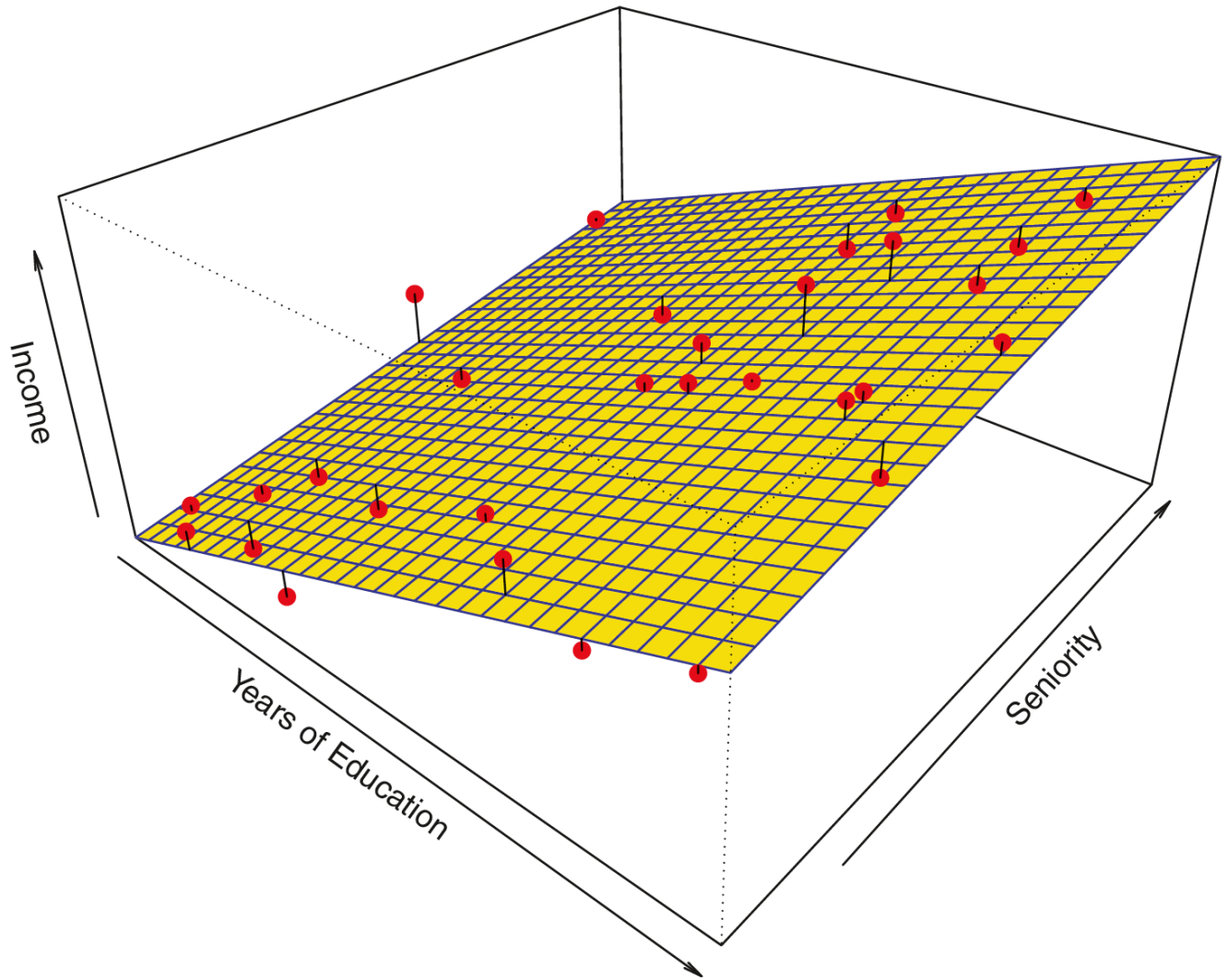
$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon \hat{f}(X)]^2 \\ &= [f(X) - \hat{f}(X)]^2 + \mathrm{Var}(\epsilon) \end{aligned}$$

The focus is on minimizing the reducible error - finding that model $\hat{f}$ that closely matches f.

Our goal is to apply a statistical learning method to the training data in order to estimate the unknown function f.

Parametric methods

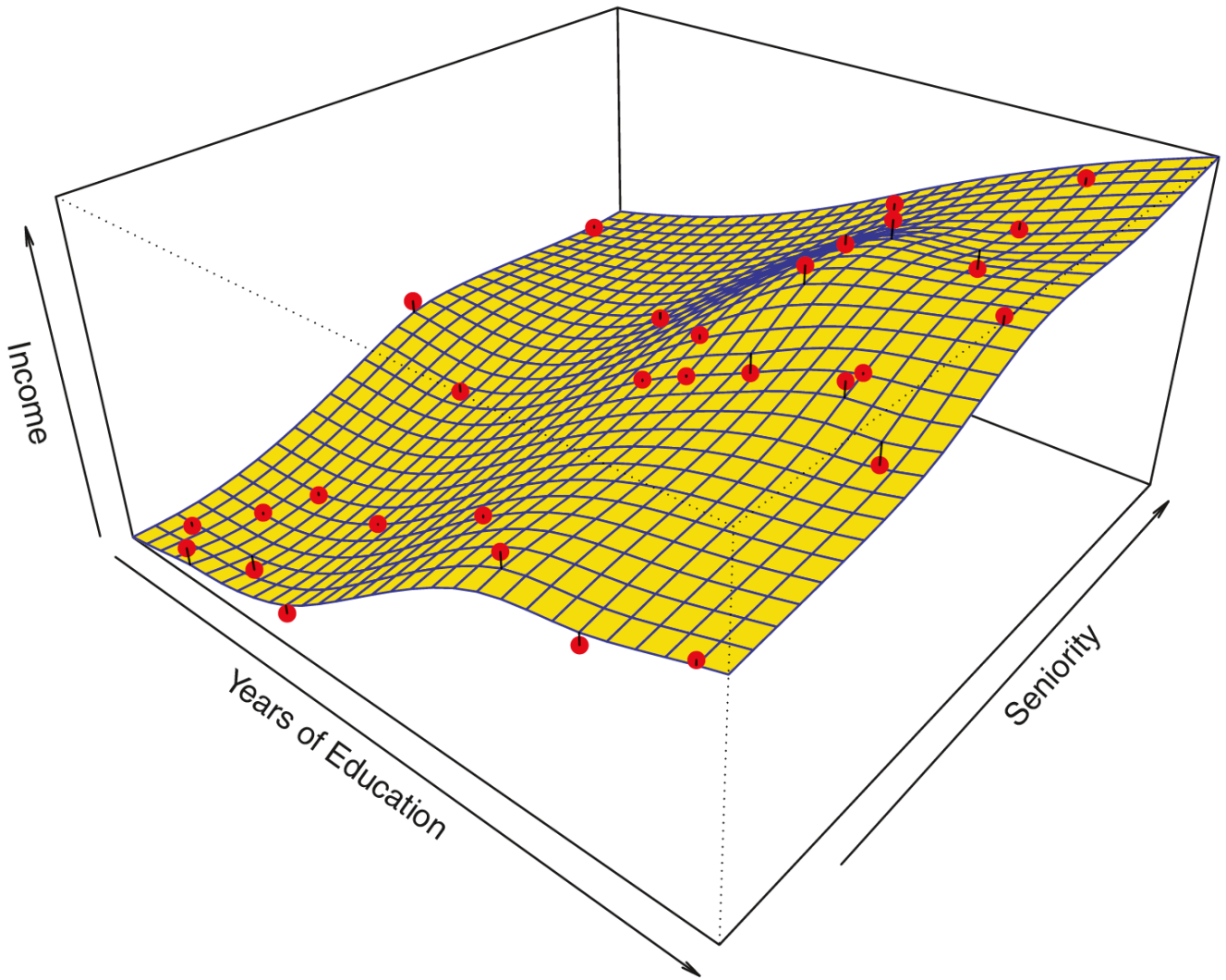
- An assumption about the functional form is made $Y = \beta_0 + \beta_1 X_1 + \ldots$.
- Since mode form is linear, fitting is easier and usually fast.
- Need to estimate parameters. Many ways to do this. Most popular is least squares.
- Unlikely that the true form of f is linear.
- **Flexible models** that can fit many different possible functional forms for f may offer a solution to this problem, where it involve greater number of parameters.
- Potential to overfit - memorizing data by following noise


Income
Years of Education
Seniority

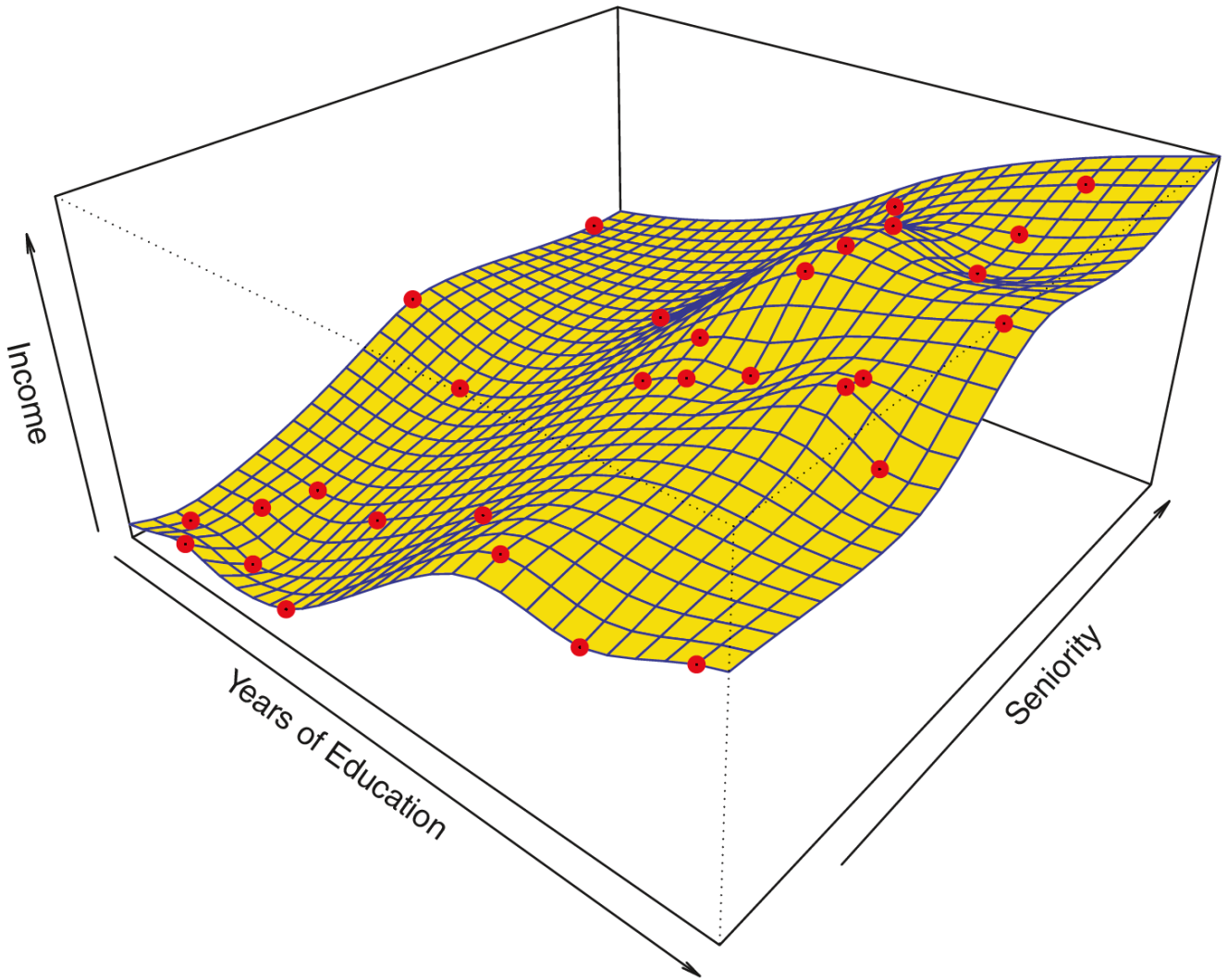
Nonparametric methods

- No functional form of f is given. Meaning they can wiggle all over the place.
- A very large number of parameters is needed.
- Fitting is more computationally intensive.
- Potential to overfit

Smooth thin-plate spline fit



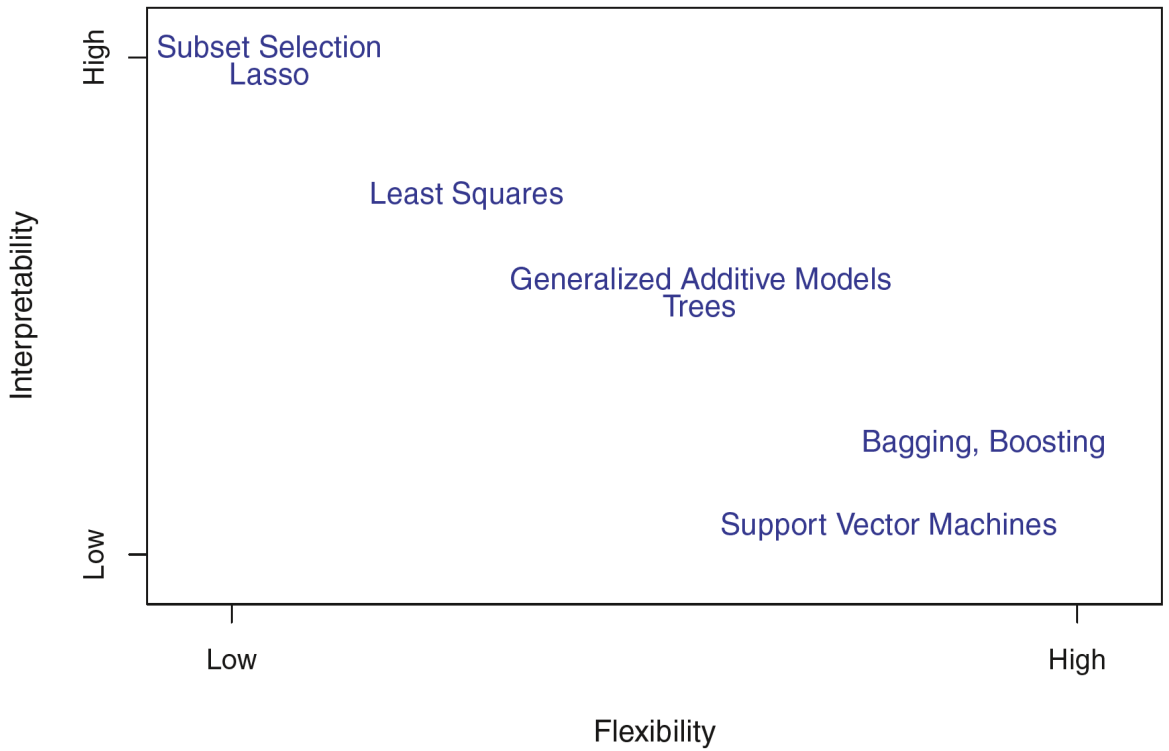
Rough thin-plate spline fit



OVERFIT!!!

Trade-off between interpretability and flexibility.

- The more flexible a model is, the less-likely you are to be able to interpret it.
- It becomes increasing more difficult to determine how the parameters are affecting the model the more flexible it is.
- Lasso regression is very inflexible as it forces some coefficients to 0.



Regression vs Classification Problems

Variables are classified as quantitative or qualitative.

- **Quantitative** - Take on numerical values
- **Qualitative** - Take on categorical value

- Regression problems are those with quantitative responses.
- Classification problems are those with qualitative responses.

Regression vs Classification Problems

- Some confusion can arise. For instance, logistic regression is used for classification problems yet obviously has the term 'regression' in it.
- This is because logistic regression outputs probabilities that each observation is in a certain class.
- Probabilities are quantitative values between 0 and 1, and thus you have a 'regression'.

- Some algorithms such as K nearest neighbor, random forests, support vector machines can be used for both classification or regression purposes

Section 2

Assessing model accuracy

There is no free lunch in statistics

- No one model works best.
- No one method dominates all others over all possible data sets.
- Selecting the best approach are the most challenging parts of performing machine learning.

Measuring quality of fit

- There are different ways to measure how well a supervised learning problem fit the data.
- For a classification problem we can simply find the percentage of observations that had the correct class predicted.
- For regression problems we could simply take the absolute value of the difference between the prediction, $\hat{f}(x)$, and the actual output, y and take the average of this.

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

Mean Squared Error(MSE)

Although the mean absolute error is a good intuitive metric, the standard is mean squared error(MSE).

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

Typical software regression implementations find the value of the parameters that minimize the MSE.

Training Data

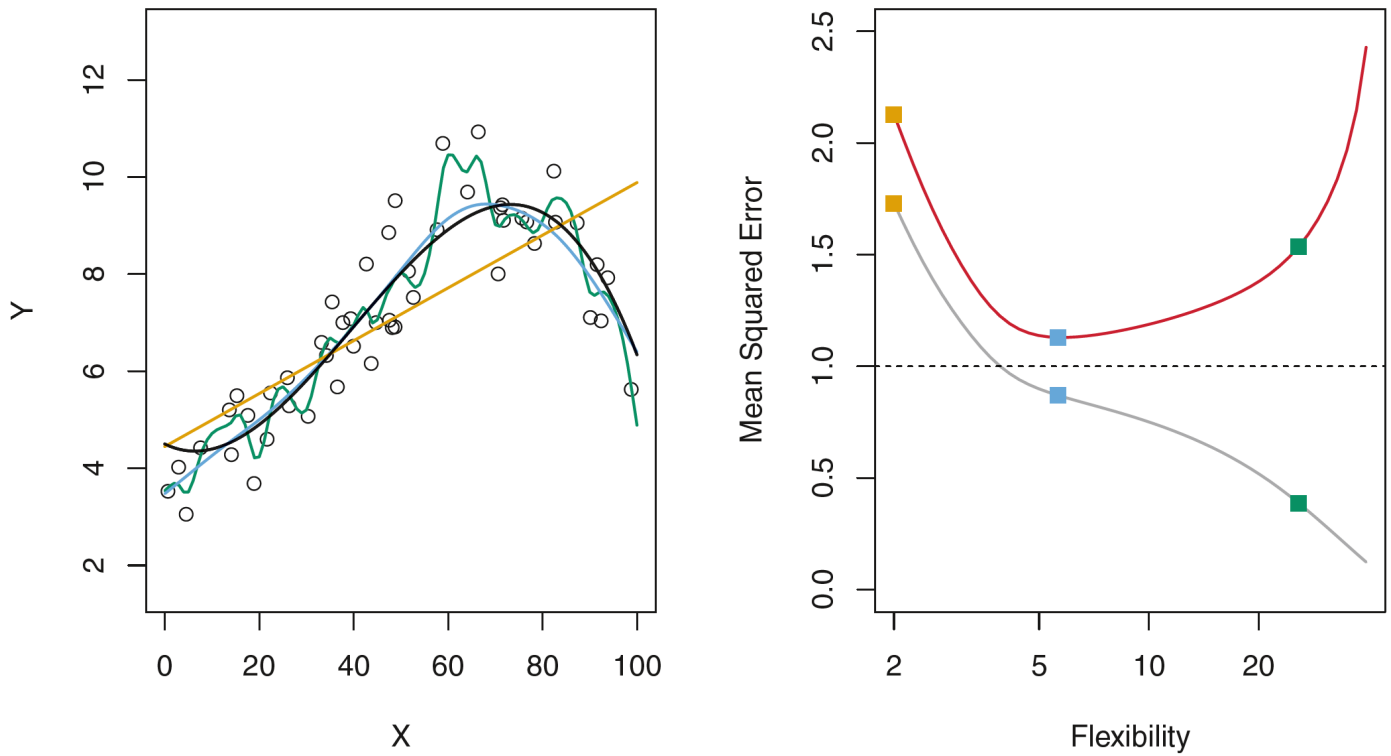
- Data used to build a prediction model.
- Should not be used to validate the model.

Testing Data

- Data used to determine the usefulness of the model.
- Validates the model.
- This data is unseen during model building phase.

- We are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.
- We want lowest test MSE. Don't care too much about training MSE.
- During the training phase it's usually possible to fit a model that has no error. Such a model would be highly flexible and not fit well on new data is it simply memorized the noise/error.
- There is no guarantee that a model with low training MSE will have the same test MSE.

Example



Left: True f (black), linear regression line (orange), two smoothing spline fits (blue and green).
Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line).

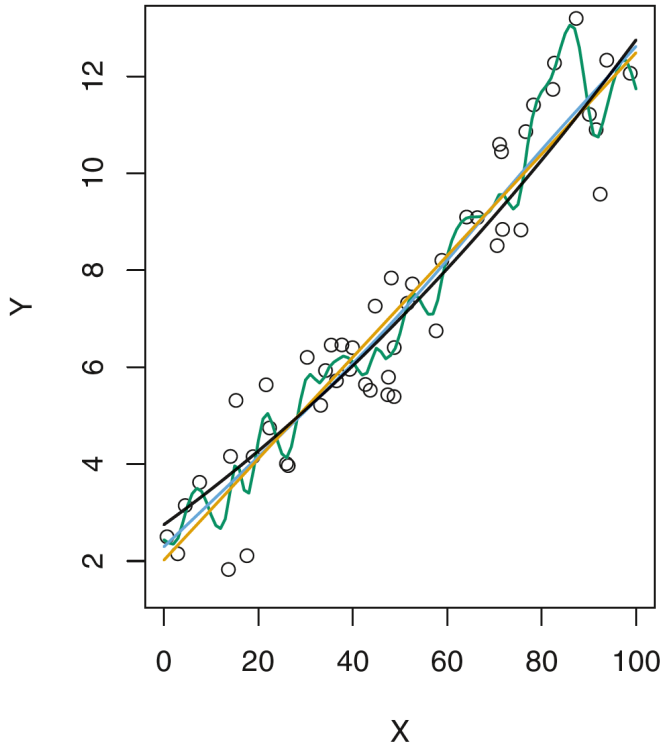
Training MSE will always decrease with more flexibility.

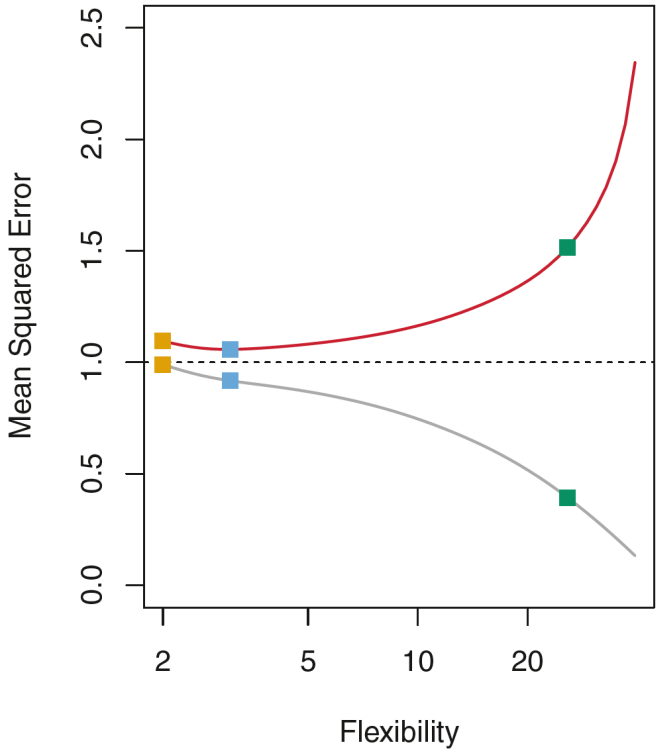
- As a model has more access to features - i.e. is more flexible, it will be able to fit the data better, so no matter what, the training MSE will always decrease as more feature are added to the model.

Overfitting

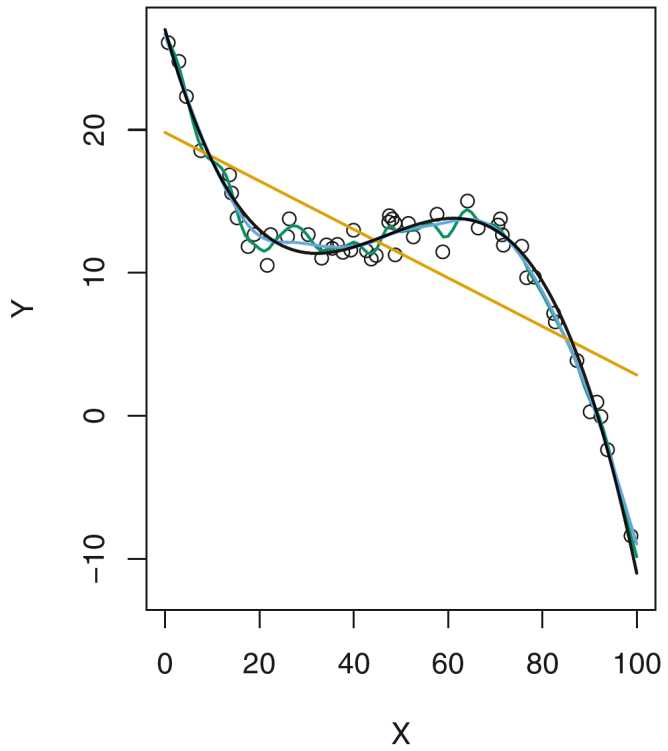
- When the test MSE of a more flexible model surpasses the test MSE of a lesser flexible model, overfitting is occurring.
- Random patterns are beginning to be picked up by the model.

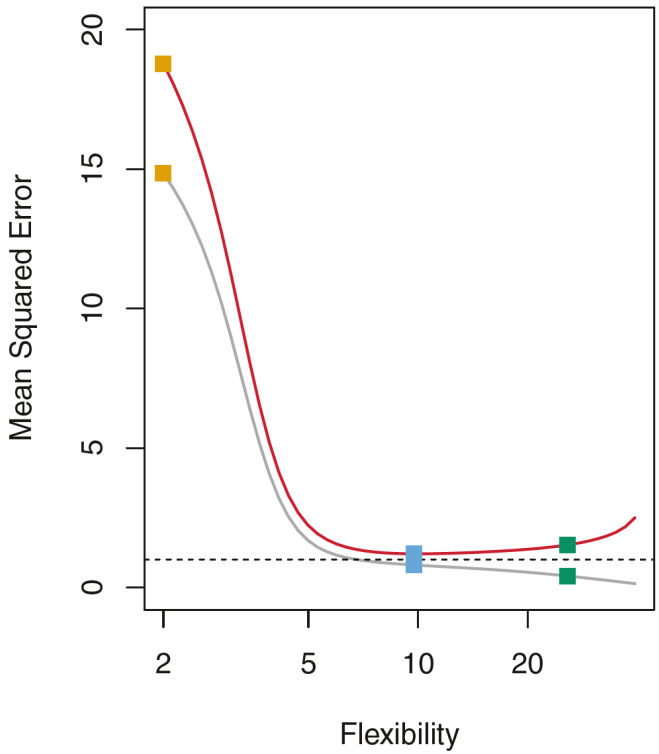
Example


Y
X
2
4
6
8
10
12
0
20
40
60
80
100


Mean Squared Error
Flexibility
0.0
0.5
1.0
1.5
2.0
2.5
2
5
10
20

Example


Y
X
−10
0
10
20
0
20
40
60
80
100

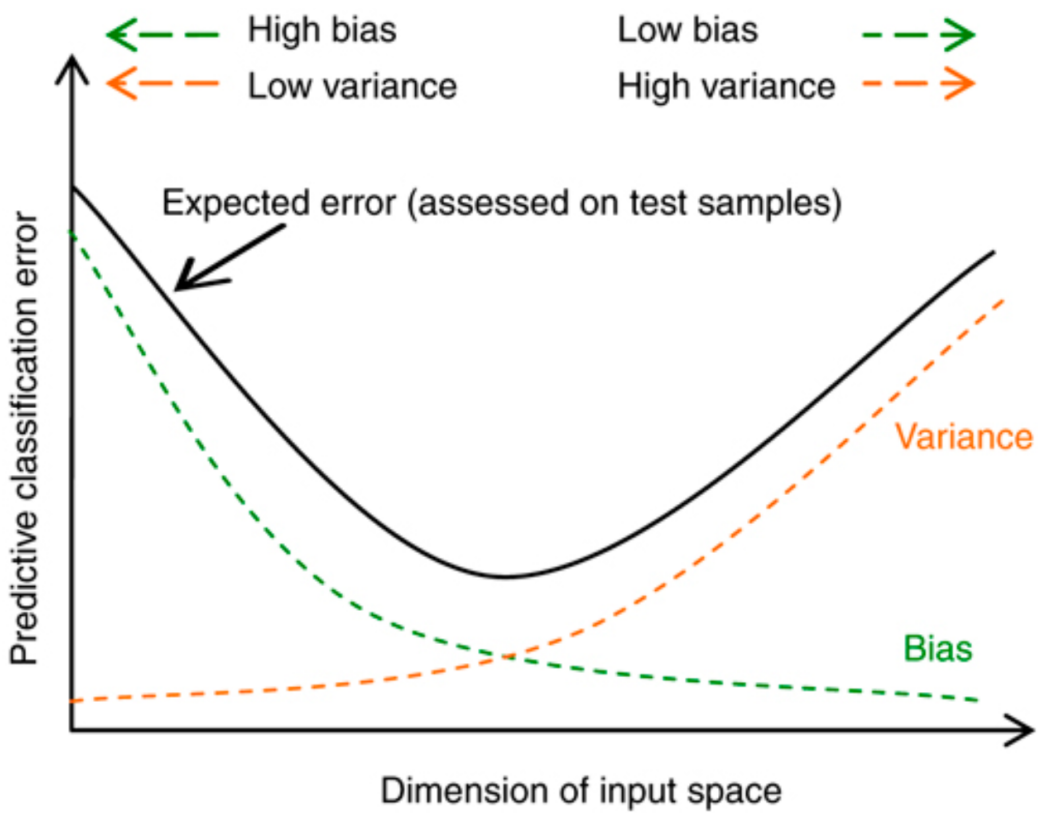

Mean Squared Error
Flexibility
0
5
10
15
20
2
5
10
20

Subsection 1

Bias Variance Tradeoff

- The word **bias** typically means your inherent prejudice against someone or some thing.
- In statistical learning, the term **bias** is used we mean the measure of how much a simpler model deviates from the actual theoretical 'truth'.
- So **low bias** would simply mean that the chosen model does a good job approximating the current model complexities and **high bias** would mean a model unable to explain model complexities.
- **Variance**, in statistical learning, represents the degree to which your model, $\hat{f}$, would change when given new training data.
- Think about coming up with 100 different models for 100 different training sets and plotting them all on one plot. The more scatter in the plot, the higher the variance.

- **Aim:** low bias and variance
- **low bias** $\rightarrow$ the model approximates the true relationship, f well
- **low variance** $\rightarrow$ we don't want our model changing much depending on which data it gets trained on.
- In reality, it's very difficult to achieve both.
  - A low bias model can be generated by fitting a highly flexible model with many dof and the error will be minimized. But the model will likely highly overfit to the randomness of the data and look very different for a new training data.
  - To achieve a model with low variance is quite easy. A model with a very low number of dof, like a simple linear regression, should not change much from training sample to training sample. This stability from model to model would represent low variance.



As model complexity grows, bias decreases (sometimes to 0) and variance increases (sometimes to infinity!). The goal is to find a balance.

Subsection 2

Classification Error

- So far, we have discussed on regression problems, those with numbers as outputs.
- Those with categories as output are known as classification problems.
- A very simple and intuitive way to assess your model for classification is find the percentage of observations that you classified correctly.
- This is just like seeing how many answers you got right in a test.
- The training error rate is given as

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

where the indicator variable,

$$I(y_i \neq \hat{y}_i) = \begin{cases} 1, & \text{if } y_i \neq \hat{y}_i \\ 0, & \text{if } y_i = \hat{y}_i \end{cases}$$

- However, the only real way to determine how well your model is to measure the test error rate.

Bayes Classifier

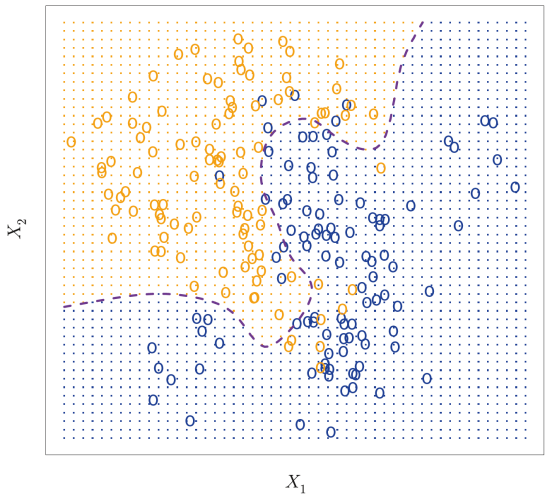
- If we know the distribution of how are observations were created - f has a known probability distribution - we can simply use basic conditional probability to determine what class is the most likely.
- For instance, if we had two fair dice, we would always guess 7 as the total for the sum of the dice since it is most likely.
- But in real life we nearly never have a situation where the probability distributions of the observations are known beforehand.

- The test error rate could be minimized by assigning each observation to the most likely class, given its predictor values.
- We should simply assign a test observation with predictor vector x_0 to the class j for which

$$Pr(Y = j|X = x_0)$$

  is largest.
- This simple classifier is called the Bayes classifier.

Example



- The orange and blue circles correspond to training observations that belong to two different classes.
- Let say we can calculate the conditional probabilities for each value of X_1 and X_2.
- The orange shaded region reflects the set of points for which $Pr(Y = \text{orange}|X) > 0.5$, while the blue shaded region indicates the set of points for which $Pr(Y = \text{orange}|X) < 0.5$.
- The dashed line is the **Bayes decision boundary** where $Pr(Y = \text{orange}|X) = 0.5$.

- The Bayes classifier's prediction is determined by the Bayes decision boundary.
- This produces the lowest possible test error rate, called the **Bayes error rate**.
- The overall Bayes rate is

$$1 - E\left(\max_j Pr(Y = j|X)\right)$$

where the expectation averages the probability over all possible values of X.

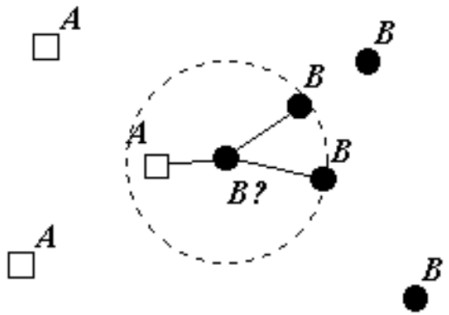
K-Nearest Neighbours

- KNN is an attempt to estimate the local probability distribution of an observation by simply counting up the classes of all of its neighbours and using that empirical tally as its probability distribution for that particular observation.
- The **K** is simply the number of neighbours each point will observe before reaching a conclusion as to what its local probability distribution will be.
- KNN is one of the simplest algorithms and requires no pre-training.

How KNN works:

1. For each observation, find its closest neighbours based on some distance function (euclidean, cosine, etc...)
2. Each of these neighbours 'casts' a vote for which class its in.
3. Tally up the votes and this is your local probability distribution.
4. Choose the highest vote getting class as a prediction for the current observation.

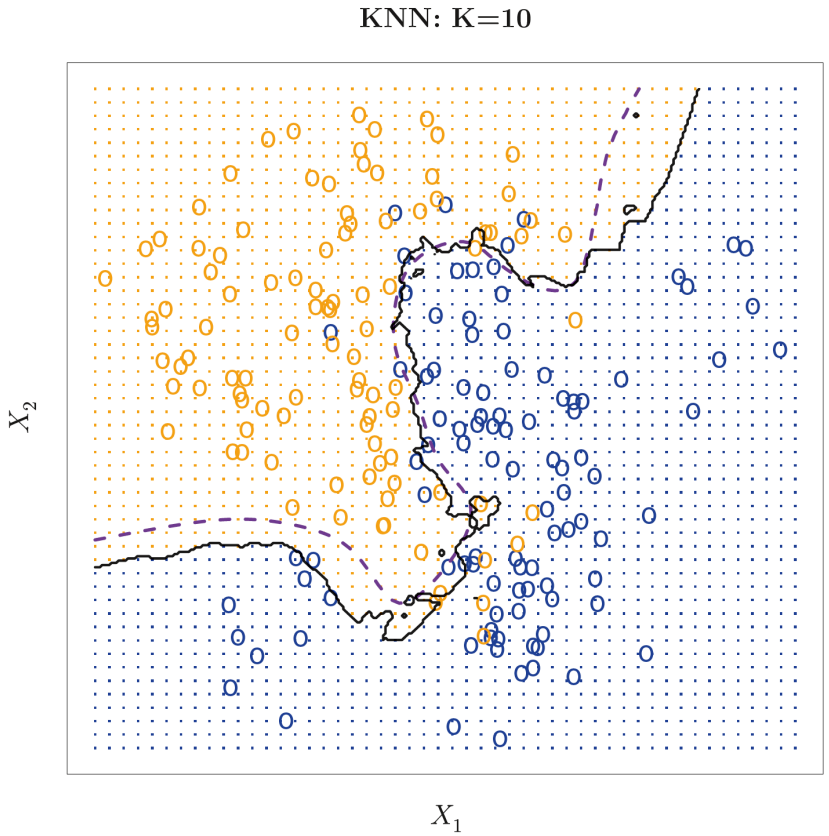
Example



- The example above shows of the $K = 3$ closest neighbors, 2 are of class B and 1 is of class A.
- Therefore the local probability distribution for the questionable point would be B: 2/3 and A: 1/3.
- KNN would predict the class to be B.

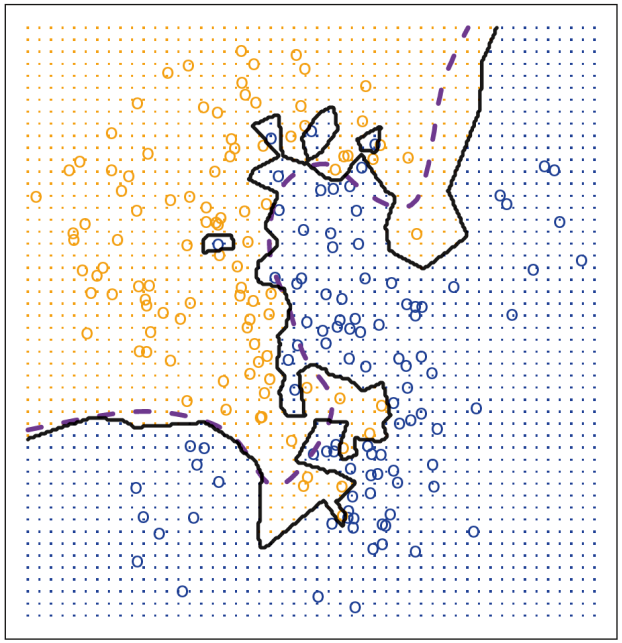
- K is usually chosen by a method called cross-validation which will be discussed later.
- But in general choosing a K to be very large, lets say as large as the number of observations, would simply have the algorithm pick the most common class and all predictions would be the same class.
- Meanwhile, choosing $K = 1$ would introduce lots of randomness (variance) as the prediction is based solely on one data point.

Example

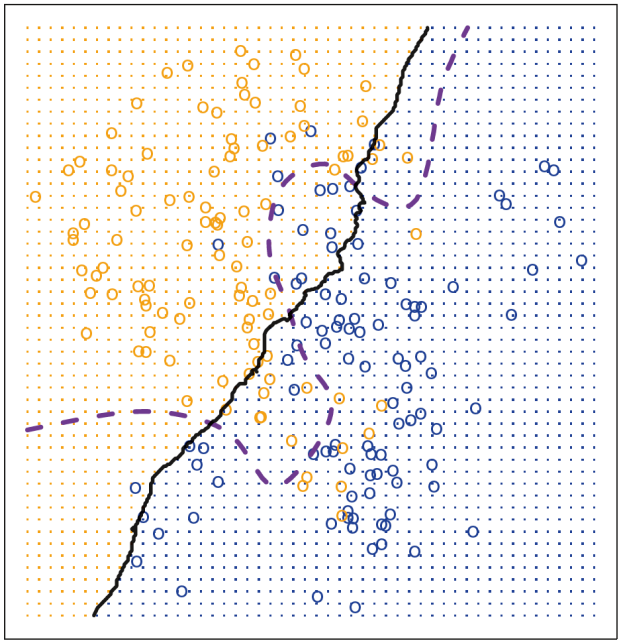


Full line (KNN) and dashed line (Bayes decision boundaries)

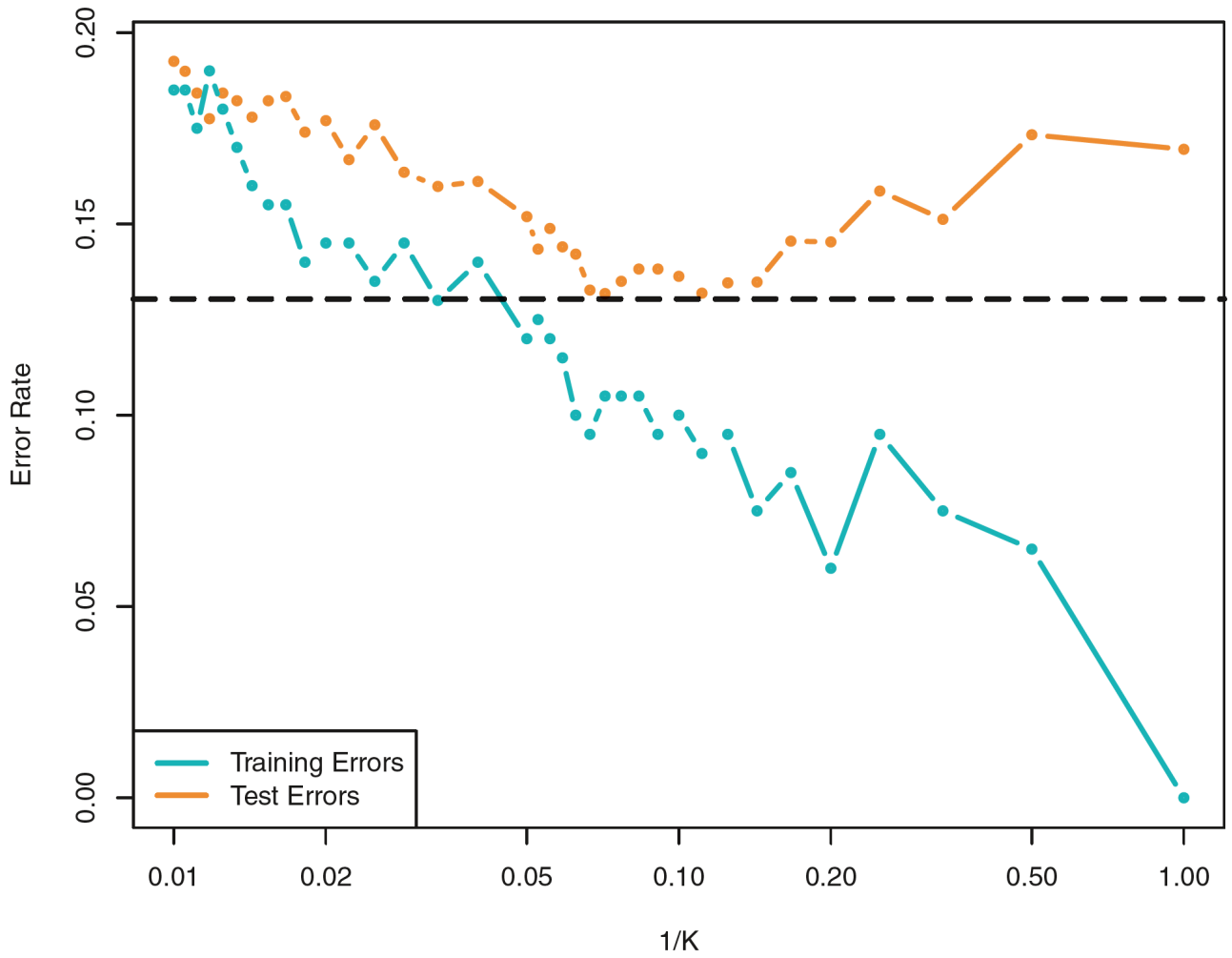
KNN: K=1



KNN: K=100



$K = 1$, over flexible
$K = 100$, not sufficiently flexible


Error Rate
0.00
0.05
0.10
0.15
0.20
Training Errors
Test Errors
0.01
0.02
0.05
0.10
0.20
0.50
1.00
1/K