

Machine Learning Overview

Key questions

- What can you do with machine learning?
(and what can't you do?)
- Where does ML fit in a data-to-decision workflow?

A short history of the world...

- The earth cooled
- (... time passes...)
- Humans arrived* and made decisions
- Language and Math/Logic is created and people write *rules* for making decisions
- Computers invented: able to make *faster* decisions
- Humans write fixed programs to make decisions using (probabilistic) distilled judgements about relationships in the data [Expert systems]
- Humans decided that *the computer would be better at learning statistical relationships* between attributes of the data [Machine Learning]

*Insert scientific and/or religious word of your choosing here

What can you do with Machine Learning?

- Infer how data elements are related
 - Does ice cream consumption depend on outdoor temperature?
- Make predictions about a target variable within a dataset
 - How much ice cream will be consumed this summer in Ohio?
- Determine the category something belongs to
 - Bird vs non-bird



<https://www.pexels.com/search/birds/>



<https://en.wikipedia.org/wiki/Strelitzia>



What can you do with Machine Learning(2) ?

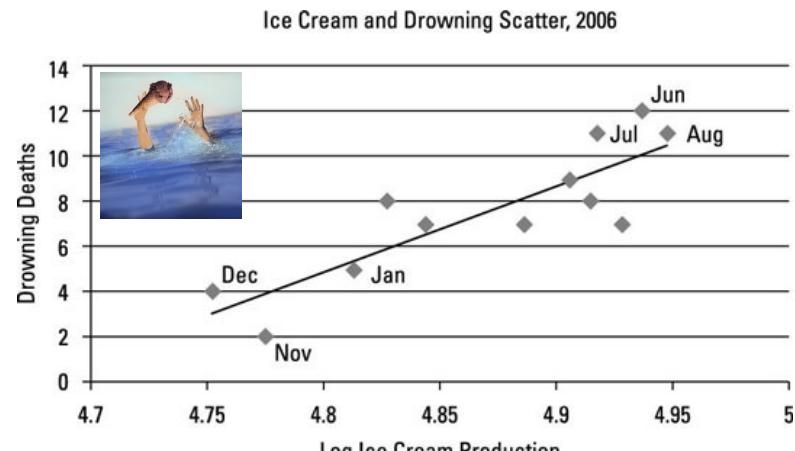
- Optical Character/Number recognition
- Translate Text between languages
- Audio<->Text
- Find Components of an image
- Describe the contents of an image
- Pick the best <restaurant; movie; product; ...> for me



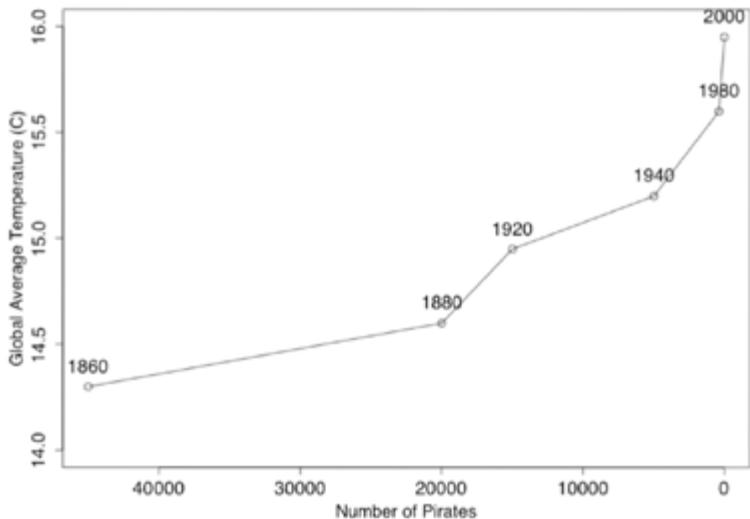
[https://www.technologyreview.com/s/523326/
how-google-cracked-house-number-
identification-in-street-view/](https://www.technologyreview.com/s/523326/how-google-cracked-house-number-identification-in-street-view/)

What can't you do with ML?

- Determine causality
 - Production of ice cream is correlated with drowning.
Which way is the causality?
- Determine whether data is Evidence or Coincidence



<http://www.dummies.com/education/economics/econometrics/the-role-of-causality-in-econometrics/>



With a decrease in the number of pirates, there has been an increase in global warming over the same period. Therefore, global warming is caused by a lack of pirates.

Even more compelling: Somalia has the highest number of Pirates AND the lowest Carbon emissions of any country.
Coincidence?

— Tim Ferriss —

AZ QUOTES

<http://sourcesandmethods.blogspot.com/2011/03/passport-ownership-cures-diabetes.html>

What can't you do with *just* ML (2)

- Find the fastest route to an address on a map
(use a cost-based pathfinding algorithm)
- Learn to Play Chess, Checkers, Go (efficiently)
(use a heuristic deep search algorithm)
- Determine geocoordinates from only a photo
(this only works for some common locations...)
- Drive a vehicle autonomously
(this requires much more than ML)

Motivation for a Data-to-Decision Workflow

- Goal is to make better decisions
- Many ways to make decisions
 - Heuristic-based Human Judgement
 - Human-built computational models (e.g. expert systems)
 - Data Analysis (correlation, trends)
- Statistical Machine Learning suggests learning from Data
 - But where does the data come from?
 - And what resulting decision activities will the data support?

Where does ML fit in a workflow?

- A workflow is a defined sequence of steps used to process something / do something / answer a question
- ML is often in the middle of the workflow.
For example, ML can help:
 - Determine what objects an image contains
 - Decide which category an observation in the data belongs to
 - Predict a value of a variable based on other observed values
- Before ML can occur, data* is gathered, wrangled, cleaned...
- After ML gives an output, decisions are made

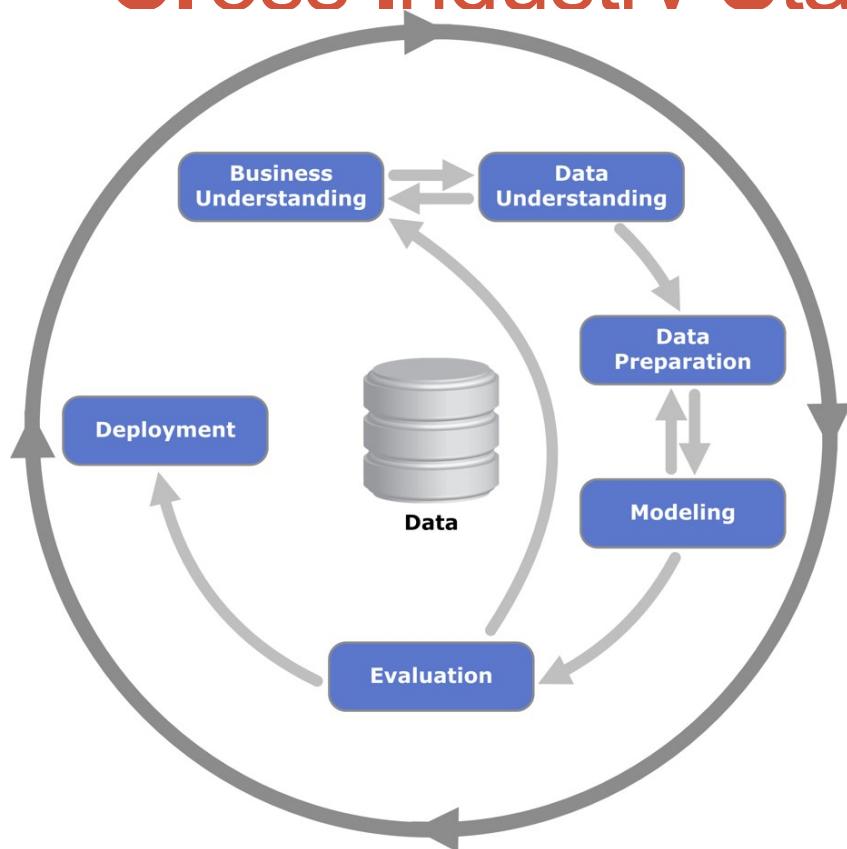
*Data can be numbers, text, audio, images, video....

Workflow Motivation

Since decisions derived from Data are only as good as the data, we must ensure the data collection, management, and analysis process is sound

- CRISP-DM
- KDD
- Audience Participation (x 2)

Cross Industry Standard Process for Data Mining (CRISP-DM)



Process diagram showing the relationship between the different phases of CRISP-DM

https://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining#/media/File:CRISP-DM_Process_Diagram.png

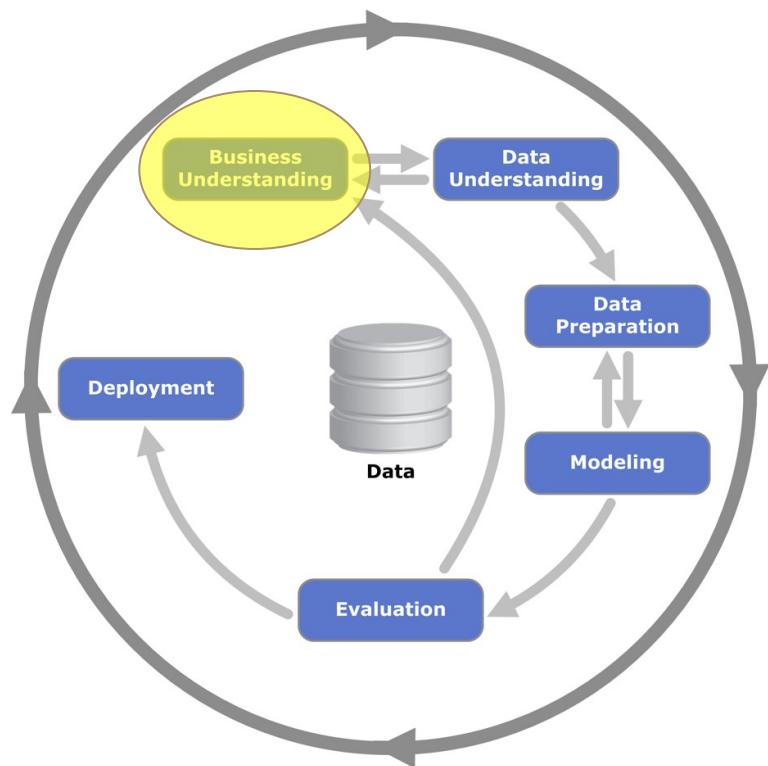
CC-SA Kenneth Jansen

Business understanding	Data understanding	Data Prep	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review Process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

Ó. Marbán et al., “A Data Mining & Knowledge Discovery Process Model,” in Data Mining and Knowledge Discovery in Real Life Applications, no. February, J. Ponce and A. Karahoca, Eds. Vienna, Austria: I-Tech, 2009, pp. 483–453.

CRISP-DM: Business Understanding

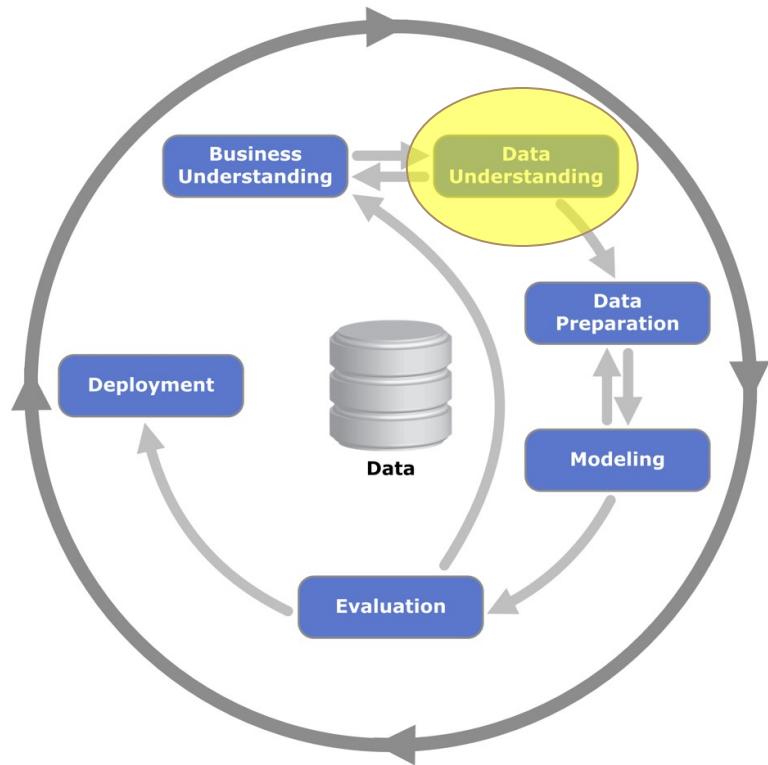
- Understand project objectives and requirements
- Develop a data mining definition and plan



Business understanding	Data understanding	Data Prep	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review Process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

CRISP-DM: Data Understanding

- Collect Data
- Document / Describe data
- Become familiar / Explore data
- Identify data quality problems
- Determine data subsets
- Form investigative hypotheses

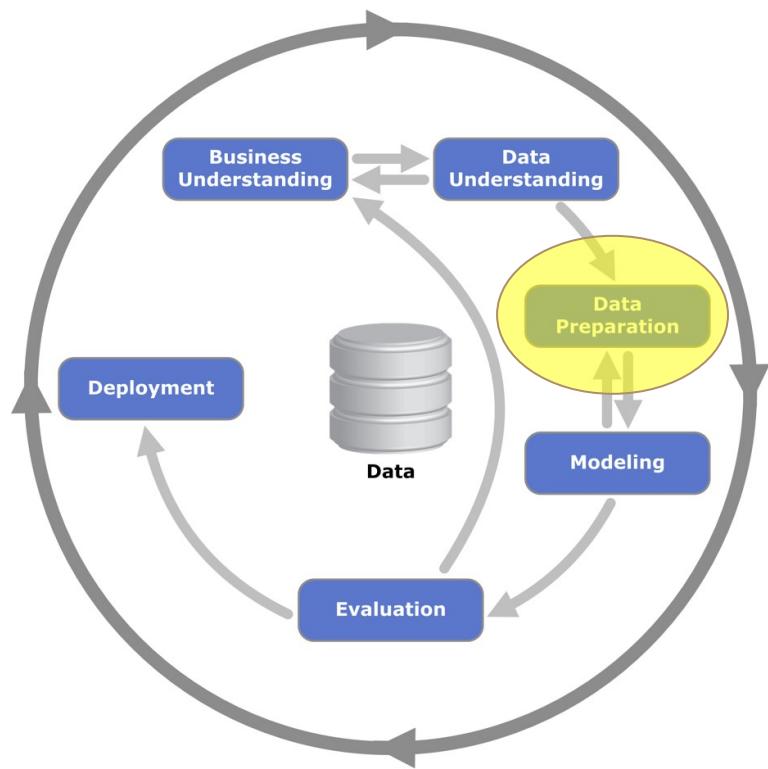


Business understanding	Data understanding	Data Prep	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review Process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

CRISP-DM: Data Preparation

- Construct final dataset from initial raw data

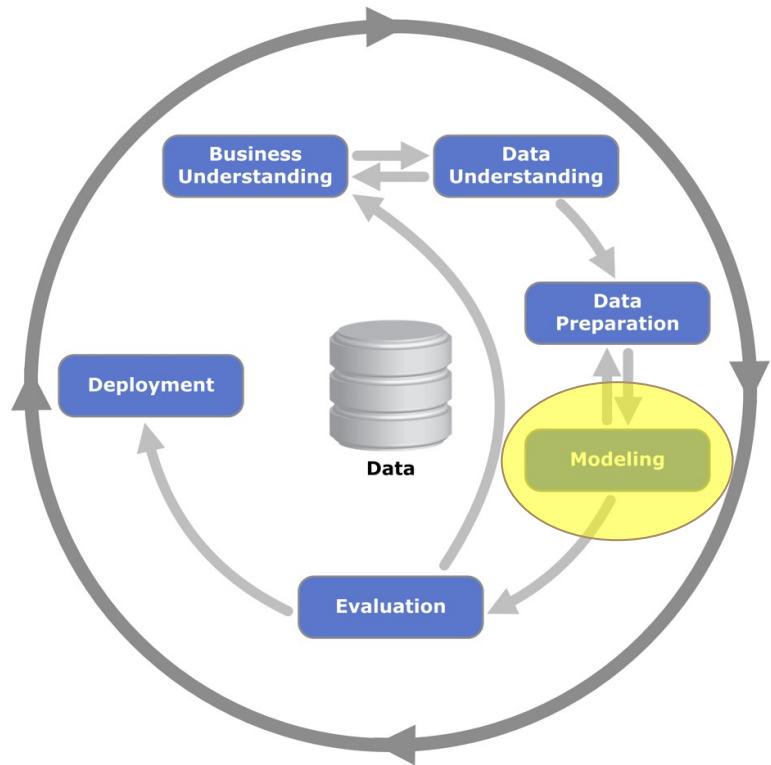
- Select / Subset / Combine / Join data
- Clean data (remove bad data / outliers?)
- Construct missing data (impute)
- Format data



Business understanding	Data understanding	Data Prep	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review Process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

CRISP-DM: Modeling

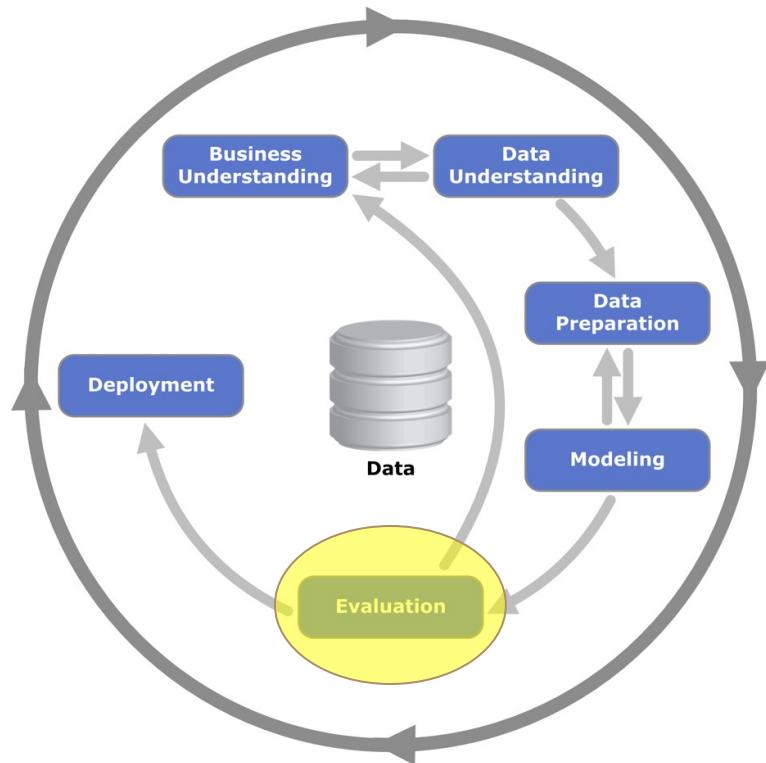
- Select modeling techniques
- Create the test/eval/assess design
- Build model
 - Fit & tune/calibrate model parameters
- Assess models



Business understanding	Data understanding	Data Prep	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review Process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

CRISP-DM: Evaluation

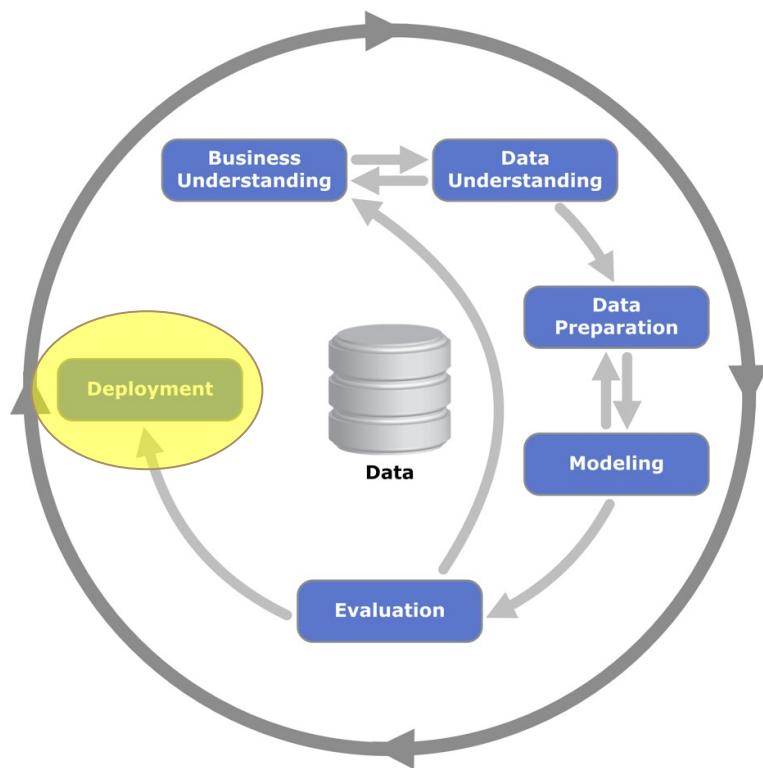
- Evaluate quality using performance criteria on unseen data
- Evaluate model's ability to answer business objectives
 - Example – model might require more data than we can collect
- Determine how to use the output of the model to make decisions



Business understanding	Data understanding	Data Prep	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review Process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

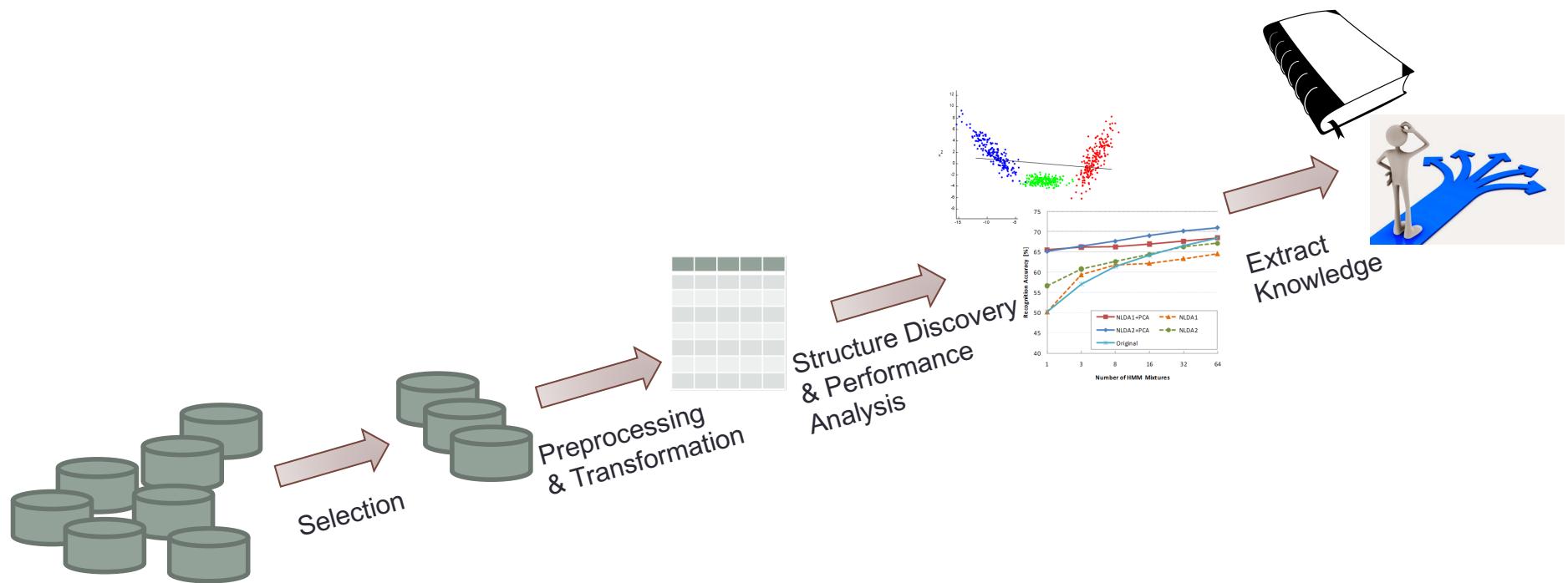
CRISP-DM: Deployment

- Integrate the model into the decision-making process
- Continuous Assessment: ensure model works well over time
- Maintenance: when to update/fix/retune the model



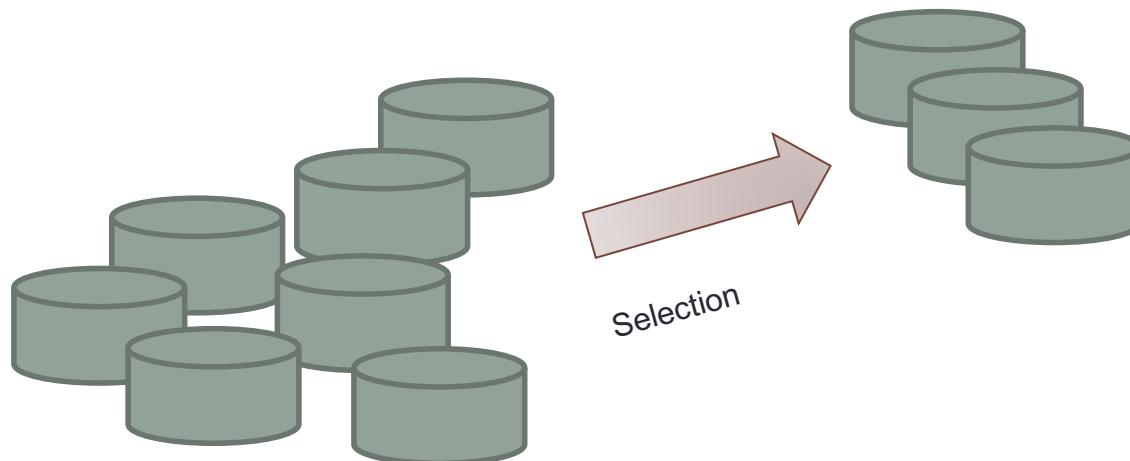
Business understanding	Data understanding	Data Prep	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review Process	Plan monitoring & maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data	Assess model		Review project
		Format data			

Knowledge Discovery in Databases



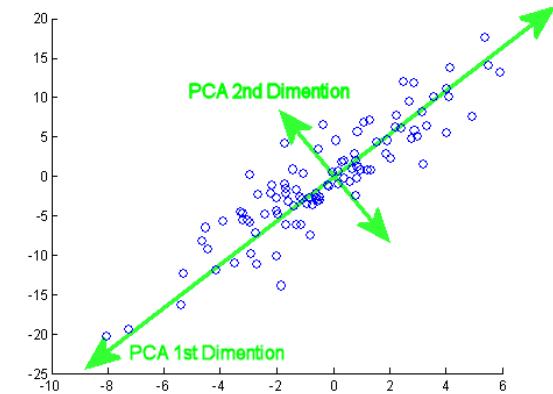
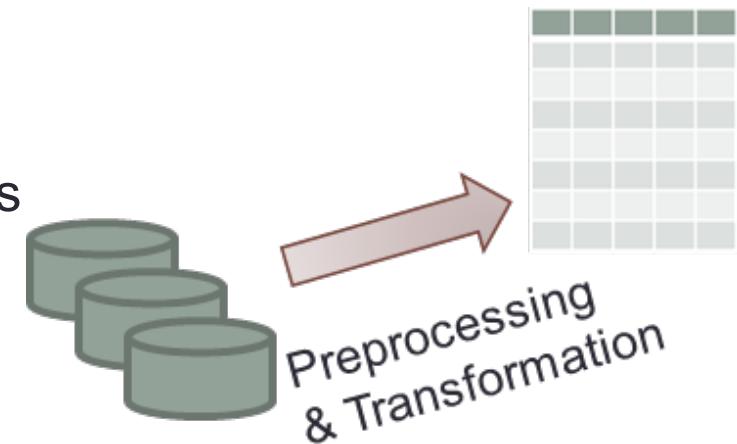
KDD: Selection

- Determine the question which needs to be answered
 - Scientific / Hypothesis Driven
- Choose the data which supports the question



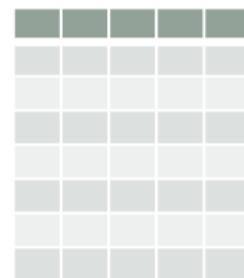
KDD: Preprocessing and Transformation (Data Wrangling)

- Preprocessing:
 - Selected Data may be in many forms
 - Raw text / Image / Video / Markup Language / Time-Series Signal
 - Extract desired *Features* from data
 - Dimensionality reduction / filtering
 - Event counts, measurements, pixel values
 - Generate *observations* (rows) with *feature values* (columns)
 - Impute missing or incorrect values?
- Transformation:
 - Scale the feature values
 - Project observations into a different space/subspace

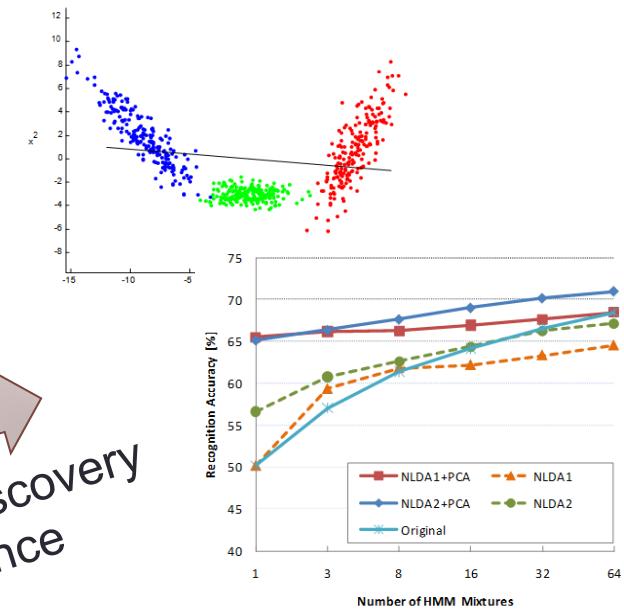


KDD: Structure Discovery and Performance Analysis

- Structure Discovery (Model Fitting)
 - Data Exploration – find trends
 - Regression – estimate a value
 - Classification – determine membership
 - Clustering – determine groupings
 - Inference – determine important features
 - Cross-Validation – tune the model
- Performance Analysis
 - Assess quality of model on unseen data

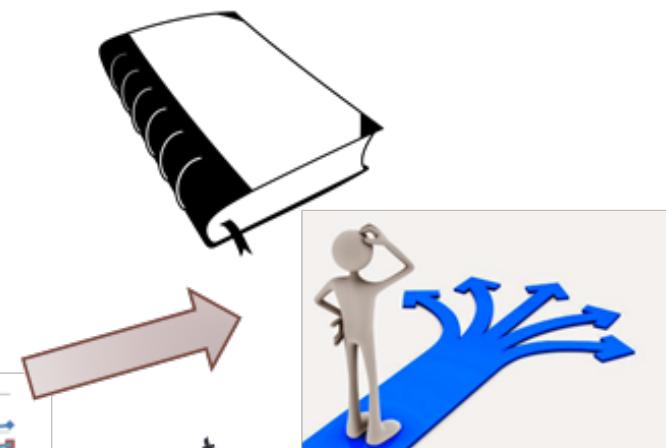
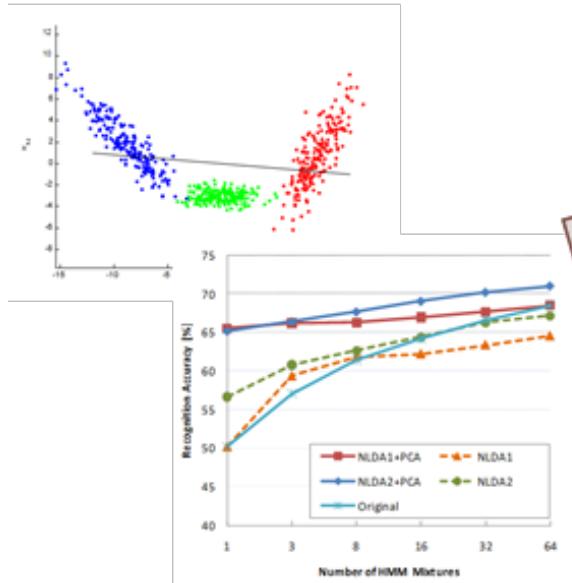


Structure Discovery
& Performance
Analysis



KDD: Extract Knowledge

- The ultimate goal of machine learning
 - Knowledge:
 - Generate understanding of the relationships within the data
 - Understand how features affect the output
 - Decision-making
 - Using knowledge to determine actions



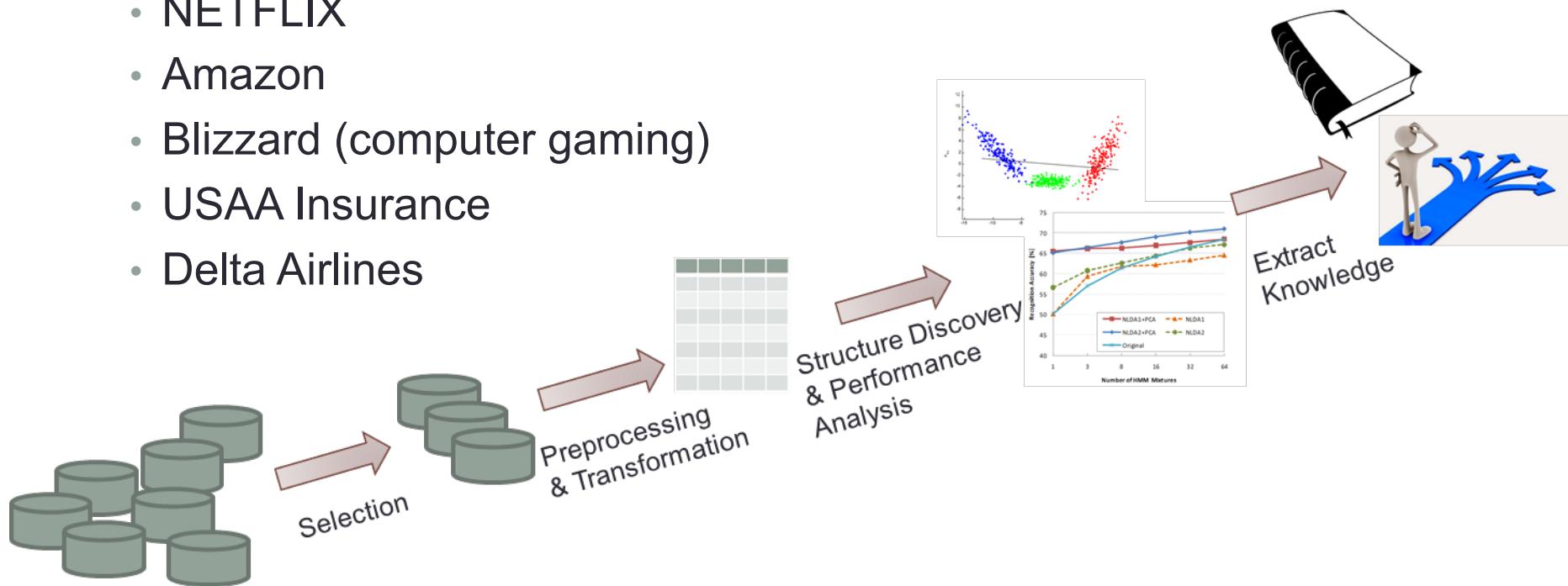
Extract
Knowledge

KDD: Challenges and Cautions

- Data mining vs. Data dredging
- Missing data
- Massive datasets / High dimensionality
- Overfitting & Statistical significance
- Concept drift / Nonstationary data

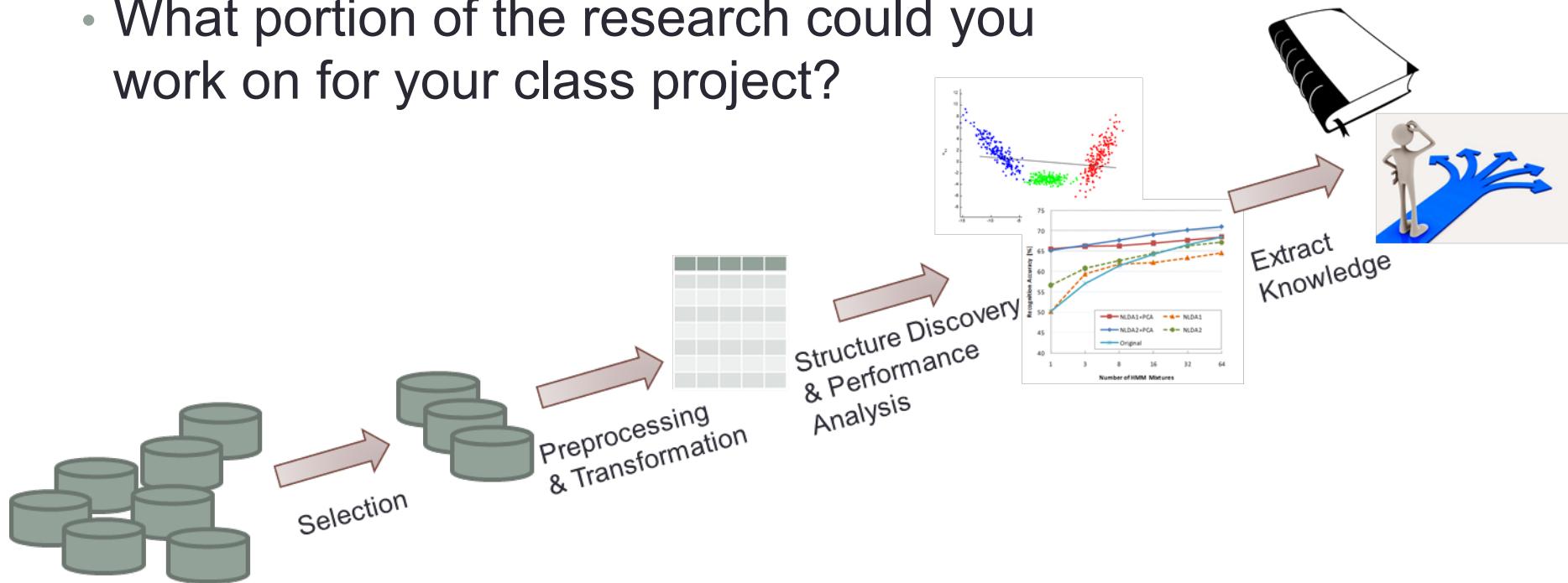
Audience Participation – Minute Paper #1

- Choose one of the following companies and describe how the company might use the KDD process to make a business decision. Describe details of the steps
 - NETFLIX
 - Amazon
 - Blizzard (computer gaming)
 - USAA Insurance
 - Delta Airlines



Audience Participation – Minute Paper #2

- How do you envision using the KDD process in *your* AFIT research?
 - Outline what needs to occur in each step
- What portion of the research could you work on for your class project?



WHAT IS STATISTICAL LEARNING?

Chapter 02 – Part I

Slides Inspired by content from IOM 530 “Applied Modern Statistical Learning Methods” – Gareth James (one of the authors of our book)

Outline

- What Is Statistical Learning?
 - Why estimate f ?
 - How do we estimate f ?
 - The trade-off between prediction accuracy and model interpretability
 - Supervised vs. unsupervised learning
 - Regression vs. classification problems

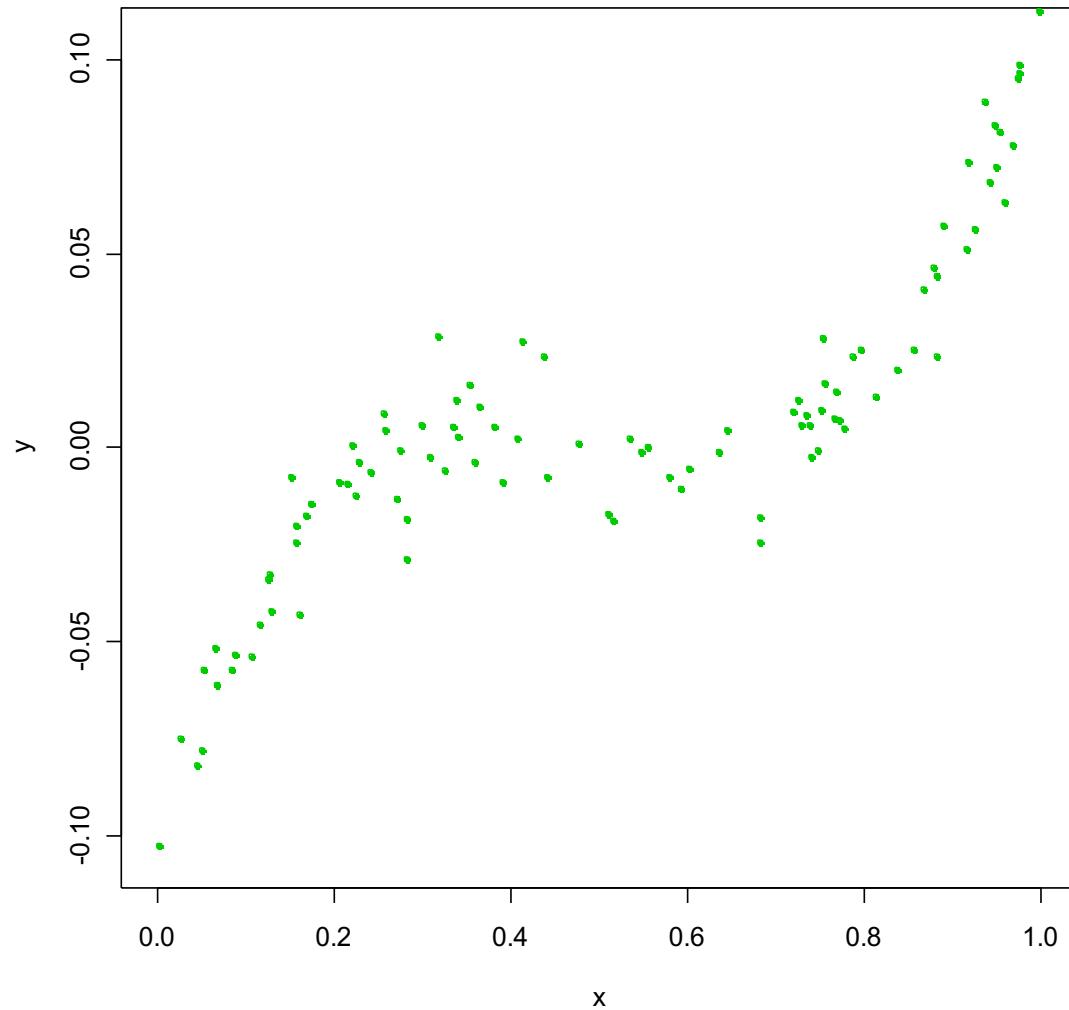
What is Statistical Learning?

- Suppose we observe Y_i and $X_i = (X_{i1}, \dots, X_{ip})$ for $i = 1, \dots, n$
- We believe that there is a relationship between Y and at least one of the X's.
- We can model the relationship as

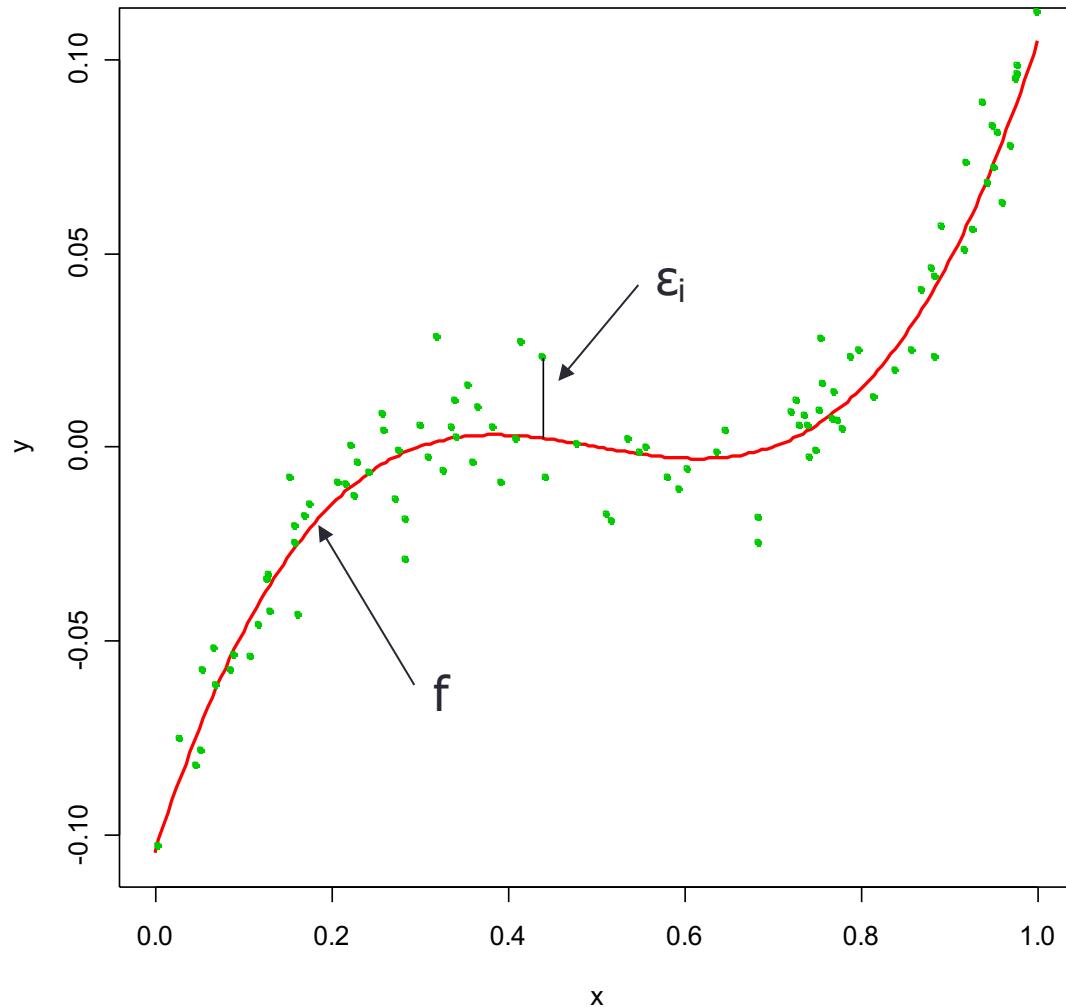
$$Y_i = f(\mathbf{X}_i) + \varepsilon_i$$

- Where f is an unknown function and ε is a random error with mean zero.

A Simple Example



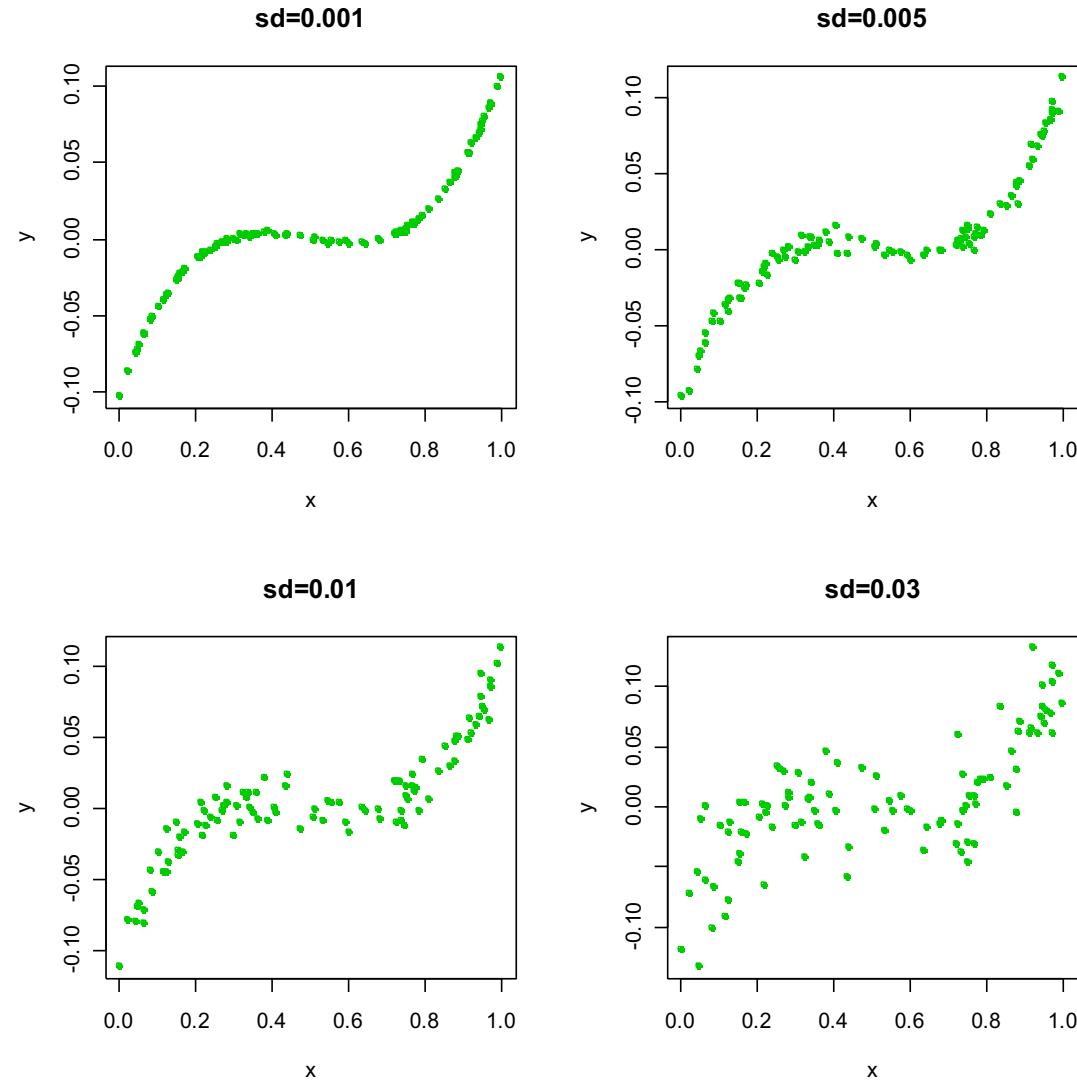
A Simple Example $Y_i = f(\mathbf{X}_i) + \varepsilon_i$



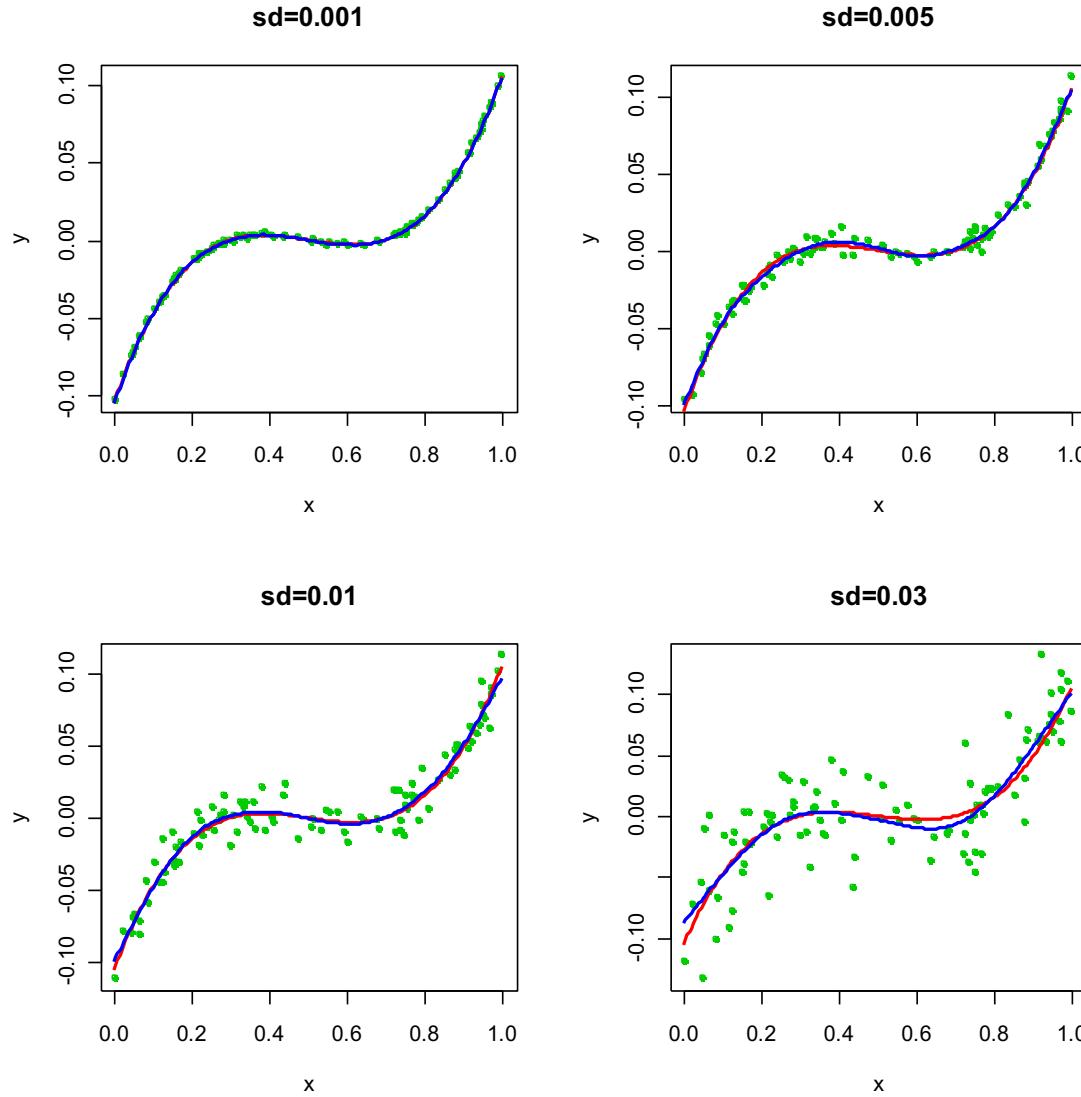
Different Standard Deviations

- The difficulty of estimating f will depend on the standard deviation of the ε 's.

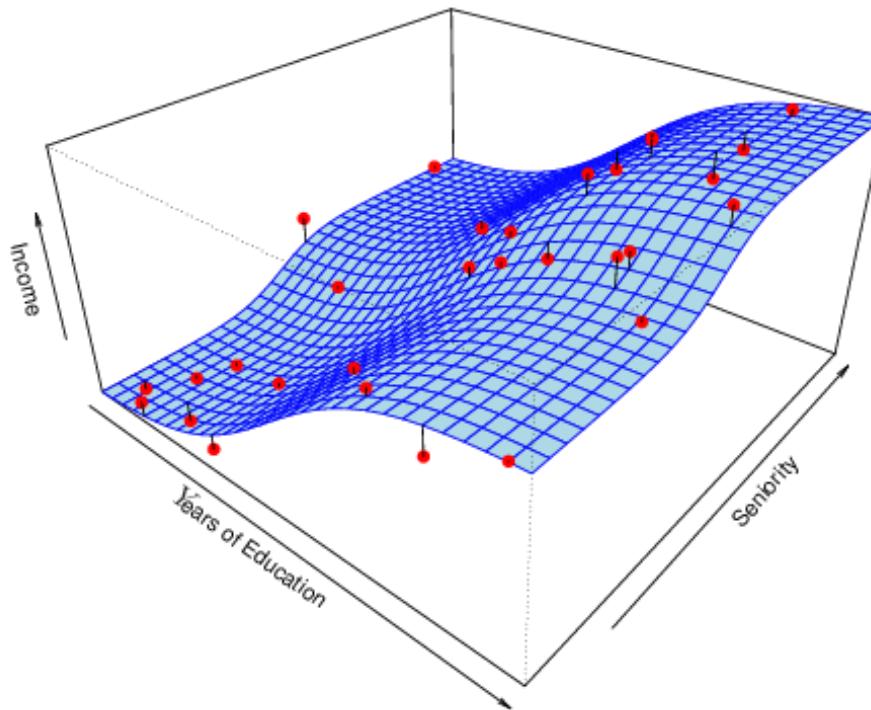
$$Y_i = f(\mathbf{X}_i) + \varepsilon_i$$



Different Estimates For f



Income vs. Education & Seniority



- Shown above is the “true” relationship between the variables Years of Education, Seniority, and Income.
- **CONCEPT CHECK: Describe the relationship between income, years of education and seniority that you see here**

Why Do We Estimate f ?

- Statistical Learning, and this course, are all about how to estimate f .
- The term statistical learning refers to using the data to “learn” f .
- Why do we care about estimating f ?
- There are 2 reasons for estimating f ,
 - **Prediction (Estimation)**
 - **Inference (Explanation)**

Prediction (Estimation)

- If we can produce a good estimate for f (and the variance of ε is not too large) we can make accurate predictions for the response, Y , based on a new value of X .

Prediction / Estimation Example: Direct Mailing Decision

- How much money an individual will donate to a charity?
- Data:
 - \mathbf{X} : 400 characteristics about each person
 - Y : How much they donated.
- Business Question: For a given individual should I send out a mailing?
 - Is the expected value of taking the action greater than the cost of the action?
- Assume that there is no desire to know what features are associated with people who contribute.

Inference (Explanation)

- We may also be interested in the type of relationship between Y and the X's.
- For example,
 - Which particular predictors (features) actually affect the response?
 - Is the relationship positive or negative?
 - Is the relationship a simple linear one or is it more complicated?

Inference Example: Understanding Home prices

- How do characteristics affect home prices
- Housing data:
 - X: 14 characteristics (e.g. number of beds; baths; square feet)
 - Y: cost of the home
- Business Question:
 - How would altering the variables affect my home's value?
- For example
 - Would installing an in-ground pool increase my home's value?
 - What is the financial impact of turning my 1-car garage into a woodworking shop?
 - What is the most cost-effective thing I could do to improve my home's value before I sell it?

How Do We Estimate f ?

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

$$\{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \dots, (\mathbf{X}_n, Y_n)\}$$

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

- Statistical Learning Methods:

- Parametric Methods
- Non-parametric Methods

|Observations| = n

	$\mathbf{X}_{:,1}$	$\mathbf{X}_{:,2}$...	$\mathbf{X}_{:,m}$	Target Label
\mathbf{X}_1					Y_1
\mathbf{X}_2					Y_2
...					...
\mathbf{X}_n					Y_n

Parametric Methods

- Reduces the problem of estimating f to estimating a set of parameters.
- Two-step model based approach

STEP 1:

Make some assumption about the functional form of f , i.e. come up with a model. For example, a linear model:

$$f(\mathbf{X}_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 \cdot X_{i2} + \dots + \beta_p X_{ip}$$

Parametric Methods (cont.)

STEP 2:

Use the training data to fit the model i.e. estimate f or equivalently the unknown parameters such as $\beta_0, \beta_1, \beta_2, \dots, \beta_p$.

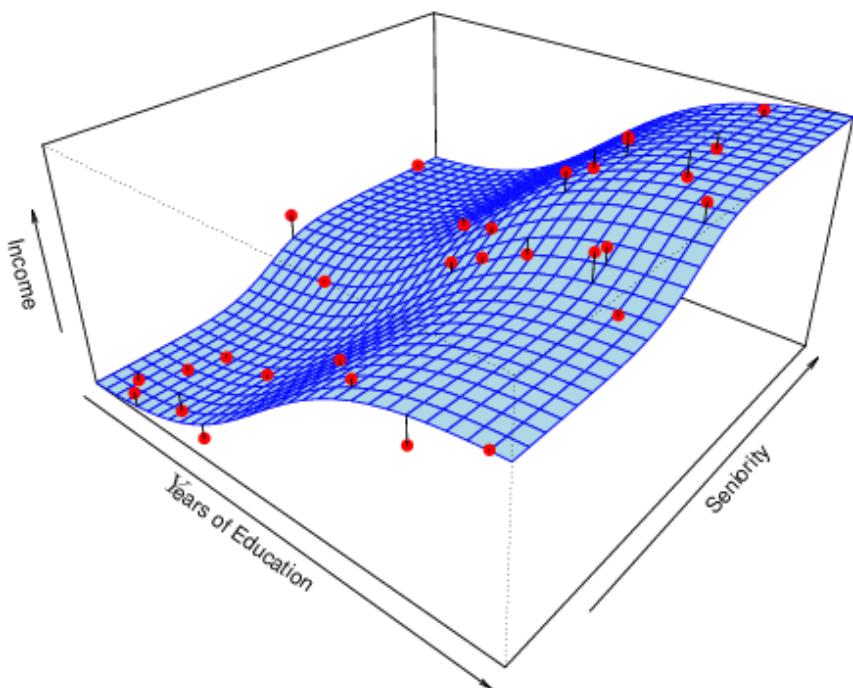
Individual prediction error terms can be computed:

$$\varepsilon_i = Y_i - f(\mathbf{X}_i)$$

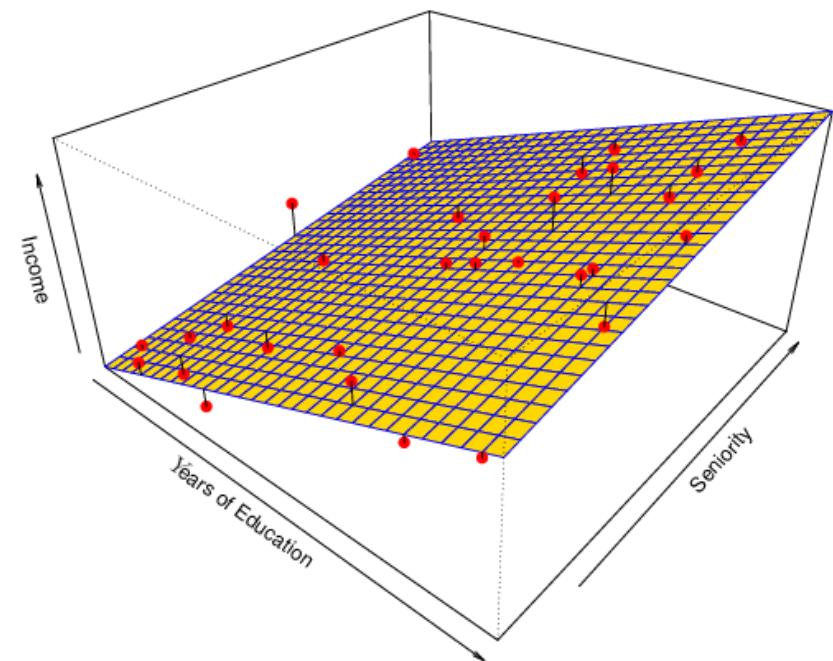
- One common approach for estimating the parameters in a linear model is ordinary least squares (OLS) – which minimizes the square of the sum of the error terms.
 - Has limitations due to computational tractability of inverting a matrix
 - That there are other approaches

META: Why do you think we would want to use OLS in a linear model? (hint – why is *squaring* the error terms mathematically important?)

Parametric Example: Linear Regression



True Phenomenon



Linear Model Approximation

$$f = \beta_0 + \beta_1 \times \text{Education} + \beta_2 \times \text{Seniority}$$

In-Class coding exercise

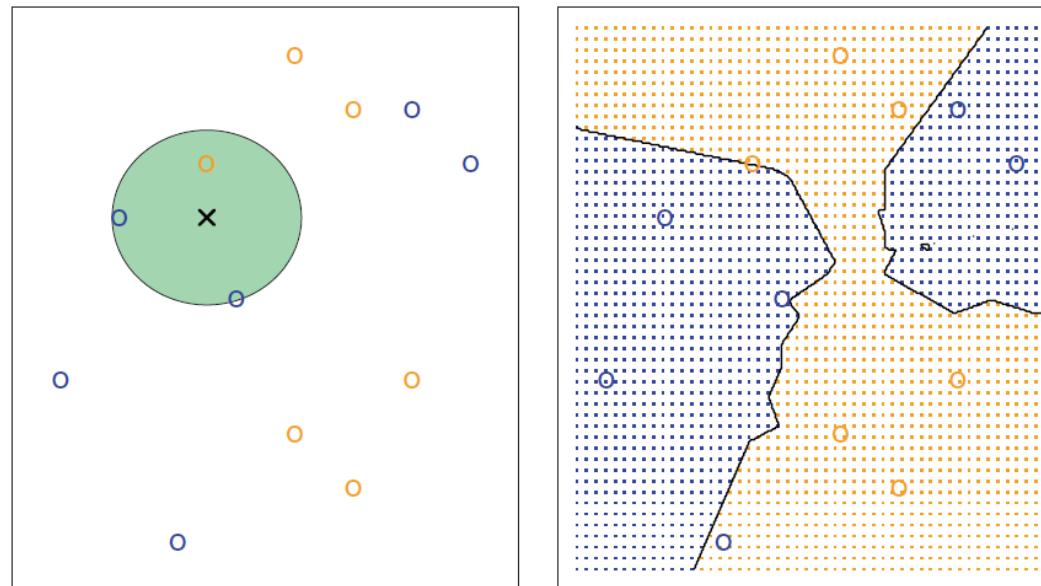
- This exercise will explore 1-dimensional linear regression (one feature is used to predict the target variable)
- You will manually fit the 2 parameter model by guessing and testing different betas until you get a low error
- Directions:
 - Obtain the python starter code from canvas (“in class” tab)
 - Read the directions for “Simple Linear Regression as Matrix Algebra, part 1”
 - Complete the steps 1 – 6 individually – ask your neighbor or the instructor for help if needed

Non-parametric Methods

- Non-parametric methods do not make explicit assumptions about the functional form of f
 - There is no parametric model, and no model parameters are fit from the data during the training process
 - Instead, (some) data observations from the training set are stored and used (directly) during prediction
- Advantages: accurately fit a wider range of possible f
- Disadvantages: Slower to train; Risk of overfitting; A very large number of observations are required to obtain an accurate estimate of f

Non-parametric Example: K-Nearest Neighbors

- Datapoints from the training set are stored
- A new datapoint's membership depends on what training set members it is close to (k members are considered)



ISLR page 40 / Figure 2.14

Model Flexibility

- Flexibility refers to a model's capacity to represent a complex mapping between the underlying data and the target variable
- Low flexibility models make the assumption that the relationship between the data and the target variable is simpler (e.g. linear)
- Higher-flexibility models allow for more elaborate relationships (e.g. polynomial)

Model Flexibility Tradeoff (1/2)

- Why not just use a more flexible method if it has a higher capacity?

Reason 1: Interpreting is easier with less flexible model

A simple method such as linear regression produces a model which is much easier to interpret (Inference is easier). For example, in a linear model, β_j is the average increase in Y for a one unit increase in X_j holding all other variables constant.

Model Flexibility Tradeoff (2/2)

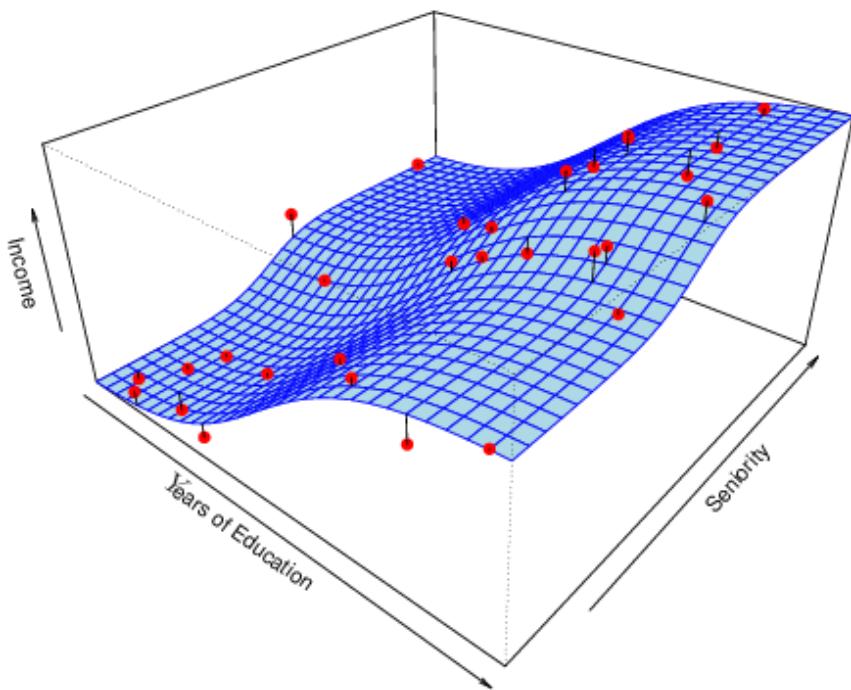
- Why not just use a more flexible method if it has a higher capacity?

Reason 2: Risk of overfitting during training

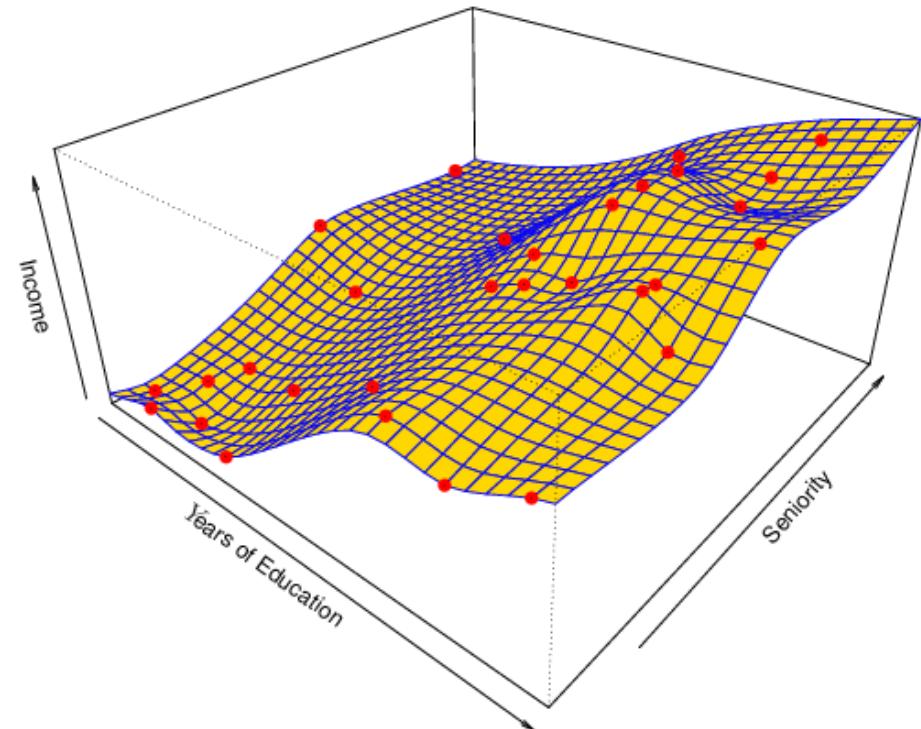
When data availability is limited, it is often possible to get more generalizable predictions with a simple model... a complicated model requires more data to properly train. With less data the higher capacity model essentially replicates a lookup function.

Overfitting

- A model can be too flexible, and make poor estimates of f on unseen data. This is also known as failure to *generalize*



True Phenomenon in Blue



Overfit model
(Fitted to the noise in the data)

Supervised vs. Unsupervised Learning

➤ Supervised Learning:

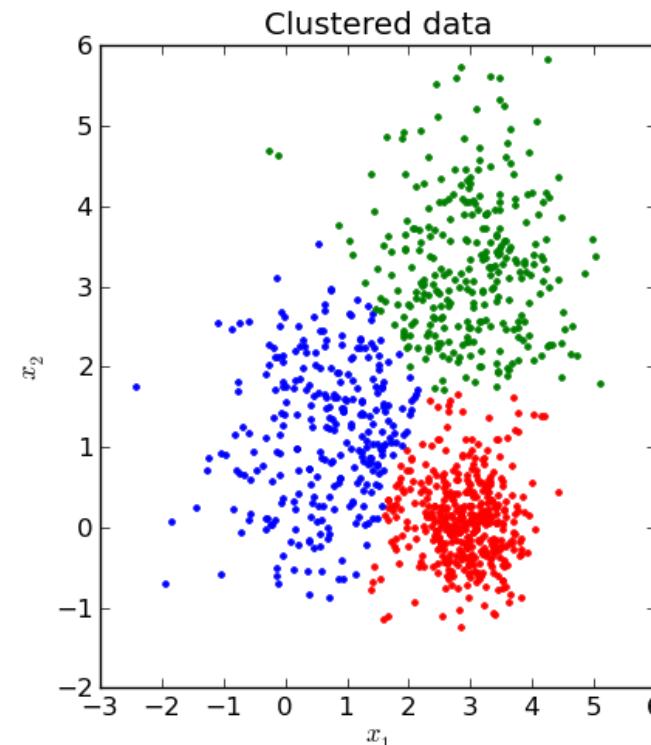
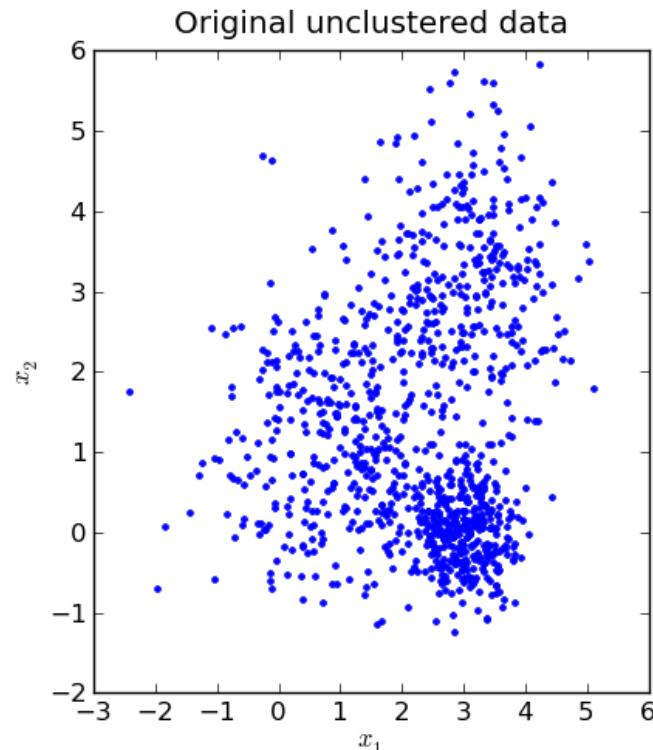
- Supervised Learning is where the predictors, X_i , and the response, Y_i , are observed
- Example task: Income prediction
- Example technique: linear regression

➤ Unsupervised Learning:

- Only the X_i 's are observed.
- We need to use the relationships among the X_i 's to draw conclusions about the data
- Example task: market segmentation - divide potential customers into groups based on their characteristics
- Example technique: clustering

Clustering Example

- Clustering requires a distance measure to be defined for the data elements so that closeness can be determined in the original data feature space
- Clustering algorithms are sometimes evaluated using intra-member cohesion and inter-member separation



Regression vs. Classification

- Supervised learning problems can be further divided into regression and classification problems.
- Regression: Y is continuous/numerical:
 - Predicting the value of a stock 6 months from today.
 - Predicting the price of a given house based on characteristics.
- Classification: Y is categorical:
 - Is this email SPAM or not?
 - Is this a picture of a cat, a dog, or a mouse?

ASSESSING MODEL ACCURACY

Chapter 02 – Part II

Slides Inspired by content from IOM 530 “Applied Modern Statistical Learning Methods” – Gareth James (one of the authors of our book)

Outline

- Assessing Model Accuracy
 - Measuring the Quality of Fit
 - The Bias-Variance Trade-off
 - The Classification Setting

In-class exercise Part 1

- Complete the in-class exercise worksheet front side for Day 4 (problems 1 – 7)

Measuring Quality of Fit

- How do we evaluate a regression model's performance?
- One way: mean squared error (MSE):

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

- Where \hat{y}_i is the prediction our method gives for the observation y_i in our training data.
- **CONCEPT CHECK: What is n in the equation?**
- **Which is better – a higher, or a lower MSE?**
- **Why do we use mean squared error instead of mean error?**

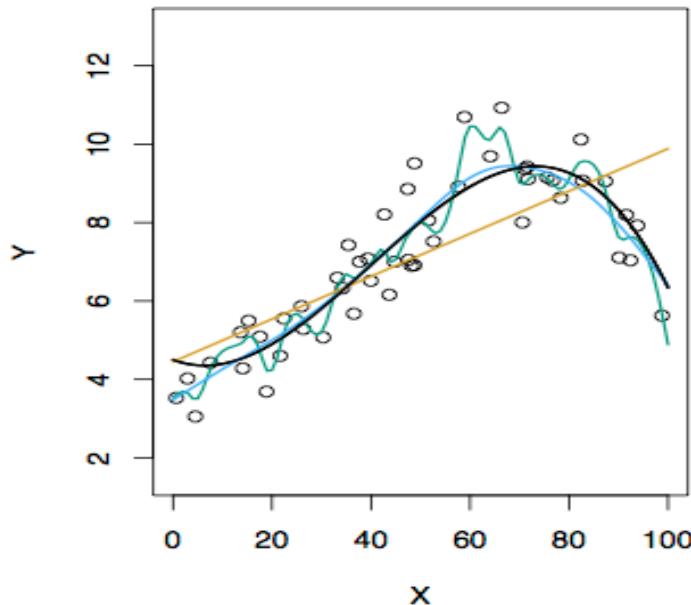
Training vs. Test set performance

- Model Fitting: Choose model parameters such that MSE is minimized on the **Training Data**
- What we really care about is how well the method works on data that was not used for training (i.e. **Test Data**). Test data performance indicates the model's ability to *generalize*.
- There is no guarantee that the method with the smallest *training* MSE will have the smallest *test* MSE. If training performance is good and test performance is bad, the model has *failed to generalize*.

Training vs. Test errors

- In general the more flexible a method is the lower its training MSE ... it will “fit” or explain the training data very well.
- In a more flexible model, the test MSE may be higher than in a less flexible model.
- **CONCEPT CHECK: Why would test MSE be larger than train MSE in a flexible model?**

Examples with Different Levels of Flexibility: Example 1

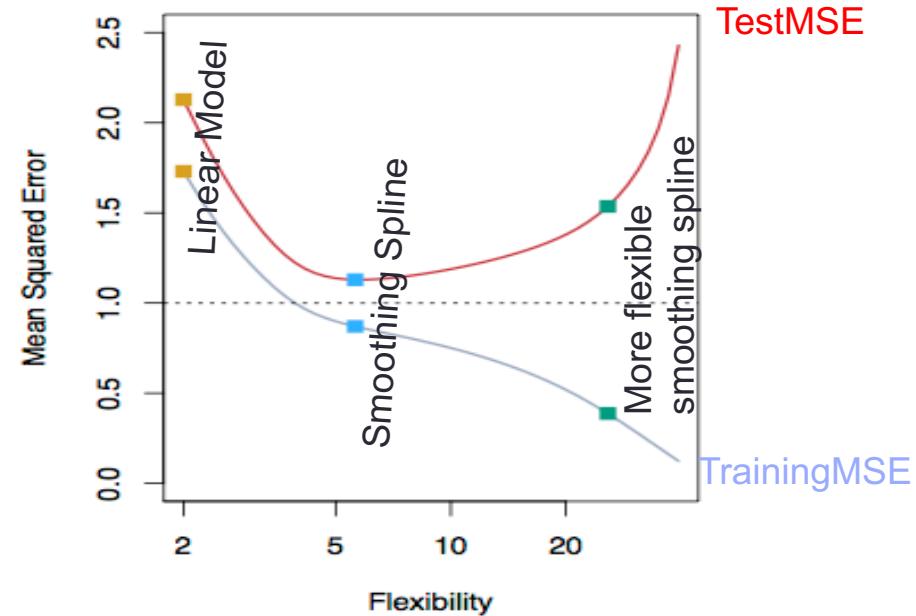


Black: Truth

Orange: Linear Estimate

Blue: smoothing spline

Green: smoothing spline (more flexible)



RED: Test MSE

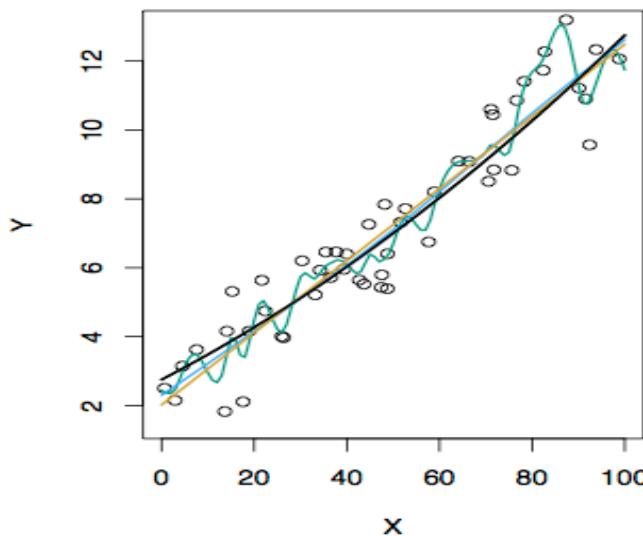
Grey: Training MSE

Dashed: Minimum possible test MSE (irreducible error)

CONCEPT CHECK:

Where does “irreducible error” come from?

Examples with Different Levels of Flexibility: Example 2

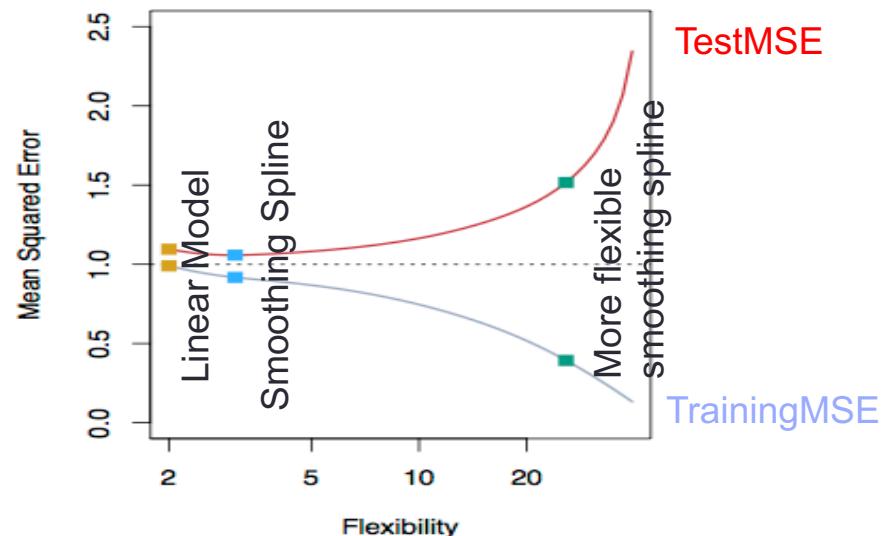


Black: Truth

Orange: Linear Estimate

Blue: smoothing spline

Green: smoothing spline (more flexible)

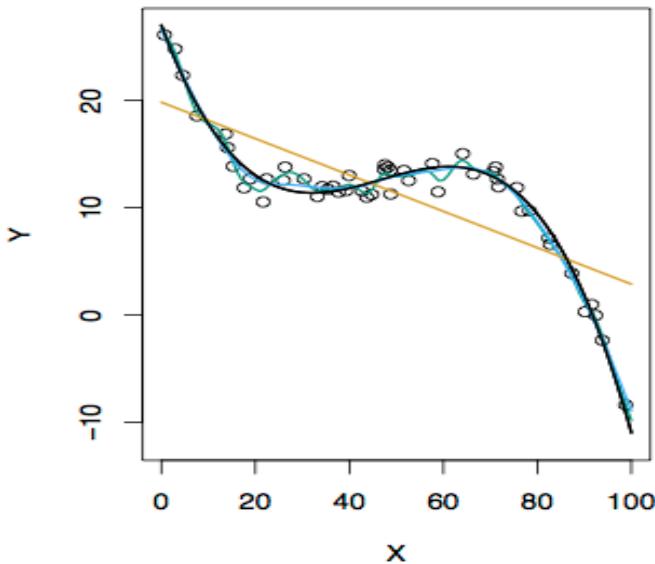


RED: Test MSE

Grey: Training MSE

Dashed: Minimum possible test
MSE (irreducible error)

Examples with Different Levels of Flexibility: Example 3

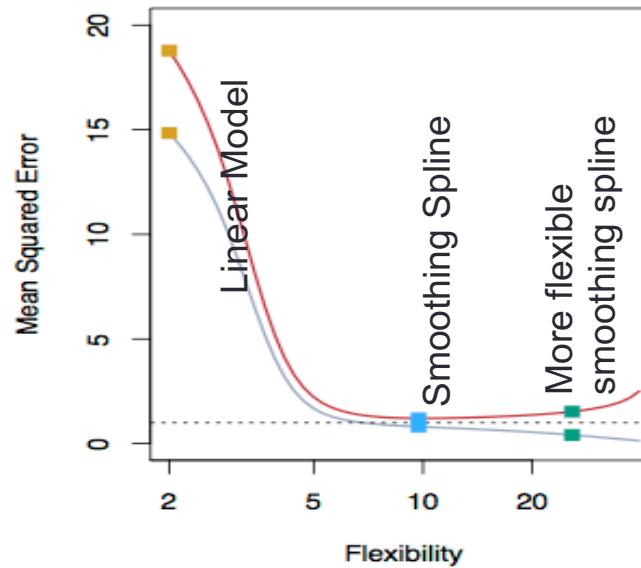


Black: Truth

Orange: Linear Estimate

Blue: smoothing spline

Green: smoothing spline (more flexible)



TestMSE

TrainingMSE

RED: Test MSE

Grey: Training MSE

Dashed: Minimum possible test
MSE (irreducible error)

Bias / Variance Tradeoff

- The previous graphs of **test** versus **training** MSE's illustrate a very important tradeoff that governs the choice of statistical learning methods.
- There are always two competing forces that govern the choice of learning method: *bias* and *variance*.

Bias of Learning Methods

- The *Inductive Bias* of a (machine) learning algorithm is the set of assumptions used to predict outputs given inputs it has not encountered (Tom Mitchell, 1980).
- Intuition- Higher Bias: quicker to generalize well... but more likely to overgeneralize
- In our text, **Bias** refers to the error that is introduced by modeling a real life phenomenon by a model that does not match the phenomenon.
- Often we are talking about a real world phenomenon that is complicated *being modeled by a much simpler model.*
- For example, linear regression assumes that there is a linear relationship between Y and X. It is unlikely that, in real life, the relationship is exactly linear so *some bias* will be induced when we select a linear model.
- The more flexible/complex a method is the less bias it will generally have.

Variance of Learning Methods

- Variance refers to the model's sensitivity to error caused by small fluctuations in the training data.
- Intuition – Variance tells us how sensitive the model is to having trained with a different set of training data from the same original phenomenon
- **Concept check: Why does choosing a higher variance model lead to a performance improvement on the training set but worse performance on the test set?**

Bias-Variance Trade-off

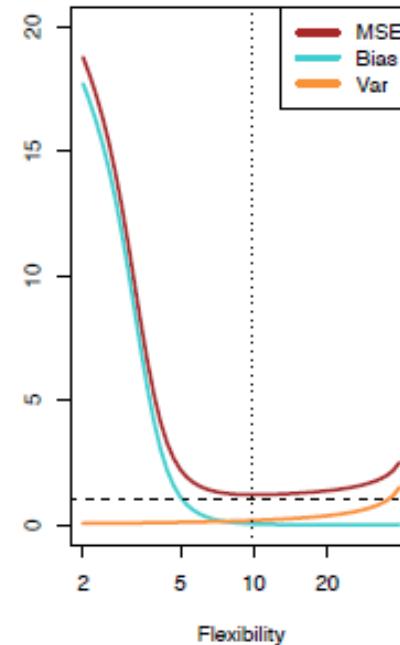
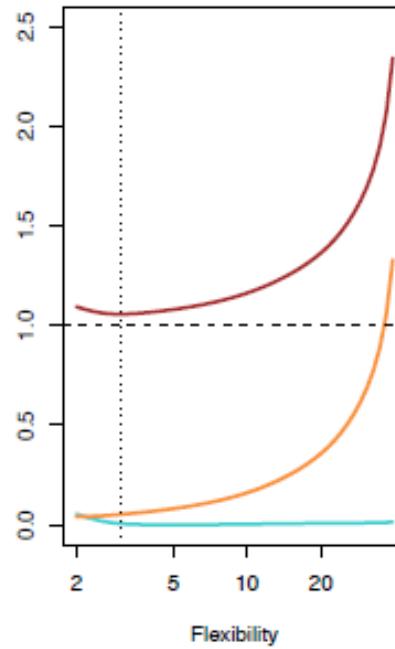
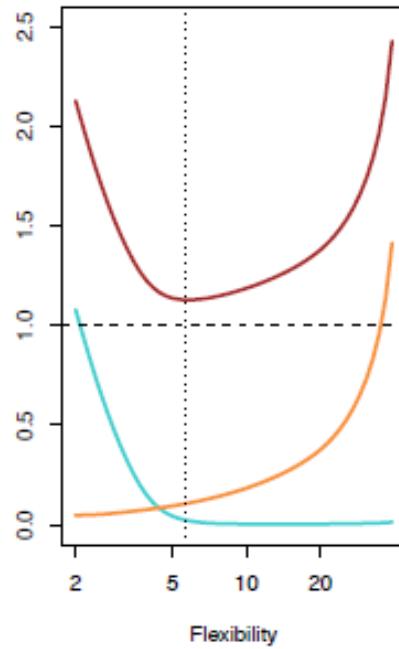
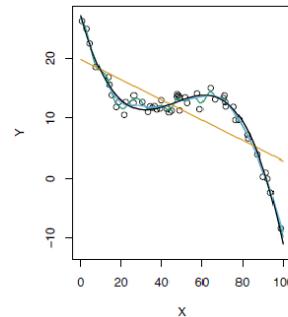
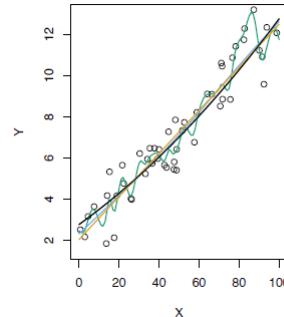
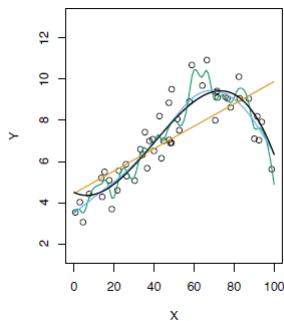
- It can be shown that for any given, $X=x_0$, the expected test MSE for a new Y at x_0 will be equal to

$$\text{ExpectedTestMSE} = E(Y - f(x_0))^2 = \text{Bias}^2 + \text{Var} + \sigma^2$$

Irreducible error

- ... As a modeling method gets more flexible the bias will decrease and the variance will increase but expected test MSE may go up or down
- The mathematical details of the derivation are in the “Elements of Statistical Learning” book but are not covered in our course
(<https://web.stanford.edu/~hastie/ElemStatLearn/>)

Test MSE, Bias and Variance



Assessing *Classification* Performance

- For a regression problem, we used the MSE to assess the accuracy of the statistical learning method
- For a classification problem we can use the error rate i.e.

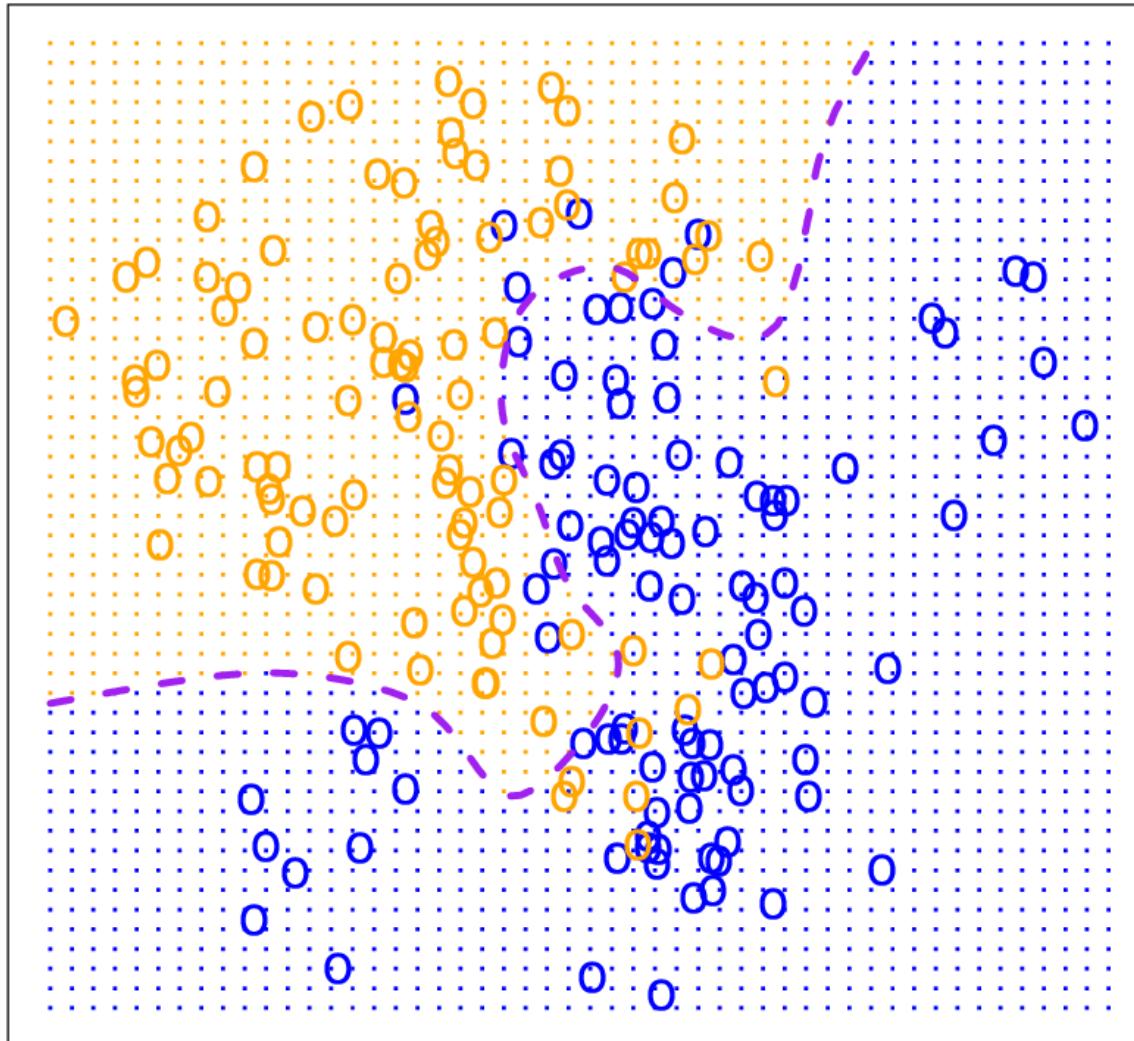
$$\text{Error Rate} = \sum_{i=1}^n I(y_i \neq \hat{y}_i) / n$$

- $I(y_i \neq \hat{y}_i)$ is an *indicator function*, which will give 1 if the condition $(y_i \neq \hat{y}_i)$ is true, otherwise it gives a 0.
- Thus the error rate represents the fraction of incorrect classifications, or misclassifications

Bayes Error Rate

- The Bayes error rate refers to the lowest possible error rate that could be achieved if somehow we knew exactly what the “true” probability distribution of the data looked like.
- On test data, no classifier (or statistical learning method) can get lower error rates than the Bayes error rate.
- *In many real life problems the Bayes error rate can't be calculated exactly. Why not?*

Bayes Optimal Classifier



K-Nearest Neighbors (KNN)

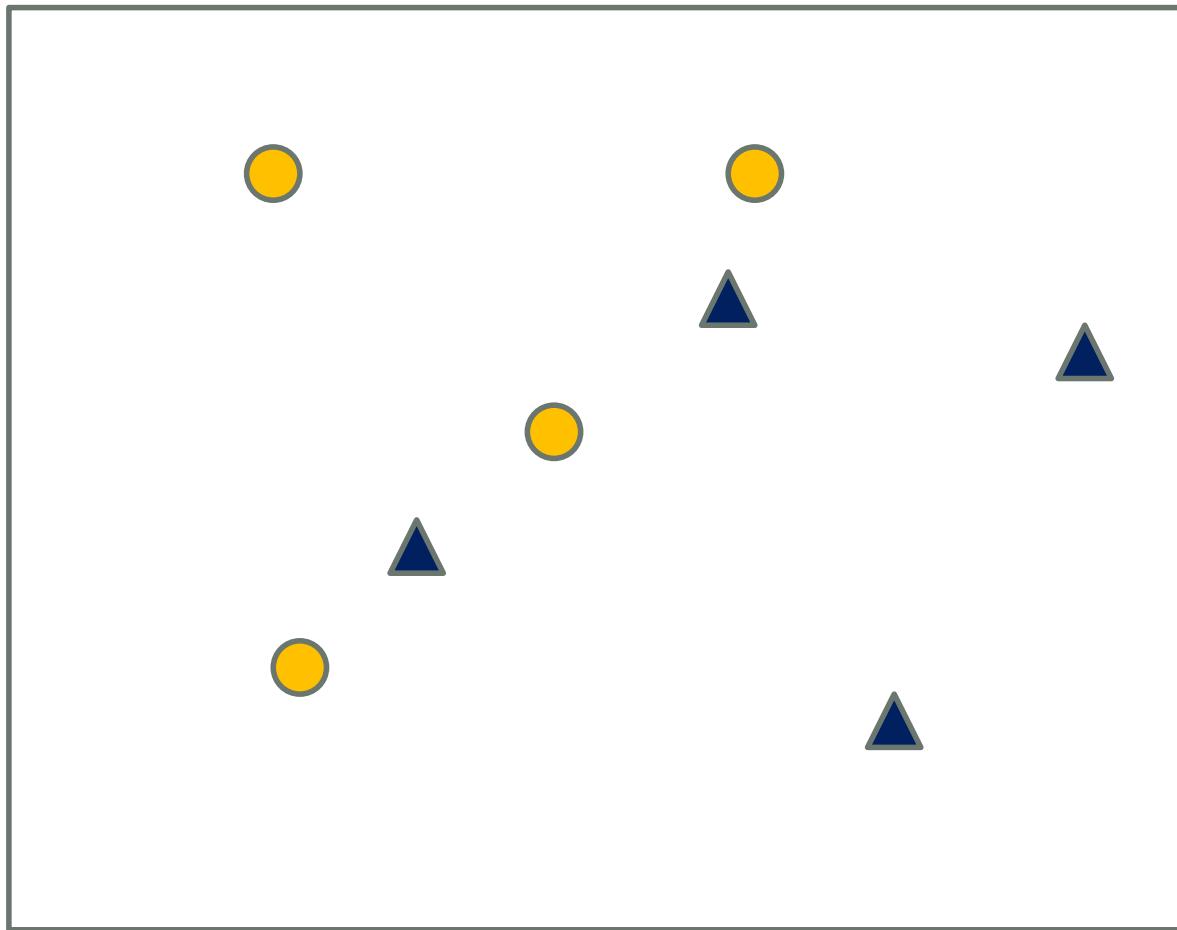
- K-Nearest Neighbors is a flexible approach for classification
 - can be used to *estimate* the Bayes Classifier.
- For any given X_i we find the k closest neighbors to X_i in the training data, and examine their corresponding Y labels.
- The class of X_i is predicted to be the class of the majority (or plurality if more than 2 classes) of its neighbors
- The smaller that k is the more flexible the method will be.

KNN Worksheet

- Using your knowledge of test and training set performance, complete problem # 8 on the worksheet for Day 4

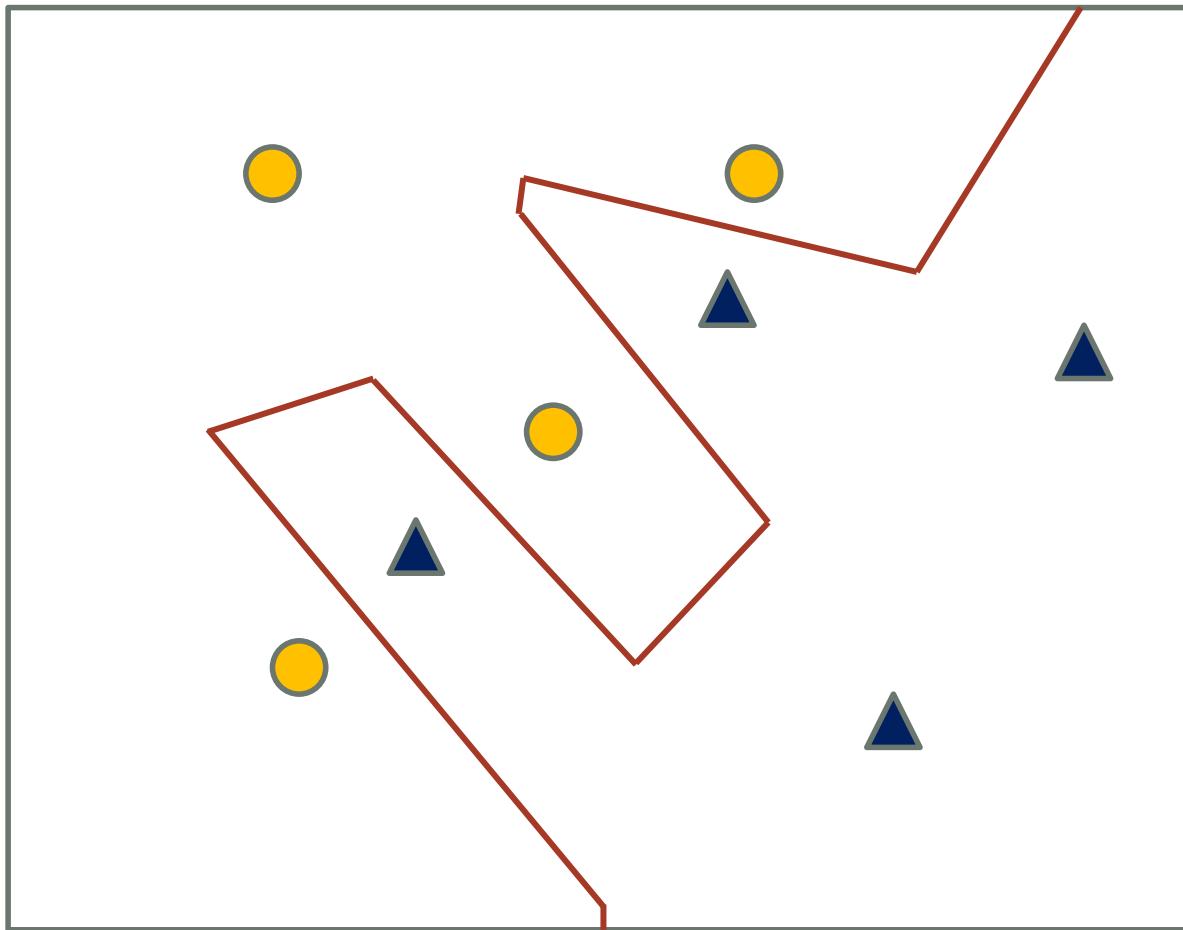
KNN visual – decision boundaries

- What do the decision boundaries look like when K=1?

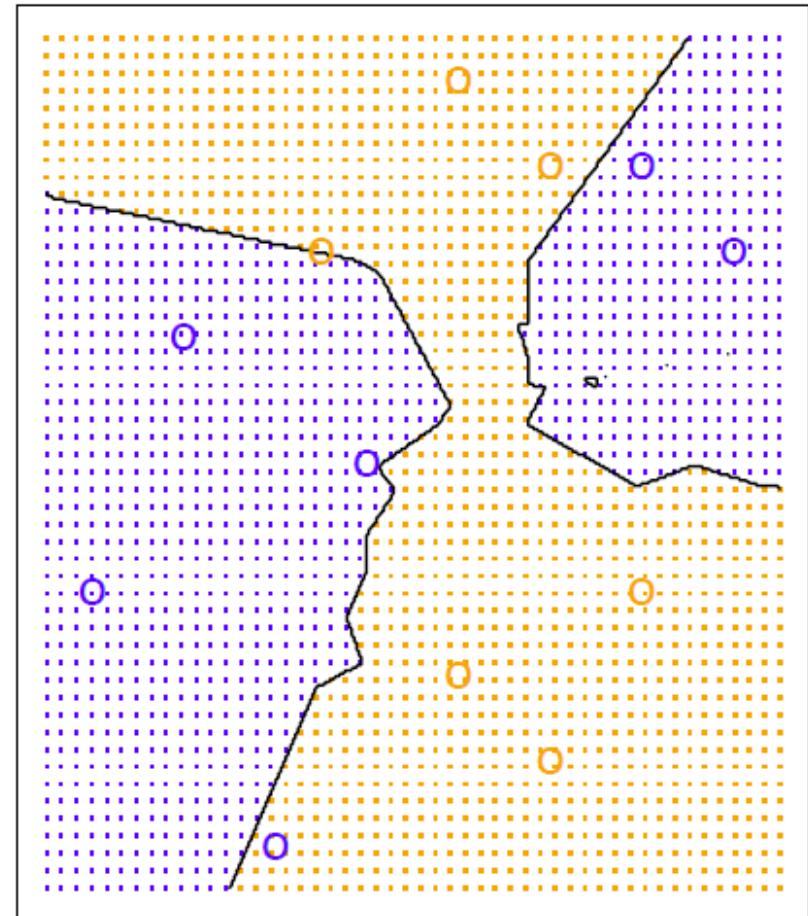
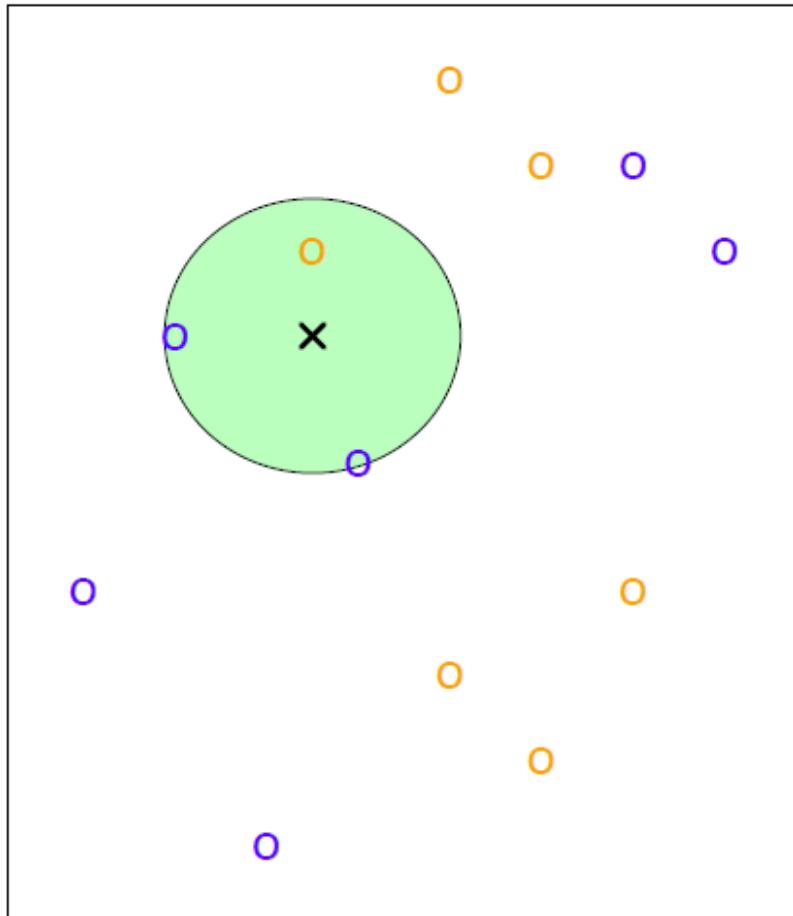


KNN Demo – decision boundaries

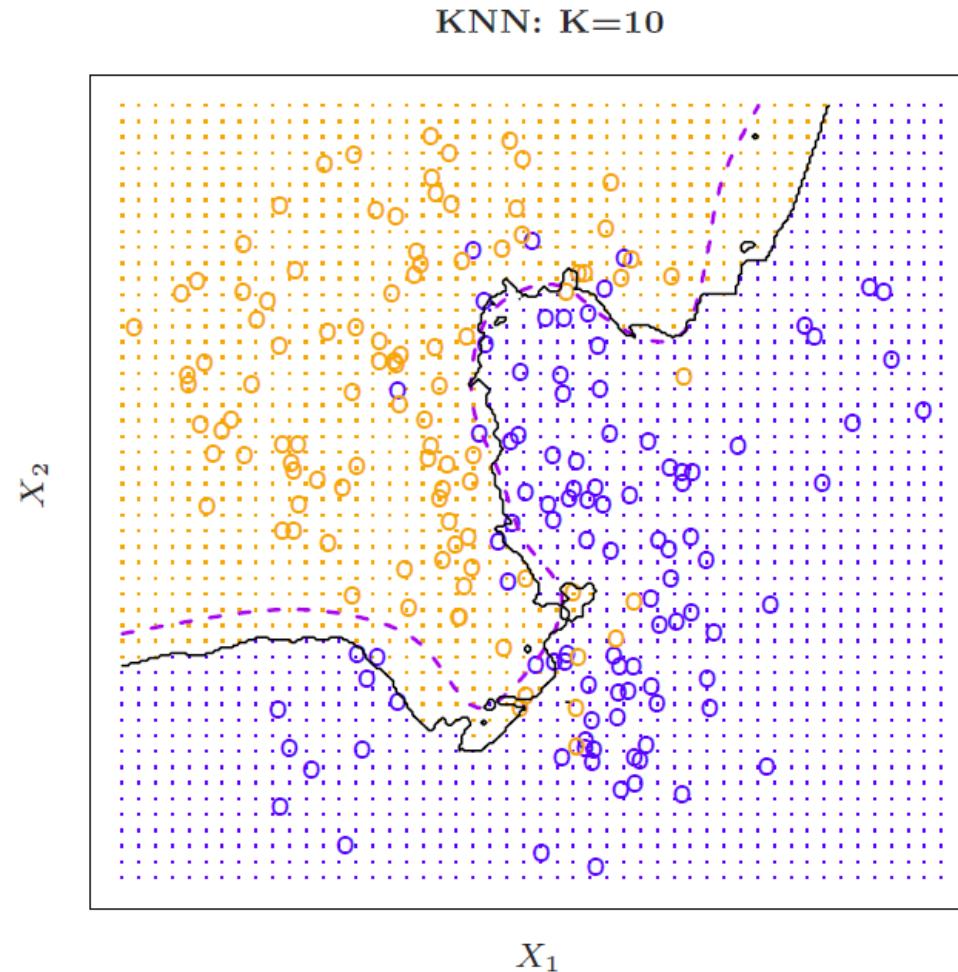
- What do the decision boundaries look like when K=1?



KNN Example with $k = 3$

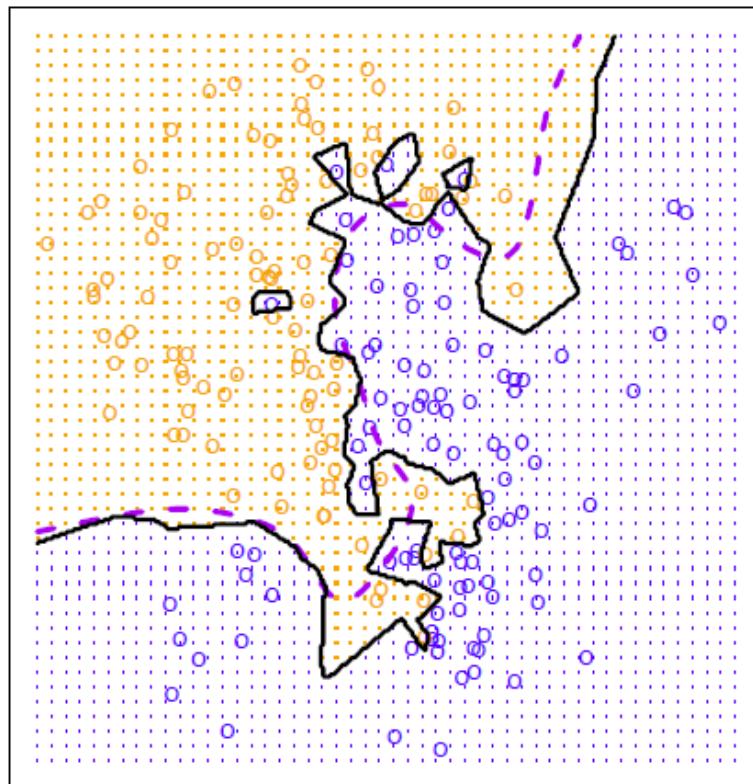


Simulated Data: K = 10

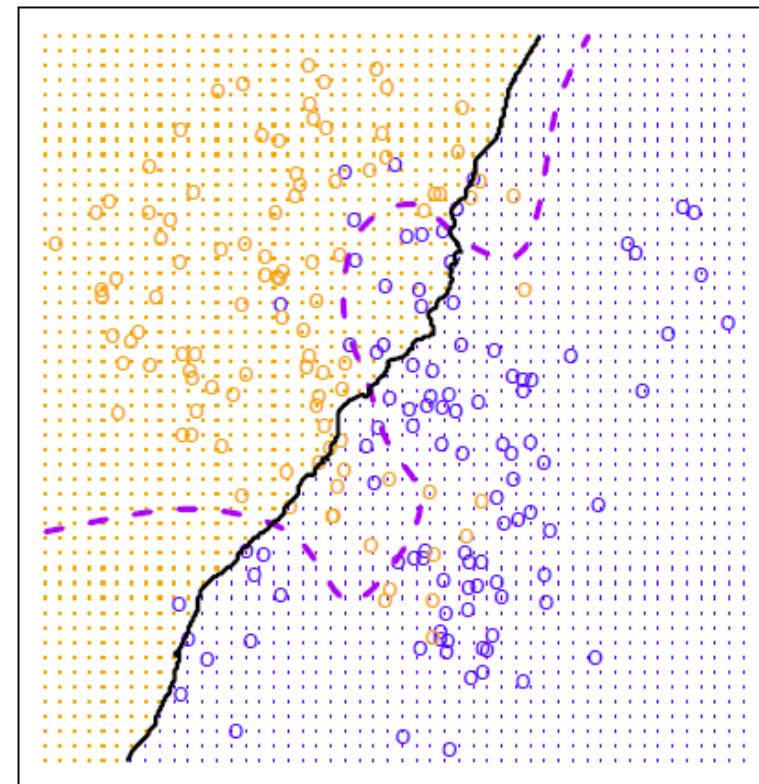


K = 1 and K = 100

KNN: K=1

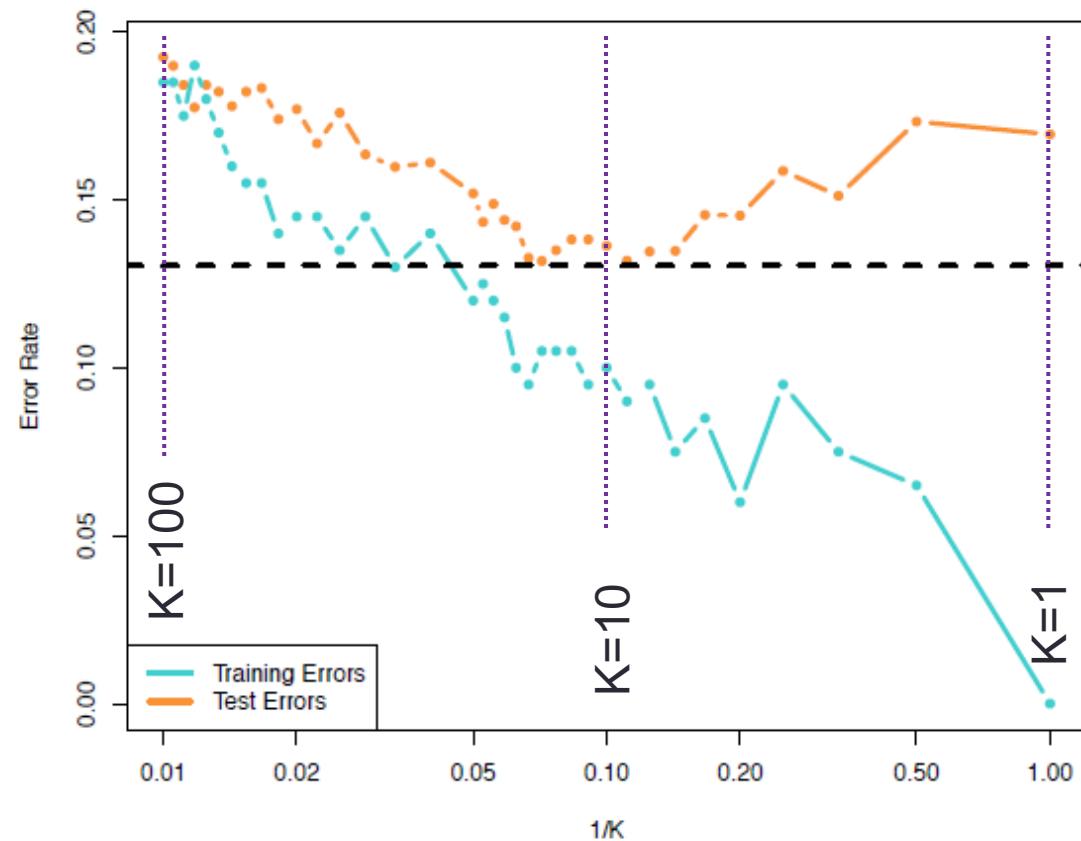


KNN: K=100



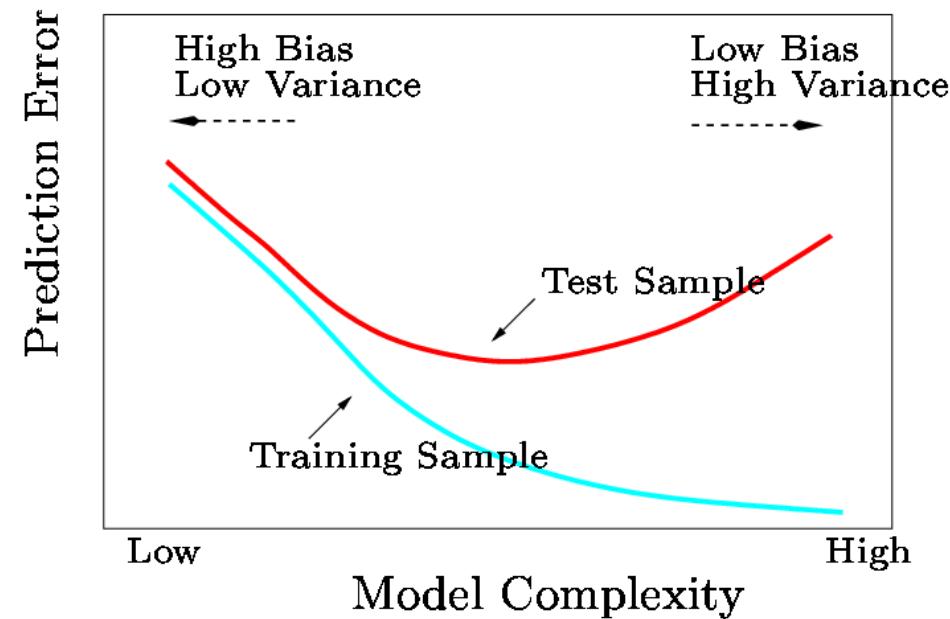
Training vs. Test Error Rates on the Simulated Data

- Notice that training error rates keep going down as k decreases or equivalently as the flexibility increases.
- However, the test error rate at first decreases but then starts to increase again. **WHY?**



Model complexity & Performance

- As model complexity increases, ***training*** error declines*
- As complexity increases, ***test*** errors will decline at first (as reductions in bias dominate) but will then start to increase again (as increases in variance dominate)
- Where test error is minimized, the model has a good complexity



Find the model with the *right* complexity

More flexible/complicated is not always better

LINEAR REGRESSION

Chapter 03

Outline

- The Linear Regression Model
 - Least Squares Model Fitting
 - Measures of Fit
 - Inference in Regression
- Other Considerations in Regression Model
 - Qualitative Predictors
 - Interaction Terms
- Potential Fit Problems
- Linear vs. KNN Regression

Outline

- The Linear Regression Model
 - Least Squares Model Fitting
 - Measures of Fit
 - Inference in Regression
- Other Considerations in Regression Model
 - Qualitative Predictors
 - Interaction Terms
- Potential Fit Problems
- Linear vs. KNN Regression

The (multiple) Linear Regression Model

$$y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi} + U_i$$

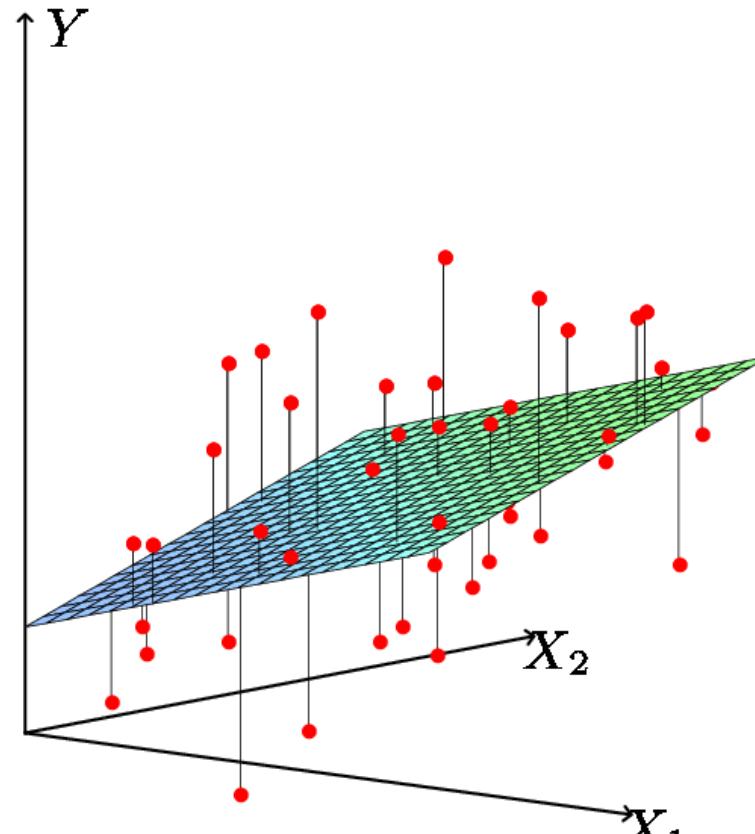
- The parameters in the linear regression model are very easy to interpret.
- β_0 is the intercept (i.e. the average value for Y if all the X's are zero), β_j is the slope for the j^{th} variable X_j
- β_j is the average increase in Y when X_j is increased by one and **all other X's are held constant**.

Least Squares Fit

- Estimate the parameters using least squares
- The best coeff's are the ones which minimize the cost

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

$$= \frac{1}{n} \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \dots + \hat{\beta}_p X_{pi}))^2$$



Concept Check:

What is the difference between RSS and MSE?

Relationship between population and least squares fit

Population	$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + U$
	
Least Squares fit	$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \dots + \hat{\beta}_p X_{pi}$

- Would like to know β_0 through β_p : the population line.
Instead we know $\hat{\beta}_0$ through $\hat{\beta}_p$: the least squares line.
- Use $\hat{\beta}_0$ through $\hat{\beta}_p$ as guesses for β_0 through β_p and \hat{y}_i as a guess for y .

Least Squares Pseudocode Exercise

- Write pseudocode for a primitive method for determining the least-squares model fit in 1-variable linear regression (to find β_0 & β_1)
 - Your observations are stored in matrix X. For each observation, assume you are given x_1 and the corresponding y.
 - Hint: If you want to do gradient descent, you could compute a “local gradient” near a value of β_i by computing the RSS change occurring from an epsilon increase of the coefficient:
 - RSS when using $(\beta_i + \varepsilon)$ minus RSS when using $(\beta_i - \varepsilon)$
 - Think: how would you use these local gradients to search for a best set of beta values?
- How would you extend your idea to a general multiple linear regression model fitting algorithm?

Least Squares Python Exercise

- Write python code for a primitive method for determining the least-squares model fit in 1-variable linear regression (to find β_0 & β_1)
 - Your observations are stored in matrix X. For each observation, assume you are given x_1 and the corresponding y.
- Your portion of the code needs to compute a “local gradient” near a value of β_i by computing the RSS change occurring from an epsilon increase of the coefficient (for each coefficient):
 - $\text{RSS}(f(X \text{ at } \beta_0 + \varepsilon, \beta_1)) - \text{RSS}(f(X \text{ at } \beta_0 - \varepsilon, \beta_1))$
 - $\text{RSS}(f(X \text{ at } \beta_0, \beta_1 + \varepsilon)) - \text{RSS}(f(X \text{ at } \beta_0, \beta_1 - \varepsilon))$

Evaluation Criteria Worksheet

There are a number of evaluation criteria for linear regression models. Fill out the first side of the handout per the instructions

RSS	p -value
MSE	R^2
TSS	Correlation(X,Y)
Var & SE	F-statistic
RSE	Leverage statistic
t -statistic	VIF

We will discuss a subset of these in class

Measure of Lack of Fit: Residual Standard Error (RSE)

- RSE is an estimate of the standard deviation of the irreducible error ε .
- Roughly the average amount that the response will deviate from the true regression line (because of ε)

$$RSE = \sqrt{\frac{RSS}{n-2}} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}}$$

RSE is sensitive to the Y scale of the data since it is measured in units of y .

Measures of Fit: R²

- Some of the variation in Y can be explained by variation in the X's and some cannot.
- R² is a proportion of the variance and is scale invariant
- R² tells you the fraction of variance that can be explained by X.

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \approx 1 - \frac{\text{Ending Variance}}{\text{Starting Variance}}$$

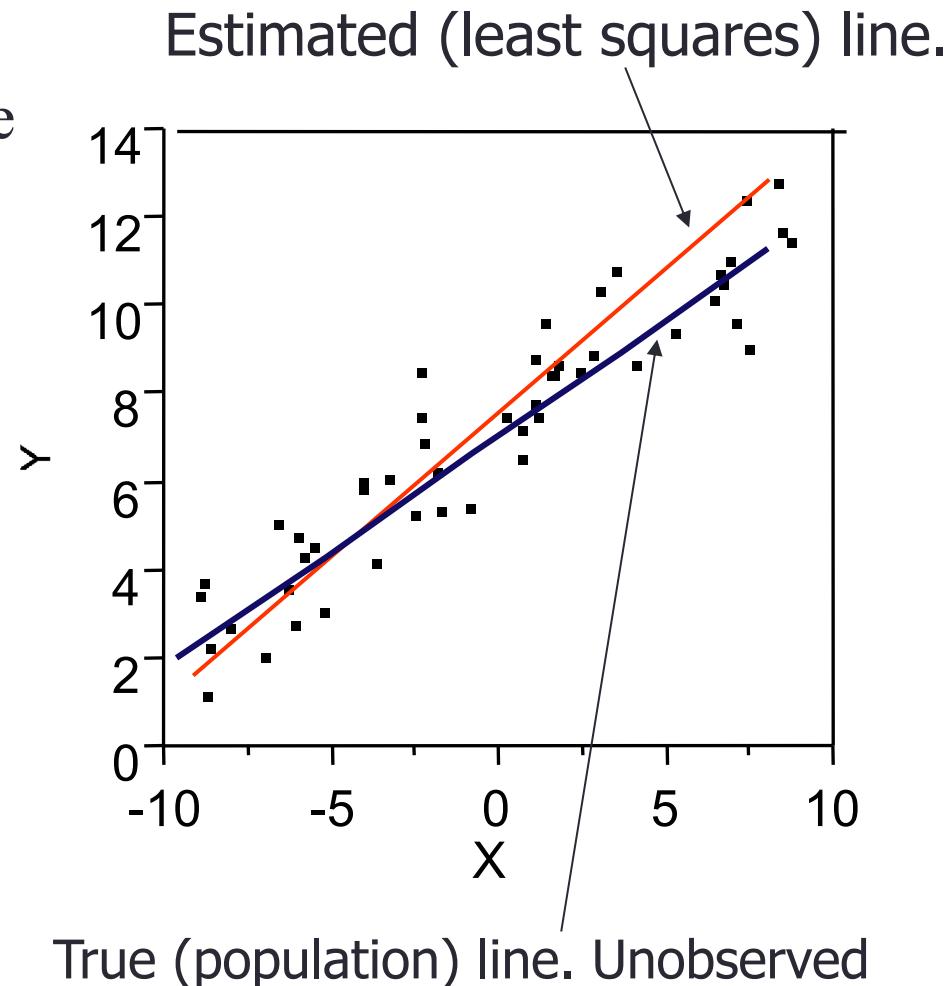
R² is always between 0 and 1. Zero means no variance of the response (Y) has been explained by the model. One means all the variance in the response Y has been explained (perfect fit to the data).

Prediction & Inference in Regression

➤ The regression line from the sample is not the regression line from the population.

➤ What we want to do:

- Guess what value Y would take for a given X value
- Assess how well the line describes the plot.
- Guess the slope of the population line.



Feature (Predictor) Relevance

- Can we be sure that at least one of our X variables is a useful predictor? [i.e. not the case that $\beta_1 = \beta_2 = \dots = \beta_p = 0$]
- Do all the predictors help to explain Y , or are only a subset useful?
 - In other words, is $\beta_j = 0$ or not? We can use a hypothesis test to answer this question.
 - Feature Selection: If we can't be sure that $\beta_j \neq 0$ then there is no point in using X_j as one of our predictors.

Evaluating the regression model (1/2)

➤ Test for:

- H_0 : all slopes = 0 ($\beta_1=\beta_2=\dots=\beta_p=0$),
- H_a : at least one slope $\neq 0$
- p predictors (features) and n observations
- Compute the F statistic

$$F = \frac{\left(\frac{(TSS - RSS)}{p} \right)}{\left(\frac{RSS}{(n - p - 1)} \right)}$$

When F is close to 1 there is no relationship between the response and the predictors
 When $F > 1$, we can consider rejecting H_0
 The amount above 1 required depends on n .
 The larger n is, the less F has to be to reject H_0
 Note: $p < n$ for this to be useful

Evaluating the regression model (2/2)

➤ Test for:

- H_0 : all slopes = 0 $(\beta_1 = \beta_2 = \dots = \beta_p = 0)$, $F = \frac{\frac{(TSS - RSS)}{p}}{\frac{RSS}{(n - p - 1)}}$
- H_a : at least one slope $\neq 0$

Answer comes from the F test in the ANalysis Of VAriance (ANOVA) table.

The ANOVA table has many pieces of information. What we care about is the F-Ratio and the corresponding p -value.

ANOVA Table

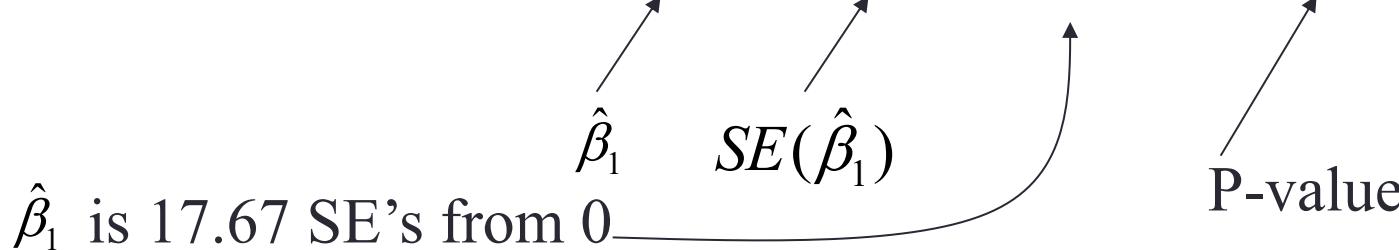
Source	df	SS	MS	F	p-value
Explained	2	4860.2347	2430.1174	859.6177	0.0000
Unexplained	197	556.9140	2.8270		

Given a passing F-test, Is $\beta_j \neq 0$? is X_j an important variable?

- We use a hypothesis test to answer this question
- $H_0: \beta_j = 0$ vs $H_a: \beta_j \neq 0$
- Calculate
$$t = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$
 ← Number of standard deviations away from zero.
- If t is large (equivalently p -value is small) we can be sure that $\beta_j \neq 0$ and that there is a relationship

Regression coefficients

	Coefficient	Std Err	t-value	p-value
Constant	7.0326	0.4578	15.3603	0.0000
TV	0.0475	0.0027	17.6676	0.0000



Testing Individual Variables & Conditional Relationships

Example: Is there a (statistically detectable) linear relationship between Newspapers and Sales given all the other variables have been accounted for? What about if Newspaper is the only available media?

Regression coefficients

	Coefficient	Std Err	t-value	p-value
Constant	2.9389	0.3119	9.4223	0.0000
TV	0.0458	0.0014	32.8086	0.0000
Radio	0.1885	0.0086	21.8935	0.0000
Newspaper	-0.0010	0.0059	-0.1767	0.8599

big p-value: NO

Regression coefficients

	Coefficient	Std Err	t-value	p-value
Constant	12.3514	0.6214	19.8761	0.0000
Newspaper	0.0547	0.0166	3.2996	0.0011

Small p-value in simple regression

Interpretation: Newspaper doesn't add much given that TV and Radio are used. Decision: If we can use TV & Radio, we should, but if they are not available, Newspaper still affects sales.

Outline

- The Linear Regression Model
 - Least Squares Model Fitting
 - Measures of Fit
 - Inference in Regression
- Other Considerations in Regression Model
 - Qualitative Predictors
 - Interaction Terms
- Potential Fit Problems
- Linear vs. KNN Regression

Two-way Qualitative Predictors

- Suppose you have a “gender” feature. How do you code “male” and “female” (category listings) into a regression equation?
- Option 1:
Code them as indicator variables (“dummy” variables)
 - For example we can “code” Males=0 and Females= 1.
- Option 2:
Code them as +1/-1 variables For example we can “code” Males= -1 and Females= 1.

Two-way Qualitative: Zero-One Coding

- Suppose we want to include income and gender to determine bank balance.
- Two genders (male and female). Let

$$\text{Gender}_i = \begin{cases} 0 & \text{if male} \\ 1 & \text{if female} \end{cases}$$

- then the regression equation is

$$Y_i \approx \beta_0 + \beta_1 \text{Income}_i + \beta_2 \text{Gender}_i = \begin{cases} \beta_0 + \beta_1 \text{Income}_i & \text{if male} \\ \beta_0 + \beta_1 \text{Income}_i + \beta_2 & \text{if female} \end{cases}$$

- Interpretation of β_2 : The average extra balance each month that females have for given income level. Males

Regression coefficients

	Coefficient	Std Err	t-value	p-value
Constant	233.7663	39.5322	5.9133	0.0000
Income	0.0061	0.0006	10.4372	0.0000
Gender_Female	24.3108	40.8470	0.5952	0.5521

Two-way Qualitative: Other Coding Schemes

- There are different ways to code categorical variables.
- Two genders (male and female). Let

$$Gender_i = \begin{cases} -1 & \text{if male} \\ 1 & \text{if female} \end{cases}$$

- then the regression equation is

$$Y_i \approx \beta_0 + \beta_1 Income_i + \beta_2 Gender_i = \begin{cases} \beta_0 + \beta_1 Income_i - \beta_2, & \text{if male} \\ \beta_0 + \beta_1 Income_i + \beta_2, & \text{if female} \end{cases}$$

- Interpretation of β_2 : The average amount that females are above the average, for any given income level. β_2 is also the average amount that males are below the average, for any given income level.

Multi-way Qualitative: Other Coding Schemes

- How would you code if there were more than 2 classes of a categorical variable
 - Example: color = {Red, Green, or Blue}
- Design a coding scheme and then explain how to interpret the resulting coefficients of your coding variables

Other Issues Discussed

➤ Interaction terms

➤ Non-linear effects

➤ Multicollinearity

➤ Model Selection

Interaction

- The effect on Y of increasing X_1 depends on another data feature (e.g. X_2)
- Example
 - The effect on Salary (Y) when increasing Position (X_1) also depends on gender (X_2)
 - Maybe as they get promoted, Male salaries go up faster (or slower) than Females.
- Advertising example:
 - TV and radio advertising both increase sales.
 - Perhaps due to synergy, spending money on both of them may increase sales more than spending the same amount on one alone?

Interaction in advertising

$$Sales = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \beta_3 \times TV \times Radio$$

$$Sales = \beta_0 + (\beta_1 + \beta_3 \times Radio) \times TV + \beta_2 \times Radio$$

- Spending \$1 extra on TV increases average sales by $0.0191 + 0.0011 \times Radio$

Interaction Term
TV & Radio together

$$Sales = \beta_0 + (\beta_2 + \beta_3 \times TV) \times Radio + \beta_2 \times TV$$

- Spending \$1 extra on Radio increases average sales by $0.0289 + 0.0011 \times TV$

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	6.7502202	0.247871	27.23	<.0001*
TV	0.0191011	0.001504	12.70	<.0001*
Radio	0.0288603	0.008905	3.24	0.0014*
TV*Radio	0.0010865	5.242e-5	20.73	<.0001*

Should we consider interaction effects?

- Example: Relationship between job position and salary for men and women.
- Because we used a +1 / -1 dummy variable (gender), and did not include interaction terms, our model has forced the line for men and the line for women to be parallel.
- Parallel lines suggest that promotions have the same salary benefit for men as for women (even if that is not true in reality).
- Non-parallel line would suggest promotions affect men's and women's salaries differently

Parallel Regression Lines

Expanded Estimates

Nominal factors expanded to all levels

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	112.77039	1.454773	77.52	<.0001
Gender[female]	1.8600957	0.527424	3.53	0.0005
Gender[male]	-1.860096	0.527424	-3.53	0.0005
Position	6.0553559	0.280318	21.60	<.0001

Regression equation

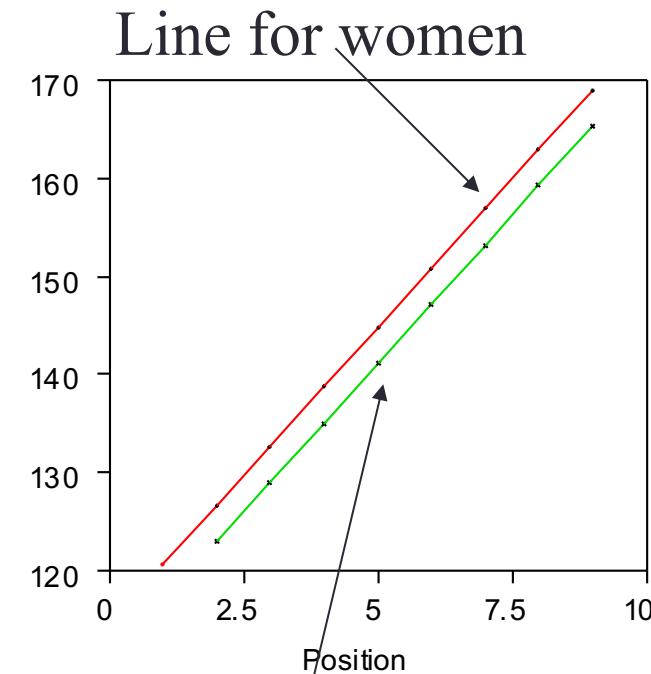
$$\text{female: salary} = 112.77 + 1.86 + 6.05 \times \text{position}$$

$$\text{males: salary} = 112.77 - 1.86 + 6.05 \times \text{position}$$

Different
intercepts

Same
slopes

Parallel lines have the same slope.
Dummy variables give lines different intercepts, but their slopes are still the same.

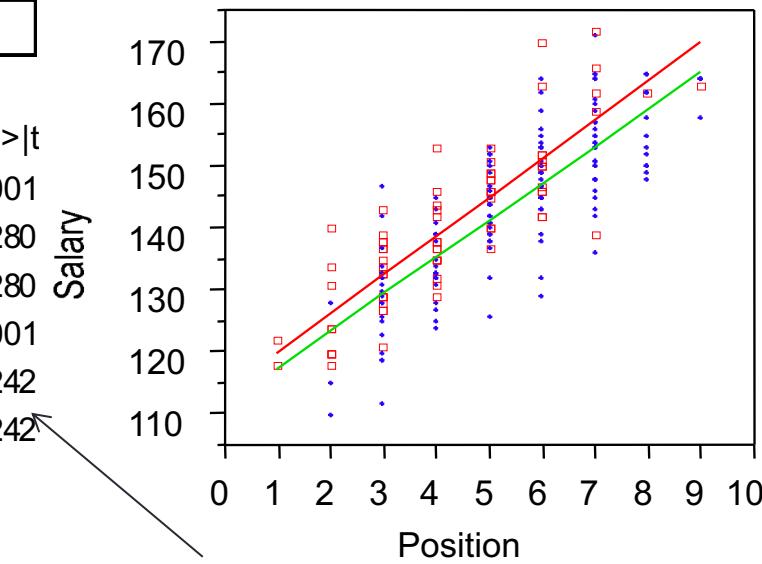


Should the Lines be Parallel?

Expanded Estimates

Nominal factors expanded to all levels

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	112.63081	1.484825	75.85	<.0001
Gender[female]	1.1792165	1.484825	0.79	0.4280
Gender[male]	-1.179216	1.484825	-0.79	0.4280
Position	6.1021378	0.296554	20.58	<.0001
Gender[female]*Position	0.1455111	0.296554	0.49	0.6242
Gender[male]*Position	-0.145511	0.296554	-0.49	0.6242



Interaction between gender and position

Interaction is not significant

Procedure: Add interaction terms. Check for significance of coefficients.

Significant coeffs in this example are Intercept and Position.

Since gender-position interactions are not significant, no reason to reject parallel lines as a reasonable assumption

Interpretation: income increase due to promotions does not depend on gender

Outline

- The Linear Regression Model
 - Least Squares Model Fitting
 - Measures of Fit
 - Inference in Regression
- Other Considerations in Regression Model
 - Qualitative Predictors
 - Interaction Terms
- Potential Fit Problems
- Linear vs. KNN Regression

Potential Fit Problems Worksheet

There are a number of possible problems that one may encounter when fitting the linear regression model. Fill out the second side of the handout per the instructions

1. Non-linearity of the data
2. Dependence of the error terms
3. Non-constant variance of error terms
4. Outliers
5. High leverage points
6. Collinearity

See Section 3.3.3 for more details.

Outline

- The Linear Regression Model
 - Least Squares Model Fitting
 - Measures of Fit
 - Inference in Regression
- Other Considerations in Regression Model
 - Qualitative Predictors
 - Interaction Terms
- Potential Fit Problems
- **Linear vs. KNN Regression**

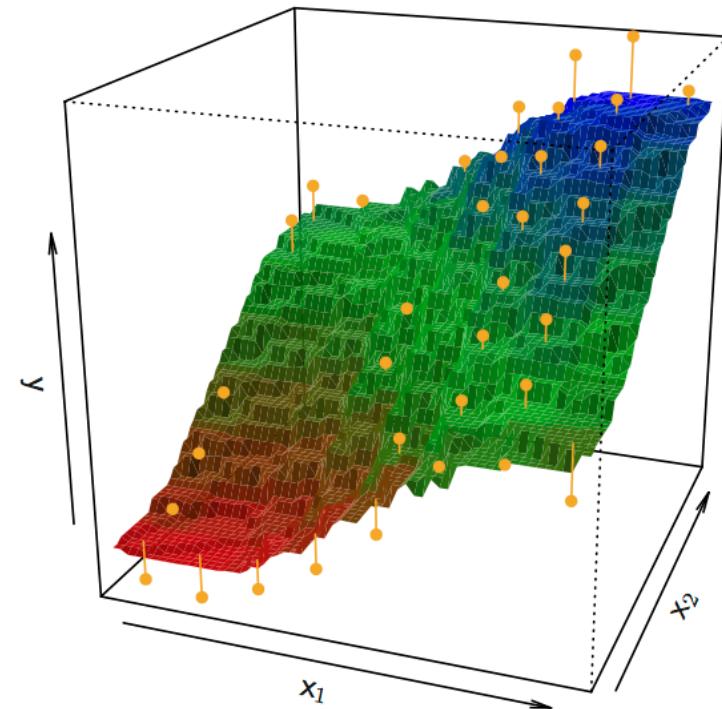
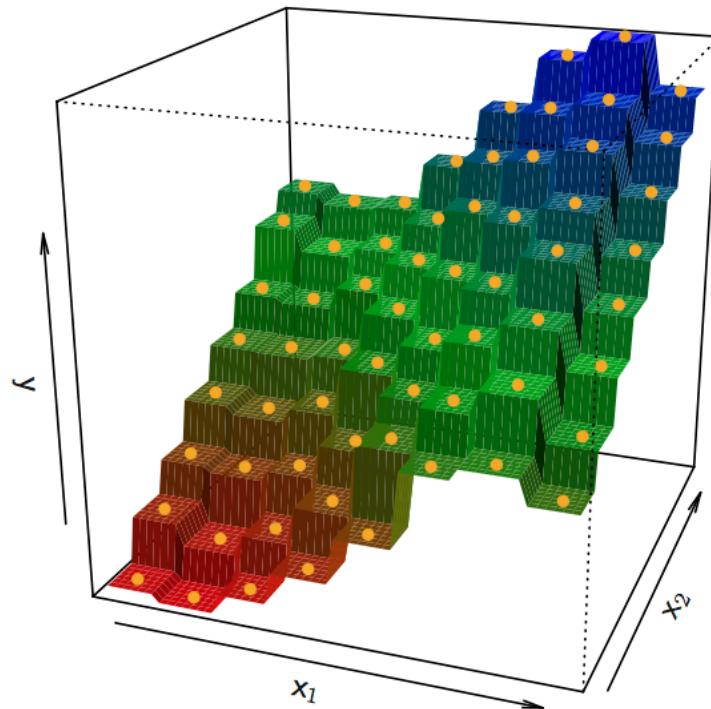
KNN Regression

- kNN Regression is similar to the kNN classifier.
- To predict Y for a given value of X, consider k closest points to X in training data and take the average of the responses. i.e.

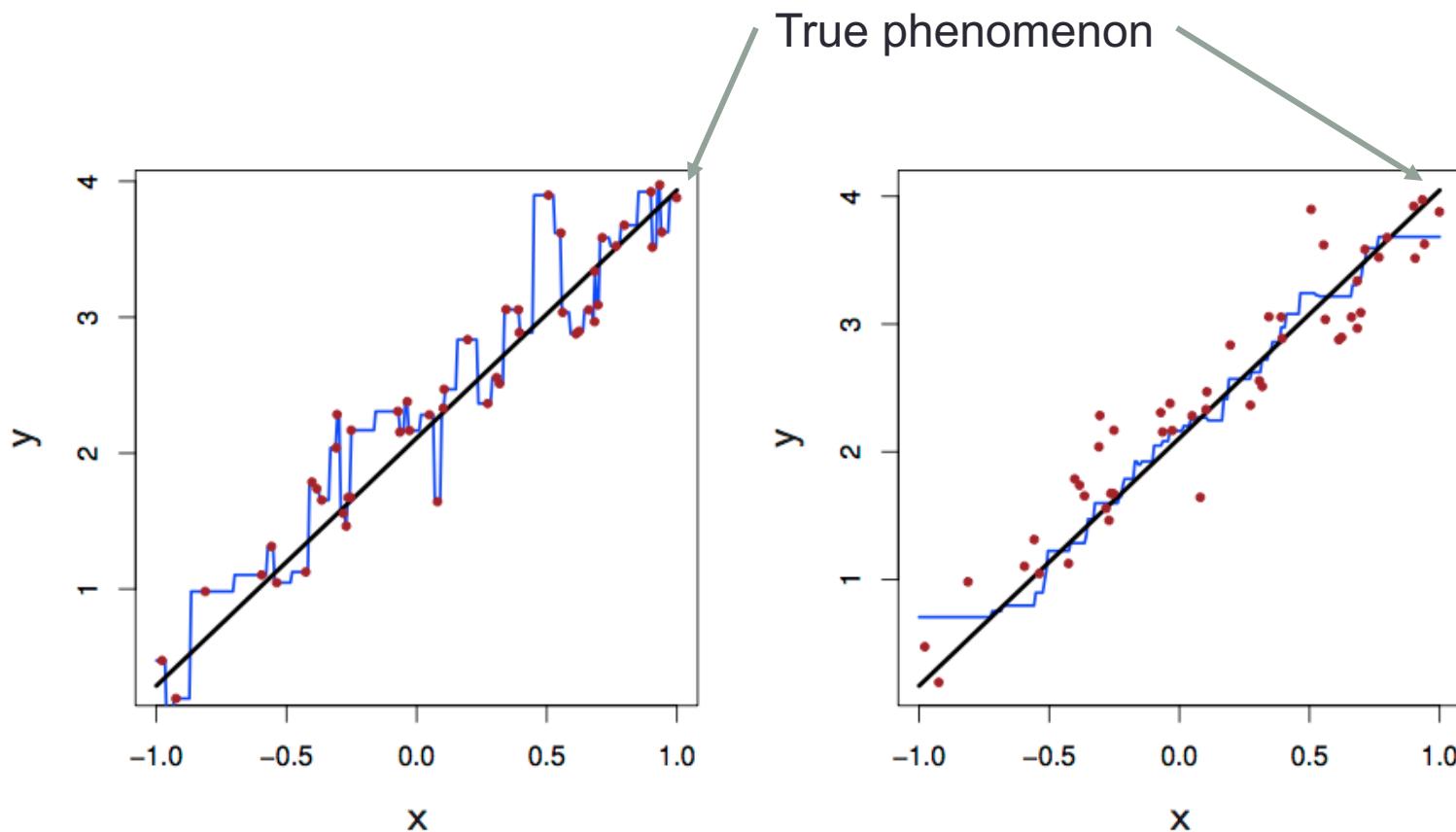
$$f(x) = \frac{1}{K} \sum_{x_i \in N_i} y_i$$

- If k is small kNN is much more flexible than linear regression.
- Is that better?

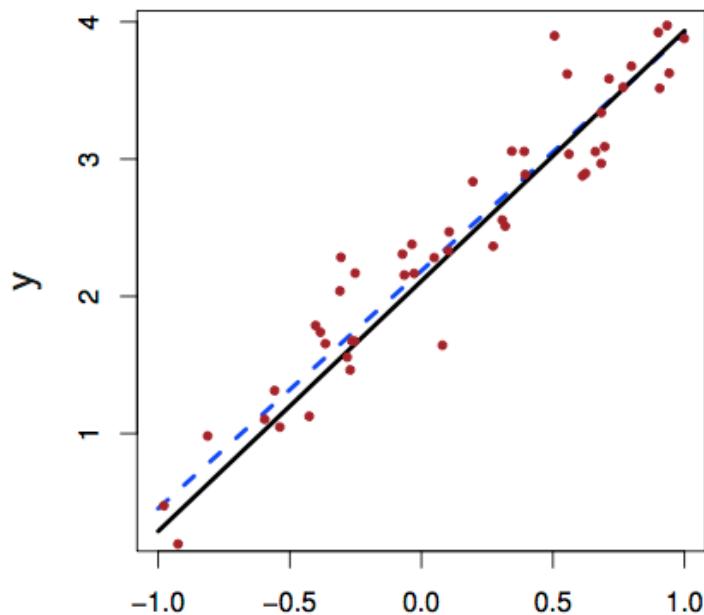
KNN Fits for $k = 1$ and $k = 9$



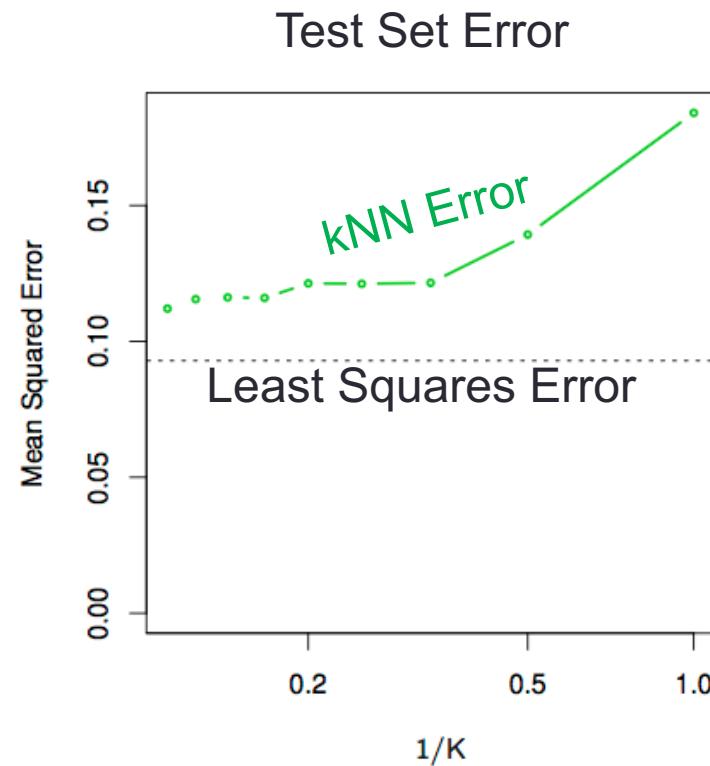
KNN Fits in One Dimension ($k = 1$ and $k = 9$)



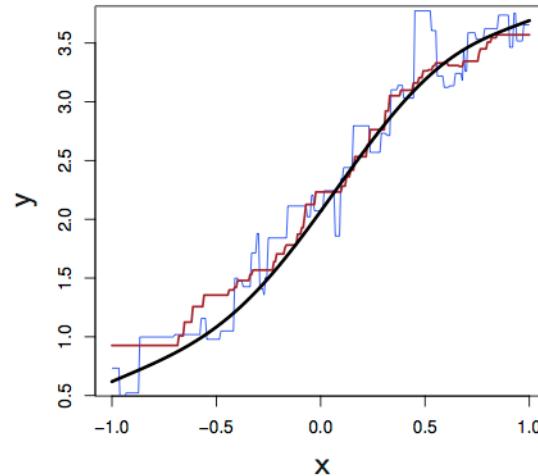
Linear Phenomenon: Linear Regression Fit vs. kNN



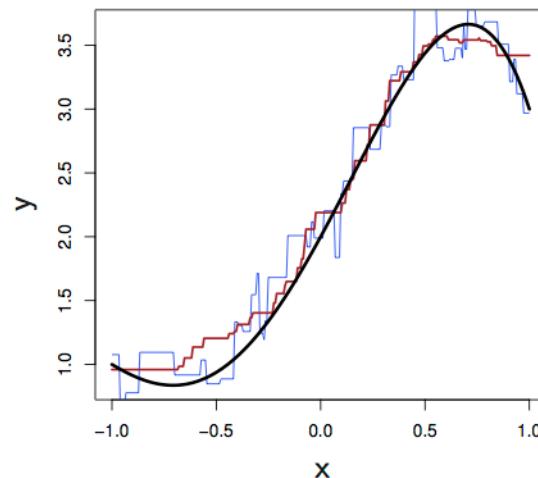
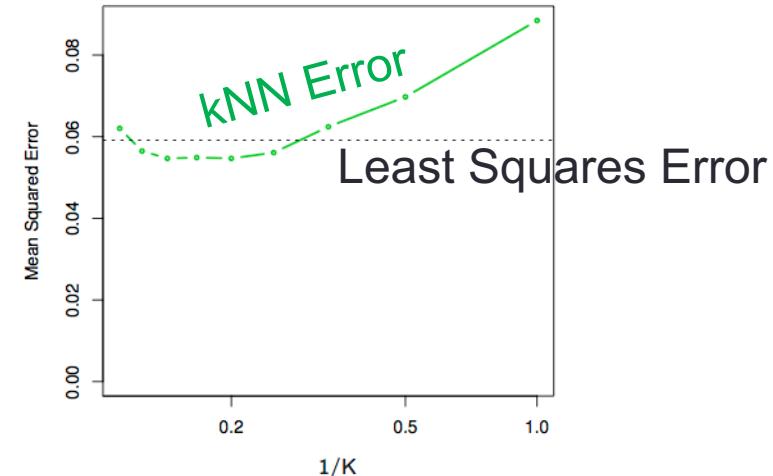
Concept Check:
Why is kNN getting worse as k goes from big to small?



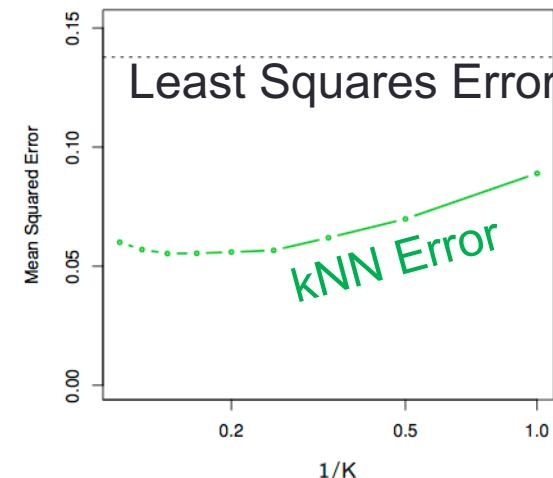
Nonlinear phenomenon: kNN vs. Linear Regression



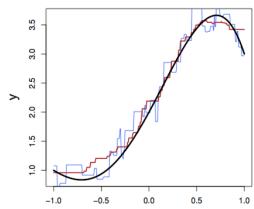
Test Set Error



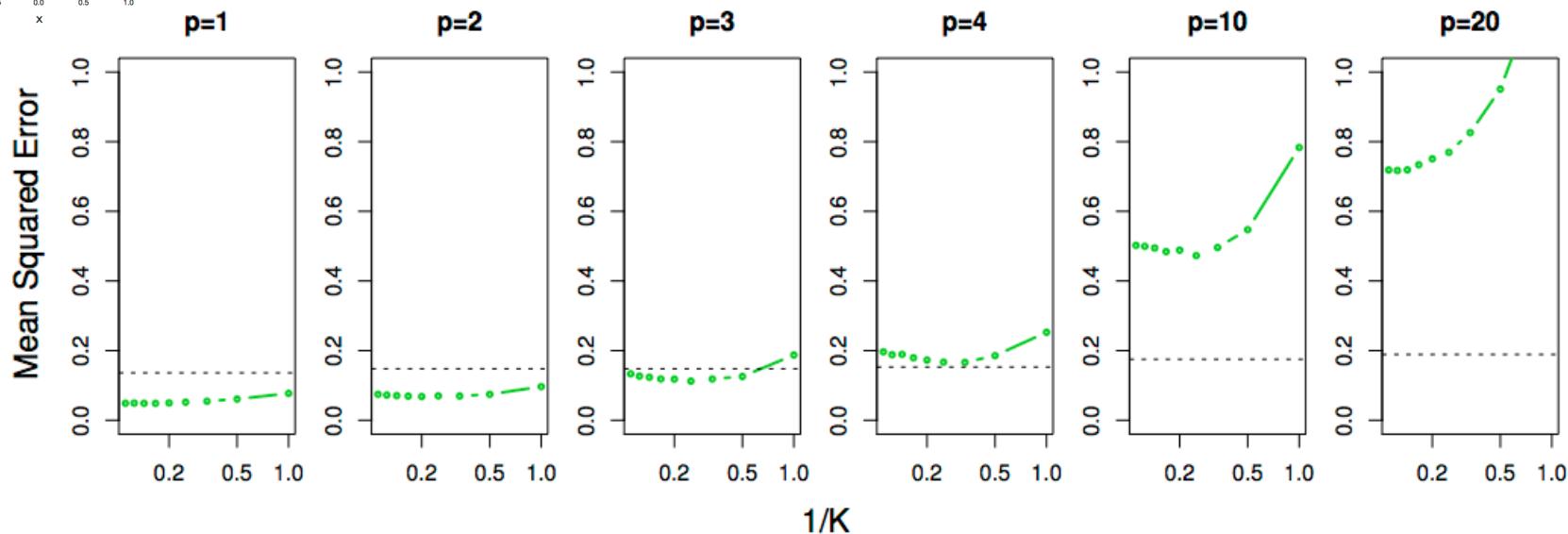
Test Set Error



kNN is Not So Good in High Dimensional Situations



one feature is relevant & nonlinear,
but additional features are irrelevant (noise)



Concept Check: Why does kNN perform ever worse than linear regression as we increase the number of (irrelevant) features?

This behavior is evidence of the phenomenon known as
“The Curse of Dimensionality”

CLASSIFICATION METHODS

Chapter 04 (part 01) – Logistic Regression

Outline

- Classification problem examples
 - What's wrong with using linear Regression?
- Simple Logistic Regression
 - Logistic Function
 - Interpreting the coefficients
 - Making Predictions
 - Adding Qualitative Predictors
- Multiple Logistic Regression

Classification

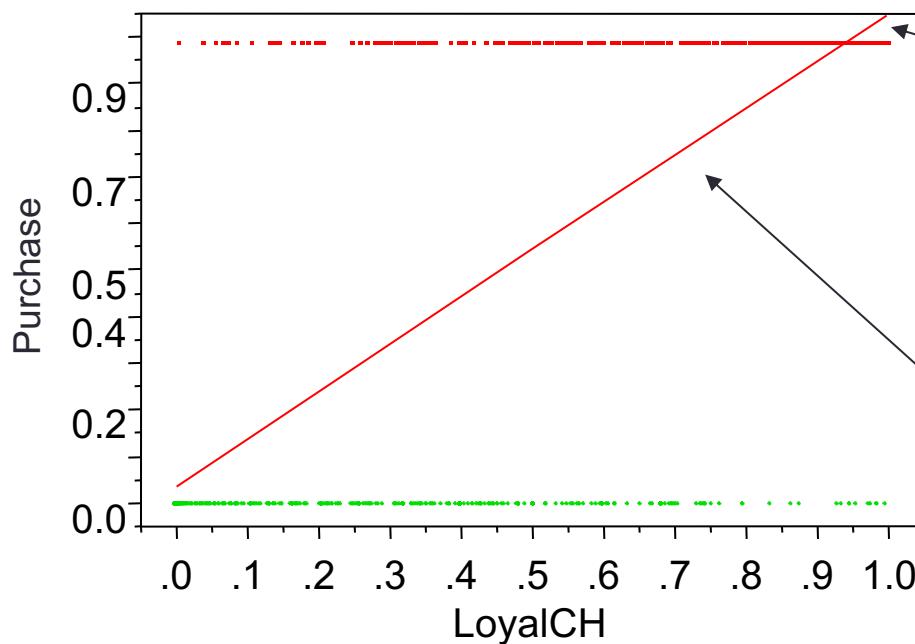
- Recall that in the regression problem our goal is to estimate the (real) number based on observed features
 - Output Type = Cardinal
- Classification: Estimating the category (class) to which something belongs
- Classes often have no direct underlying numerical relationship but we might use numbers as output values
 - Output Type = Nominal
 - Example: Tank = 1, Non-Tank = 0

OJ Classification Example

- Goal: predict what customers will buy:
Citrus Hill orange juice or Minute Maid orange juice
(based on their brand loyalty to various juice types)
- Y (Purchase CH) is categorical: 0 (no) or 1 (yes)
- X (LoyalCH) numerical (between 0 and 1) which specifies how loyal customers are to the Citrus Hill (CH) brand (0 = not loyal... 1 = completely loyal)
- Could we use Linear Regression when Y is categorical?

Why not use Linear Regression For category estimation?

- Regression forms a line...



How do we interpret values greater than 1?

How do we interpret values of Y between 0 and 1?

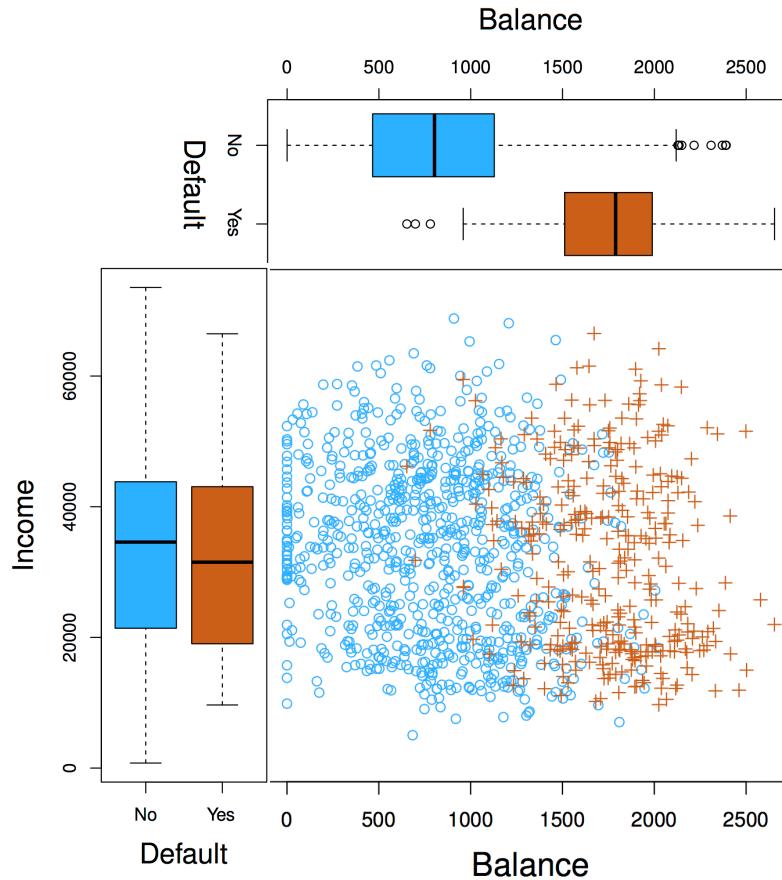
Problems with linear regression for Classification

- The regression line $\beta_0 + \beta_1 X$ can take on any value between negative and positive infinity
- In the orange juice classification problem, Y should only take on two possible values: 0 or 1.
- Therefore the regression line almost always predicts the wrong value for Y in classification problems

Classification Example 2: Credit Card Default Data

- We would like to be able to predict customers that are likely to default (not pay off their card)
- Possible X variables are:
 - Annual Income
 - Monthly credit card balance
- The Y variable (Default) is categorical: Yes or No
- How do we check the relationship between Y and X?

Exploring the (credit card) Default Dataset

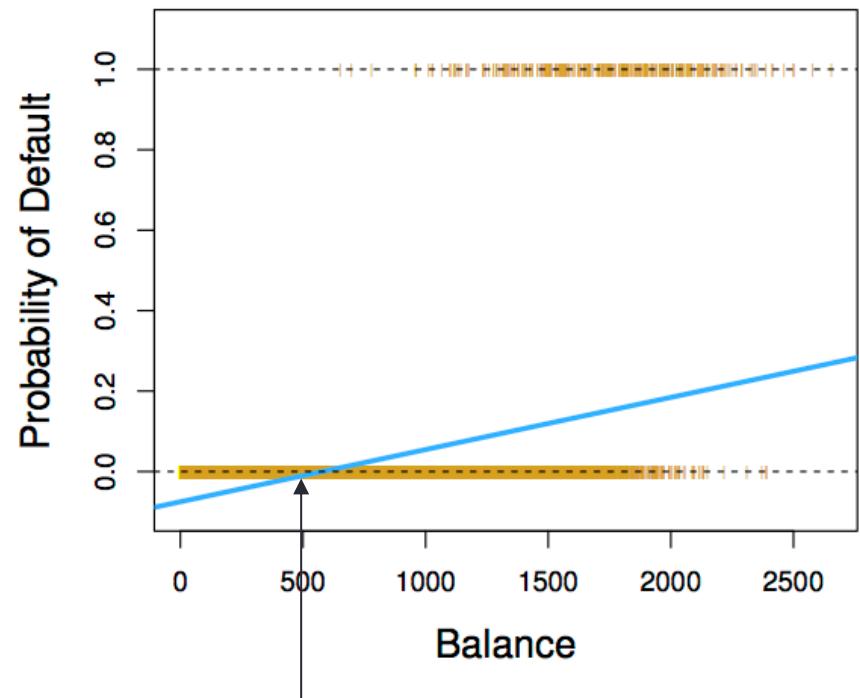


Concept Check:

Is there a meaningful relationship between Balance and Defaulting?
Is there a meaningful relationship between Income and defaulting?

Why not Linear Regression?

- For very low balances we predict a negative probability
- For high balances we predict a probability above 1

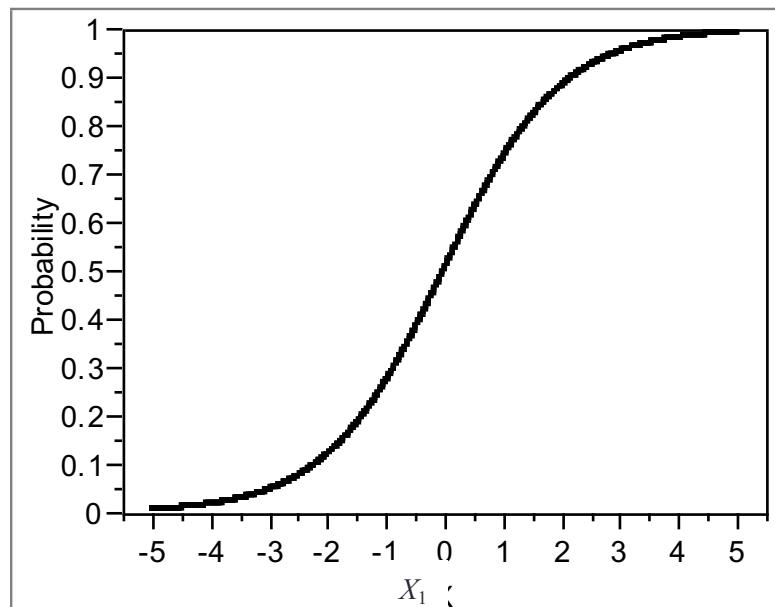


When $\text{Balance} < 500$, $\Pr(\text{default})$ is negative!

Solution: Use Logistic Function

- Instead of trying to predict Y, let's try to predict $P(Y = 1)$, i.e., the probability a customer buys Citrus Hill (CH) juice.
 - Model $P(Y = 1)$ with a function that gives outputs between 0 and 1.
 - Determine the Boolean answer by thresholding p
- **Logistic** function: **Logistic Regression**

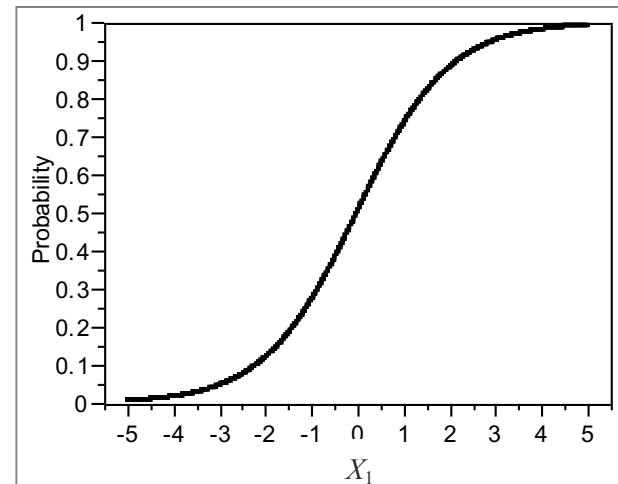
$$p = P(y = 1) = \frac{e^{\beta_0 + \beta_1 X_1}}{1 + e^{\beta_0 + \beta_1 X_1}}$$



Logistic Function: Thinking & Coding Practice

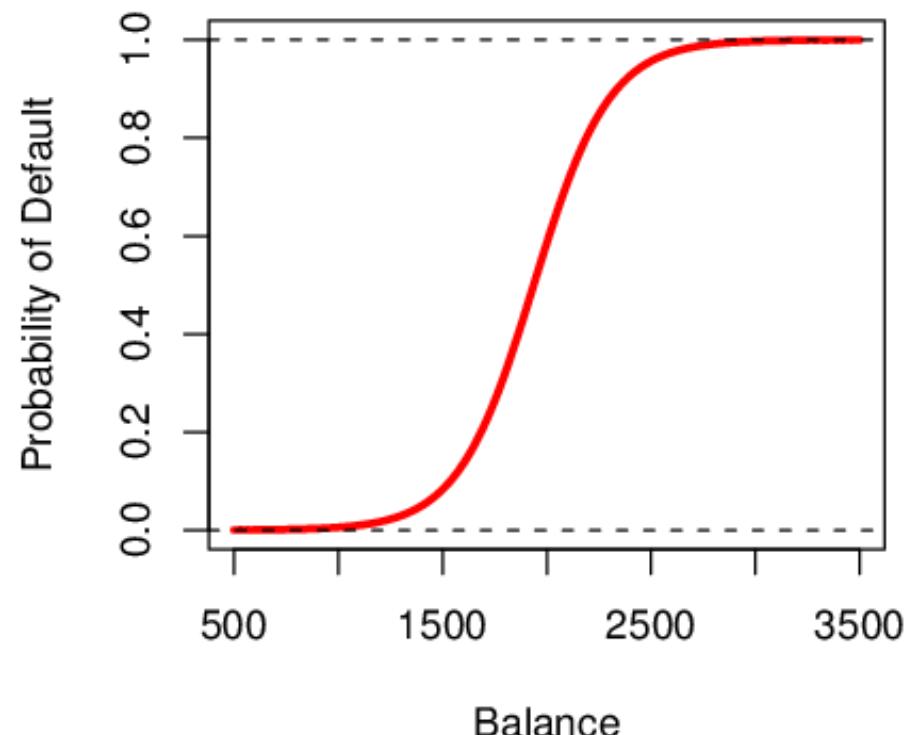
- What do you think happens to the shape of the curve as you alter the size and sign of β_0 ? β_1 ?
- Write a function which accepts β_0 , β_1 , and X and returns $P(Y=1)$
 - β_0 and β_1 are scalars
 - X is a $(n \times 1)$ matrix.
 - $P(Y=1)$ is a $(n \times 1)$ matrix
- Plot the results and see what happens when you alter the betas. Does it match your intuition?

$$p = P(y = 1) = \frac{e^{\beta_0 + \beta_1 X_1}}{1 + e^{\beta_0 + \beta_1 X_1}}$$



Logistic Function on Bank Default Data

- The probability of default is close to, but not less than zero for low balances.
- ... and close to, but not above 1 for high balances

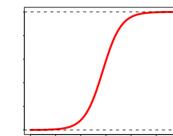


Interpreting β_1

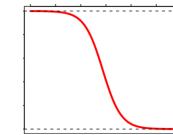
$$p = P(y = 1) = \frac{e^{\beta_0 + \beta_1 X_1}}{1 + e^{\beta_0 + \beta_1 X_1}}$$

- Interpreting what β_1 means is not very easy with logistic regression, simply because we are predicting $P(Y)$ and not Y
- If $\beta_1 = 0$, no relationship between Y and X

- If $\beta_1 > 0$, when X gets larger Y approaches 1



- If $\beta_1 < 0$, when X gets larger Y approaches 0



- But how much bigger or smaller depends on where we are on the slope

• Concept Check:

- How is the logistic line altered by changing β_0 ?

Logistic Regression Assessment: Are the coefficients significant?

- We still want to perform a hypothesis test to see whether we can be sure that are β_0 and β_1 significantly different from zero.
- We use a Z test instead of a T test (due to the process used to compute the coefficients), but that doesn't change the way we interpret the *p*-value
- Here the *p*-value for balance is very small, and $\hat{\beta}_1$ is positive, so we are sure that if the credit balance increases, then the probability of default will increase as well.

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Logistic Regression

Making Predictions

- Suppose an individual has an average balance of \$1000.
What is their probability of default?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 1000}}{1 + e^{-10.6513 + 0.0055 \times 1000}} = 0.00576$$

- The predicted probability of default for an individual with a balance of \$1000 is less than 1%.
- For a balance of \$2000, the probability is much higher, and equals to 0.586 (58.6%).

Logistic Regression

Encoding Qualitative Predictors

- We can predict if an individual will default by checking if she is a student or not. Thus we can use a qualitative variable “Student” coded as (Student = 1, Non-student =0).
- $\hat{\beta}_1$ is positive: This indicates students tend to have higher default probabilities than non-students

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student [Yes]	0.4049	0.1150	3.52	0.0004

$$\widehat{Pr}(\text{default=Yes} | \text{student=Yes}) = \frac{e^{-3.5041+0.4049 \times 1}}{1 + e^{-3.5041+0.4049 \times 1}} = 0.0431,$$

$$\widehat{Pr}(\text{default=Yes} | \text{student=No}) = \frac{e^{-3.5041+0.4049 \times 0}}{1 + e^{-3.5041+0.4049 \times 0}} = 0.0292.$$

Multiple Feature Logistic Regression

- We can fit multiple logistic regression coefficients

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p}}.$$

Multiple Feature Logistic Regression

Credit Card Default Data

- Predict Default using:
 - Balance (quantitative)
 - Income (quantitative)
 - Student (qualitative)

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student [Yes]	-0.6468	0.2362	-2.74	0.0062

Making Predictions with multiple-feature Logistic Regression

- A student with a credit card balance of \$1,500 and an income of \$40,000 has an estimated probability of default

$$\hat{p}(X) = \frac{e^{-10.869 + 0.00574 \times 1500 + 0.003 \times 40 - 0.6468 \times 1}}{1 + e^{-10.869 + 0.00574 \times 1500 + 0.003 \times 40 - 0.6468 \times 1}} = 0.058.$$

Interpreting multiple-feature Logistic Regression

Explain what happened here...

- The sign of the student coefficient changes when adding more features – Why?

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student [Yes]	0.4049	0.1150	3.52	0.0004

Positive



	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student [Yes]	-0.6468	0.2362	-2.74	0.0062

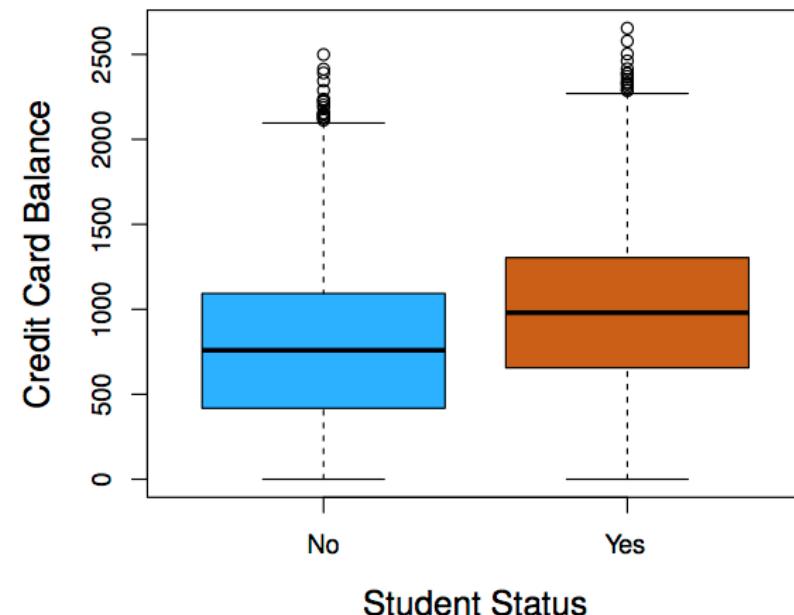
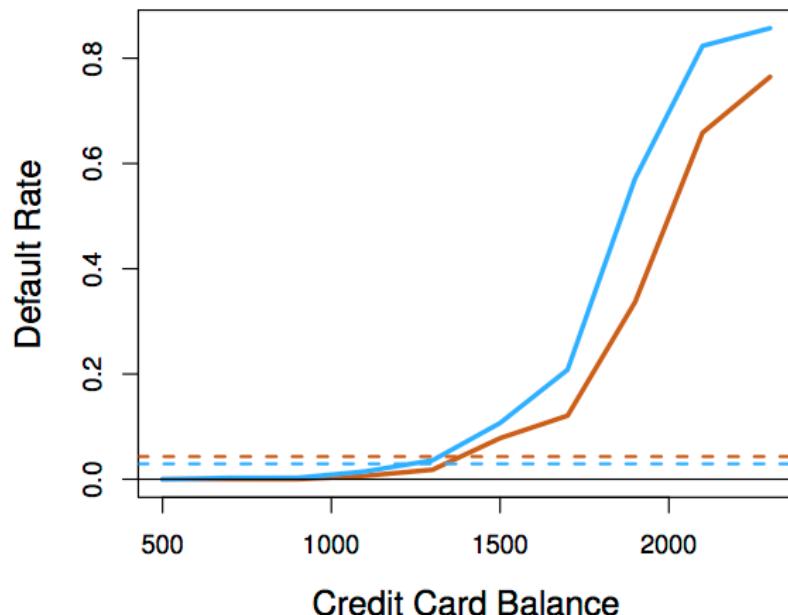
Negative



Interpreting multiple-feature Logistic Regression

To whom should credit be offered?

- A student (orange) is risker than non students (blue) *if no information about the credit card balance is available*



- However, for two individuals *with the same credit card balance*, *the student is less risky* than a non student

CLASSIFICATION METHODS

Chapter 04 (part 02)

LINEAR DISCRIMINANT ANALYSIS (LDA) & QUADRATIC DISCRIMINANT ANALYSIS (QDA)

Outline

- Overview of LDA
- Why not Logistic Regression?
- Estimating Bayes' Classifier
- LDA formulation
- Alternative LDA formulation
- 2-class performance measures
- Overview of QDA
- Comparison between LDA and QDA

Linear Discriminant Analysis

- Goal: Classify observations
 - Will a consumer buy a product or not?
 - Will a customer be satisfied or not?
 - Which candidate will a voter vote for?
- LDA Key intuition:
 - Represent each class as a simple distribution with parameters
 - Predict the class of a new observation by which class distribution has the highest probability at that observation's feature values

Assumptions of LDA

- The observations are an unbiased random sample (*i.i.d.*) of the population
- Each predictor variable is normally distributed
- All classes share common (co)variance parameters

Why not Logistic Regression?

- Logistic regression parameter values are unstable when the classes are well separated
 - **Work on in-class Problem #1**
- In the case where n is small, and the distribution of predictors X is approximately normal, then LDA is more stable than Logistic Regression
- LDA is also more popular than logistic regression when there are more than two response classes

Bayes' Classifier

- Bayes' classifier is the golden standard. Unfortunately, it is usually not determinable unless we are using synthetic data from a known distribution
- **Concept check: What is the property associated with data points along the Bayes Classification Boundary?**
- So far, we have *estimated* Bayes classifier with two methods:
 - KNN classifier
 - Logistic Regression

Estimating Bayes' Classifier

- With Logistic Regression we modeled the probability of Y being from the k^{th} class as

$$p(X) = \Pr(Y = k \mid X = x) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- However, Bayes' Theorem states for a K -class problem,

$$p_k(X) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)}$$

π_k : Prior Probability of coming from class k

$f_k(x)$: Unknown density function for x given that x is an observation from class k (we can choose this function depending on our model)

Idea: Model classes using distributions, then use Bayes Theorem to make classification decisions

Bayes requires estimating π_k and $f_k(x)$

- We need to estimate π_k and $f_k(x)$ to compute $p(x)$

$$p_k(X) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)}$$

- The most common model for $f_k(x)$ is the Normal Density

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)$$

- Using the normal density, we only need to estimate three parameters to compute $p(x)$:

$$\mu_k \quad \sigma_k^2 \quad \pi_k$$

Use Training Data set for Estimation

- The mean $\hat{\mu}_k$ could be estimated by the feature-wise average of all training observations from the k^{th} class.
- The variance $\hat{\sigma}^2$ could be estimated as the weighted average of variances of all K classes. In LDA we make the assumption that the variances for each class are equal. $\sigma_k^2 = \sigma^2$
- Estimate, $\hat{\pi}_k$ as the proportion of the training observations that belong to the k^{th} class.

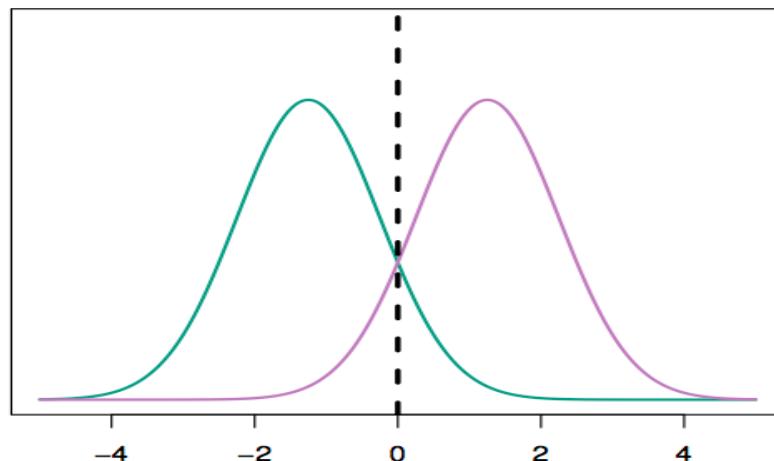
$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^K \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

$$\hat{\pi}_k = \frac{n_k}{n}$$

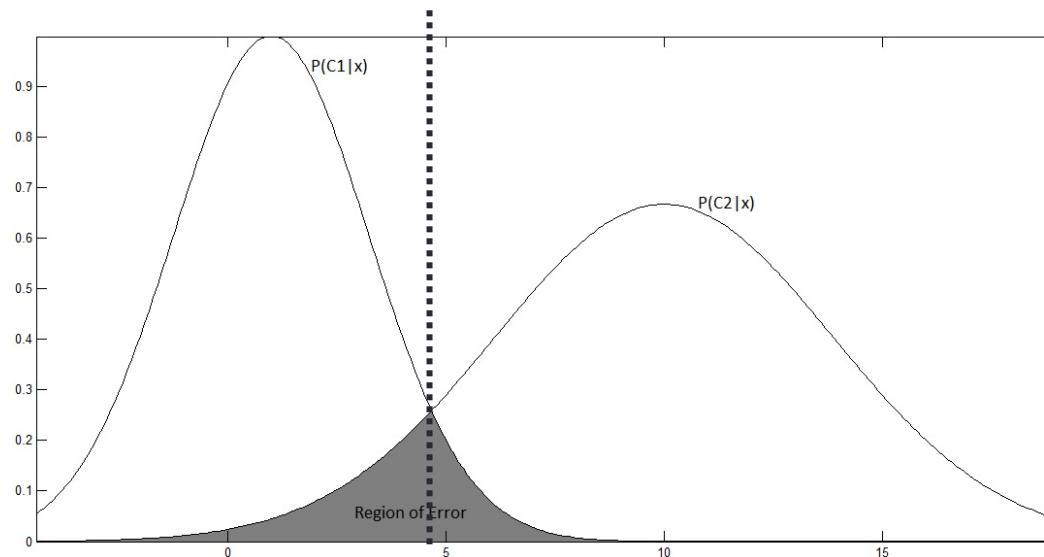
Bayes Example with One Predictor ($p = 1$)

- Suppose we have only one predictor ($p = 1$)
- Two normal density function $f_1(x)$ and $f_2(x)$, represent two distinct classes
- The two density functions overlap, so there is some uncertainty about the class to which an observation with an unknown class belongs
- The dashed vertical line represents Bayes' decision boundary



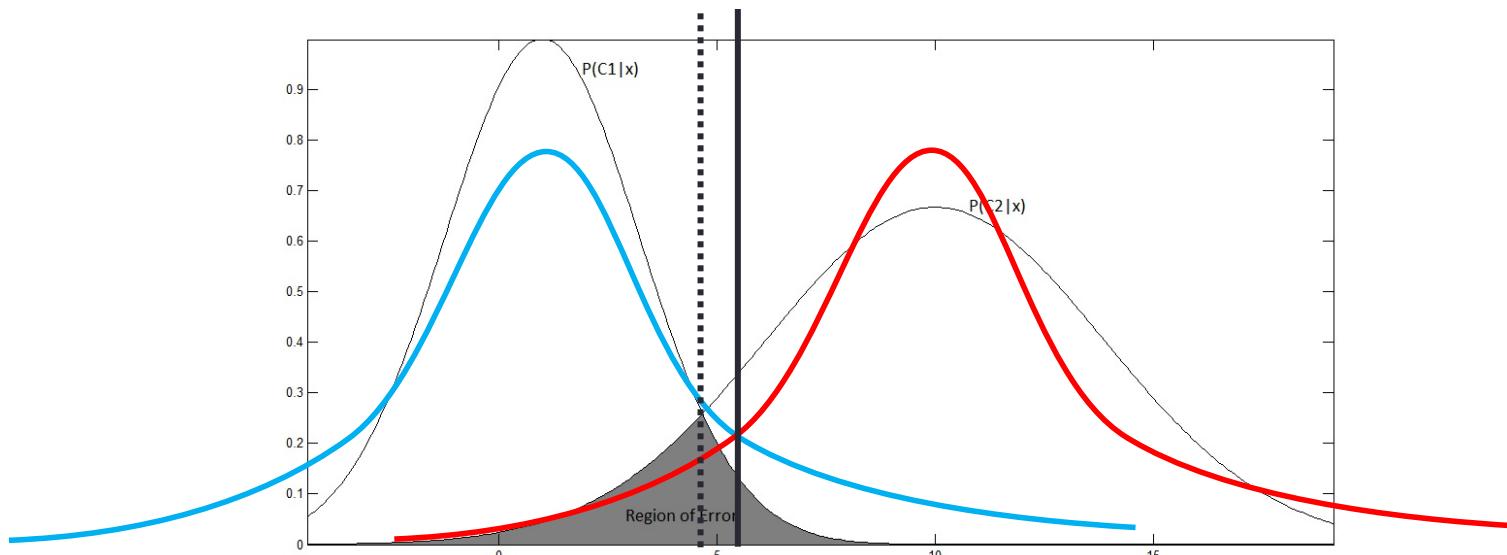
More complex example with one predictor (2-class, single feature)

- A discriminator is established at the point of equal probability...
- With a true Bayes classifier, this discriminator is not necessarily exactly between the class means

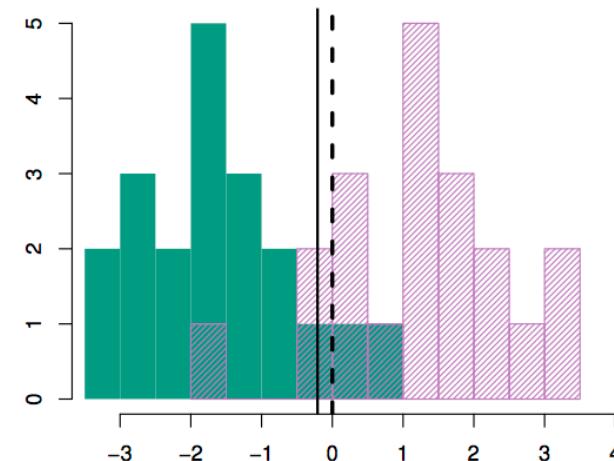
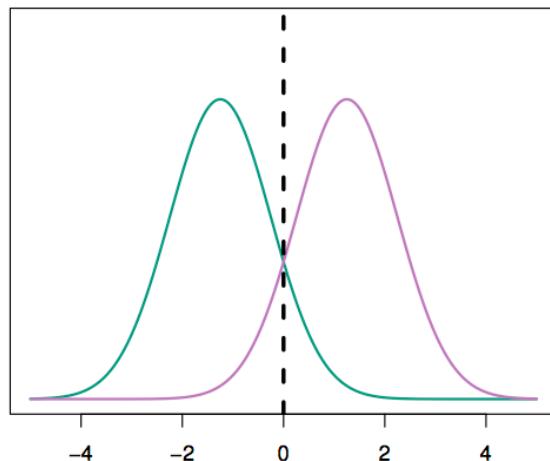


LDA intuition (2-class, single feature)

- LDA assumes that the observations in each class are Gaussian with the *same variance* but *different means*
- Model each class using
 - sample mean
 - (average) sample variance of each class
- Bayes classifier & LDA not necessarily equal



- Differences between Bayes and LDA performance are also due to sampling issues which are used to estimate class means and variances
 - 20 observations were drawn from each of the two classes
 - The dashed vertical line is the Bayes' decision boundary
 - The solid vertical line is the LDA decision boundary
 - Bayes' error rate: 10.6%
 - LDA error rate: 11.1%



Apply LDA

- LDA assumes that each class has a normal distribution with one mean per class but the same variance for every class $\hat{\mu}_k \quad \hat{\sigma}$
- The key variables are estimated from the training data

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i \quad \hat{\pi}_k = \frac{n_k}{n} \quad \hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^K \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

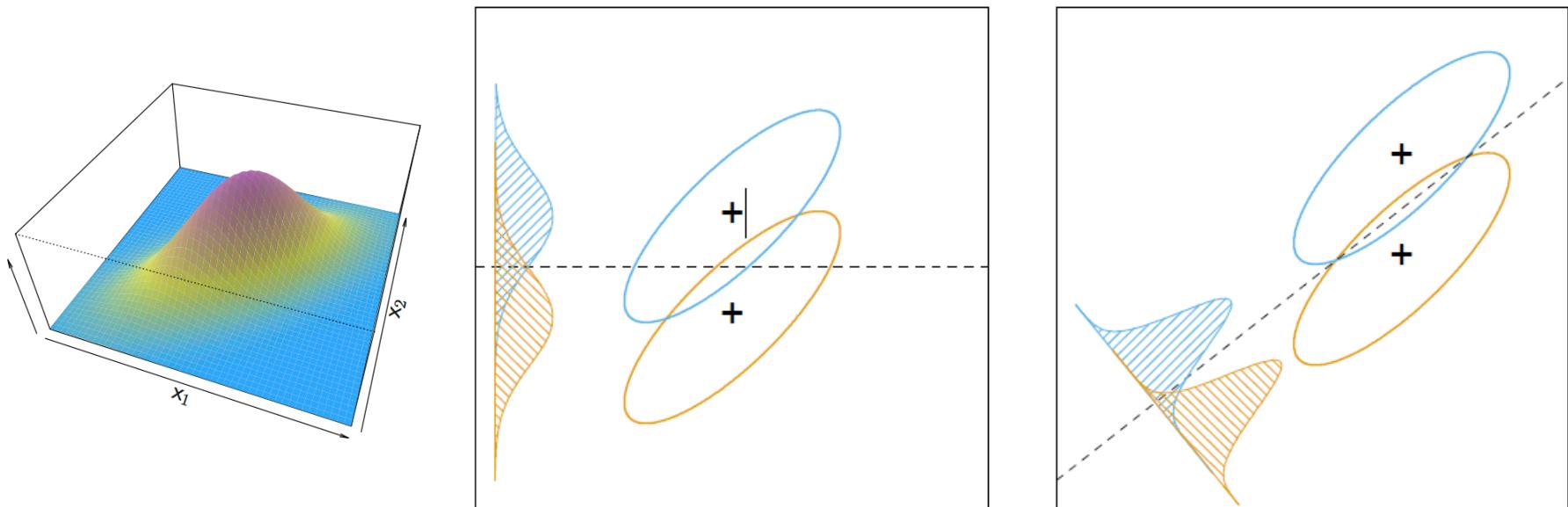
$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2\right)$$

- Bayes' theorem is used to compute p_k and the observation is assigned to the class with the maximum probability among all K probabilities

$$p_k(X) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)}$$

LDA intuition (more than 1 feature)

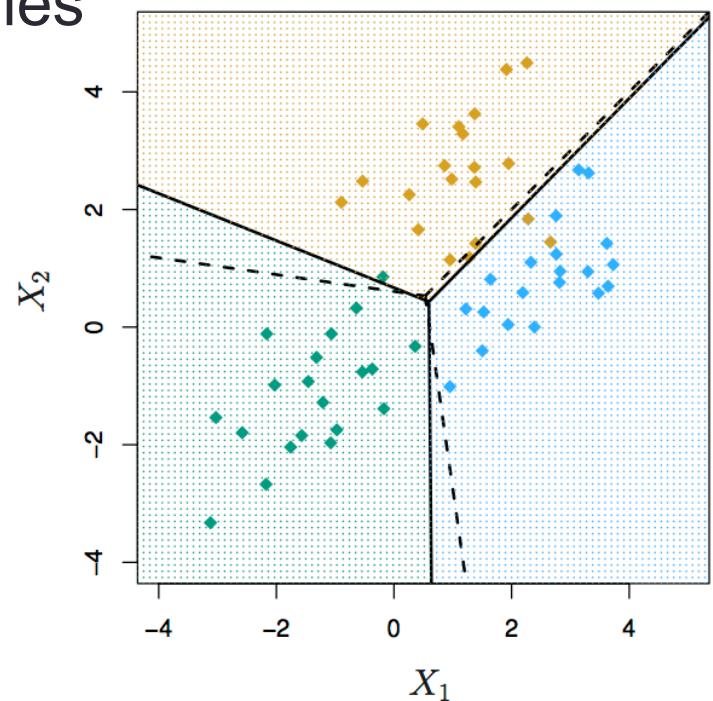
- If X is multidimensional ($p > 1$), we use exactly the same approach except the density function $f(x)$ is modeled using the multivariate normal density
- Need to find the direction for which a projection into fewer dimensions yields the most information for discrimination of the LDA (Bayes-like) classifier using Covariance



Elements of Statistical Learning – Figure 4.9

Multiclass LDA

- Three classes & Two predictors ($p = 2$)
- 20 observations were generated from each class
- The dashed lines are Bayes' boundaries
- The solid lines are LDA boundaries



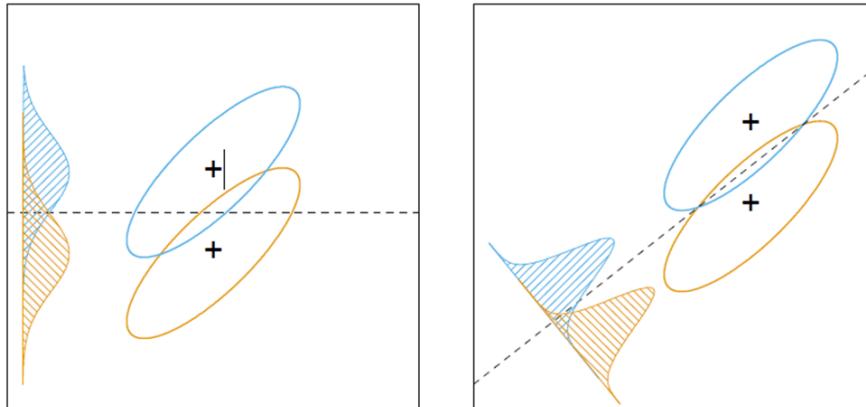
Alternative LDA formulation

- LDA involves the determination of linear equation (just like linear regression) that will predict which **class** the case belongs to.

$$D = w_0 + w_1 X_1 + w_2 X_2 + \dots + w_i X_i$$

- D : discriminant hyperplane
- w : discriminant coefficients
- X : variable
- w_0 : constant (default = 0)

Alternative LDA formulation



- Goal: discriminate between the different categories
- Choose the w 's in a way to
maximize the distance between the means
of different categories
- Features which help classify observations tend to have large w 's (weight)

$$D = w_0 + w_1 X_1 + w_2 X_2 + \dots + w_i X_i$$

Alternative LDA computation (2 class, 1 feature)

- Select the discriminator line D such that

$$D = w_1 X_1 + w_0$$

where

$$w_1 = \frac{\mu_{c1} - \mu_{c0}}{\sigma}$$

w_0 is default 0, or selectable to maximize training performance

Thus, select class {0,1} according to: $w_1 X_1 > w_0$

Alternative LDA computation (2 class, multi-feature)

- Select the discriminator line D such that

$$D = w^T X + w_0$$

where

$$w = \frac{\mu_{c1} - \mu_{c0}}{\Sigma^{-1}}$$

Σ^{-1} is the inverse (shared) covariance matrix of the classes

w_0 is default 0, or selectable to maximize training performance

Thus, select class {0,1} according to: $w^T X > w_0$

Alternative LDA in practice

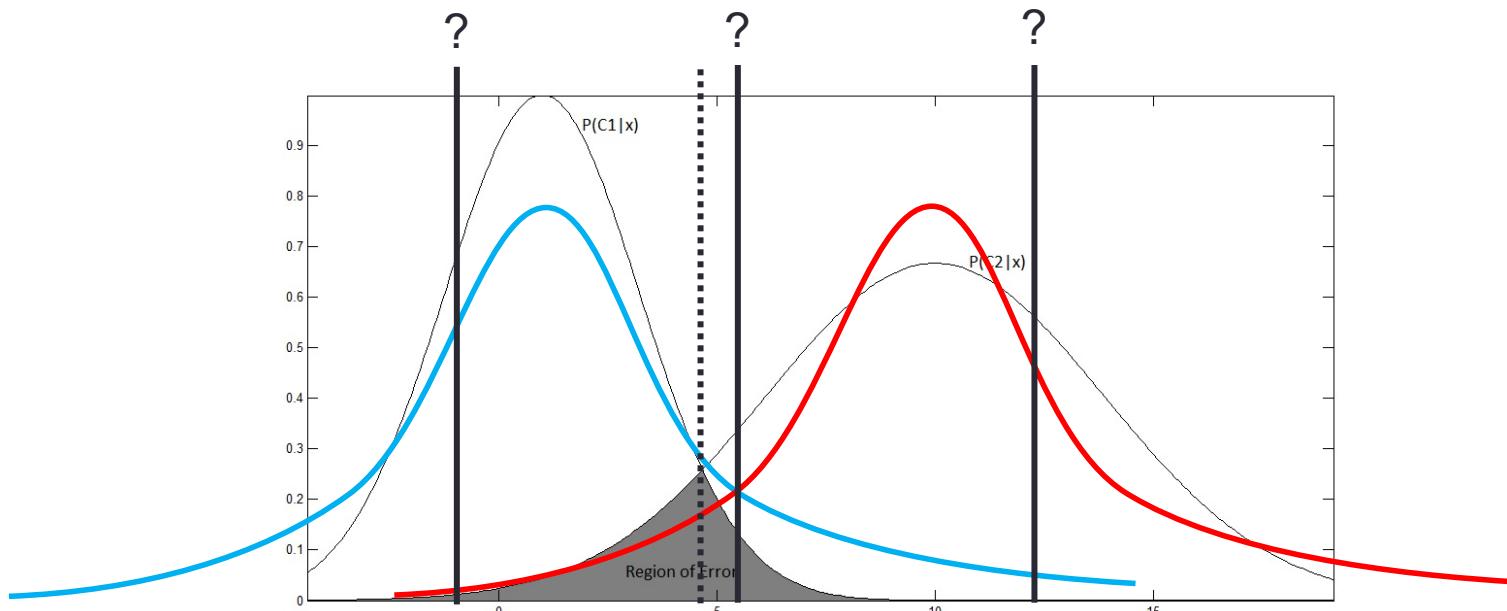
- In practice LDA is often combined with a feature reduction technique to reduce the effective dimensionality of the space
- When using LDA packages, select a smaller “components” parameter to enact dimensionality reduction
- `sklearn.discriminant_analysis.LinearDiscriminantAnalysis(solver='svd', shrinkage=None, priors=None, n_components=None, store_covariance=False, tol=0.0001) [source]`
 - `n_components` : int, optional
 - Number of components (< `n_classes` - 1) for dimensionality reduction.
- Further details in Elements of Statistical Learning

2-class Performance Measures

- Altering the decision boundary
- Confusion Matrix
- ROC

Altering the decision boundary

- Sometimes the (approximate) Bayes decision boundary may not be adequate for the business case
- **Audience participation:** Give an example of this and explain why



Classifier performance on Default Data (at $p(y|x) > 0.5$ as Threshold for Default)

		<i>True Default Status</i>		Total
		No	Yes	
<i>Predicted Default Status</i>	No	9644	252	9896
	Yes	23	81	104
Total	9667	333		10000

- LDA makes $252 + 23 = 275$ mistakes on 10000 predictions (2.75% misclassification error rate)
- But LDA miss-predicts $252/333 = 75.5\%$ of defaulters
- We shouldn't use 0.5 as threshold for predicting default if this will cost the bank a lot of money

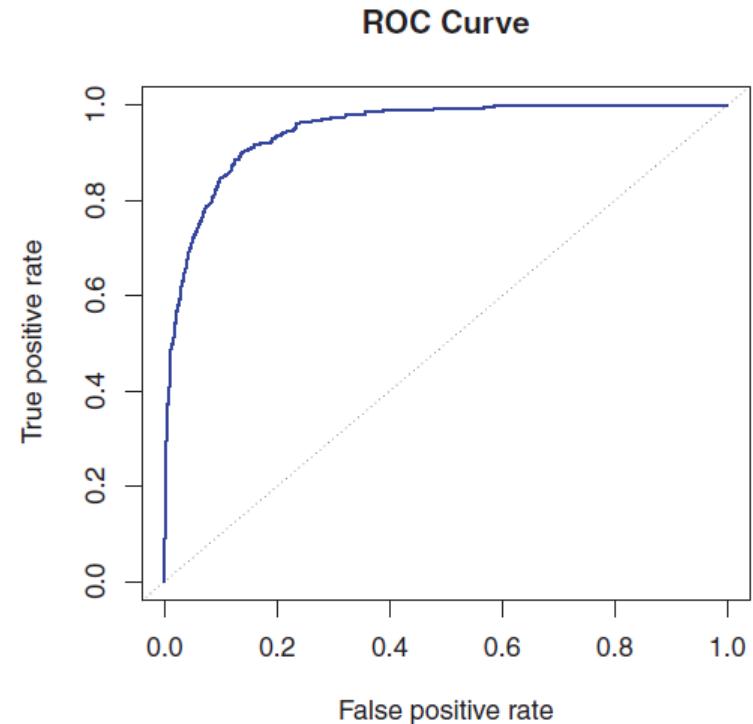
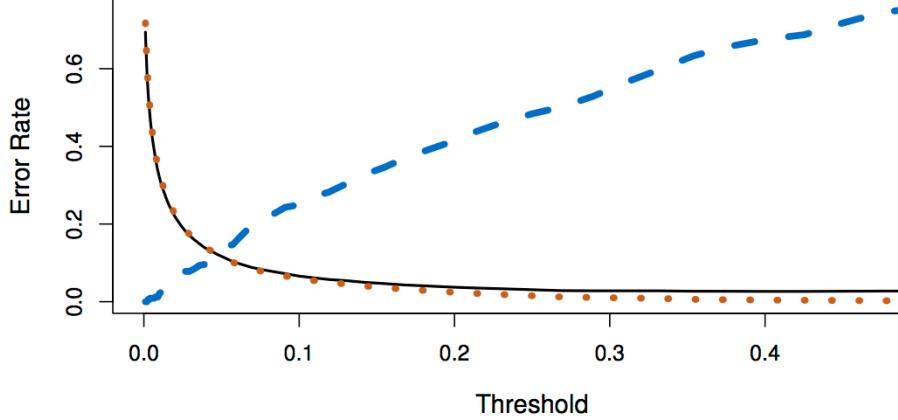
Use $p(y|x) > 0.2$ as Threshold for Default?

- Now the total number of mistakes is $235+138 = 373$
(3.73% misclassification error rate)
- But we only miss-predicted $138/333 = 41.4\%$ of defaulters

		<i>True Default Status</i>			
		No	Yes	Total	
<i>Predicted Default Status</i>	No	9432	138	9570	
	Yes	235	195	430	
	Total	9667	333	10000	

Threshold Values, Error Rates, ROC

- Black solid: overall error rate
- Blue dashed: Fraction of defaulters missed
- Orange dotted: non defaulters incorrectly classified

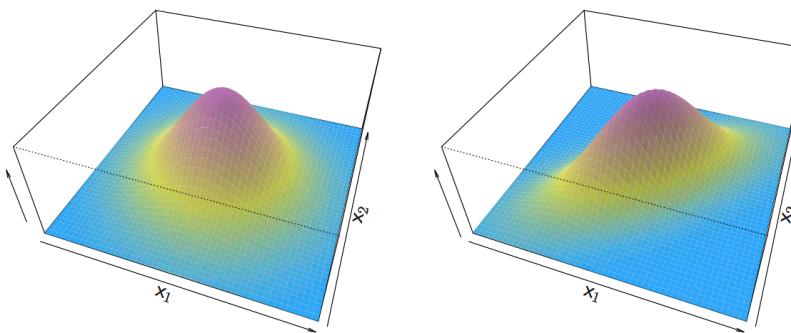


In Class Work – Classifier Performance and ROC

- Work on problem 2 now

Quadratic Discriminant Analysis (QDA)

- LDA assumed that every class has the same variance/ covariance
- However, LDA may perform poorly if this assumption is far from true
- QDA works identically as LDA except that it estimates separate variances/ covariance for each class

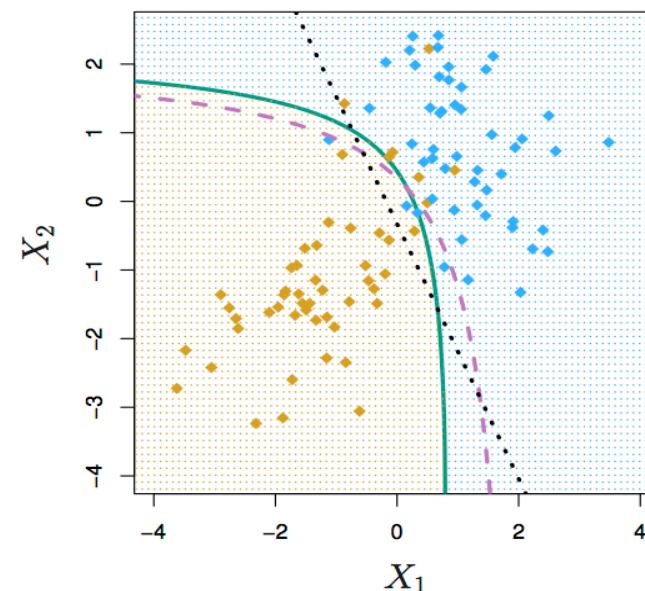
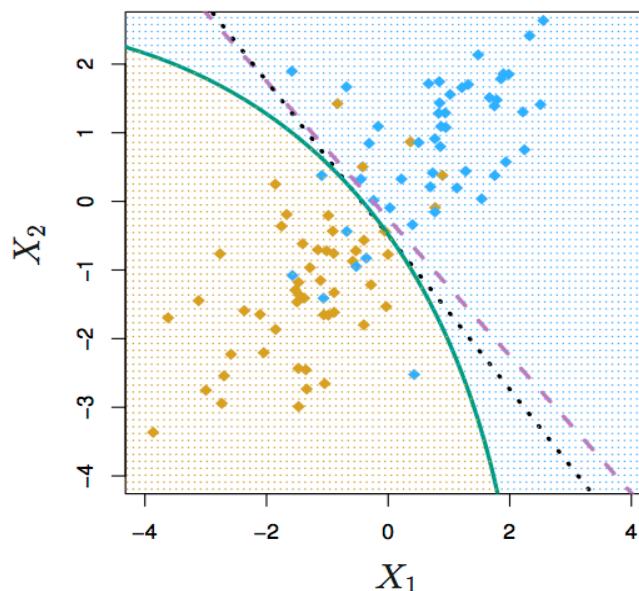


Which is better? LDA or QDA?

- Since QDA allows for different variances among classes, the resulting boundaries become quadratic
- Which approach is better: LDA or QDA?
 - QDA may work better when the variances are very different between classes and we have enough observations to accurately estimate the variances
 - LDA may work better when the variances are similar among classes or we don't have enough data to accurately estimate the true differences in the variances

Comparing LDA to QDA

- Black dotted: LDA boundary
- Purple dashed: Bayes' boundary
- Green solid: QDA boundary
- Left: variances of the classes are equal (LDA is better fit)
- Right: variances of the classes are not equal (QDA is better fit)



Comparison of Classification Methods

- KNN (Chapter 2)
- Logistic Regression (Chapter 4)
- LDA (Chapter 4)
- QDA (Chapter 4)

Logistic Regression vs. LDA

- Similarity: Both Logistic Regression and LDA produce linear boundaries
- Difference: LDA assumes that the observations are drawn from the normal distribution with common variance in each class, while logistic regression does not have this assumption.
 - LDA would do better than Logistic Regression if the assumption of normality holds,
 - otherwise logistic regression can outperform LDA

KNN vs. (LDA and Logistic Regression)

- KNN takes a completely different approach
- KNN is completely non-parametric: No assumptions are made about the shape of the decision boundary
- Advantage of KNN: We can expect KNN to dominate both LDA and Logistic Regression when the decision boundary is highly non-linear
- Disadvantage of KNN: KNN does not tell us which predictors are important (no table of coefficients)

QDA vs. (LDA, Logistic Regression, and KNN)

- QDA is a higher variance parametric model which offers a compromise in performance between non-parametric KNN method and linear methods such as LDA and logistic regression
- If the true decision boundary is:
 - Linear: LDA and Logistic outperforms
 - Moderately Non-linear: QDA outperforms
 - More complicated: KNN is superior

RESAMPLING METHODS

Chapter 05

Validation for Decisionmaking

- The Validation Set Approach
- Leave-One-Out Cross Validation
- K-fold Cross Validation
- Bias-Variance Trade-off for k-fold Cross Validation
- Cross Validation on Classification Problems

What are resampling methods?

- Tools that involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain more information about the fitted model
 - Model Assessment: estimate error rates on unseen data
 - Model Selection: select appropriate model hyperparameters (e.g. level of model flexibility)
- They are computationally expensive! But these days we have powerful computers
- Two resampling methods:
 - Cross Validation
 - Bootstrapping

Borghetti's “Golden Rule” of Performance Reporting

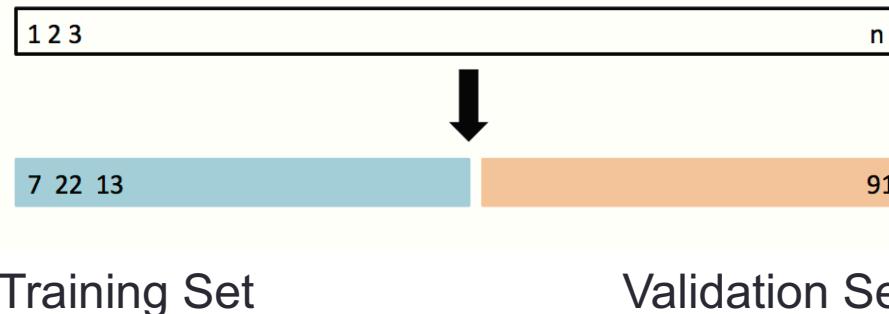
- You already know that:
 - Training sets are used to fit models, and training set accuracy may vary greatly from test set performance, so it is inappropriate to report model quality based on the training set performance
- Golden Rule: If you use an observation as part of your decision-making process, then you should NOT use it as part of a set of data you are using to *report* performance
 - Decision-making includes things like fitting a model, choosing hyperparameters, and selecting which model to use
- If you need to choose hyperparameters or select the best model, don't use the test-set data to make the decision!

Training v. Validation v. Test sets

- TEST SET: Used for performance prediction **only**. It estimates performance of a model on unseen data.
Sequester the test set before ML!
- NON-TEST-SET DATA:
 - Training Set: Used to “fit” parameterized models (e.g. find coefficients/weights)
 - Validation Set: Use when considering multiple models or making hyperparameter decisions
 - Estimate of each models quality from non-training data once the model has been fit on the training data
 - Used to make selection decisions (e.g. pick the best k in KNN; select whether LDA or QDA model works best)

5.1.1 Model Selection using The Validation Set Approach

- Goal: select the best model (e.g. LDA vs. QDA)
- If we have a large data set, randomly* split the non-test data into *training* and *validation* sets
- Use the training set to build each possible model (i.e. the different combinations of variables)
- Select the model that gave the lowest error rate when applied to the validation set



*be careful when randomizing selection from time-series or sequence-based data

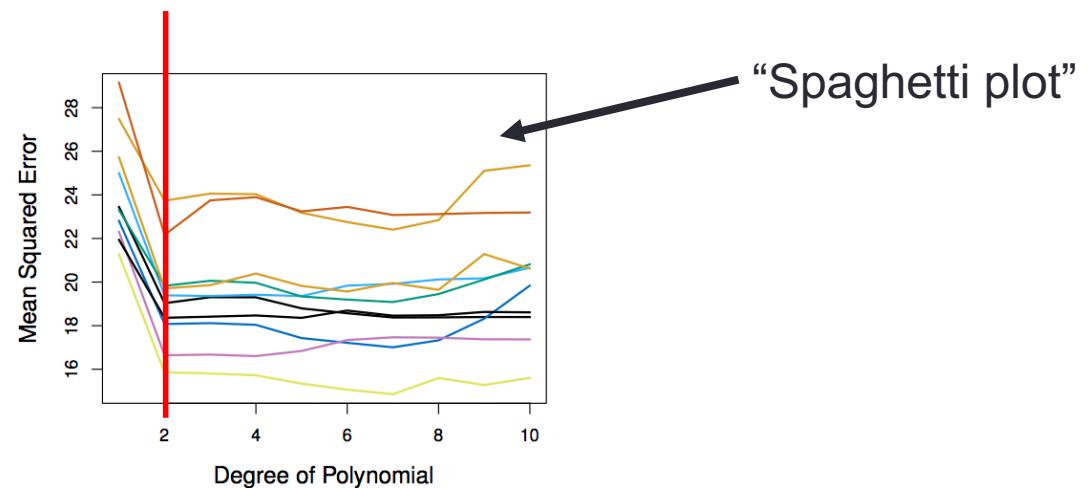
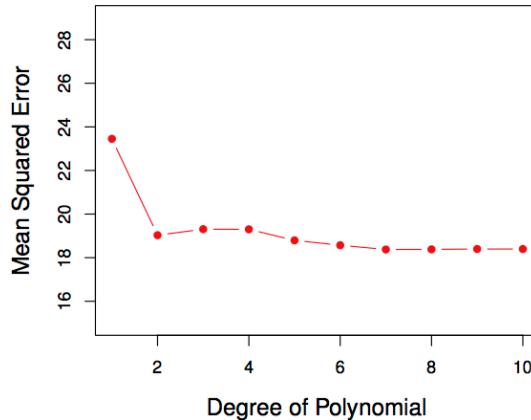
Model Selection using Validation Set

Example: Auto Data

- Suppose that we want to predict `mpg` from `horsepower`
- Goal: select the best of 10 possible models:
 - Model 1: features = horsepower (Linear only)
 - Model 2: features = horsepower + horsepower² (Linear + Quadratic)
 - ...
 - Model 10: features = horsepower + horsepower² +...+ horsepower¹⁰
- Which model gives a better fit?
 - Randomly split `Auto` data set into training (196 obs.) and validation set (196 obs.)
 - Fit both models using the training data set
 - Evaluate MSE on each model using the validation data set
 - Select model with the lowest validation MSE
- **Question: If we use the validation set to determine which model is the better fit, is the MSE of the winning validation set a good estimate for MSE on novel/unseen data?**

Validation Set Variability

- Model selection task: which order of polynomial model fits best (from polynomials with orders 1:10)
- Left: Validation error rate for a single (random) 50/50 split
- Right: Validation method repeated 10 times, each time the split is done randomly
- There is a lot of variability among the MSE's...



Validation Set Approach in summary

- Advantages:
 - Simple
 - Easy to implement
- Disadvantages:
 - The validation MSE of a single split can be highly variable
 - Only a subset of observations are used to fit the model (training data). Statistical methods tend to perform worse when trained on fewer observations

Cross Validation

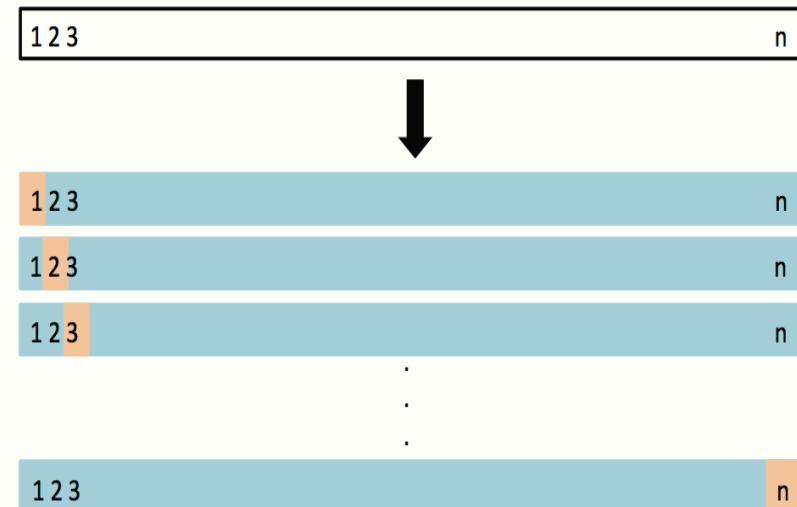
- Goal: reduce variability of the results of the validation set method
- Intuition: repeated resampling (*bootstrapping*) can yield better statistical properties on the estimate of a value
- Procedure:
 - Repeatedly conduct a train-then-evaluate process using different subsets of the data.
 - Estimate the performance as the mean of the (lower variance) performance estimates

Bootstrap (5.2)

- In class exercise

5.1.2 Leave-One-Out Cross Validation (LOOCV)

- This method is similar to the Validation Set Approach, but it tries to address the ValSet's disadvantages
- For each suggested model, do:
 - Split the data set of size n into
 - Training data set (blue) size: n - 1
 - Validation data set (beige) size: 1
 - Fit the model using the training data
 - Validate model using the validation data, and compute the corresponding MSE
 - Repeat this process n times
 - The MSE for a model is computed as follows:



$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n \text{MSE}_i$$

LOOCV vs. the Validation Set Approach

- LOOCV has less chance of statistical sample bias
 - We repeatedly fit the statistical learning method using training data that contains $n-1$ observations - almost all the data set is used for training
- LOOCV produces a single MSE
 - The validation set approach produces different MSE when applied repeatedly due to randomness in the splitting process, while performing LOOCV multiple times will always yield the same results, because we split based on 1 obs. each time
- LOOCV is computationally intensive (disadvantage)
 - We fit each model n times

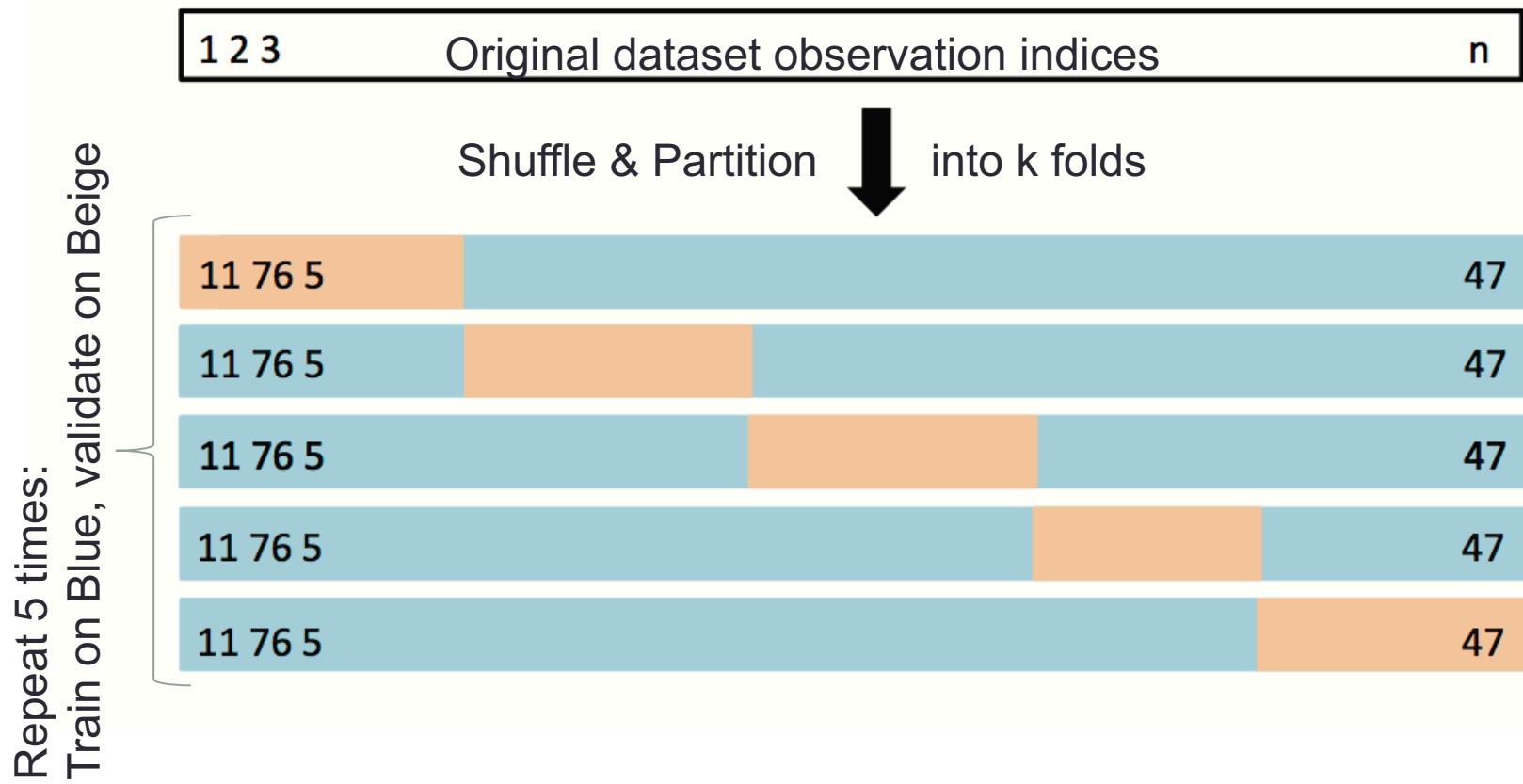
5.1.3 *k*-fold Cross Validation

- LOOCV is computationally intensive, and the validation set approach has high variability... is there a hybrid approach?
- ***k*-fold Cross Validation:**
- Randomly divide the data set into k different partitions (e.g. $k = 5$, or $k = 10$) known as “folds”
- Repeat a train-validate process k times using those folds:
 - In the i^{th} iteration, we train using all of the folds *except* the i^{th} and we validate on the data from the i^{th} fold.
- By averaging the k different MSE’s we get an estimated error rate for unseen observations

$$\text{CV}_{(k)} = \frac{1}{k} \sum_{i=1}^k \text{MSE}_i$$

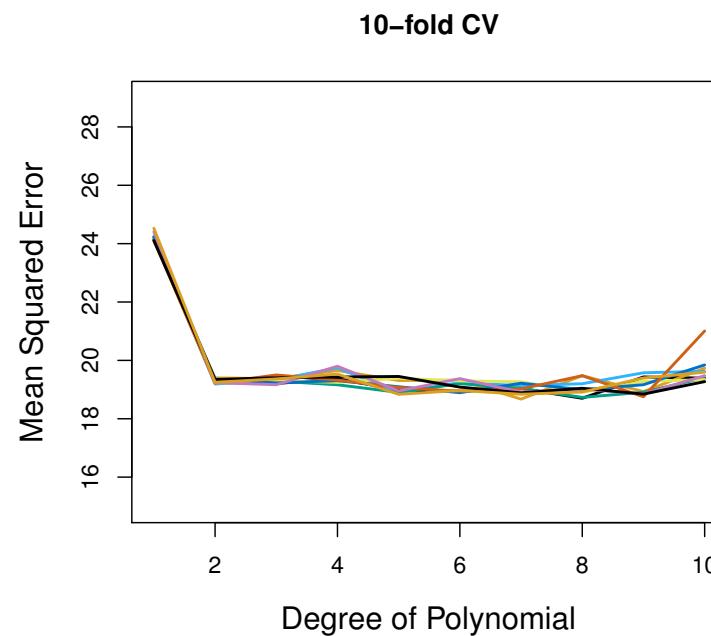
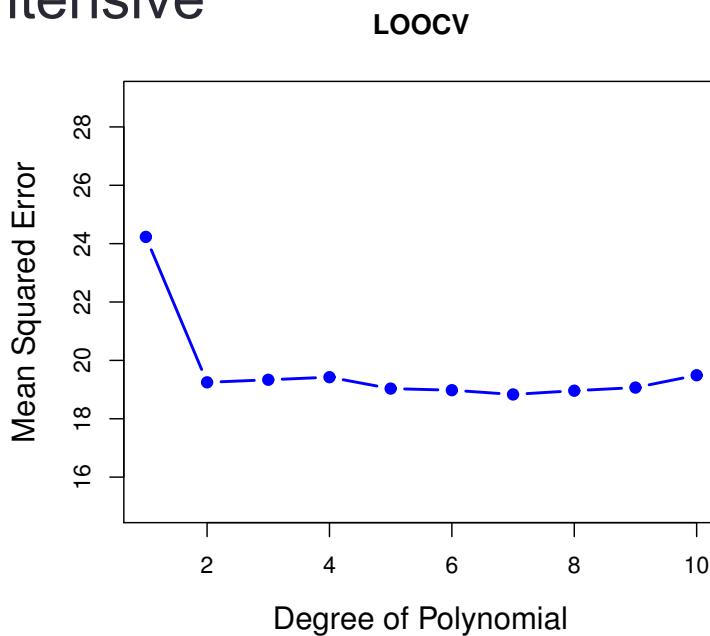
k-fold Cross Validation

Example: $k = 5$



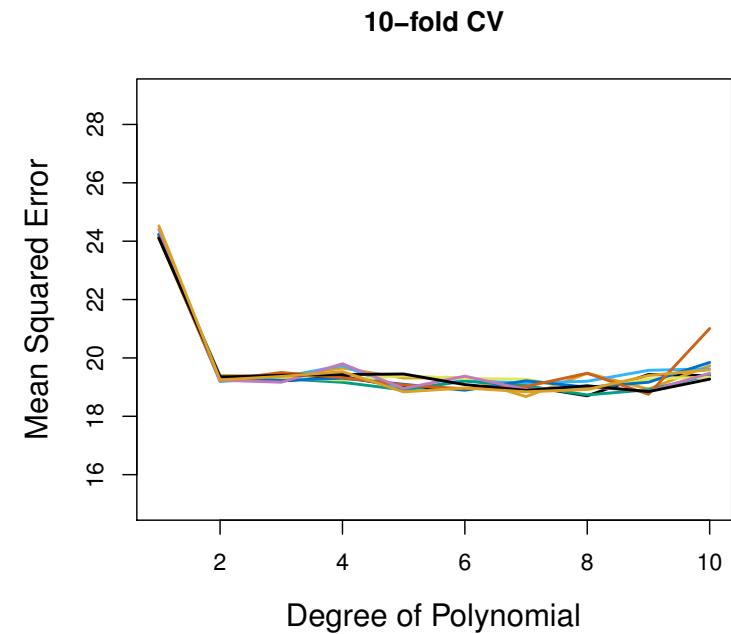
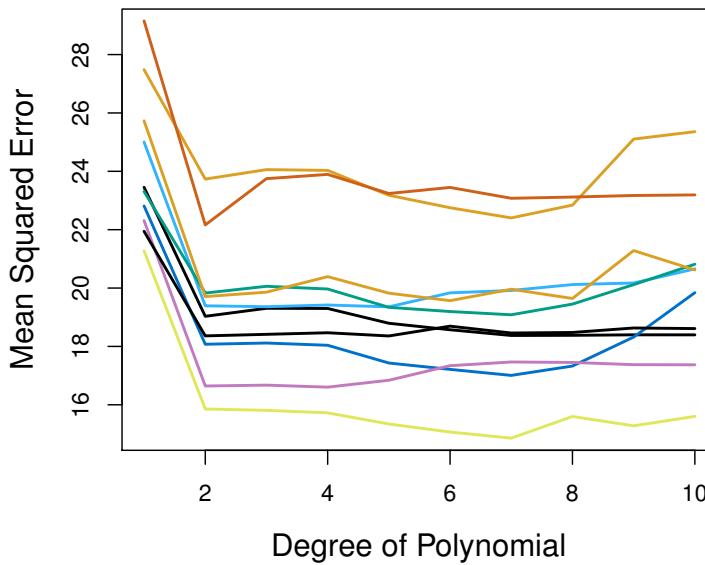
Auto Data: LOOCV vs. k -fold CV

- Left: LOOCV error curve
- Right: 10-fold CV was repeated 9 times, and the figure shows the slightly different CV error rates
- LOOCV is a special case of k -fold, where $k = n$
- They are both stable, but LOOCV is more computationally intensive



Auto Data: Validation Set Approach vs. k -fold CV Approach

- Left: Validation Set Approach
- Right: 10-fold Cross Validation Approach
- 10-fold CV is more stable



k -fold Cross Validation on Three Simulated Data Sets

Black: True Model

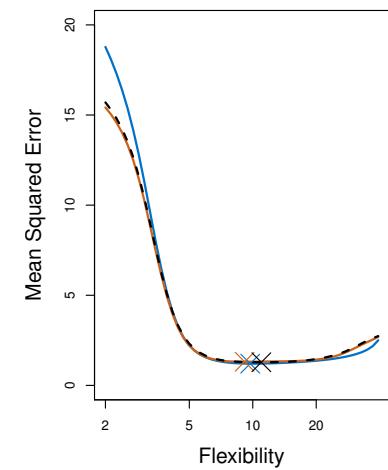
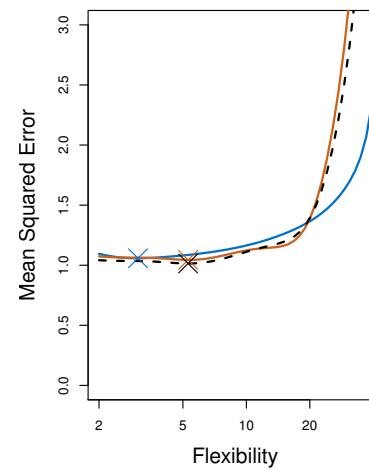
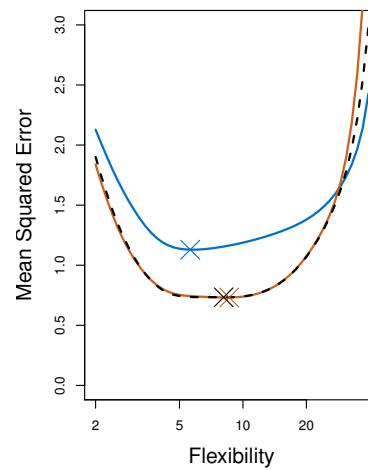
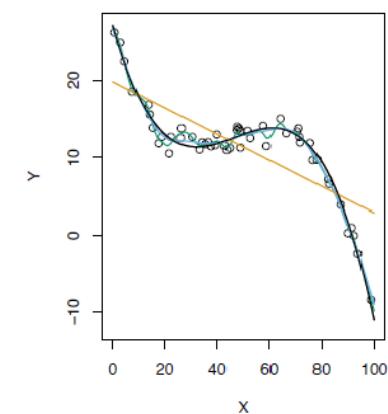
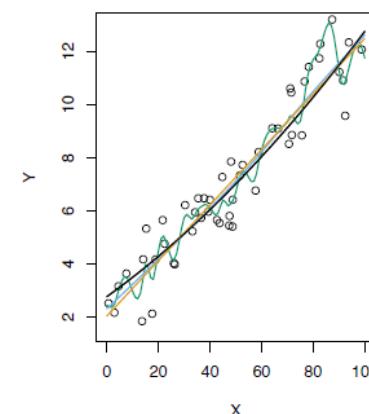
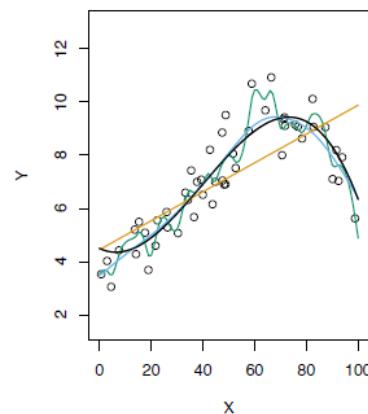
Orange: Linear Regression

Blue: Low Flexibility Spline

Green: High Flexibility Spline

From Chapter 2

Fig 2.9, 2.10, and 2.11



- Blue: True Test MSE
- Black-dashed: LOOCV MSE
- Orange: 10-fold MSE
- Refer to chapter 5 for the bottom graphs, Fig 5.6 page 182

5.1.4 Bias- Variance Trade-off for k -fold CV

- Putting aside that LOOCV is more computationally intensive than k -fold CV... Which is better LOOCV or k -fold CV?
 - LOOCV is less bias than k -fold CV (when $k < n$)
 - But, LOOCV has higher variance than k -fold CV (when $k < n$)
 - Thus, there is a trade-off between what to use
- Conclusion:
 - We tend to use k-fold CV with ($k = 5$ and $k = 10$)
 - It has been empirically shown that they yield test error rate estimates that suffer neither from excessively high bias, nor from very high variance

5.1.5 Cross Validation on Classification Problems

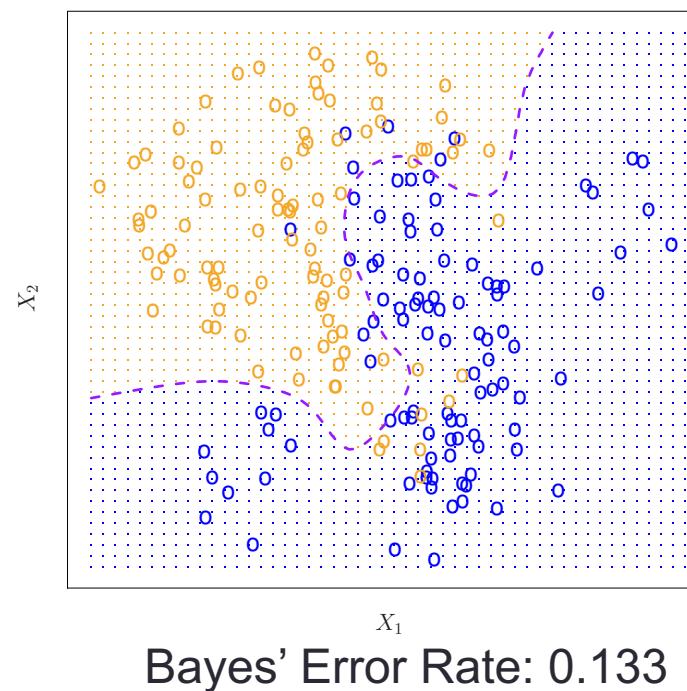
- So far, we have been dealing with CV on regression problems
- We can use cross validation in a classification situation in a similar manner
 - Divide data into k parts
 - Repeat k times:
 - Hold out one part, fit the model using the remaining data and compute the error rate on the hold out data
 - CV error rate is the average over the k error rates we have computed

Cross Validation in Practice

- CV can help estimate MSE or Classification Accuracy
- If we can estimate MSE before running the actual MSE how could we use it in our *model selection* process?
- What types of things could we decide about our Machine Learning modeling process?
- **MINUTE PAPER:** Brainstorm and write down your specific uses for CV.

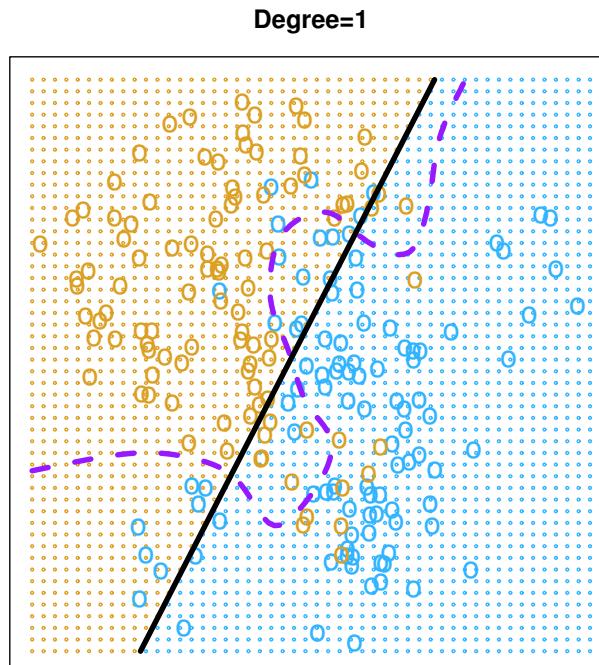
Model Selection Example: Use CV to Select Classification Model

- Model Choice includes Logistic Regression (with terms of $\text{deg} = 1, 2, \dots, 10$) and KNN with many choices for K
- The data set used is simulated (refer to Fig 2.13)
- The purple dashed line is the Bayes' boundary

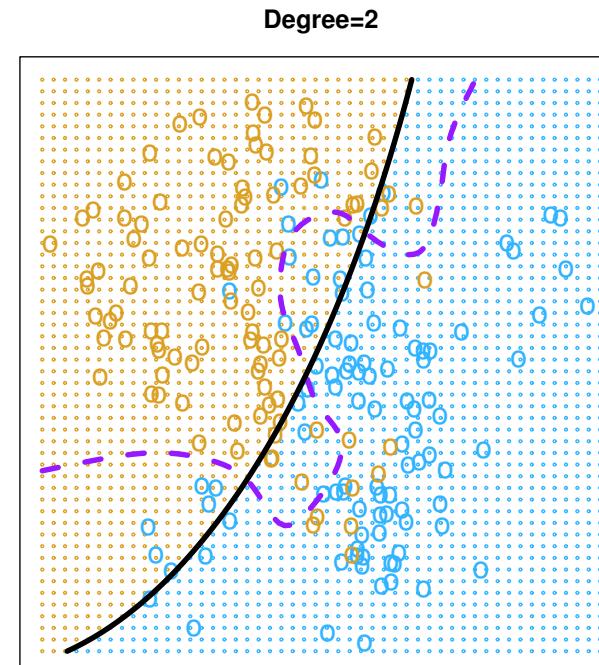


Use CV to Select Classification Model

- Linear Logistic regression (Degree 1) is not able to fit the Bayes' decision boundary
- Quadratic Logistic regression does better than linear



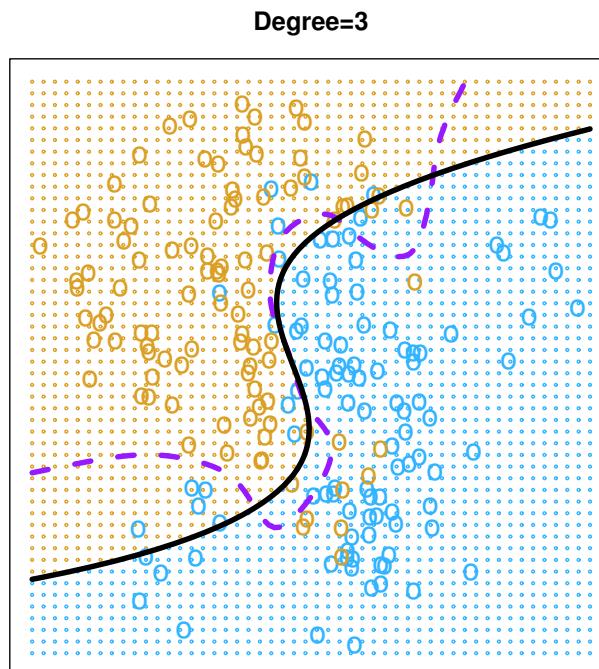
Error Rate: 0.201



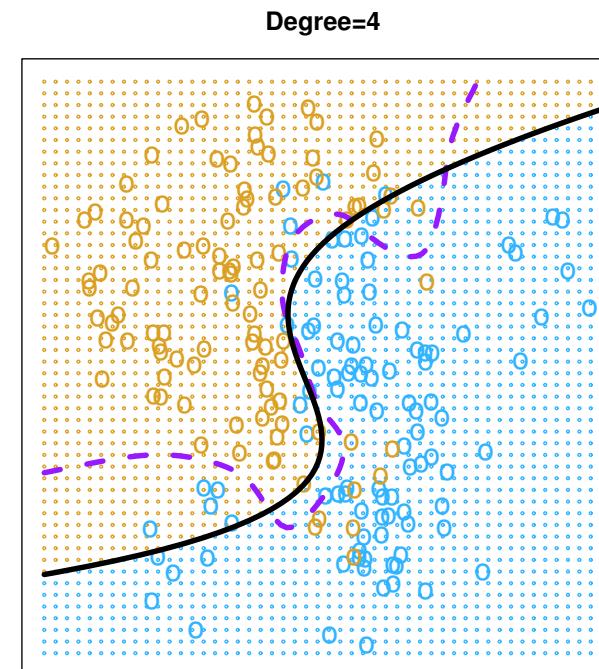
Error Rate: 0.197

Use CV to Select Classification Model

- Using cubic and quartic predictors, the accuracy of the model improves



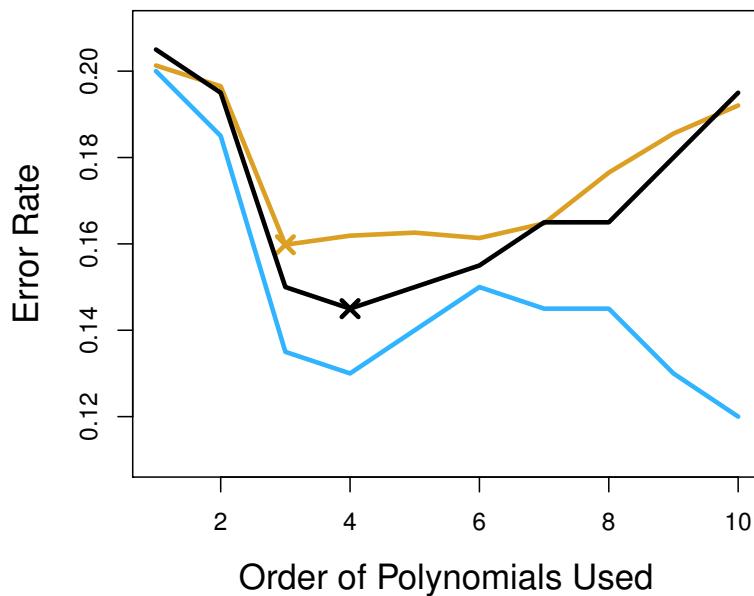
Error Rate: 0.160



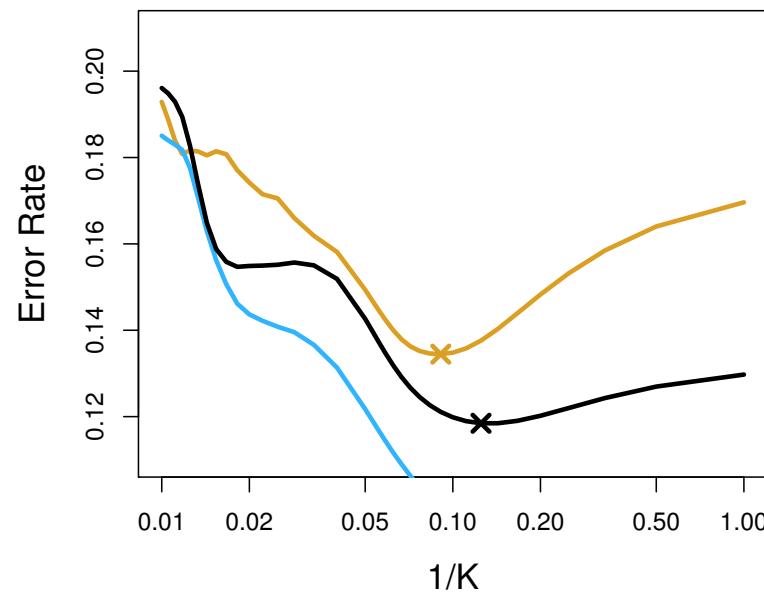
Error Rate: 0.162

Use CV to Select Classification Model

Logistic Regression



KNN



Blue: Training Error
Black: 10-fold CV Error
Brown: Test Error

Concept Check:
How do we interpret the results of these graphs?
What value polynomial should we choose?

LINEAR MODEL SELECTION AND REGULARIZATION

Chapter 06

Given a set of available features, how do we build the best set of features for our model?

- Subset Selection
 - Best Subset Selection
 - Stepwise Selection
 - Choosing the Optimal Model
- Shrinkage Methods
 - Ridge Regression
 - The Lasso

Improving on the Least Squares Regression Estimates for models with many features

- Given a set of observations, in Linear Regression, the cost can be expressed as MSE or RSS or R^2
- Either least-squares fitting process or an iterative optimization picks coefficients that minimize this cost
- There are 2 reasons that coefficients selected using these cost estimates may not be ideal:
 1. Prediction Accuracy on non-training data
 2. Model Interpretability for features

Prediction Accuracy Problems

- The Linear Regression estimate has low variability especially when the relationship between Y and X is linear and the number of observations n is much larger than the number of predictors p ($n \gg p$)
- But, when $n \approx p$, then the fit can have high variance and may result in overfitting and poor estimates on unseen observations – poor generalizability
- And, when $n < p$, then the variability of fit increases dramatically, and the variance of these estimates are unacceptable

Model Interpretability Problems

- When we have a large number of features in the model there will be many that have little or no influence on Y
- Leaving these variables in the model makes it harder to determine “important variables”
- The model would be easier to interpret by removing the unimportant variables

Solution Concepts

- Subset Selection
 - Identify a subset of all p predictors which best predict the response Y , and then fit the model using only this subset
 - Methods: *best subset selection* and *stepwise selection*
- Regularization through coefficient Shrinkage
 - Penalize the model (new cost function element) for having non-zero estimates of coefficients -> pushes coefficients towards zero
 - This shrinkage *reduces the variance* **WHY?**
 - Some of the coefficients may shrink to exactly zero – helps with variable selection/interpretation
 - Methods: *Ridge regression* and the *LASSO*
- Dimension Reduction
 - Project all p predictors into an M -dimensional space where $M < p$, and then fit a linear regression model
 - Example: Principle Components Regression

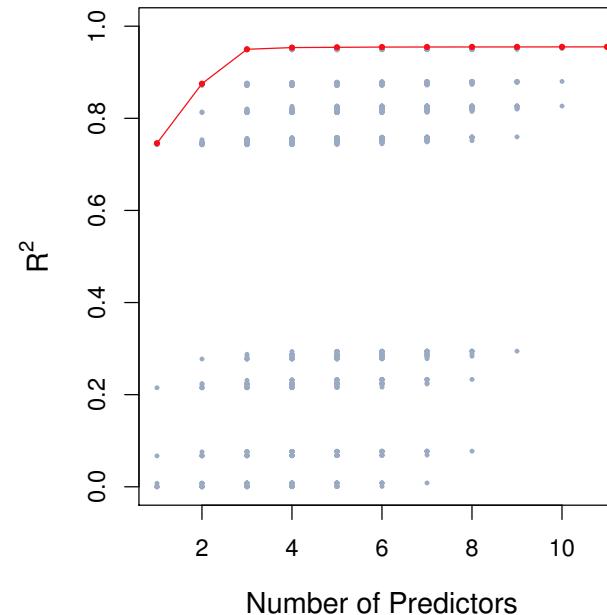
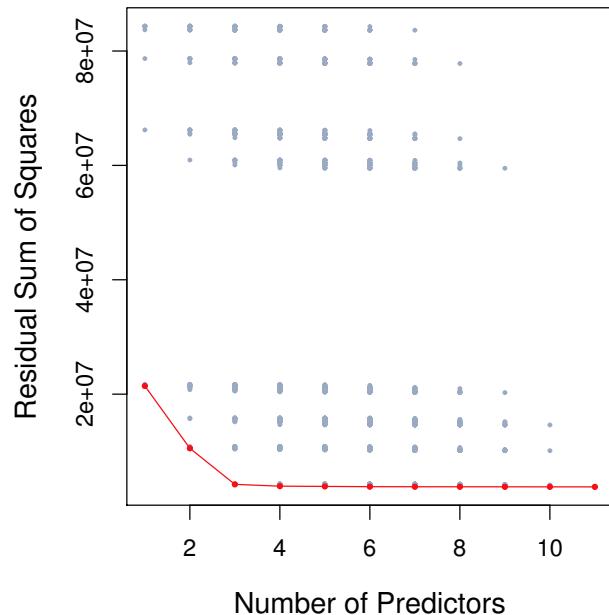
6.1 SUBSET SELECTION

6.6.1 Best Subset Selection

- Fit a linear regression model for each possible combination of the X predictors
- How do we judge which subset is the “best”?
- One simple approach is to take the subset with the smallest RSS or the largest R^2
- Unfortunately, one can show that the model that includes all the variables will always have the largest R^2 (and smallest RSS) **Why do you think this is?**

Credit Data: R^2 vs. Subset Size

- RSS will never increase (and R^2 will never decrease) as the number of variables increase - not very useful



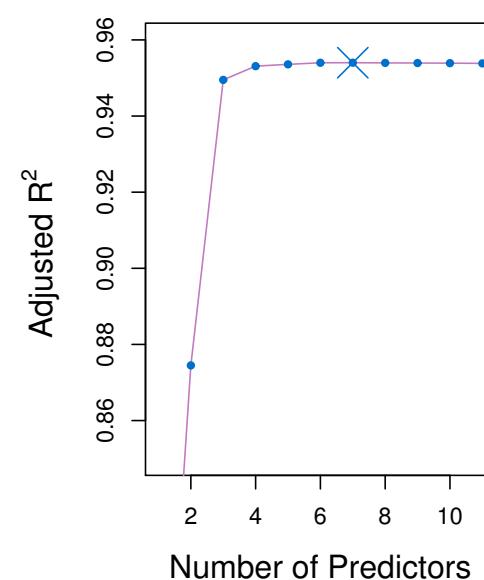
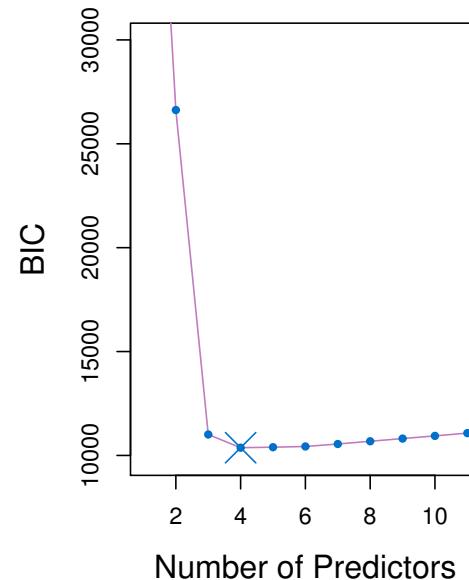
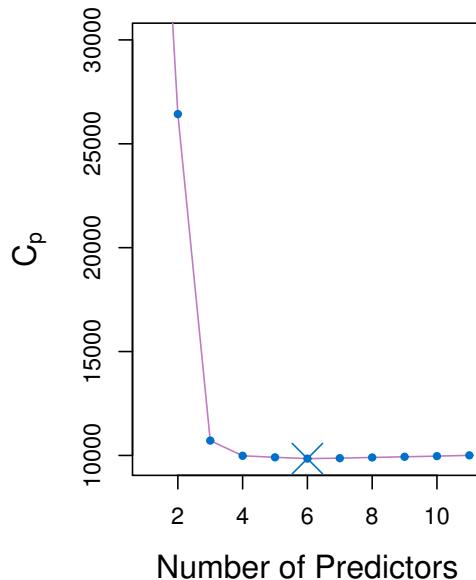
- Grey dots – actual performance of various subset models
- red line: the best model for a given number of predictors, according to RSS and R^2

Measures for *Estimating* model Performance on unseen data from *training* data fit

- To compare different models, adjust the RSS of the *training* data model fit based on some penalty for number of features
(p211-212):
 - Adjusted R²
 - AIC (Akaike information criterion)
 - BIC (Bayesian information criterion)
 - C_p (Mallow's C_p: Proportional to AIC)
- These methods add penalty to RSS for the number of variables (i.e. complexity) in the model
- Estimates are made using model's fit of *training* data
- All are estimates...None are perfect

Credit Data: C_p , BIC, and Adjusted R^2

- A small value of C_p or BIC indicates a low error, and thus (hopefully) a better model
- A large value for the Adjusted R^2 indicates a better model



Feature Selection through Best Subset Selection

- Best Subset Selection considers all possible subsets of available features to find the optimal fit using validation data
- Select the model using the subset of features which yields the best performance on the (cross) validation data
 - E.g. best MSE or lowest classification error
- Concept Check: Compute $O(\cdot)$ for best subset selection as a function of p ...
What is the number of possible feature subsets when there are p features available?

Feature Selection via Stepwise Selection

- Best Subset Selection is computationally intensive especially when we have a large number of predictors (large p)
- More computationally-attractive methods:
 - Forward Stepwise Selection: Begins with the model containing no predictor, and then adds one predictor at a time that *improves the model the most* until no further improvement occurs
 - Backward Stepwise Selection: Begins with the model containing all predictors, and then deleting one predictor at a time where the predictor chosen at each step is the *feature that causes the least degradation* to model performance when removed.
- Compute $O(\cdot)$ for these methods as a function of p :
This can be thought of as a search (CSCE 523-style)
What is the number of computations needed when there are p features available?

REGULARIZATION (Parameter Shrinkage) METHODS

6.2.1 Ridge Regression

- Ordinary Least Squares (OLS) estimates β 's by minimizing

$$\text{RSS} = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2.$$

- Ridge Regression uses a slightly different minimization equation which adds a term...

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2,$$

Math-Sense Check: Describe the influence of the last term in this equation

Ridge Regression Adds a Penalty on β 's

- The effect of this equation is to add a penalty of the form

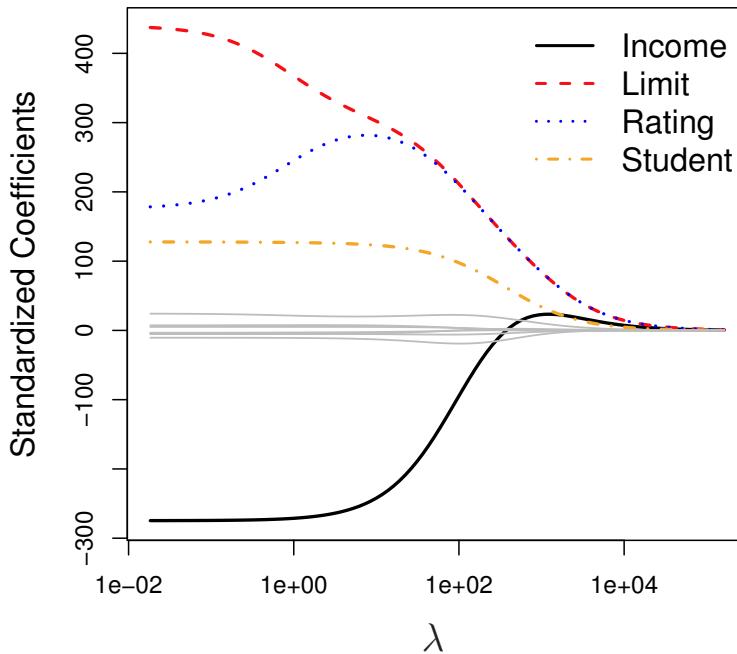
$$\lambda \sum_{j=1}^p \beta_j^2,$$

Where the tuning parameter λ is a positive value.

- This has the effect of “shrinking” large values of β 's towards zero.
- This penalty should improve the fit because shrinking the coefficients can significantly reduce their variance
- When $\lambda = 0$, we get the original RSS from Ordinary Least Squares

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2,$$

Credit Data: Ridge Regression



$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2,$$

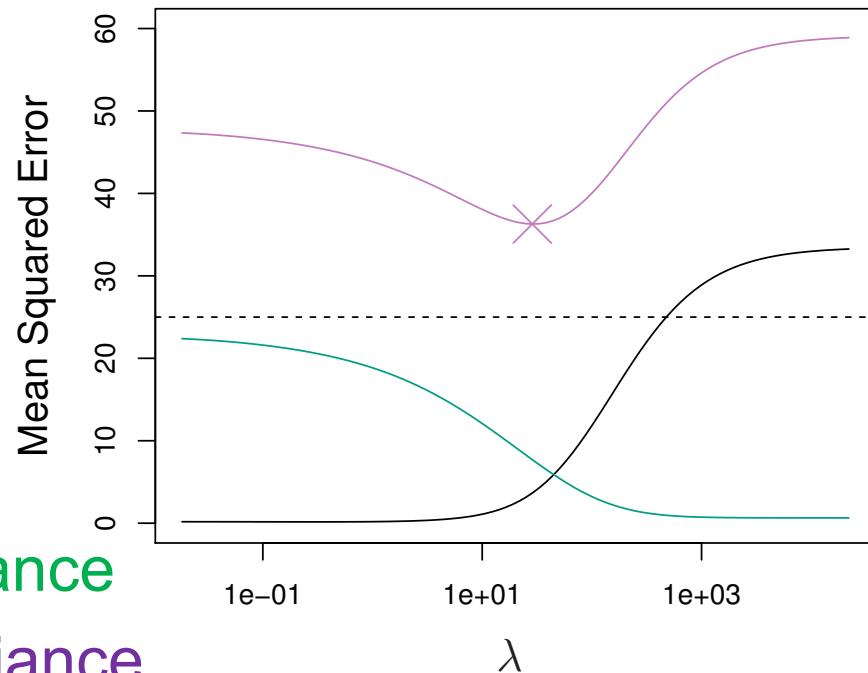
- As λ increases, the standardized coefficients shrink towards zero.
- **Will coefficients ever reach zero?**
- **If not, what are the implications with model interpretability?**

Why can shrinking towards zero be a good thing to do?

- It turns out that the parameter estimates generally have low bias but can be highly variable. In particular when n and p are of similar size or when $n < p$, then the OLS estimates will be extremely variable. **WHY?**
- The penalty term makes the ridge regression estimates biased but can also substantially reduce variance
- Thus, there is a bias / variance trade-off

Ridge Regression Bias / Variance

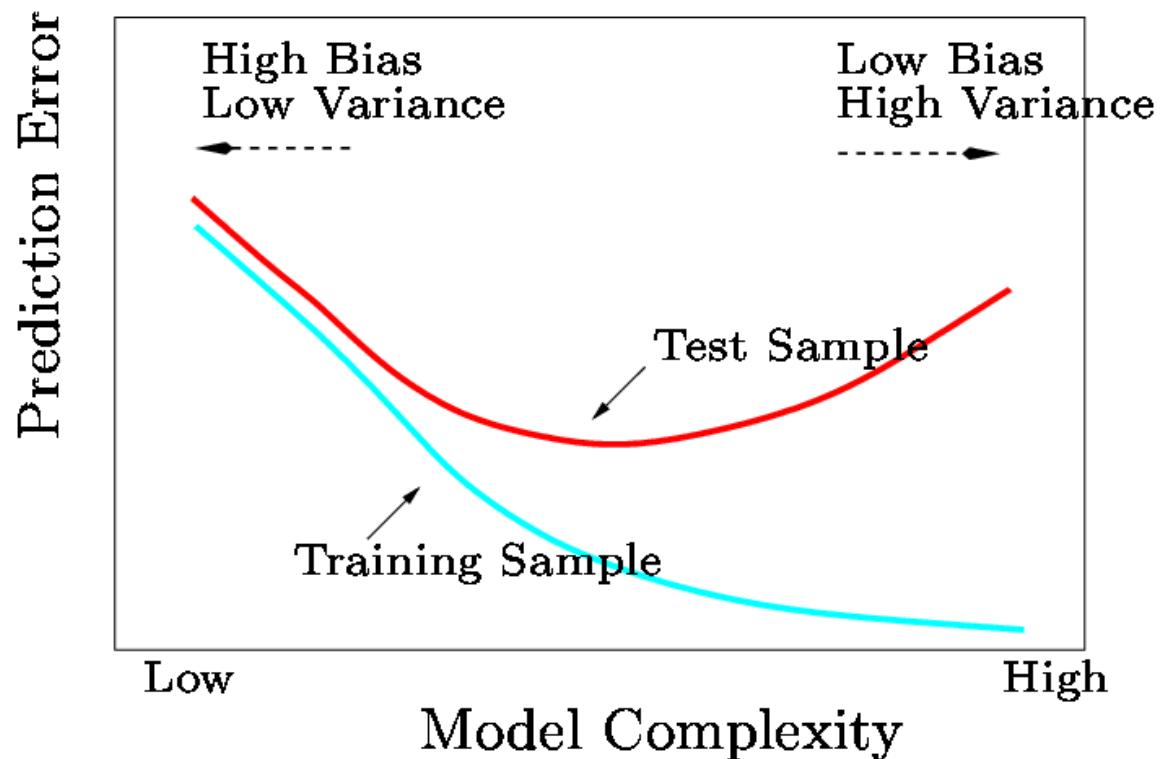
$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2,$$



- Black: MSE due to Bias
- Green: MSE due to Variance
- Purple: $\text{MSE} \sim \text{Bias} + \text{Variance}$
- Increase in λ increases bias but decreases variance

Bias / Variance Trade-off

- In general, the ridge regression estimates will be more biased than the OLS ones but have lower variance
- Ridge regression will work best in situations where the OLS estimates have high variance



Computational Advantages of Ridge Regression

- If p is large, then using the best subset selection approach requires searching through multitudes of possible models
- With Ridge Regression, for any given λ , we only need to fit one model
- Ridge Regression can even be used when $p > n$, a situation where OLS fails completely

6.2.2. The LASSO

- Ridge Regression isn't perfect
- One significant problem is that the penalty term will never force any of the coefficients to be *exactly* zero. Thus, the final model will include all variables, which makes it harder to interpret
- A more modern alternative is the LASSO:
Least **A**solute **S**hrinkage and **S**election **O**perator
- The LASSO works in a similar way to Ridge Regression, except it uses a different penalty term

Ridge Regression vs. LASSO: Penalty Term

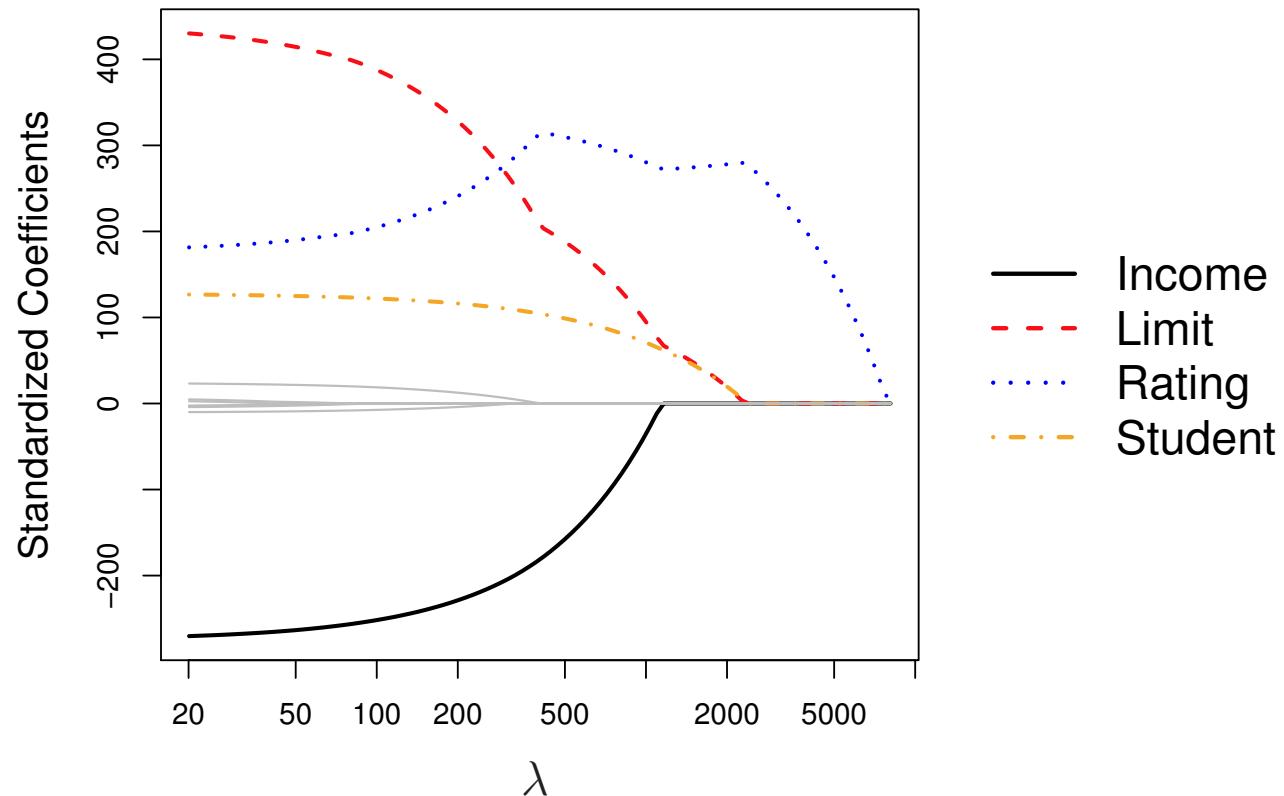
- Ridge Regression minimizes

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2,$$

- The LASSO estimates the β 's by minimizing

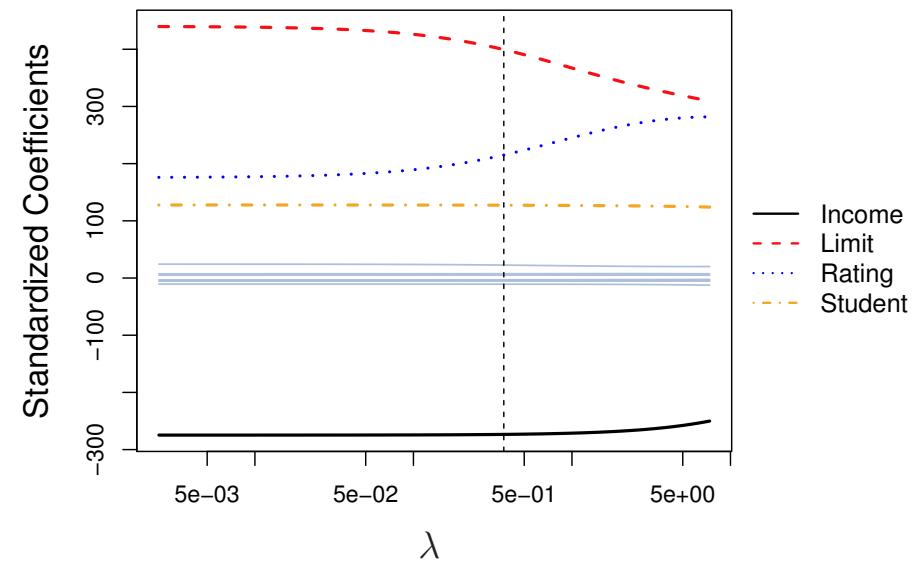
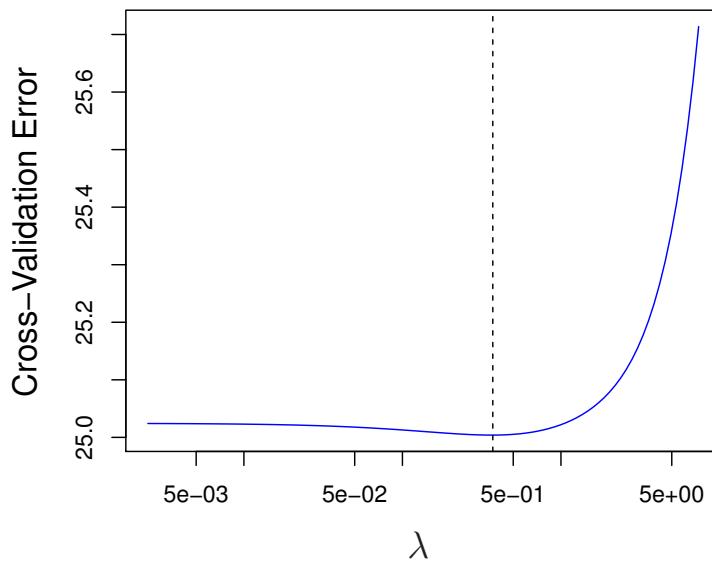
$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^p |\beta_j|.$$

Credit Data: LASSO



6.2.3 Selecting the Tuning Parameter for λ best performing model

- We need to decide on a value for λ
- Select a grid of potential values, use cross validation to determine error rate (for each value of λ) and select the lambda value that gives the lowest error rate



Benefits of LASSO

- Using this penalty, it could be proven mathematically that some coefficients end up being set to exactly zero
- With LASSO, we can produce a model that has high predictive power and it is simple to interpret because some coefficients are driven to zero
-
- In this class we will show how to do this empirically
CLASS CODING EXERCISE (Regularization)

DECISION TREES

Chapter 08 (part 01)

Outline

- The Basics of Decision Trees
 - Regression Trees
 - Classification Trees
 - Regularization via Pruning
 - Trees vs. Linear Models
 - Advantages and Disadvantages of Trees

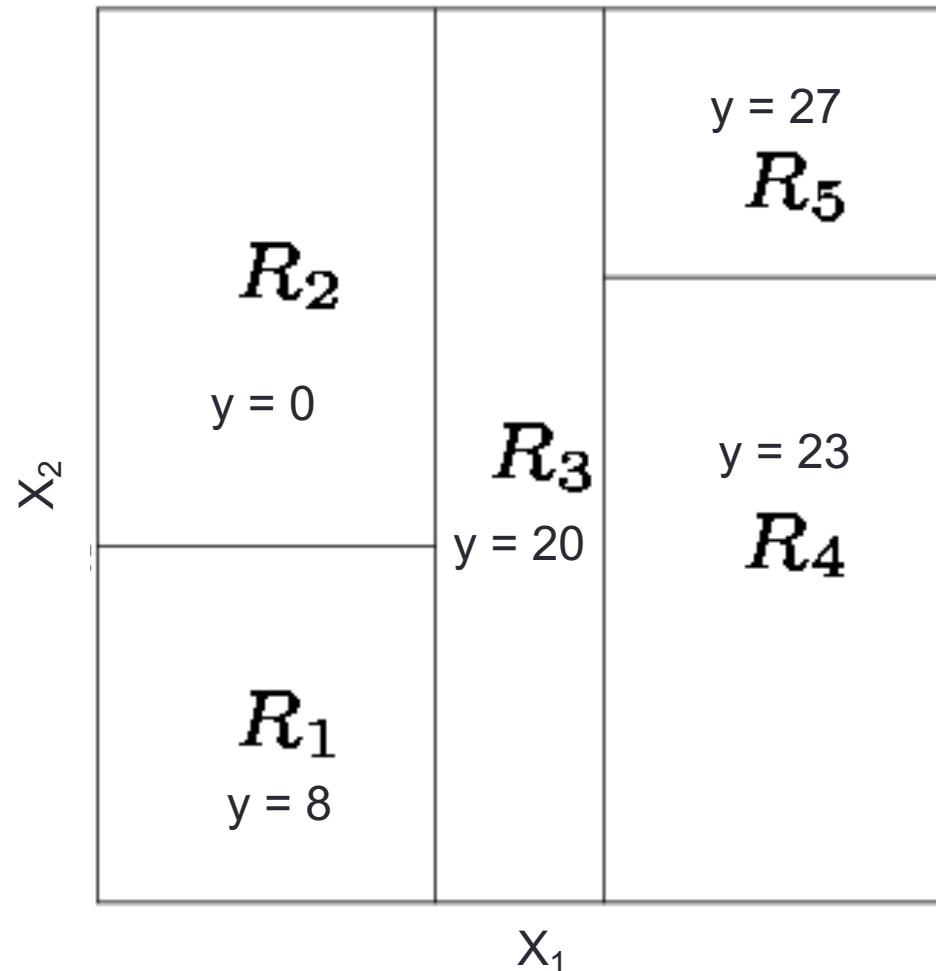
REGRESSION TREES

Prediction as Partitioning the feature space

- One way to make predictions in a regression problem is to divide the feature space (i.e. all the possible values for X_1, X_2, \dots, X_p) into distinct regions, say R_1, R_2, \dots, R_k
- Then for each observation that falls in a particular region (say R_j) we make the same prediction
 - The value of the prediction should be influenced by the response variables of the observations which are members of the region

Multi-feature example

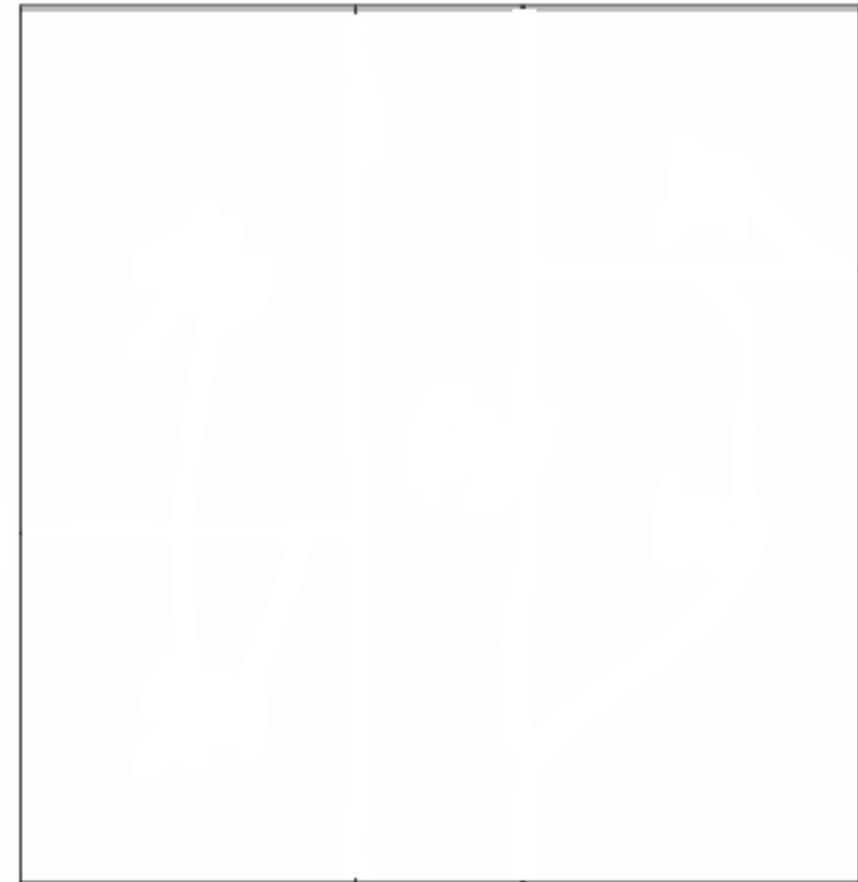
- two predictors and five distinct regions
- Depending on which region our new X comes from we would make one of five possible predictions for Y .



Splitting the X Variables

- Create the partitions by *iteratively* splitting one of the existing regions into two regions
- After the initial split, must decide which subregion to split next

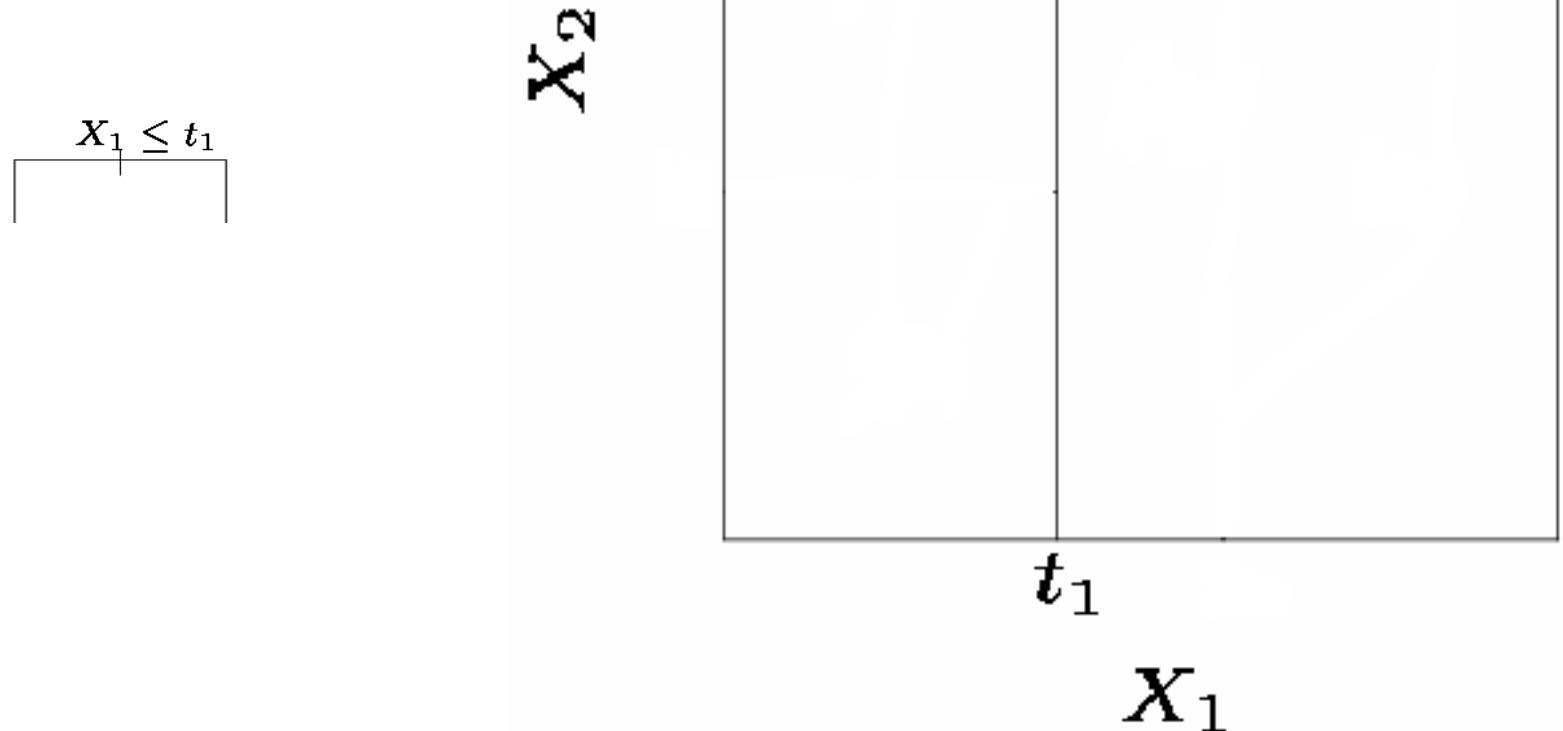
X_2



X_1

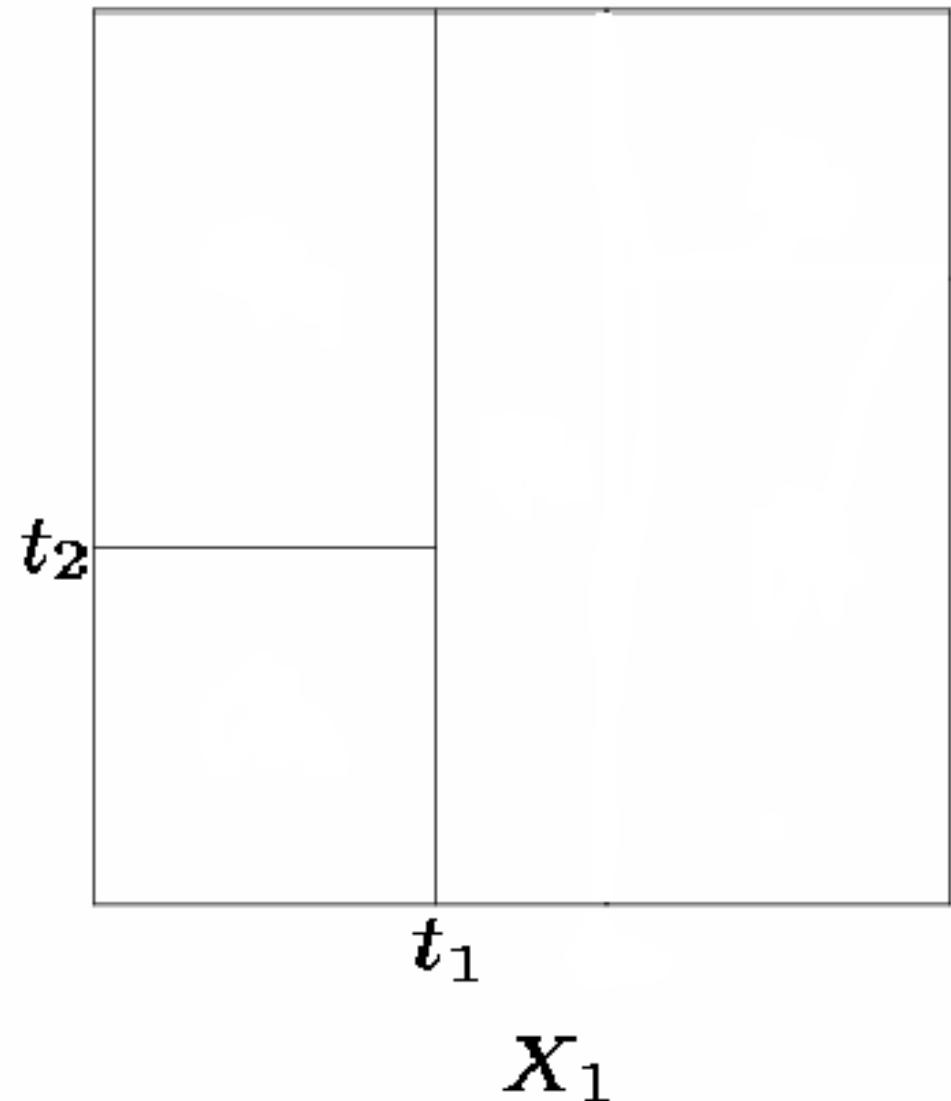
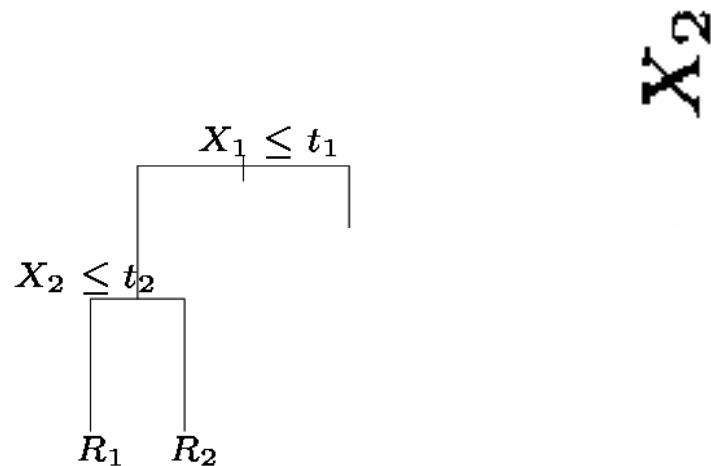
Splitting the X Variable

1. First split on $X_1 = t_1$



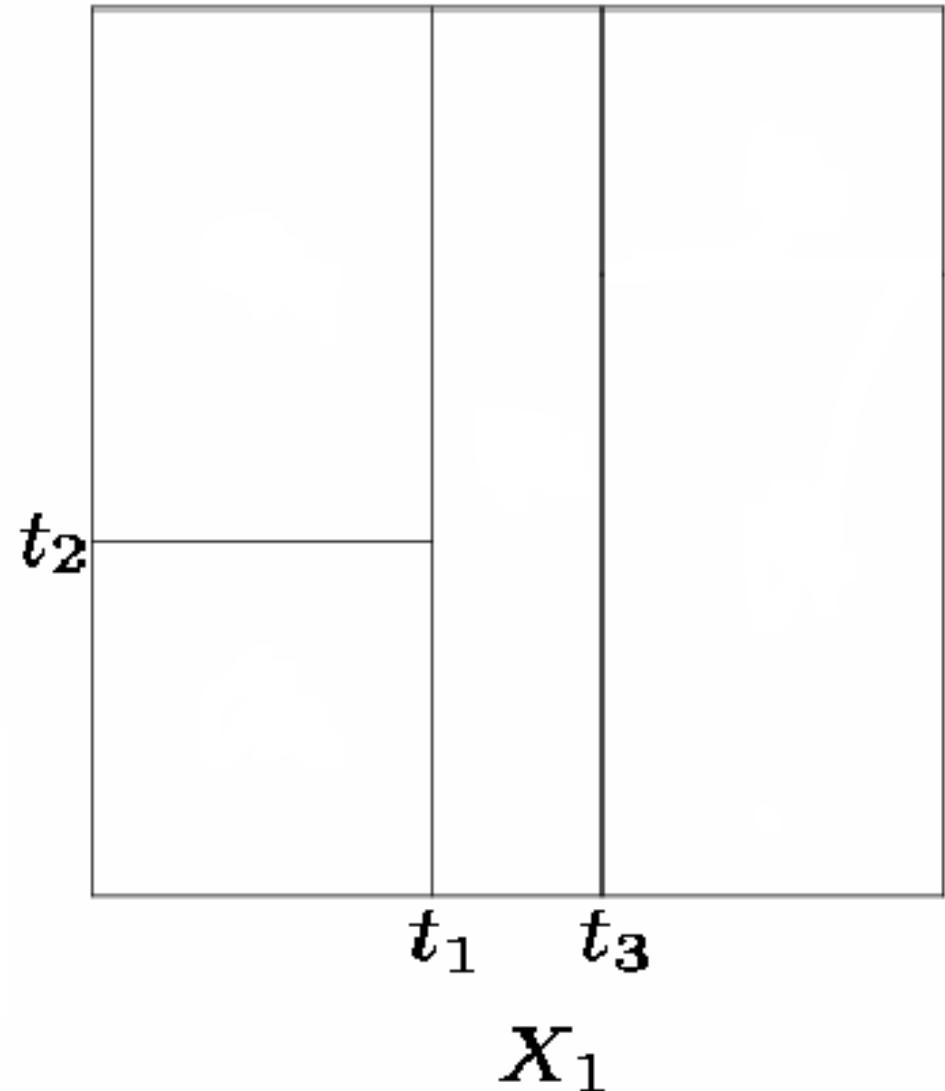
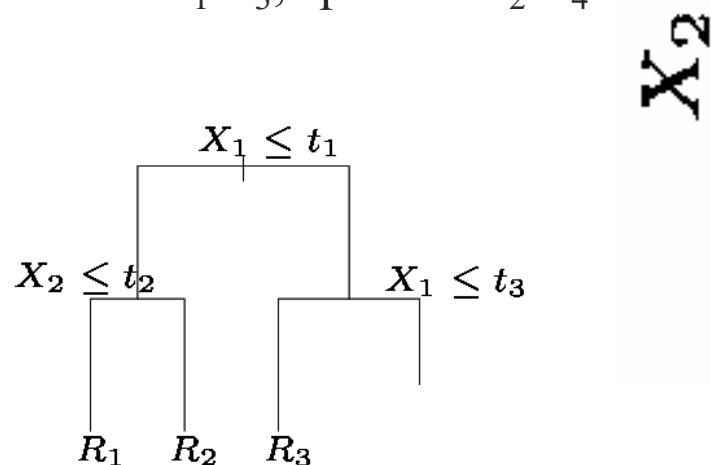
Splitting the X Variable

1. First split on $X_1 = t_1$
2. If $X_1 < t_1$, split on $X_2 = t_2$



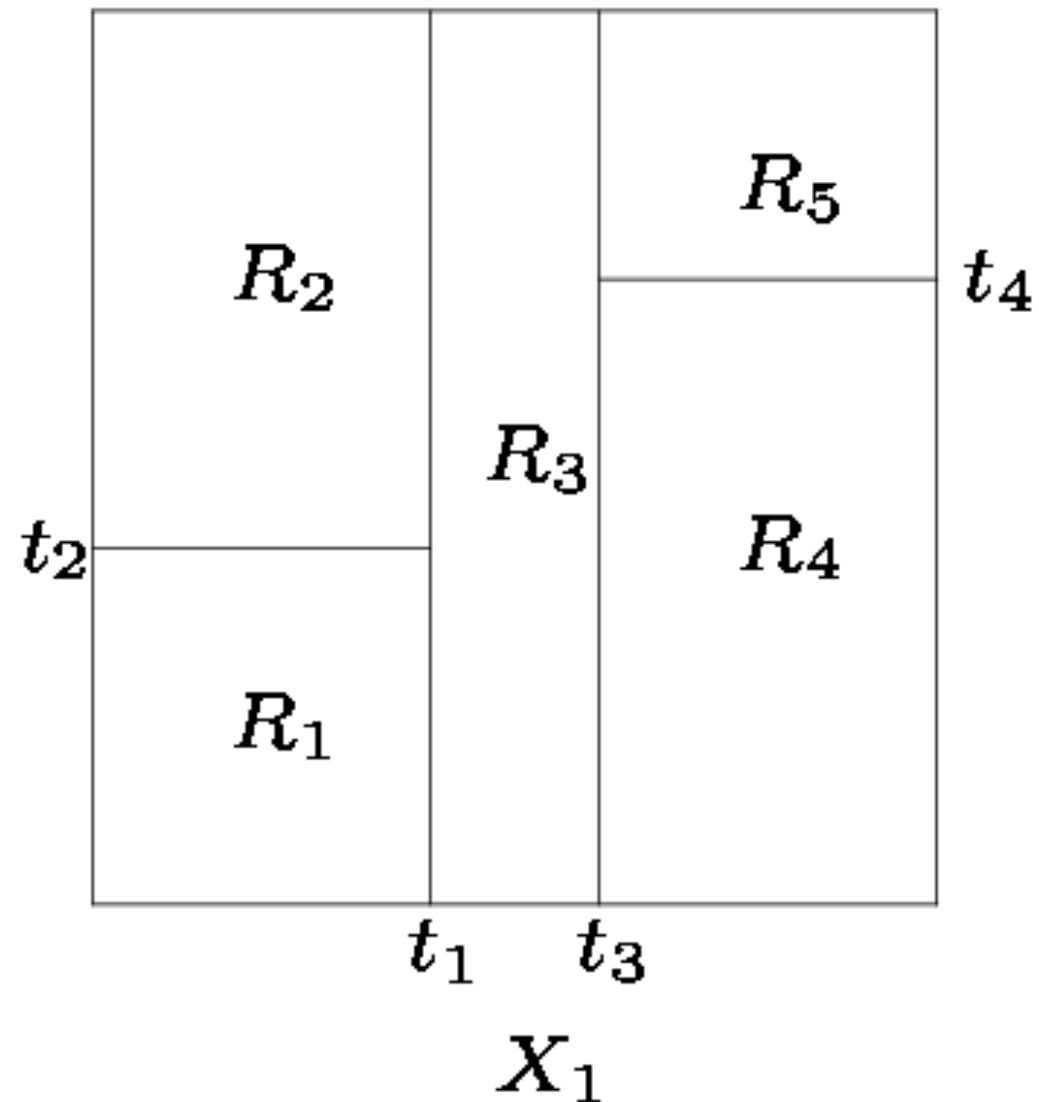
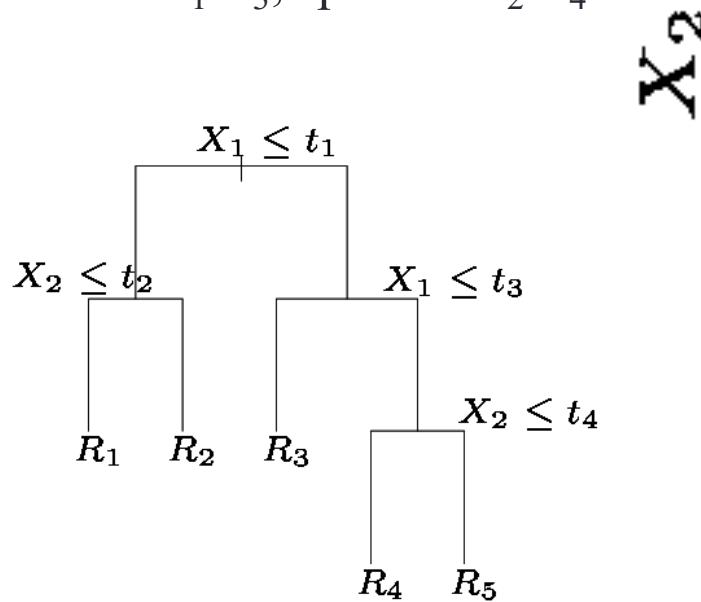
Splitting the X Variable

1. First split on $X_1 = t_1$
2. If $X_1 < t_1$, split on $X_2 = t_2$
3. If $X_1 > t_1$, split on $X_1 = t_3$
4. If $X_1 > t_3$, split on $X_2 = t_4$

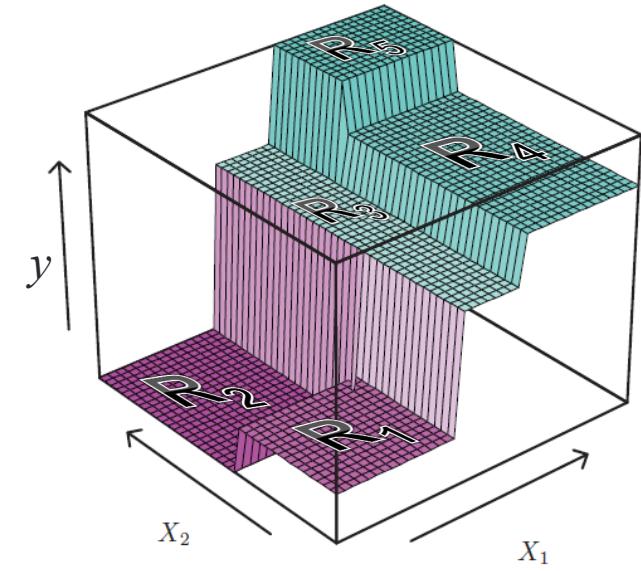
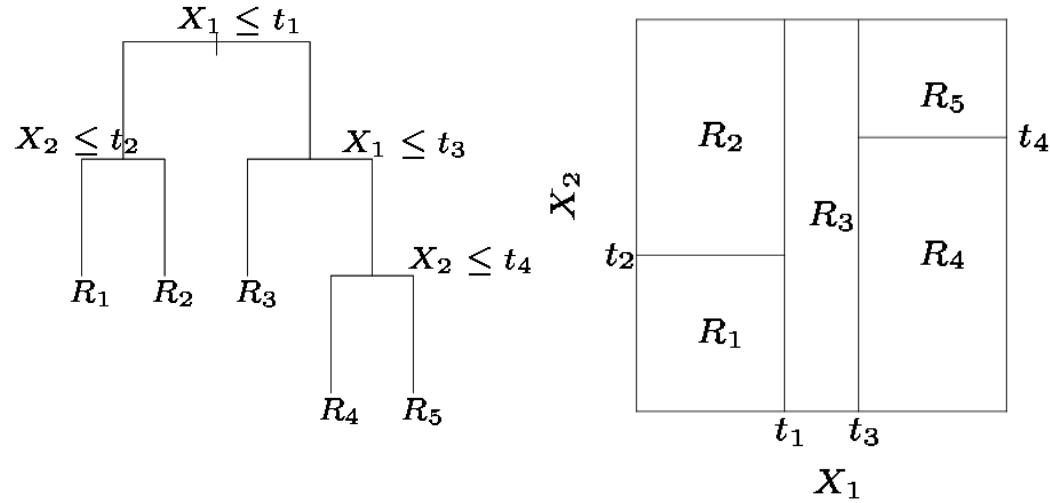


Splitting the X Variable

1. First split on $X_1 = t_1$
2. If $X_1 < t_1$, split on $X_2 = t_2$
3. If $X_1 > t_1$, split on $X_1 = t_3$
4. If $X_1 > t_3$, split on $X_2 = t_4$



Splitting the X Variable

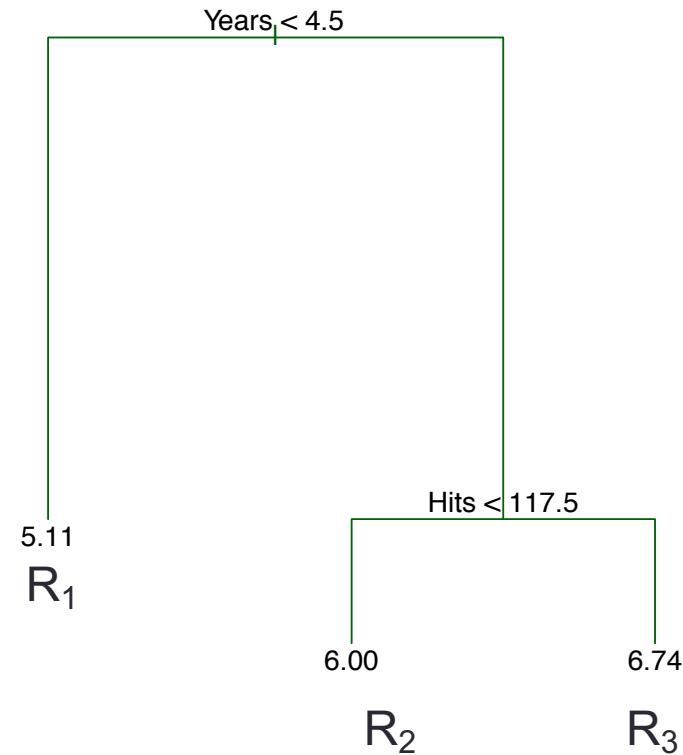


- Partitions can be represented with a tree structure.
- This provides a very simple way to interpret the model

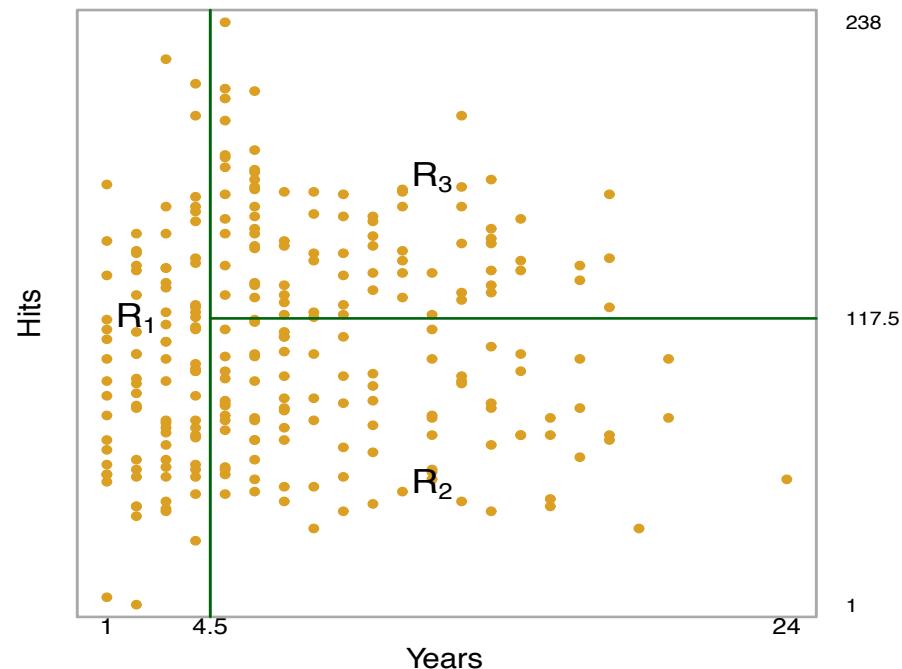
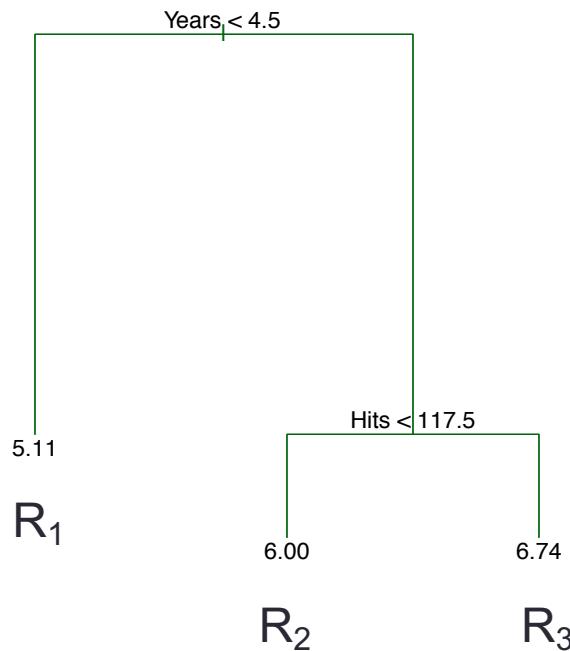
Example: Baseball Players' Salaries

- The predicted Salary is the number in each leaf node. It is the mean of the response for the observations that fall there
- Note that Salary is measured in 1000s, and log-transformed
- The predicted salary for a player who played in the league for more than 4.5 years and had less than 117.5 hits last year is

$$\$1000 \times e^{6.00} = \$402,834$$



Another way of visualizing the decision tree...



Design Decisions

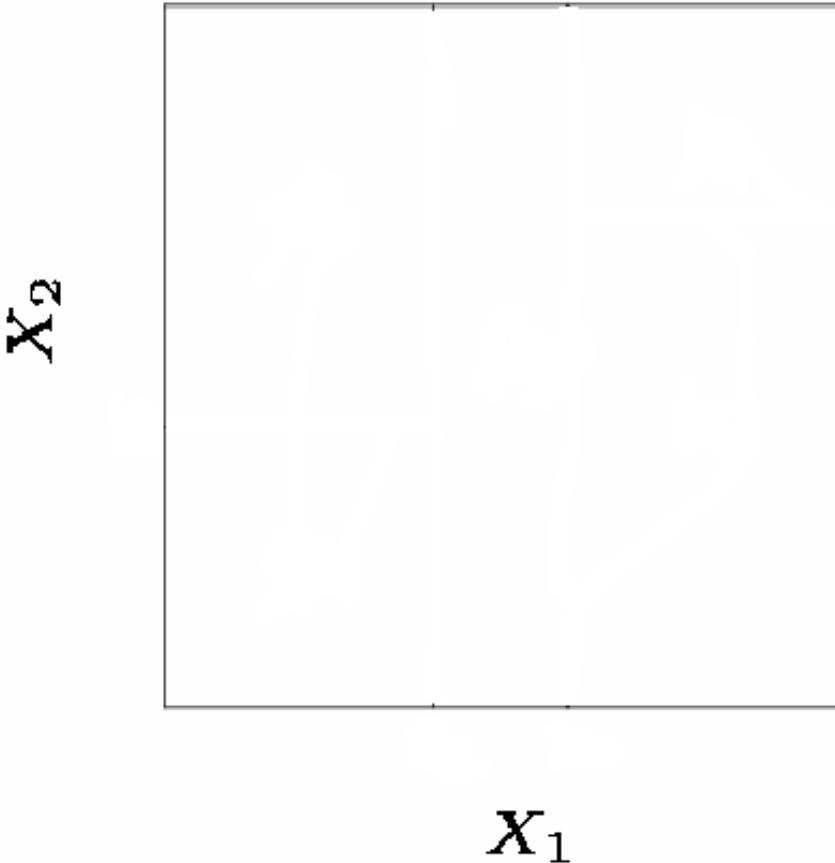
1. Where to split?

How do we decide on what regions to use i.e. R_1, R_2, \dots, R_k or equivalently what tree structure should we use?

2. What values should we use for $\mu_1, \mu_2, \dots, \mu_k$?

Where to Split?

- Consider splitting a region into two regions, $X_j > s$ and $X_j \leq s$ for all possible real values of s and feature indices j .
- One option: Choose the s and j that results in the lowest MSE on the training data.

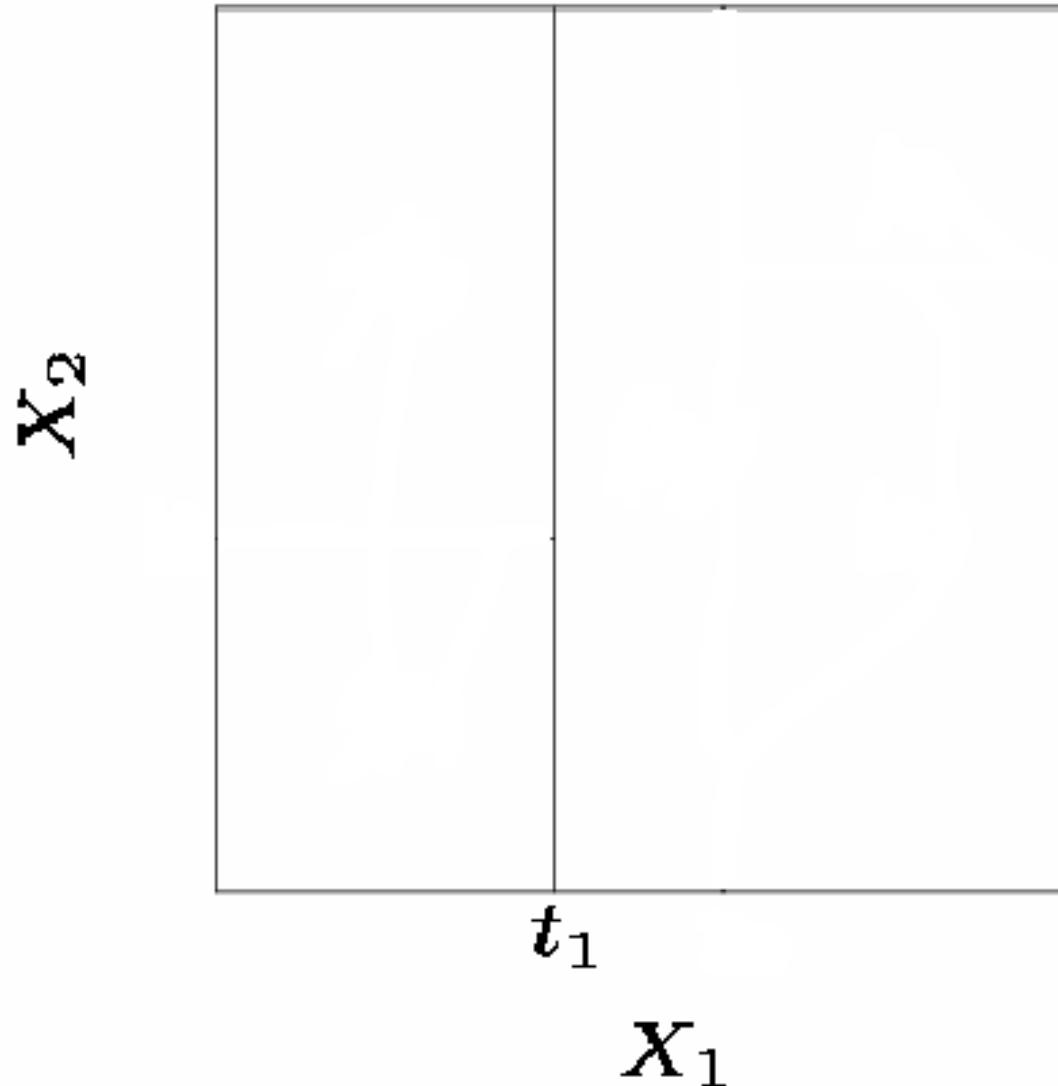


What values should we use for $\mu_1, \mu_2, \dots, \mu_k$?

- For region R_j , the best prediction is simply the average of all the responses from our training data that fell in region R_j .

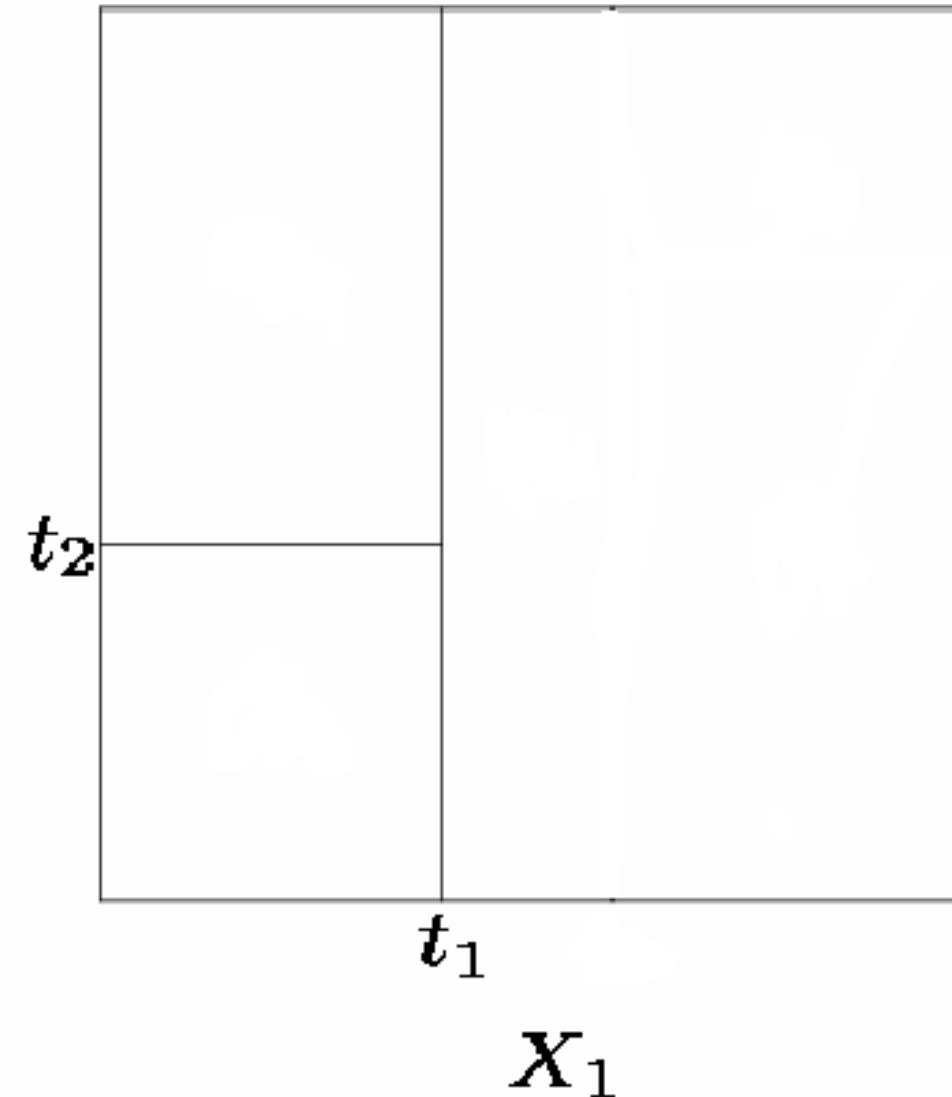
Where to Split Next?

- Suppose the optimal split was on X_1 at point t_1 .
- Now we repeat the process looking for the next best split except that we must also consider whether to split the first region or the second region up.
- Again the criteria is smallest MSE.



Where to Split?

- Here the optimal split was the left region on X_2 at point t_2 .
- This process continues until our regions have too few observations to continue e.g. all regions have 5 or fewer points.

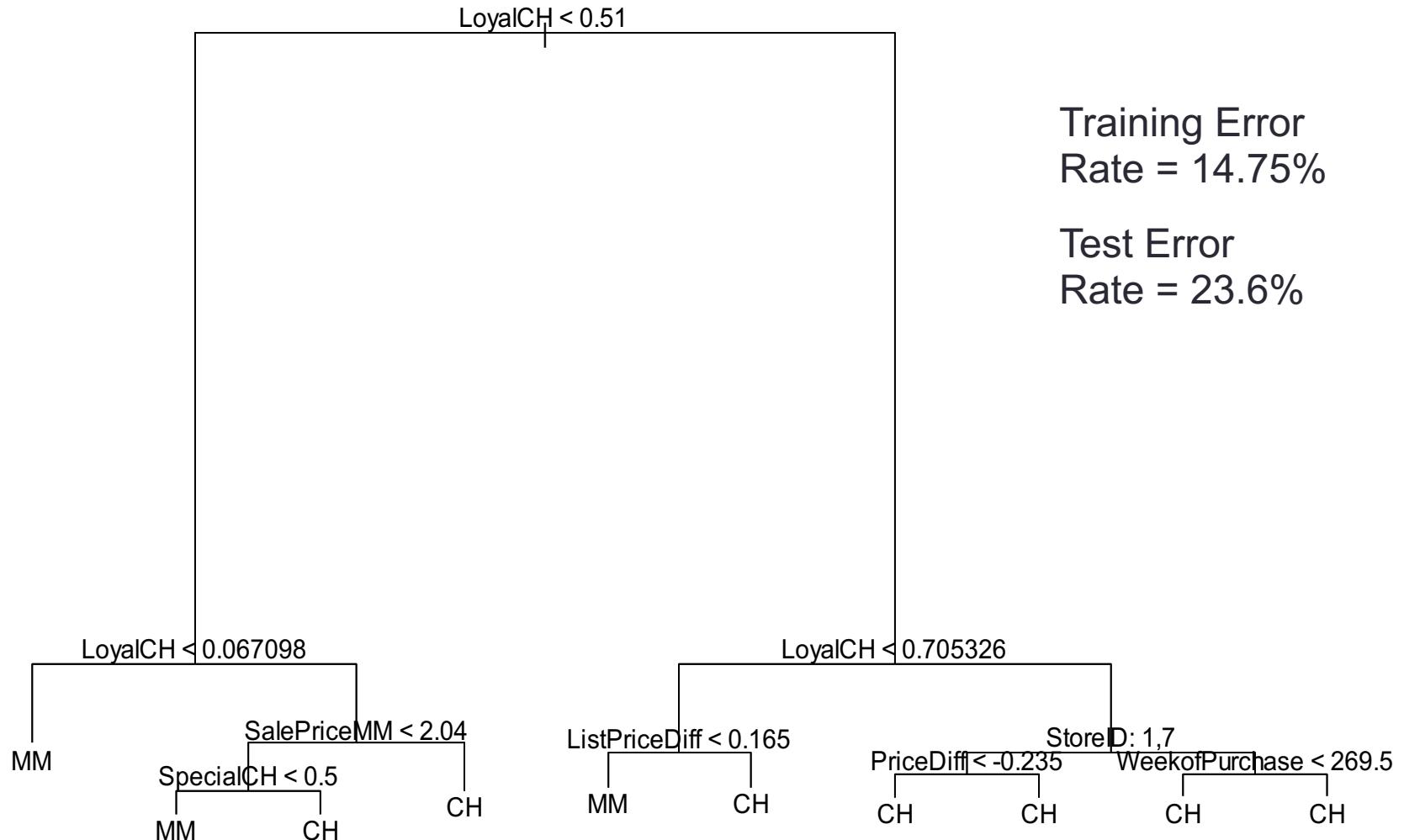


CLASSIFICATION TREES

Growing a Classification Tree

- A classification makes a prediction for a categorical Y
- For each region (or node) we predict the most common category among the training data within that region
- The tree is grown (i.e. the splits are chosen) in exactly the same way as with a regression tree except we need a different criteria to optimize
- There are several possible different criteria to use such as the “gini index” and “cross-entropy” but the easiest one to think about is to minimize the error rate.

Example: Orange Juice Preference



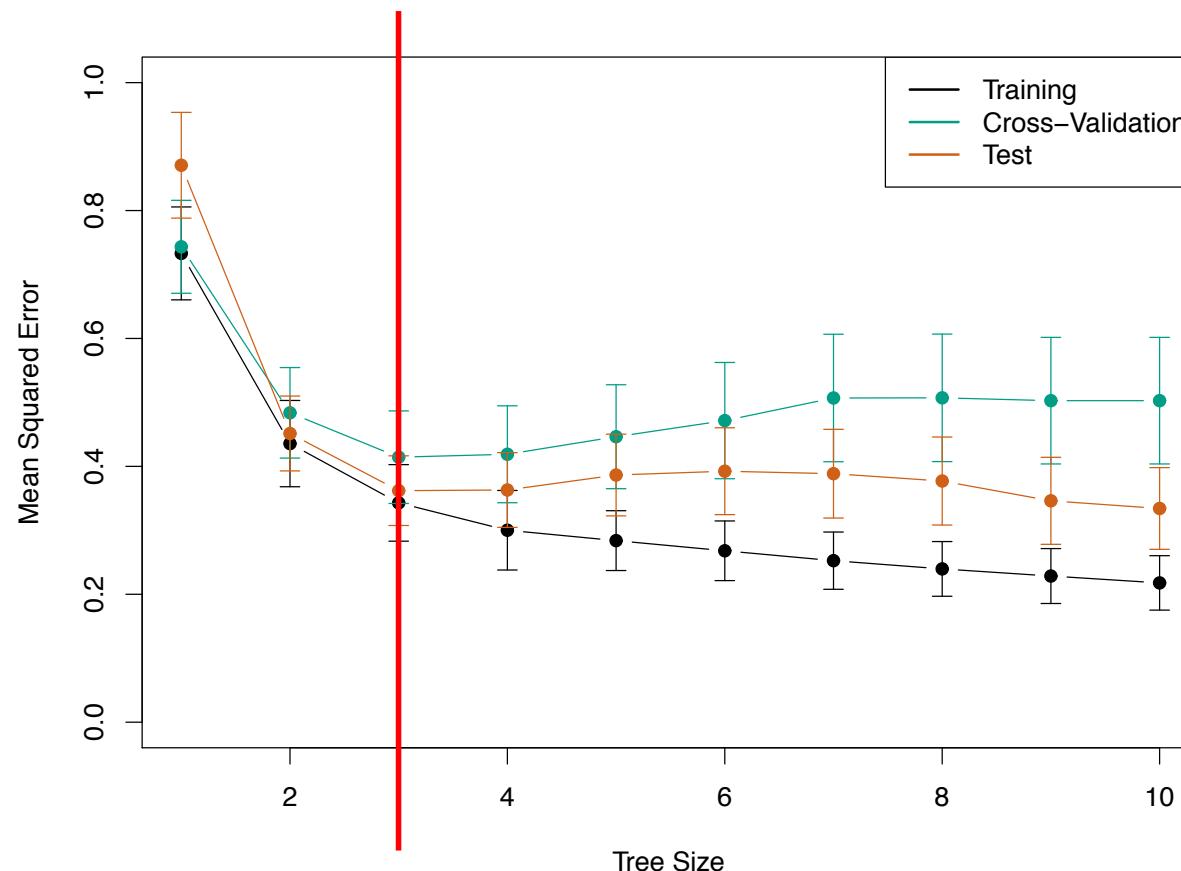
REGULARIZATION VIA PRUNING

Improving Tree Accuracy

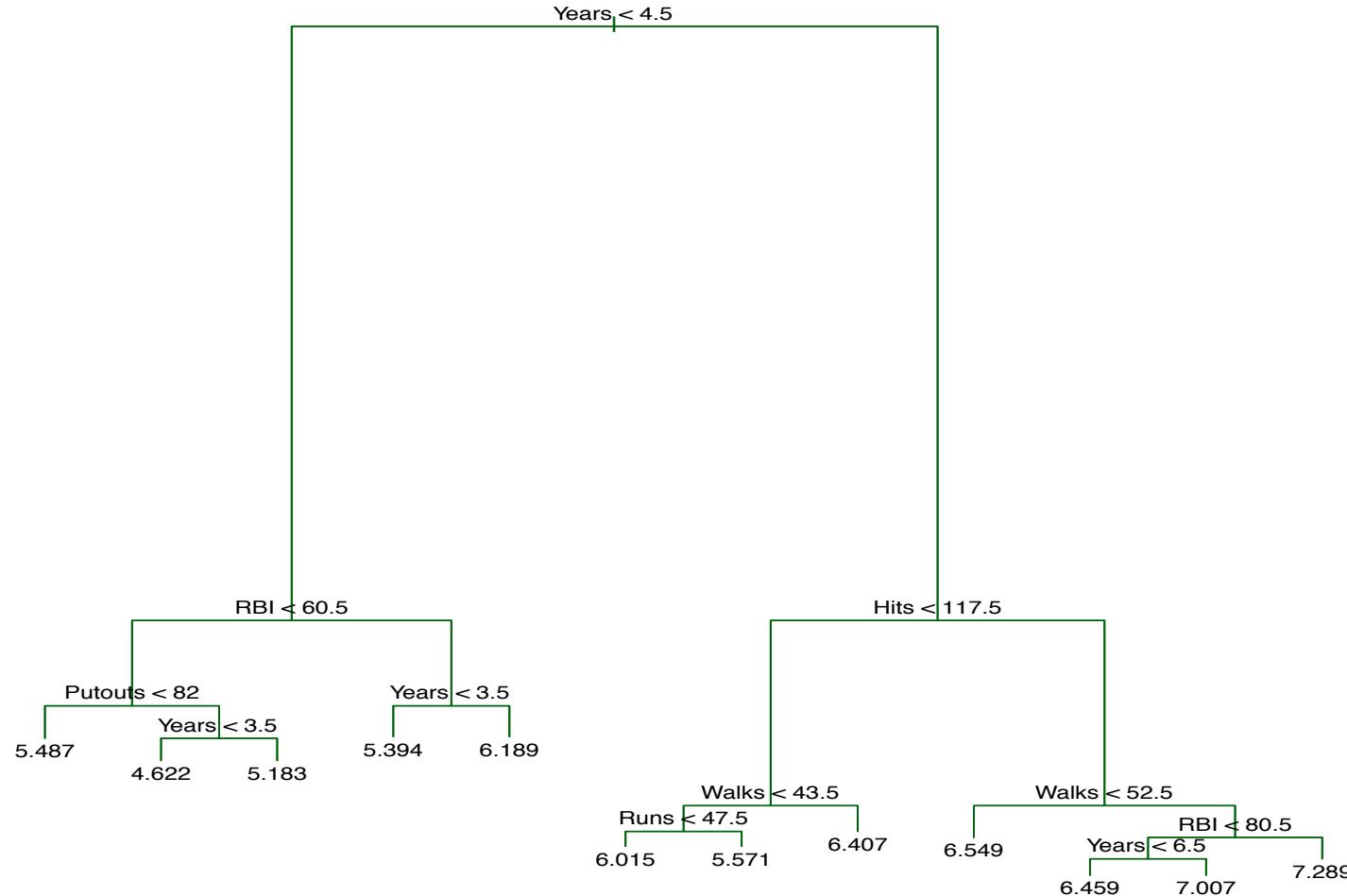
- A large tree (i.e. one with many terminal nodes) may tend to overfit the training data since you could continue to split until each observation had its own leaf node.
- Generally, we can improve estimated test set accuracy by “pruning” the tree i.e. cutting off some of the terminal nodes.
- How do we know how far back to prune the tree? We use **cross validation** to see which subtree has the lowest validation error rate.

Pruning Example: Baseball Players' Salaries

- The minimum cross validation error occurs at a tree size of 3



Pruning Example: Baseball Players' Salaries



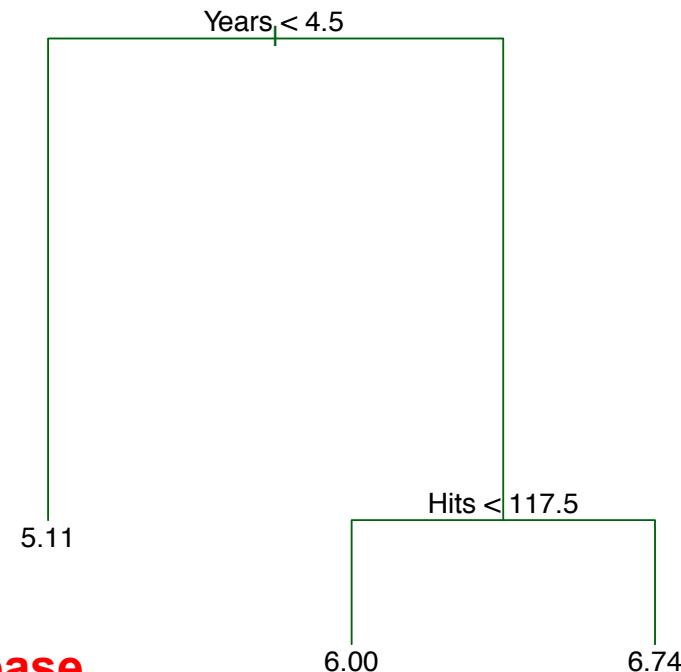
Pruning Example: Baseball Players' Salaries

- Even though training MSE continues to decrease with additional leaf nodes...
- Cross Validation indicates that the minimum CV MSE is when the tree size is three (i.e. the number of leaf nodes is 3)

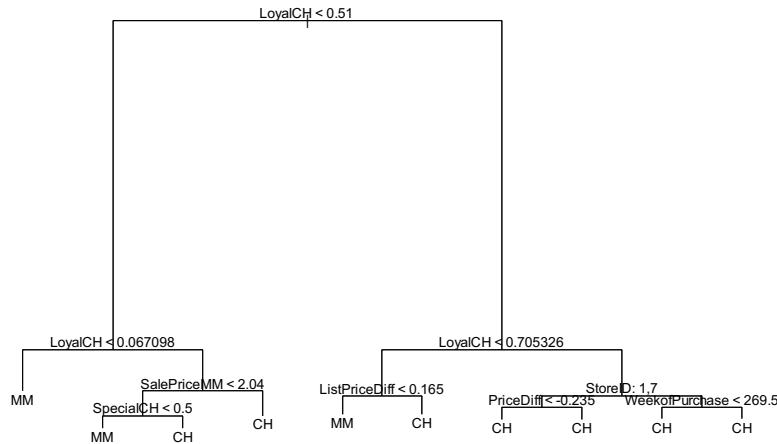
Concept Check:

Why does training MSE continue to decrease as trees gain additional leaf nodes?

Why does cross-val prefer shallower trees?



Pruning Example: Orange Juice Preference

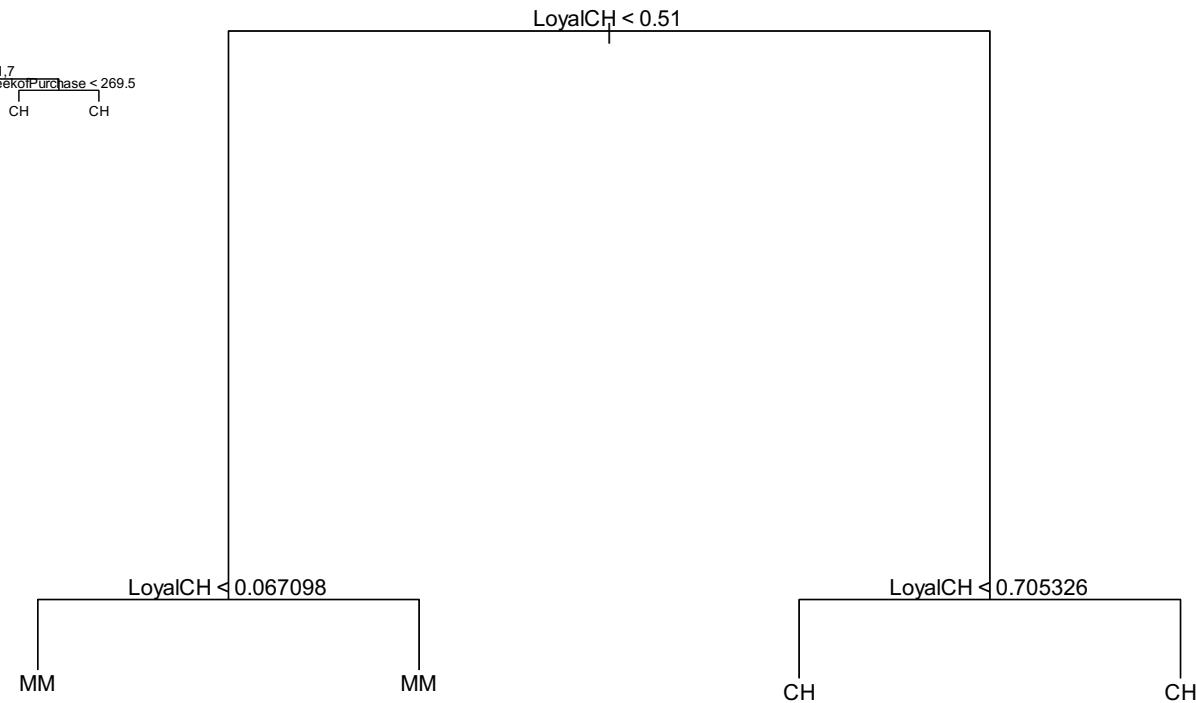


Full Tree Training
Error Rate = 14.75%

Full Tree Test Error
Rate = 23.6%

Pruned Tree

CV Tree Error Rate = 22.5%



TREES VS. LINEAR MODELS

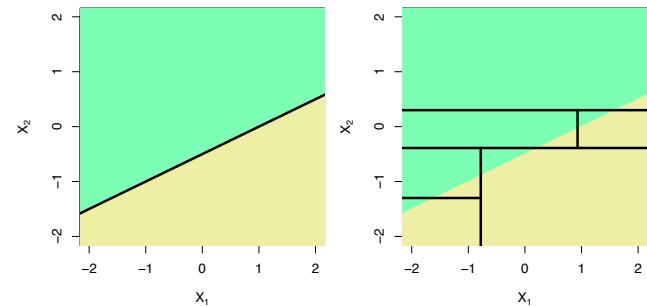
Trees vs. Linear Models

- Which model is better?
 - If the relationship between the predictors and response is linear, then classical linear models such as linear regression would outperform regression trees
 - On the other hand, if the relationship between the predictors is non-linear, then decision trees can outperform linear approaches

Trees vs. Linear Model: Classification Example

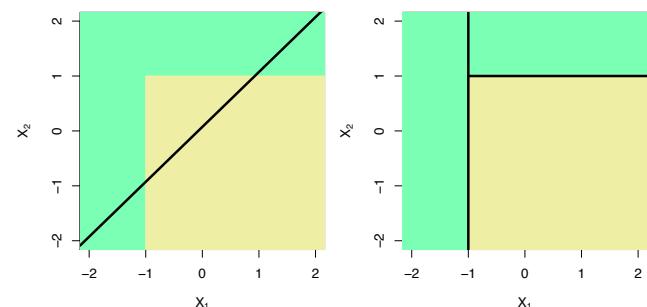
- Top row: the true decision boundary is linear

- Left: linear model (good)
- Right: decision tree



- Bottom row: the true decision boundary is non-linear

- Left: linear model
- Right: decision tree (good)



ADVANTAGES AND DISADVANTAGES OF TREES

Pros and Cons of Decision Trees

- Pros:
 - Trees are very easy to interpret (probably even easier than linear regression)
 - Trees can be plotted graphically, and are easily interpreted even by non-expert
 - Trees handle both classification and regression problems
- Cons:
 - Trees don't have the same prediction accuracy as some of the more complicated approaches in ML

In Class Exercise Part 1

- Given a set of datapoints in a region, write code for making a single splitting decision based on minimizing MSE. You should write this as a function call where you are provided
 - A matrix X of predictor values (n rows containing X_1 through X_P) which fall into the current region
 - vector of real valued labels (Y), one value per row
- Use i to index through the observations
- Use j to index through the predictors
- Use s to represent the splitting threshold for the chosen predictor
- Return the values for j and s which minimizes training set MSE
- Challenge: Can you vectorize it?
- Call your function `splitRegion(df)`
 - Return `(j, s, yLow, yHigh, dfLow, dfHigh)`

In Class Exercise Part 2

- Add code (to wrap your code from part 1) to allow consideration for determining which of the regions in the tree affords the best next split. You should write this as a function where you get:
 - The current tree which partitions the space into regions 1..r
 - The training set MSE for this tree
 - A function “region(T, k)” which returns the matrix X of observations belonging in region k of the tree T , along with a vector of true Y values for those observations.
- Return the region (k), split feature (j) and threshold (s) which minimizes the training set MSE over the choice of region, feature and threshold

BAGGING, RANDOM FORESTS, BOOSTING

Chapter 08 (part 02)

Outline

- Bagging
 - Bootstrapping
 - Bagging for Regression Trees
 - Bagging for Classification Trees
 - Out-of-Bag Error Estimation
 - Variable Importance: Relative Influence Plots
- Random Forests
- Boosting

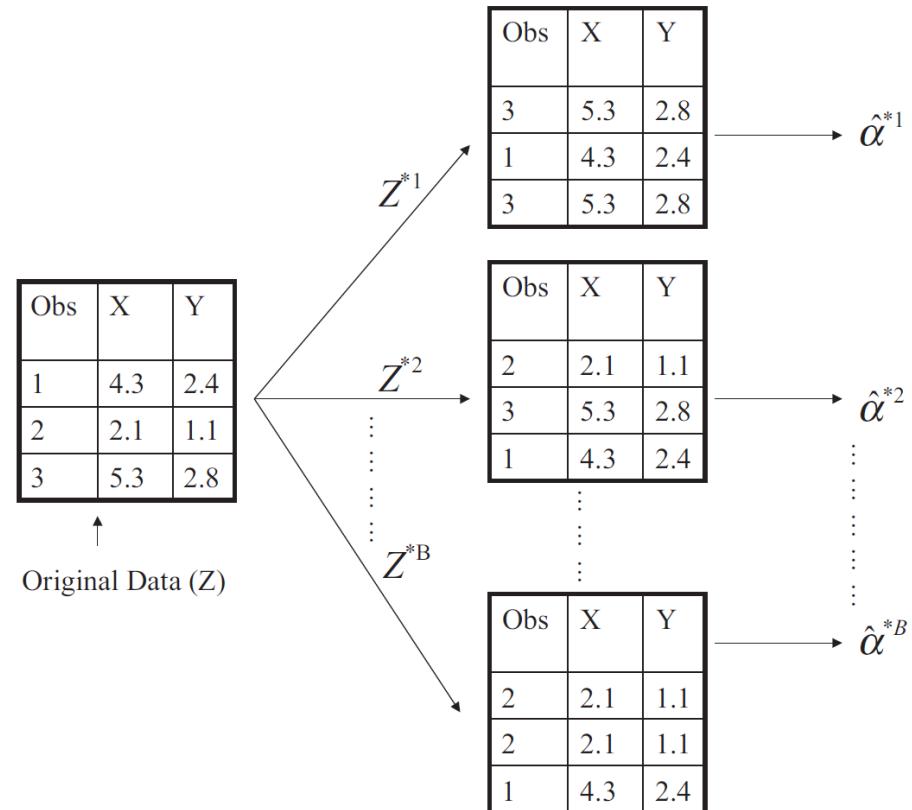
BAGGING (Bootstrap AGGregatIING)

Problem with Decision Trees

- (basic) Decision trees suffer from high variance
 - If we randomly split the training data into 2 parts, and fit separate decision trees on both parts, the results could be quite different
- We would like to have models with low variance
- To solve this problem, we can use bagging (bootstrap aggregating).
- Bagging is a general process for machine learning
 - It could be a preprocessing step for any model

Bootstrapping Revisited

- Resample the observed dataset to obtain a new set of data equal to the size to the observed dataset
- Each new observation is obtained by random sampling with replacement from the original dataset.



What is bagging?

- Bagging is an extremely powerful idea based on two things:
 - Averaging: reduce variance
 - Bootstrapping: generate many alternative training datasets
- Why does averaging reduce variance?
 - Averaging a set of observations reduces variance. Recall that given a set of n independent observations Z_1, \dots, Z_n , each with variance σ^2 , the variance of the mean \bar{Z} of the observations is given by σ^2/n

How does bagging training work?

1. Generate B different bootstrapped training datasets
2. Train the model separately on each of the B training datasets, to obtain B different models.

Bagging for Regression Trees

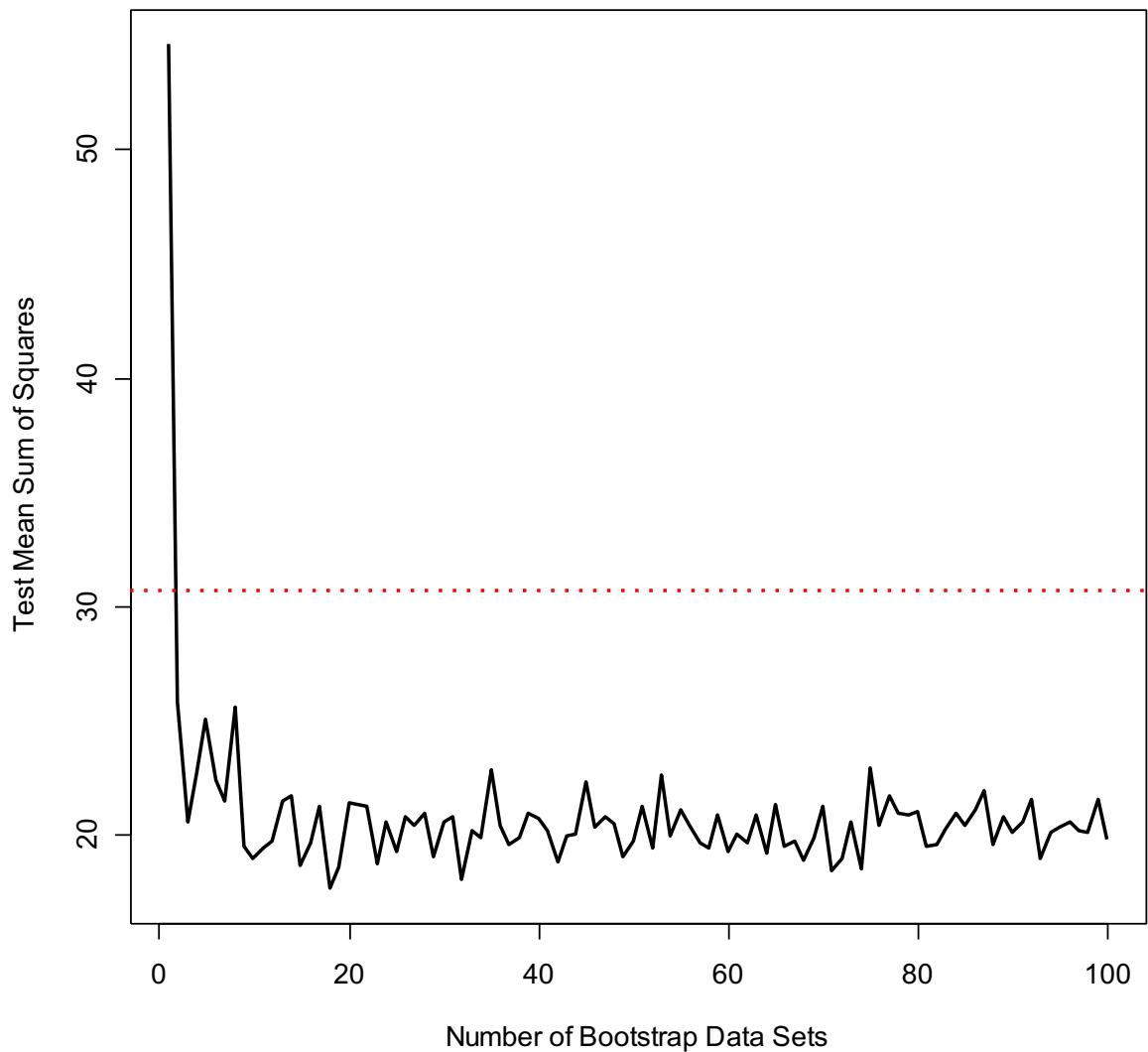
- Construct B regression trees using B bootstrapped training datasets. Do not prune the trees
- Average (or vote) to determine the resulting predictions

Bagging for Classification Trees

- Construct B regression trees using B bootstrapped training datasets
- For prediction, there are two approaches:
 1. Record the class that each bootstrapped data set predicts and provide an overall prediction to the most commonly occurring one (majority vote).
 2. If our classifier produces probability estimates we can just average the probabilities and then predict to the class with the highest probability.
- Both methods work well

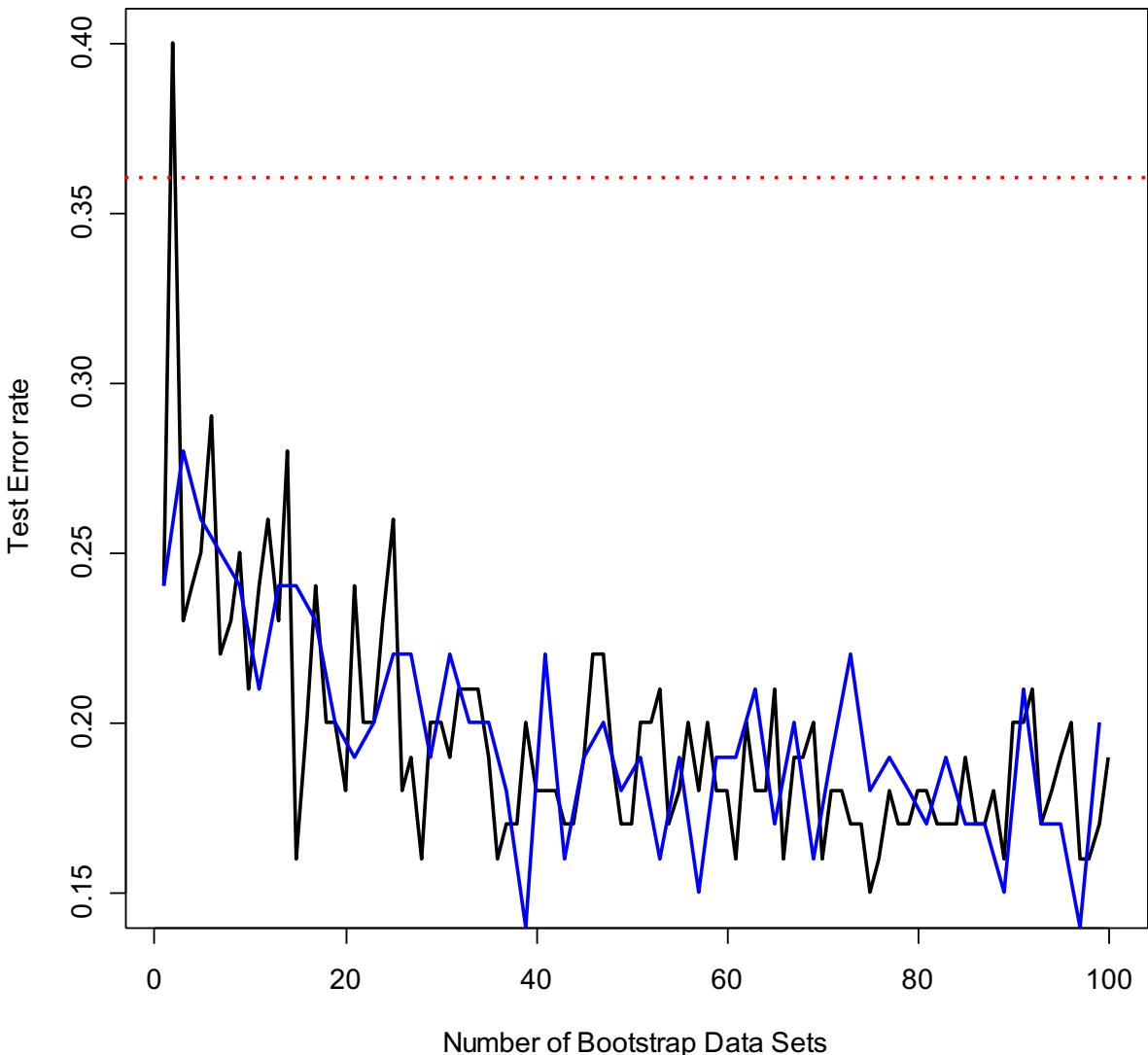
Example 1: Housing Data (Regression)

- Red line:
test mean sum of
squares using a
single tree.
- Black line:
bagging test error
rate



Example 2: Car Seat Data (Classification)

- The red line represents the test error rate using a single tree.
- The black line corresponds to the bagging error rate using majority vote
- The blue line averages the class probabilities before deciding class.



Out-of-Bag Error Estimation

- Bootstrapping randomly selects a subset of observations to train each tree
- The remaining non-selected subset could be used to estimate performance on unseen data
- On average, each bagged tree makes use of around 2/3 of the observations, so we end up having 1/3 of the observations used for estimating performance

Variable Importance Measure

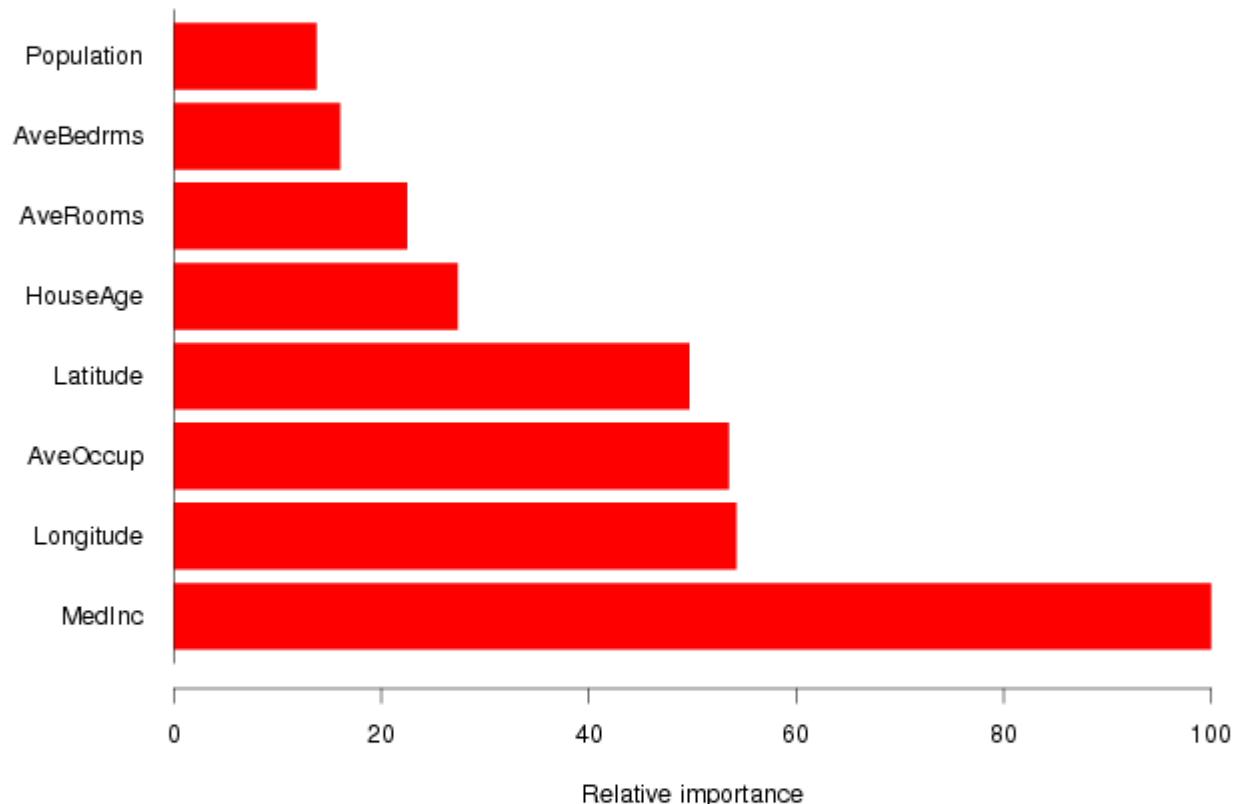
- Bagging typically improves the accuracy over prediction using a single tree, but it is harder to interpret the model
- We have hundreds of trees, and it is no longer clear which variables are most important to the procedure
- Thus bagging improves prediction accuracy at the expense of interpretability
- But, we can still get an overall summary of the importance of each predictor using Relative Influence Plots

Inference with multiple trees

- Which variables are most useful in predicting the response?
 - Relative influence plots give a score for each variable.
 - These scores represent the decrease in MSE (or Gini Index for classification) when splitting on a particular variable
 - The most influential variable is given a value of 100 and other variables are shown as a relative fraction of the influence of the most influential variable
 - The larger the score the more influence the variable has
 - A number close to zero indicates the variable is not important and could be dropped

Example: Housing Data

- Median Income is by far the most important variable.
- Longitude, Latitude and Average occupancy are the next most important.



Problems with Bagging?

- If there is a very strong predictor in the data set along with a number of other moderately strong predictors, then in the collection of bagged trees, most or all of them will use the very strong predictor for the first split
- All bagged trees will look similar. Hence all the predictions from the bagged trees will be **highly correlated**
- Averaging many highly correlated quantities does not lead to a large variance reduction
- If we want to reduce the variance, we need something to do something to break the correlation of outputs

RANDOM FORESTS

Random Forests

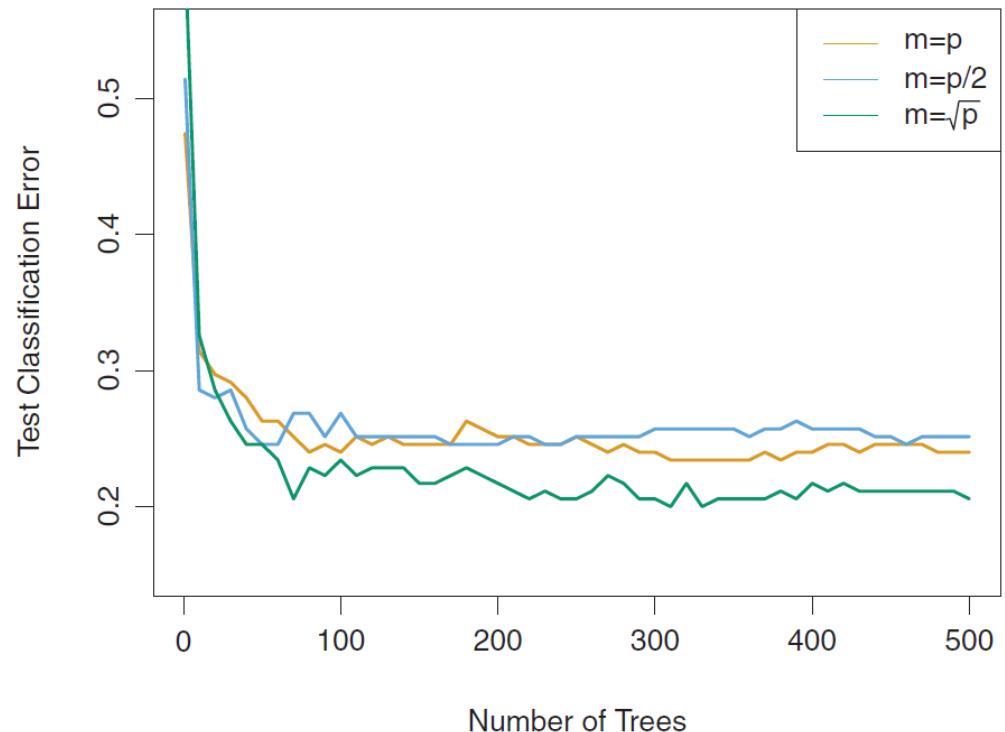
- A very efficient statistical learning method
- Builds on the idea of bagging, but it provides an improvement because it *de-correlates* the trees
- How does it work?
 - Build a number of decision trees on bootstrapped training sample, but when building these trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors (Usually $m \approx \sqrt{p}$)

Why are we considering a random sample of m predictors instead of all p predictors for splitting?

- If each tree has a different subset of p predictors, then the trees won't get stuck always picking the same best predictor if one of the p predictors was stronger than most others
- This means each tree will grow differently and a set of trees grown this way will have less correlation in their outputs
- Random forests “de-correlate” the outputs that bagged trees would have generated... leading to more reduction in variance

Random Forest Hyperparameter “m”

- m is the number of features randomly selected for use in each tree
- If random forests are set $m = p$, this is equivalent to bagging
- Empirically, \sqrt{p} often works well



Boosting – Random Forest mod

- Idea: build a forest of B trees *incrementally*, but when building the next tree, instead of fitting a model to best predict Y , attempt to best predict the *residual* error remaining in the forest
- Use crossval to determine a good size for the forest (B)

This algorithm learns
slowly
Smaller lambda leads to
slower learning... which
means lower variance

Algorithm 8.2 Boosting for Regression Trees

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \dots, B$, repeat:
 - (a) Fit a tree \hat{f}^b with d splits ($d + 1$ terminal nodes) to the training data (X, r) .
 - (b) Update \hat{f} by adding in a shrunken version of the new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \quad (8.10)$$

- (c) Update the residuals,

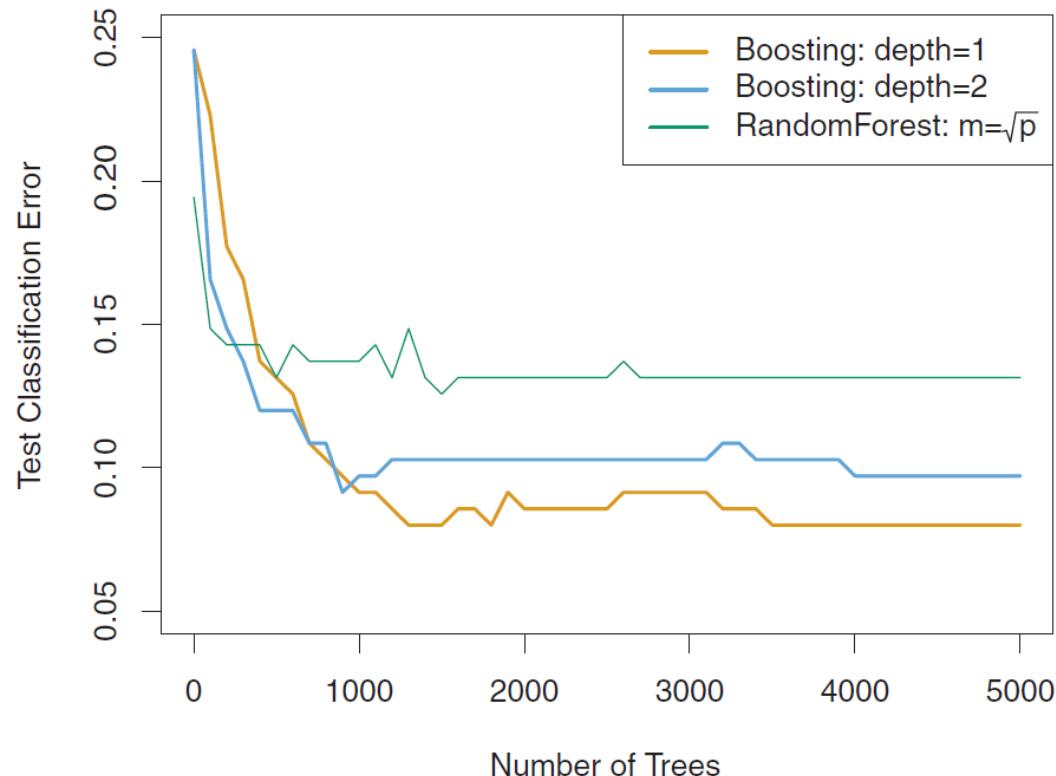
$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \quad (8.11)$$

3. Output the boosted model,

$$\hat{f}(x) = \sum_{b=1}^B \lambda \hat{f}^b(x). \quad (8.12)$$

Boosting

- d is the interaction depth
- $d = 1$ gives us a *stump* with two leaf nodes
- Boosted stumps tend to perform well



Trees & Forest Summary

- Trees and Forests offer non-linear machine learning models which can answer both prediction and inference questions
- There are *many* alternative tree & forest model architectures
- Each architecture has a set of hyperparameters to determine
- Cross-validation should be used to make model and hyperparameter decisions... but this increases the computational cost of finding a good model

Support Vector Machines (lite version)

SOURCES

The first half of
the slides are largely borrowed from
Prof. Andrew Moore's
2001 SVM tutorial at

<http://www.cs.cmu.edu/~awm/tutorials>

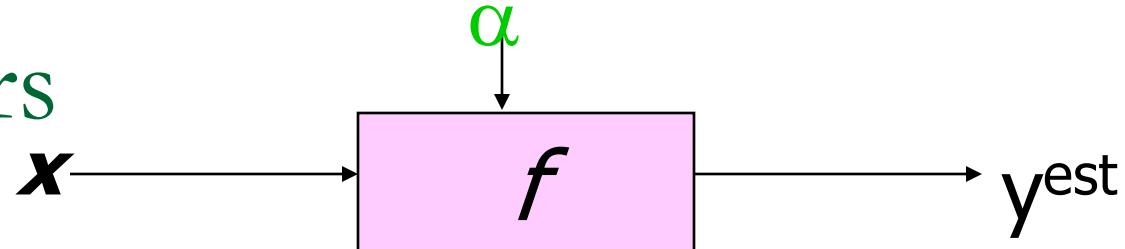
The remainder of this content
is inspired by a slide set by
Mingyue Tan
The University of British Columbia
Nov 26, 2004

www.baskent.edu.tr/~hogul/svm_tutorial.ppt

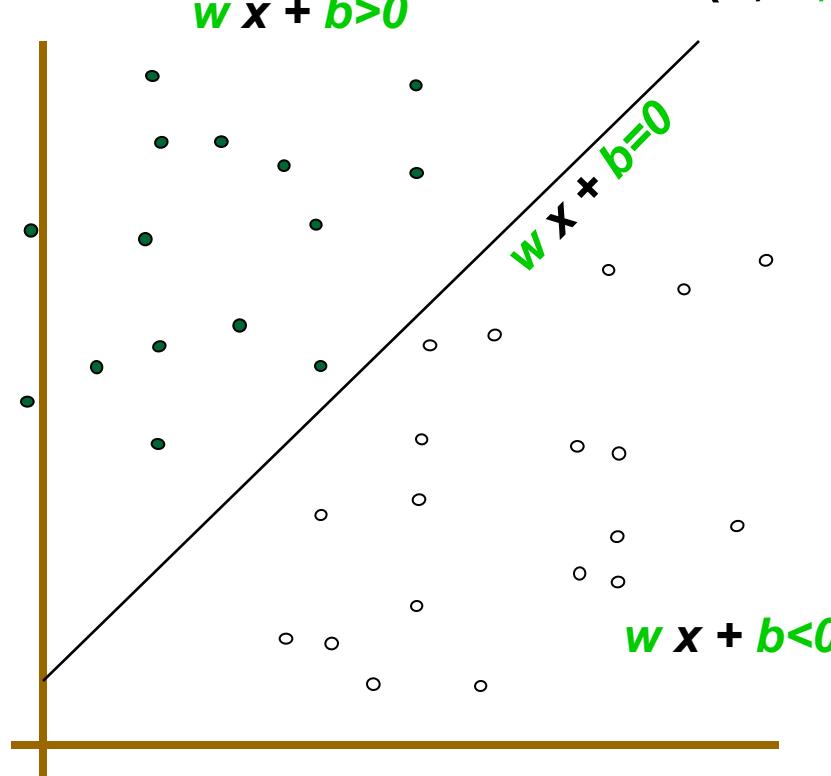
Overview

- Intro. to Support Vector Machines (SVM)
- Properties of SVM
- SVM Application considerations
- Coding Exercise

Linear Classifiers



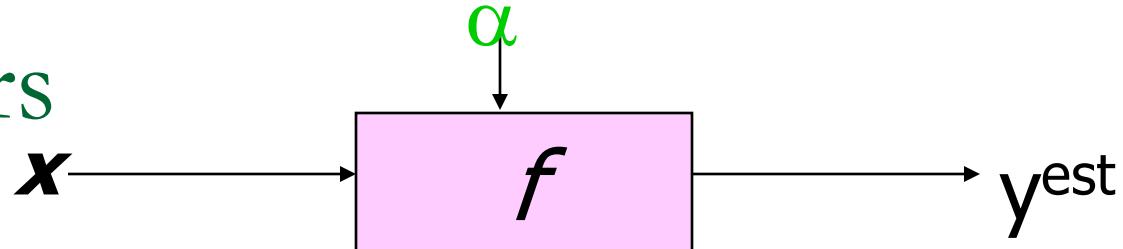
- denotes +1
- denotes -1



$$f(\mathbf{x}, w, b) = \text{sign}(w \mathbf{x} + b)$$

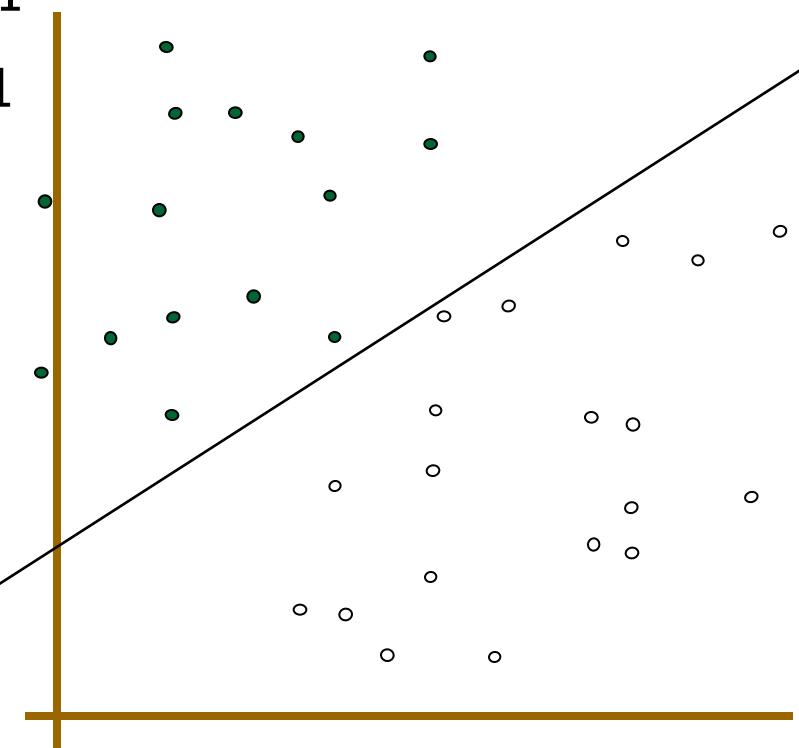
How would you
classify this data?

Linear Classifiers



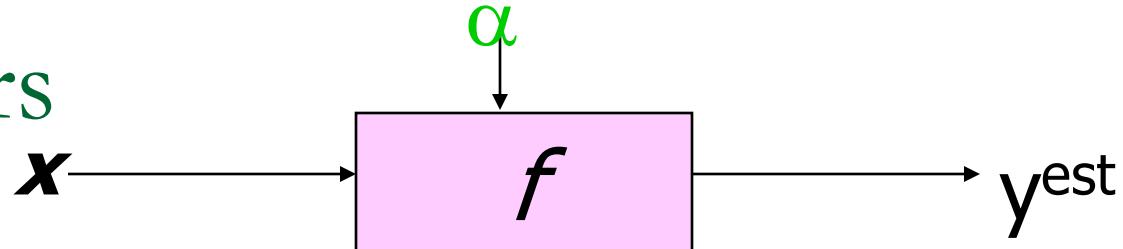
- denotes +1
- denotes -1

$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

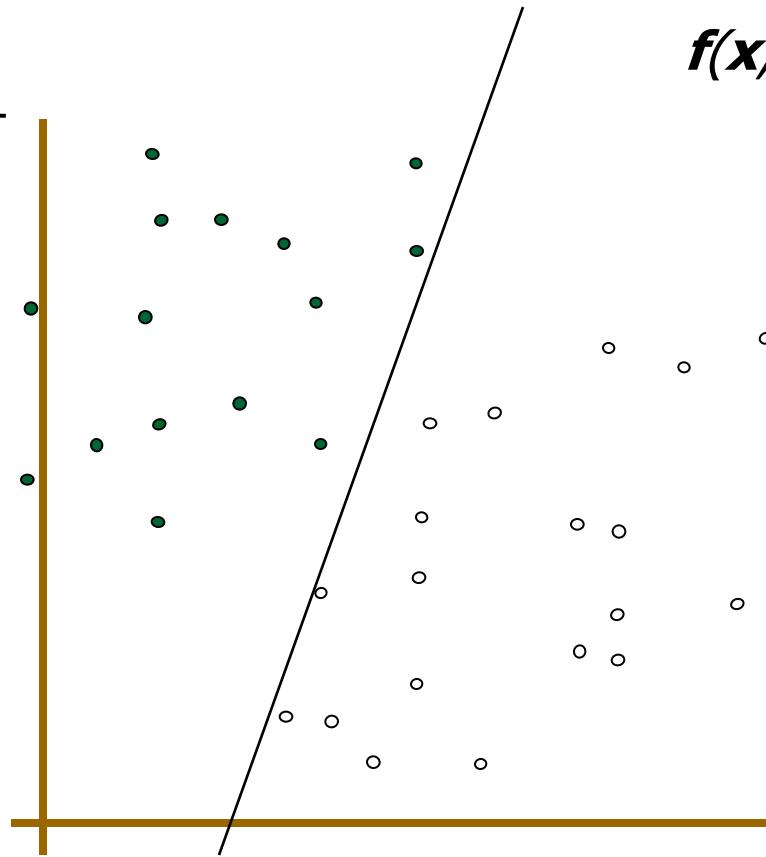


How would you
classify this data?

Linear Classifiers



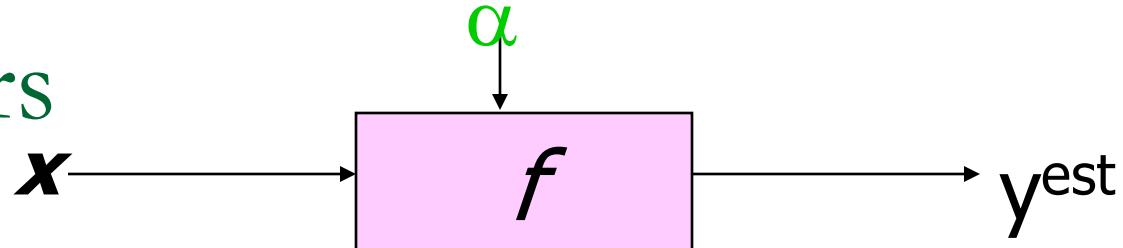
- denotes +1
- denotes -1



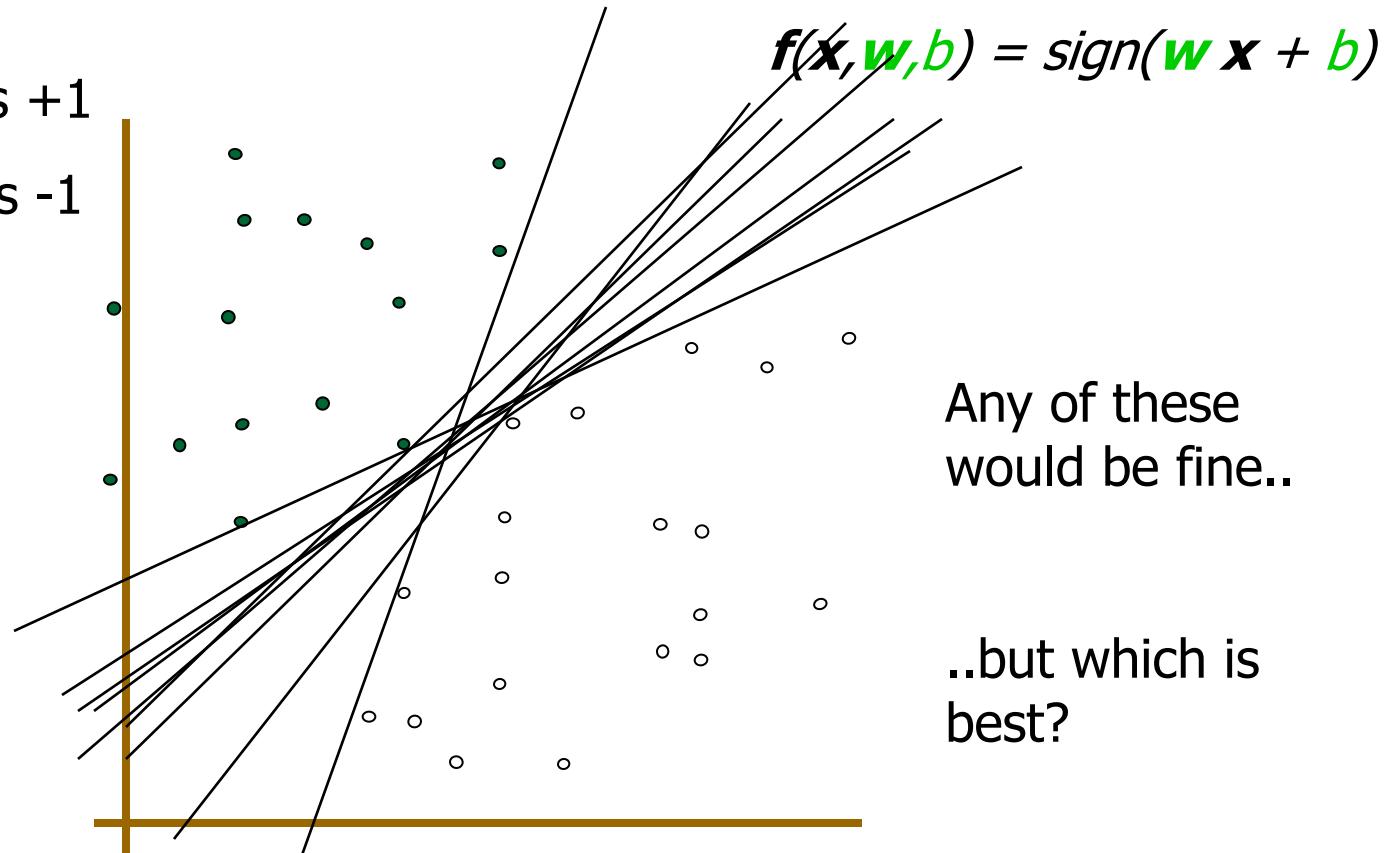
$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

How would you
classify this data?

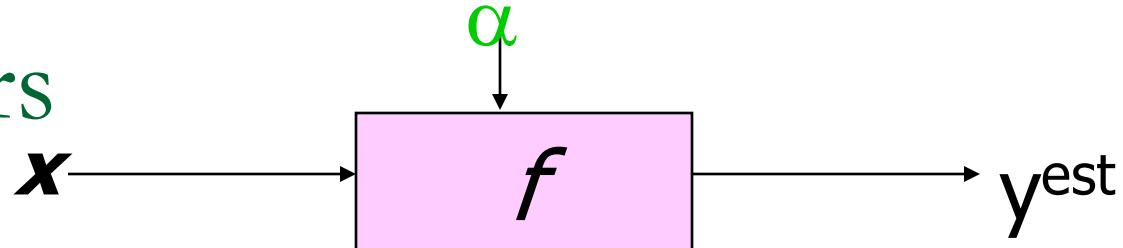
Linear Classifiers



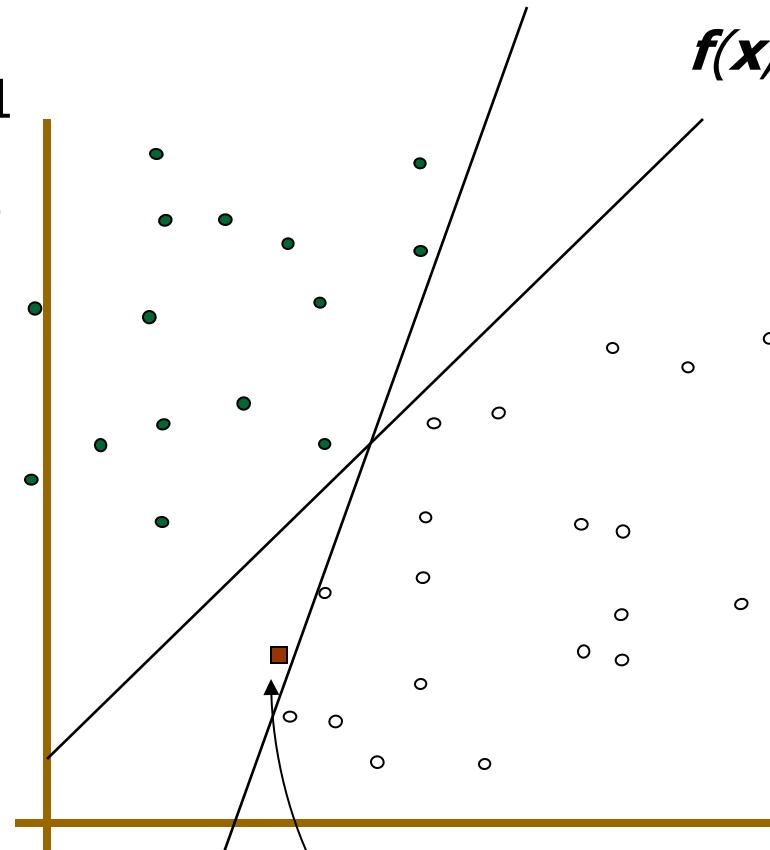
- denotes +1
- denotes -1



Linear Classifiers



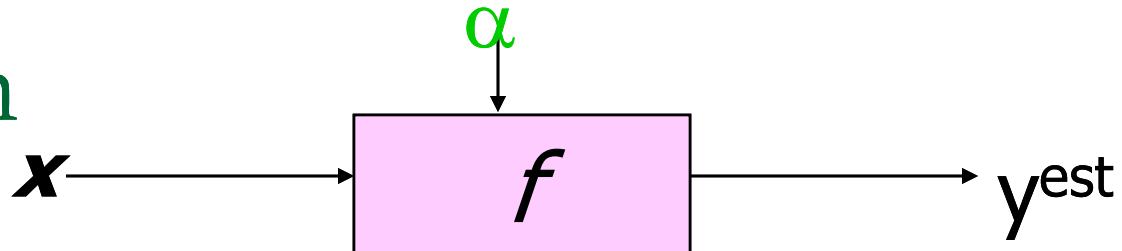
- denotes +1
- denotes -1



How would you
classify this data?

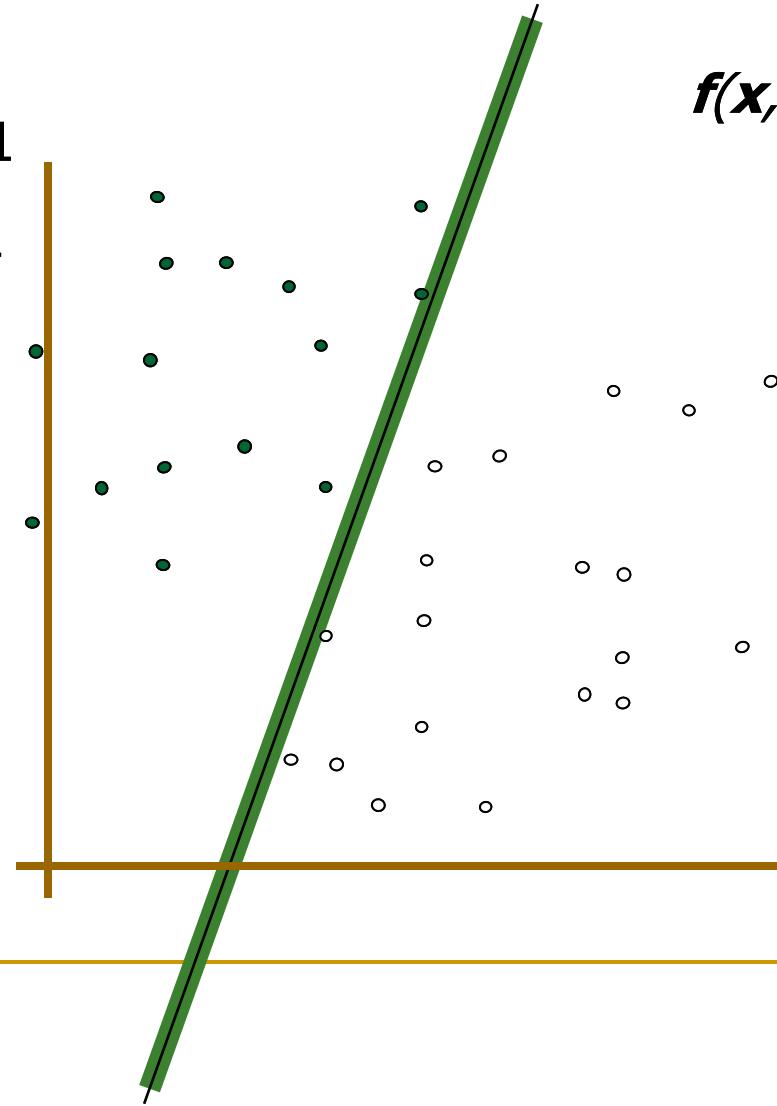
Misclassified
to +1 class

Classifier Margin



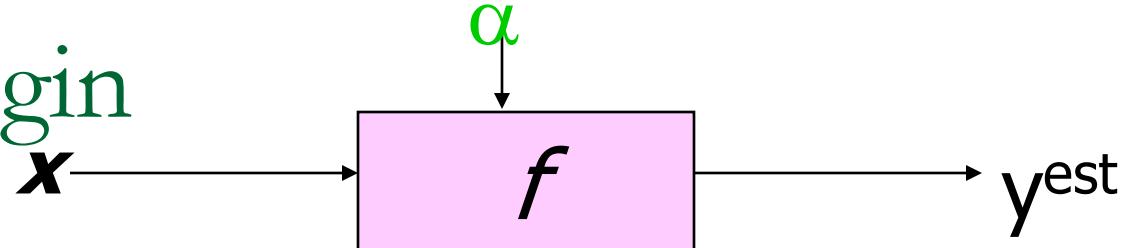
$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

- denotes +1
- denotes -1



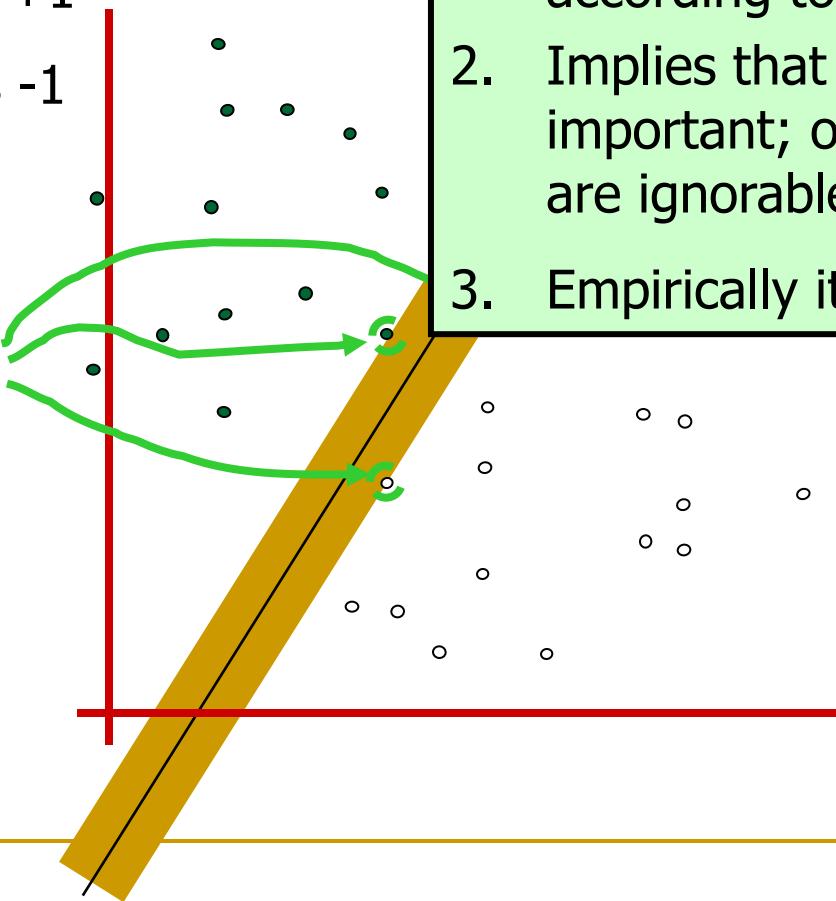
Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin



- denotes +1
- denotes -1

Support Vectors
are those
datapoints that
the margin
pushes up
against

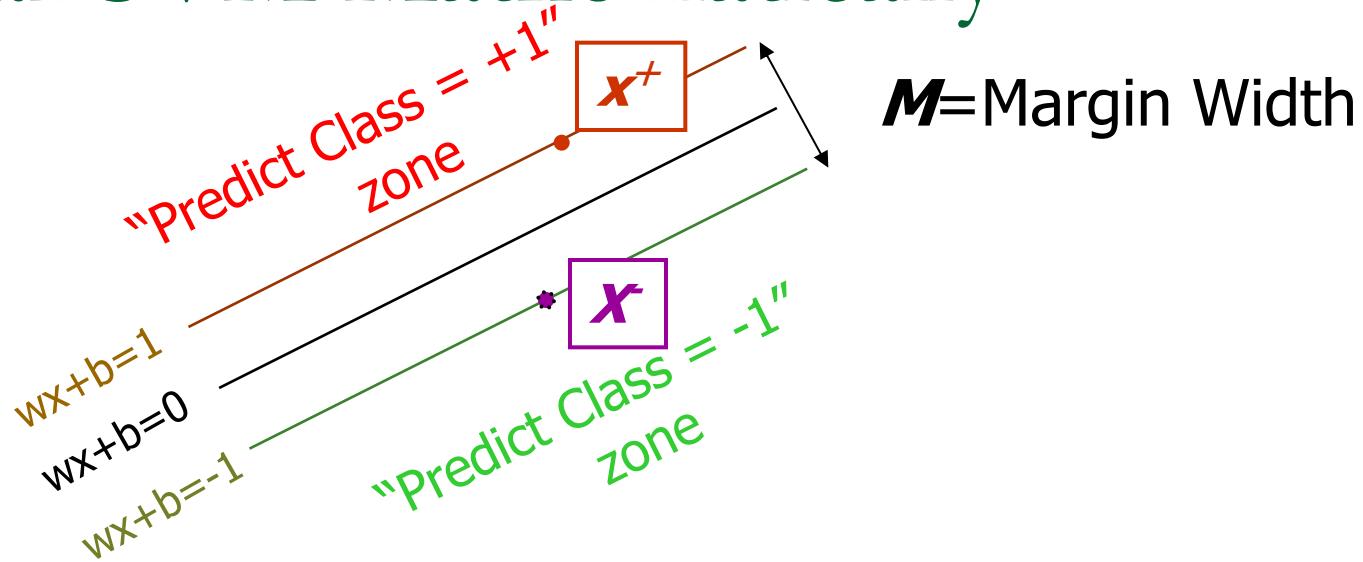


1. Maximizing the margin is good according to intuition
2. Implies that only support vectors are important; other training examples are ignorable.
3. Empirically it works very very well.

with the maximum margin.

This is the simplest kind of SVC

Linear SVM Mathematically



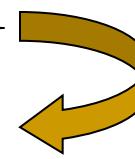
What we know:

- $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
- $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
- $\mathbf{w} \cdot (\mathbf{x}^+ - \mathbf{x}^-) = 2$

$$M = \frac{(\mathbf{x}^+ - \mathbf{x}^-) \cdot \mathbf{w}}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$$

Linear SVM Mathematically

- Goal: 1) Correctly classify all training data

$$\begin{aligned} w\mathbf{x}_i + b &\geq 1 & \text{if } y_i = +1 \\ w\mathbf{x}_i + b &\leq -1 & \text{if } y_i = -1 \\ y_i(w\mathbf{x}_i + b) &\geq 1 & \text{for all } i \end{aligned} \quad \left. \right\}$$


2) Maximize the Margin $M = \frac{2}{\|\mathbf{w}\|}$

same as minimize $\frac{1}{2} \mathbf{w}^t \mathbf{w}$

- We can formulate a Quadratic Optimization Problem and solve for w and b

Minimize $\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^t \mathbf{w}$

subject to $y_i(w\mathbf{x}_i + b) \geq 1 \quad \forall i$

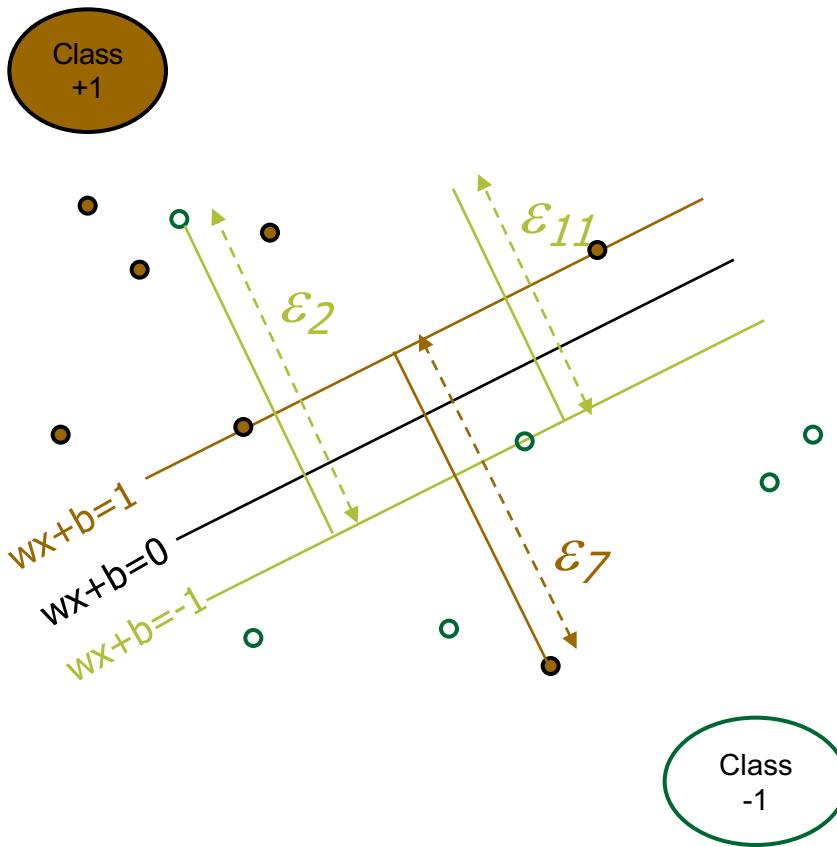
Out of scope
for this class

Margins so far...

- To this point we've considered “Hard” margins
 - During training, penalizes only the misclassified observations within the margin
 - Points within the margin are the only thing that matter for decision boundary computation
 - If the decision boundary relies on only a few observations: increasing the variance – possibility of overfitting goes up

Soft Margin Classification

Slack variables ξ_i can be added to allow misclassification of difficult or noisy examples.



What should our quadratic optimization criterion be?

Minimize

$$\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{k=1}^R \varepsilon_k$$

Hard Margin v.s. Soft Margin

- **The old formulation:**

Find \mathbf{w} and b such that

$$\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \text{ is minimized and for all } \{(\mathbf{x}_i, y_i)\}$$

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$$

- **The new formulation incorporating slack variables:**

Find \mathbf{w} and b such that

$$\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum \xi_i \text{ is minimized and for all } \{(\mathbf{x}_i, y_i)\}$$

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \text{ for all } i$$

- **Parameter C can be viewed as a way to control overfitting through regularization**

Linear SVMs: Overview

- The classifier is a *separating hyperplane*.
- Most “important” training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i .
- Both in the dual formulation of the problem and in the solution training points appear only inside dot products:

Find $\alpha_1 \dots \alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j x_i^T x_j$ is maximized and

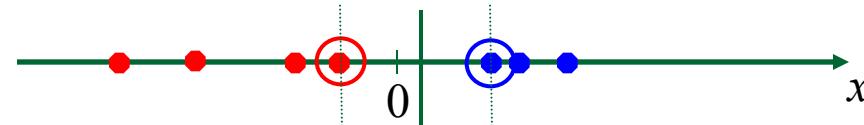
$$(1) \sum \alpha_i y_i = 0$$

$$(2) 0 \leq \alpha_i \leq C \text{ for all } \alpha_i$$

$$f(x) = \sum \alpha_i y_i x_i^T x + b$$

Non-linear SVMs

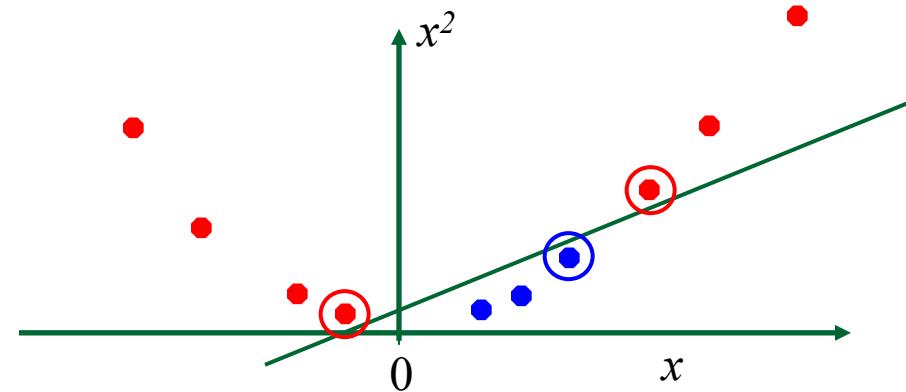
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is not linearly separable?

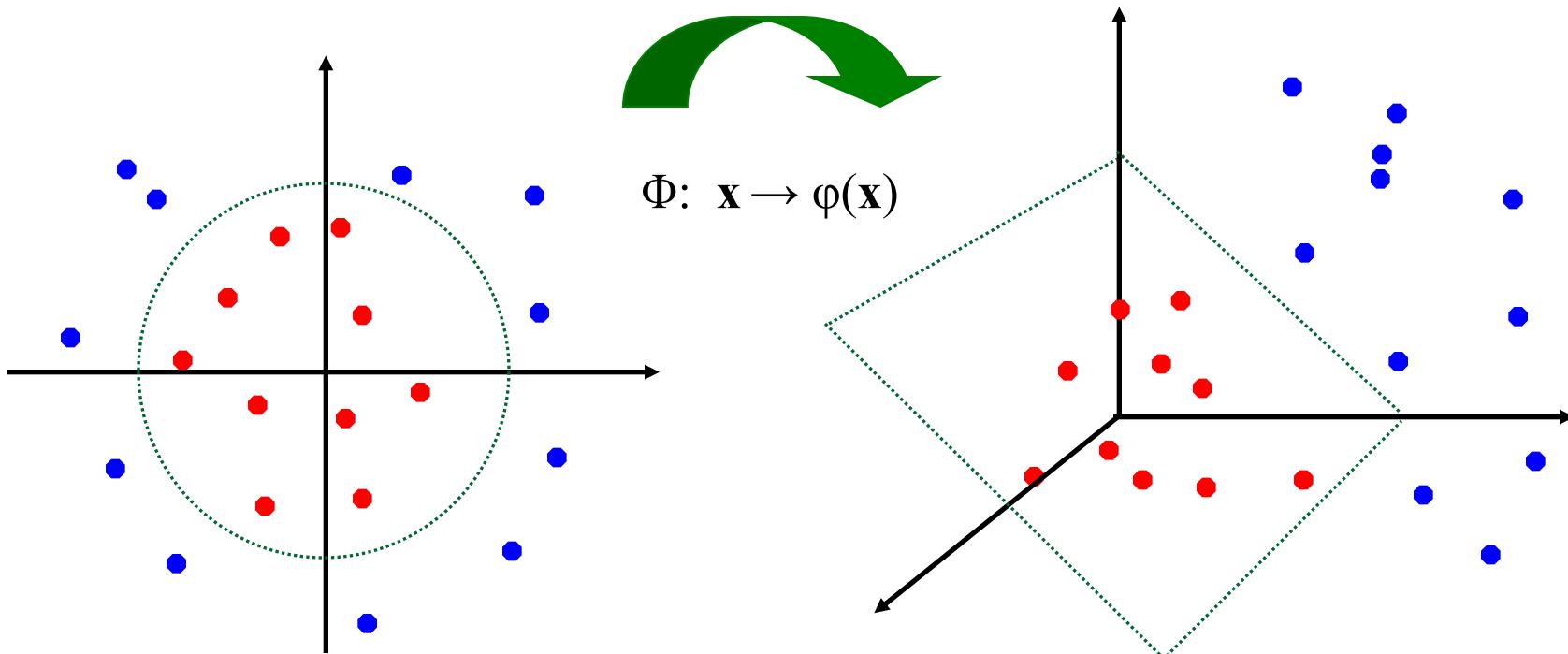


- How about... mapping data to a higher-dimensional space and *then* applying a linear classifier:



Non-linear SVMs: Feature spaces

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



The “Kernel Trick”

- The linear classifier relies on dot product between vectors $K(x_i, x_j) = x_i^T x_j$
- If every data point is mapped into high-dimensional space via some transformation $\Phi: x \rightarrow \phi(x)$, the dot product becomes:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.
- Example:

2-dimensional vectors $x = [x_1 \ x_2]$; let $K(x_i, x_j) = (1 + x_i^T x_j)^2$,

Need to show that $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$:

$$\begin{aligned} K(x_i, x_j) &= (1 + x_i^T x_j)^2, \\ &= 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2} \\ &= [1 \ x_{i1}^2 \ \sqrt{2} x_{i1} x_{i2} \ x_{i2}^2 \ \sqrt{2} x_{i1} \ \sqrt{2} x_{i2}]^T [1 \ x_{j1}^2 \ \sqrt{2} x_{j1} x_{j2} \ x_{j2}^2 \ \sqrt{2} x_{j1} \ \sqrt{2} x_{j2}] \\ &= \phi(x_i)^T \phi(x_j), \quad \text{where } \phi(x) = [1 \ x_1^2 \ \sqrt{2} x_1 x_2 \ x_2^2 \ \sqrt{2} x_1 \ \sqrt{2} x_2] \end{aligned}$$

Common Kernel Functions

- Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- Polynomial of order p : $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
- Gaussian (radial-basis function network):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

- Sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$

Non-linear SVMs Mathematically

- Dual problem formulation:

Find $\alpha_1 \dots \alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j K(x_i, x_j)$ is maximized and

(1) $\sum \alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ for all α_i

- The solution is:

$$f(x) = \sum \alpha_i y_i K(x_i, x) + b$$

- Optimization techniques for finding α_i 's remain the same!

Nonlinear SVM - Overview

- SVM locates a separating hyperplane in the feature space and classify points in that space
- It does not need to represent the space explicitly... all that is required is defining a kernel function
- The kernel function plays the role of the dot product in the feature space.

Properties of SVM

- Flexibility in choosing a similarity function
- Sparseness of solution when dealing with large data sets
only support vectors are used to specify the separating hyperplane
- Ability to handle large feature spaces
complexity does not depend on the dimensionality of the feature space
- Overfitting can be controlled by soft margin approach
- Nice math property: a simple convex optimization problem which is guaranteed to converge to a single global solution
- Feature Selection

Some Issues

- Choice of kernel
 - Gaussian or polynomial kernel is default
 - if ineffective, more elaborate kernels are needed
 - domain experts can give assistance in formulating appropriate similarity measures
- Choice of kernel parameters
 - e.g. σ in Gaussian kernel
 - σ is the distance between closest points with different classifications
 - In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters.
- Optimization criterion – Hard margin v.s. Soft margin
 - a lengthy series of experiments in which various parameters are tested

In Class Exercise

Backup Slides

- References
- Optimization Notes
- Applications

Additional Resources

- **An excellent tutorial on VC-dimension and Support Vector Machines:**
C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):955-974, 1998.
- **The VC/SRM/SVM Bible:**
Statistical Learning Theory by Vladimir Vapnik, Wiley-Interscience; 1998

<http://www.kernel-machines.org/>

Reference

- **Support Vector Machine Classification of Microarray Gene Expression Data**, Michael P. S. Brown William Noble Grundy, David Lin, Nello Cristianini, Charles Sugnet, Manuel Ares, Jr., David Haussler
- www.cs.utexas.edu/users/mooney/cs391L/svm.ppt
- **Text categorization with Support Vector Machines:
learning with many relevant features**
T. Joachims, ECML - 98

Solving the Optimization Problem

Find \mathbf{w} and b such that

$\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}$ is minimized;

and for all $\{(\mathbf{x}_i, y_i)\}$: $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

- Need to optimize a *quadratic* function subject to *linear* constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.
- The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:

Find $\alpha_1 \dots \alpha_N$ such that

$Q(\boldsymbol{\alpha}) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ is maximized and

$$(1) \quad \sum \alpha_i y_i = 0$$

$$(2) \quad \alpha_i \geq 0 \text{ for all } \alpha_i$$

The Optimization Problem Solution

- The solution has the form:

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i \quad b = y_k - \mathbf{w}^T \mathbf{x}_k \text{ for any } \mathbf{x}_k \text{ such that } \alpha_k \neq 0$$

- Each non-zero α_i indicates that corresponding \mathbf{x}_i is a support vector.
- Then the classifying function will have the form:
$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$
- Notice that it relies on an *inner product* between the test point \mathbf{x} and the support vectors \mathbf{x}_i – we will return to this later.
- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x}_i^T \mathbf{x}_j$ between all pairs of training points.

Linear SVMs: Overview

- The classifier is a *separating hyperplane*.
- Most “important” training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points x_i are support vectors with non-zero Lagrangian multipliers α_i .
- Both in the dual formulation of the problem and in the solution training points appear only inside dot products:

Find $\alpha_1 \dots \alpha_N$ such that

$Q(\alpha) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j x_i^T x_j$ is maximized and

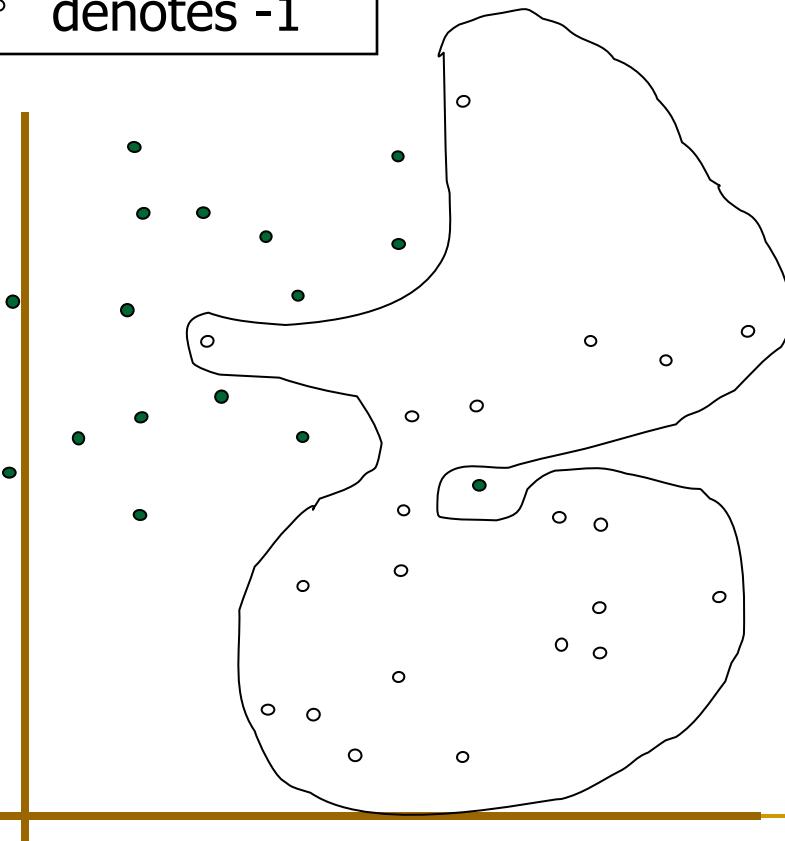
$$(1) \sum \alpha_i y_i = 0$$

$$(2) 0 \leq \alpha_i \leq C \text{ for all } \alpha_i$$

$$f(x) = \sum \alpha_i y_i x_i^T x + b$$

Dataset with noise

- denotes +1
- denotes -1



- **Hard Margin:** So far we require all data points be classified correctly
 - No training error
- **What if the training set is noisy?**
 - **Solution 1:** use very powerful kernels

OVERFITTING!

What Functions are Kernels?

- For some functions $K(\mathbf{x}_i, \mathbf{x}_j)$ checking that

$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ can be cumbersome.

- Mercer's theorem:

Every semi-positive definite symmetric function is a kernel

- Semi-positive definite symmetric functions correspond to a semi-positive definite symmetric Gram matrix:

$K =$

$K(\mathbf{x}_1, \mathbf{x}_1)$	$K(\mathbf{x}_1, \mathbf{x}_2)$	$K(\mathbf{x}_1, \mathbf{x}_3)$...	$K(\mathbf{x}_1, \mathbf{x}_N)$
$K(\mathbf{x}_2, \mathbf{x}_1)$	$K(\mathbf{x}_2, \mathbf{x}_2)$	$K(\mathbf{x}_2, \mathbf{x}_3)$		$K(\mathbf{x}_2, \mathbf{x}_N)$
...
$K(\mathbf{x}_N, \mathbf{x}_1)$	$K(\mathbf{x}_N, \mathbf{x}_2)$	$K(\mathbf{x}_N, \mathbf{x}_3)$...	$K(\mathbf{x}_N, \mathbf{x}_N)$

SVM Applications

- SVM has been used successfully in many real-world problems
 - text (and hypertext) categorization
 - image classification
 - bioinformatics (Protein classification, Cancer classification)
 - hand-written character recognition

Application 1: Cancer Classification

- High Dimensional

- $p > 1000$; $n < 100$

- Imbalanced

- less positive samples

$$K[x, x] = k(x, x) + \lambda \frac{n^+}{N}$$

- Many irrelevant features

- Noisy

SVM is sensitive to noisy (mis-labeled) data ☺

		Genes			
Patients		g-1	g-2	g-p
P-1					
p-2					
.....					
p-n					

FEATURE SELECTION

In the linear case,
 w_i^2 gives the ranking of dim i

Weakness of SVM

- It is sensitive to noise
 - A relatively small number of mislabeled examples can dramatically decrease the performance
- It only considers two classes
 - how to do multi-class classification with SVM?
 - Answer:
 - 1) with output arity m, learn m SVM's
 - SVM 1 learns "Output==1" vs "Output != 1"
 - SVM 2 learns "Output==2" vs "Output != 2"
 - :
 - SVM m learns "Output==m" vs "Output != m"
 - 2) To predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

Application 2: Text Categorization

- Task: The classification of natural text (or hypertext) documents into a fixed number of predefined categories based on their content.
 - email filtering, web searching, sorting documents by topic, etc..
- A document can be assigned to more than one category, so this can be viewed as a series of binary classification problems, one for each category

Representation of Text

IR's vector space model (aka bag-of-words representation)

- A doc is represented by a vector indexed by a pre-fixed set or dictionary of terms
- Values of an entry can be binary or weights

$$\phi_i(x) = \frac{\text{tf}_i \log (\text{idf}_i)}{\kappa},$$

- Normalization, stop words, word stems
- Doc $x \Rightarrow \Phi(x)$

Text Categorization using SVM

- The distance between two documents is $\varphi(x) \cdot \varphi(z)$
- $K(x,z) = \langle \varphi(x) \cdot \varphi(z) \rangle$ is a valid kernel, SVM can be used with $K(x,z)$ for discrimination.
- Why SVM?
 - High dimensional input space
 - Few irrelevant features (dense concept)
 - Sparse document vectors (sparse instances)
 - Text categorization problems are linearly separable

UNSUPERVISED LEARNING

Chapter 10 (part 01)

Outline

- Intro to Unsupervised Learning
- Principal Components Analysis (PCA)
 - Goals
 - Implementation - Conceptual
 - Implementation - Math
 - Interpretations
 - Uses
- Clustering (in slides for Chapter 10, part 2)

Intro to unsupervised learning

- If you don't have a response variable, you can't make a function f of inputs to outputs
- Is there anything you *can* do with a bunch observations of predictors?

Unsupervised Learning

- Even without a response variable, you can still look at relationships within the observations
- There are a few things you can do without a response variable - including:
 - Interpret (visualize) relationships among observations through linear algebraic manipulations (rotations of the feature axes): PCA
 - Compress data through dimensionality reduction: PCA
 - Lump similar observations together: Clustering
 - Compress data by substituting cluster centers and distributions for observations: Clustering / Mixture models

PCA - Goals

- Since we have no response variable, we assume that differentiation (variance) among observations captures something meaningful in the domain
- We have a p -dimensional space of features, and assume some are more meaningful than others, but all may contribute something to the interpretation
- We wish to explain or summarize these differences with as few parameters as possible. **WHY?**

PCA Intuition – Conceptual (1 of 4)

- Scale all variables (Z-scaling)
- Define a *first* axis of highest variance in feature-space of the observations (this is **component 1**)
- Then we iterate until p -axes are selected:
 - Pick one of the remaining axes which are orthogonal to all previously selected axes ...
 - ...such that this selected axis is the one which has the maximum variance among its datapoints, given the previously-selected axes are fixed
 - When an axis is selected, it is appended to an ordered list of components

PCA Intuition – Conceptual (2 of 4)

- These orthogonal axes can be described in terms of *rotations* to the p axes in the original feature dimensions
- The rotations align the new PCA axes along lines of variance, from high to low
- The “first” component is the axis expressing the most variance, and in order, successive component axes capture the ever-decreasing variance in the observations
- Each component axis can be expressed as a set of numerical *loadings* to the original p feature dimensions
- For example, the first principal component is:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

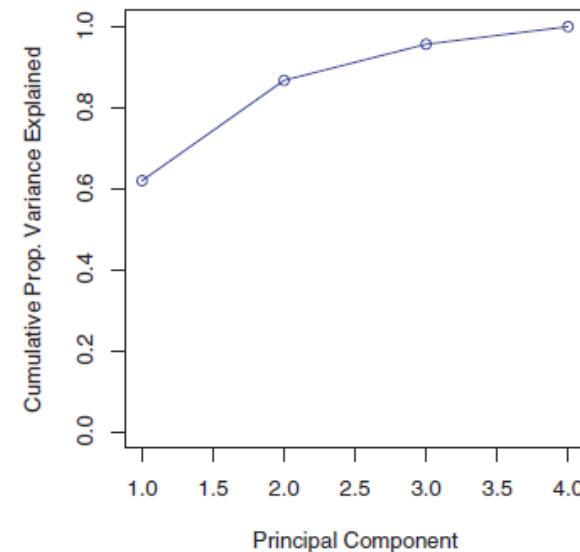
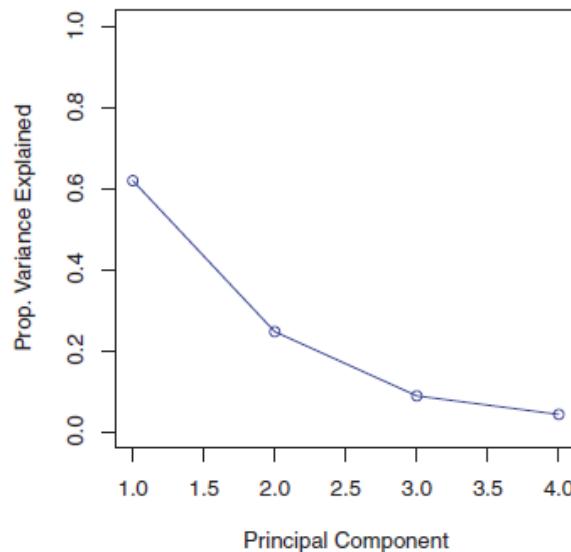
PCA Intuition – Conceptual (3 of 4)

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

- Given the first principal component loading, we can project value for an observation on that axis using the formula above
- We can repeat the process to obtain the observation's projection onto the other component axes
- The datapoint $(Z_1, Z_2, Z_3, \dots, Z_p)$ is the projection of the observation into the principal component space.

PCA Intuition – Conceptual (4 of 4)

- Each successive principal component explains less of the variance in the data
- A scree plot can be used to visualize the variance explained by the k^{th} component (or cumulative explanation of variance by the k components so far)



PCA Implementation – Code

- A linear algebra technique can provide all of the orthogonal component axes which explain the variance in the features of the observations
- Produce the covariance matrix for the dataset
`cov_mat=np.cov(X.T)`
- Singular Value Decomposition on the covariance matrix produces a $p \times p$ matrix (U) which contains the ordered loadings of the dataset:
`u, s, v = np.linalg.svd(cov_mat)`
- Each column in U corresponds to a loading vector in PCA
 - The leftmost column represents the most important component and each successive column to the right represents columns of decreasing importance

PCA Tuning

- Select a desired percentage ν of the variance to explain
- Choose k (the number of components) such that for the approximation of the datapoints:

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{appx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq \nu$$

- A matrix multiplication of the (first k columns of the) U matrix and the dataset yields the projection of the observations into component space:

```
np.dot(X, u[:, 0:componentCount])
```

- Alternately, use `sklearn.decomposition.PCA`

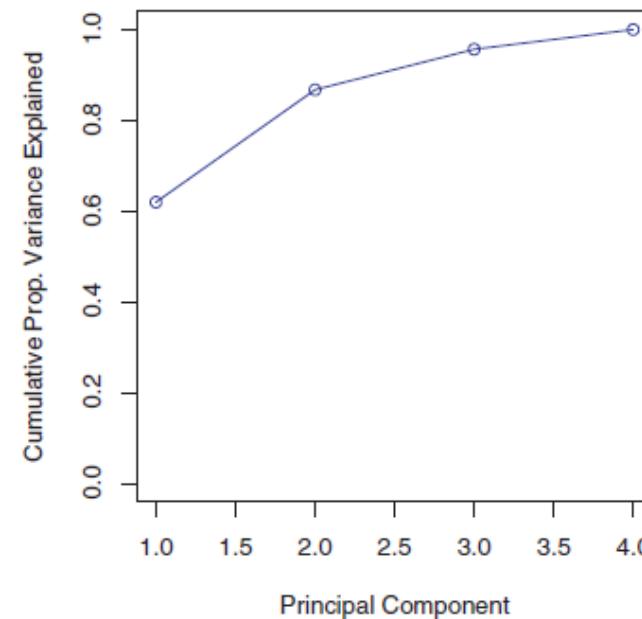
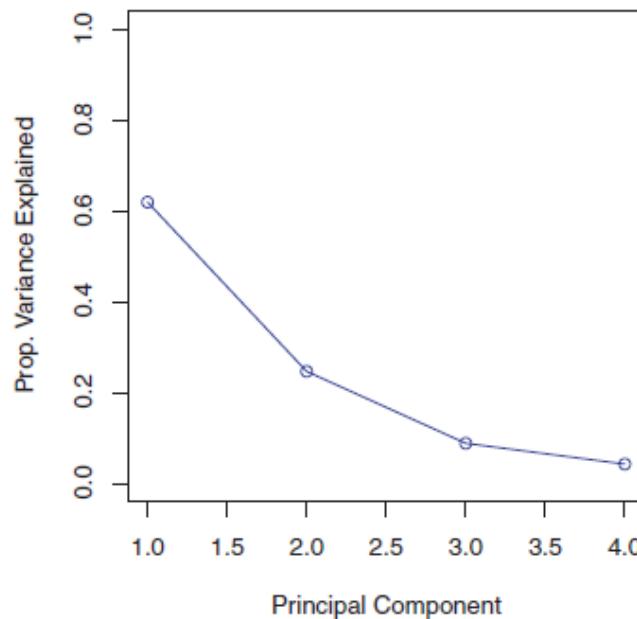
PCA Evaluation – Variance Explanation

- Variance from the m^{th} principal component

$$\frac{1}{n} \sum_{i=1}^n z_{im}^2 = \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{jm} x_{ij} \right)^2$$

Percent of Variance Explained (PVE)

$$\frac{\sum_{i=1}^n \left(\sum_{j=1}^p \phi_{jm} x_{ij} \right)^2}{\sum_{j=1}^p \sum_{i=1}^n x_{ij}^2}$$



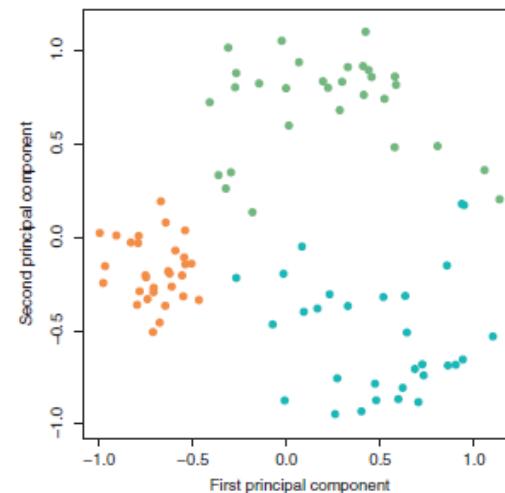
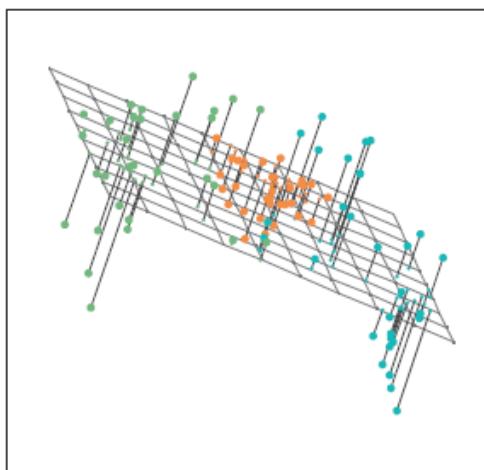
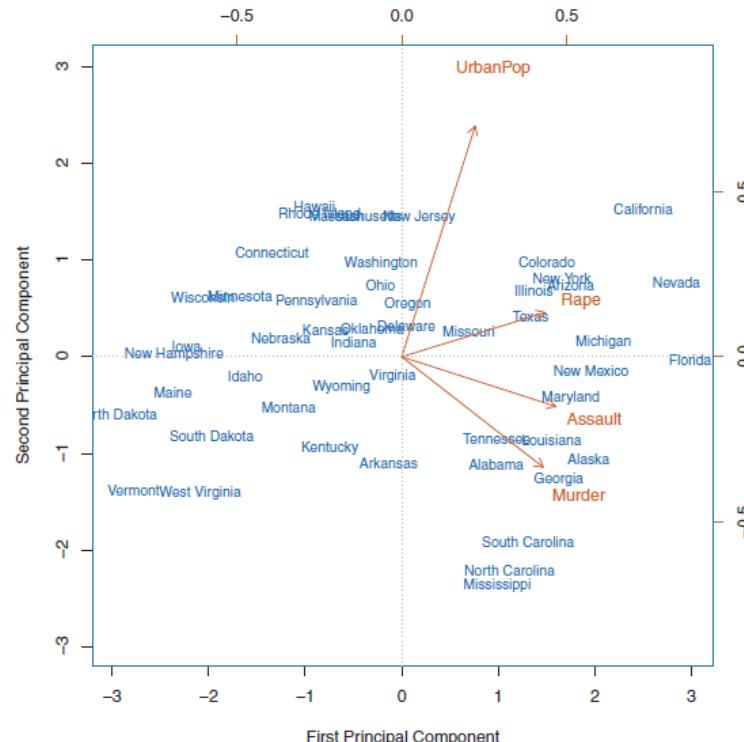
PCA Interpretation

- The projection of the observations into the first k principal components ($1..k$) represent a lossy approximation of the dataset
- In this reduced space, fewer parameters are used to approximate the data

PCA – Uses

- Visualize important data relationships (in 2D)

- Compression
(reduce number of features from p to k)



Mitigate
Collinear Features
before model fitting

More on understanding PCA

- <https://towardsdatascience.com/pca-and-svd-explained-with-numpy-5d13b0d2a4d8>
- https://medium.com/@jonathan_hui/machine-learning-singular-value-decomposition-svd-principal-component-analysis-pca-1d45e885e491
- <https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/>

CLUSTERING

Chapter 10

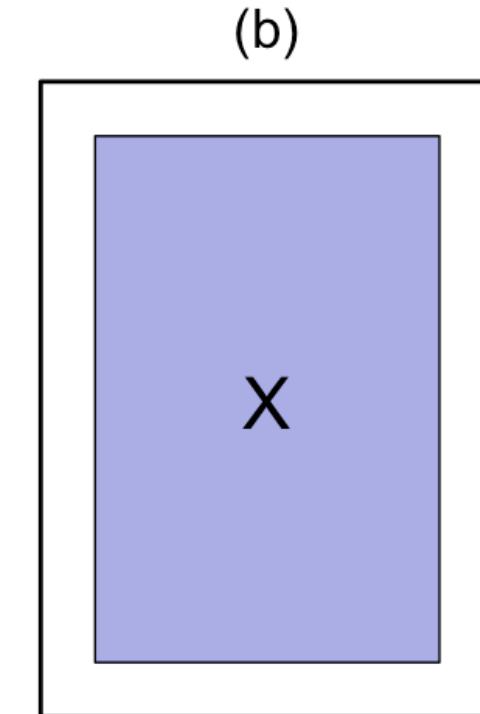
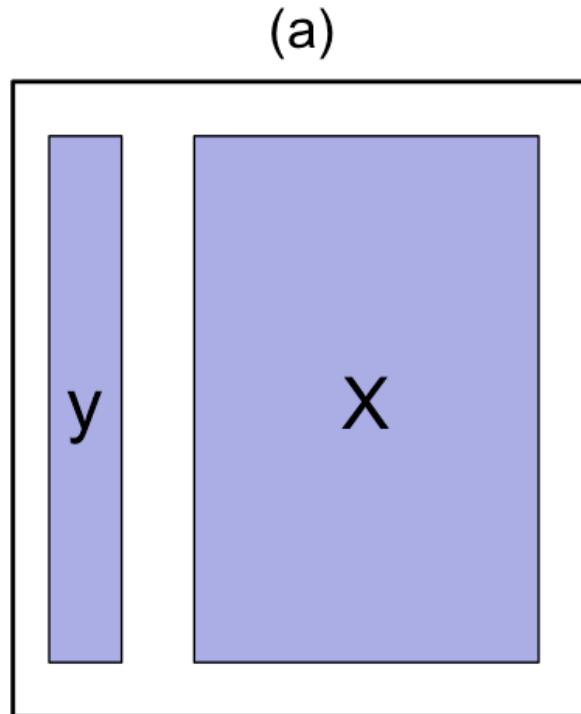
Outline

- What is Clustering?
- K-Means Clustering
- Hierarchical Clustering

WHAT IS CLUSTERING?

Supervised vs. Unsupervised Learning

- Supervised Learning: both X and Y are known
- Unsupervised Learning: only X



Supervised Learning

Unsupervised Learning

Clustering

- Clustering refers to a set of techniques for finding subgroups, or clusters, in a data set
 - *Be careful not to use the word “class” instead of cluster*
- Good clustering: when the observations within a group are similar but observations in different groups are very different
- For example, suppose we collect p measurements on each of n breast cancer patients. There may be different unknown types of cancer which we could discover by clustering the data

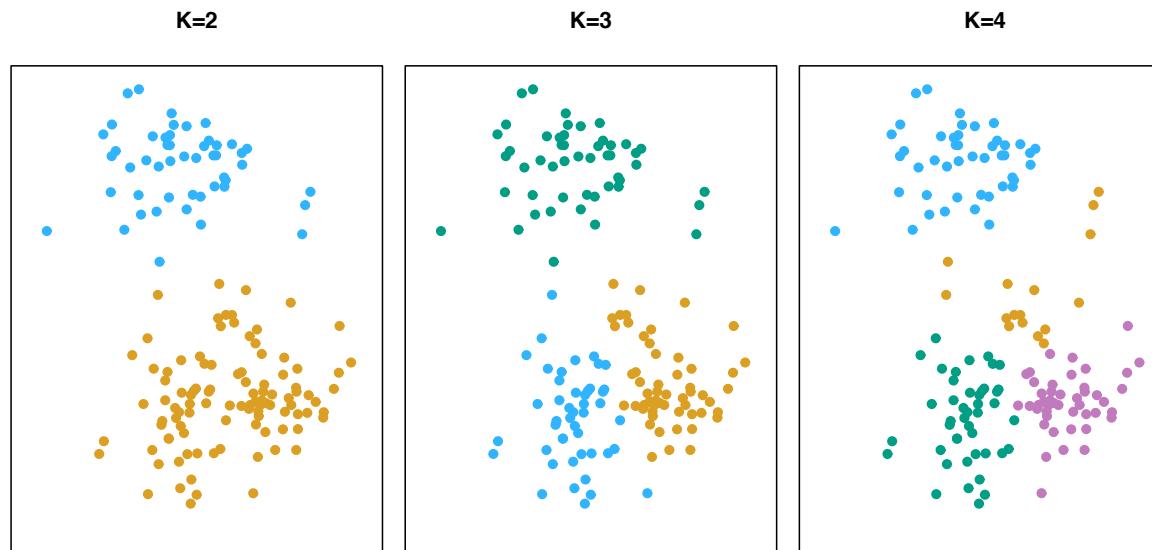
Different Clustering Methods

- There are many different types of clustering methods
- We will concentrate on two of the most commonly used approaches
 - K-Means Clustering
 - Hierarchical Clustering
- The objective is to have a
 - minimal *intra-cluster* - “within-cluster-variation”, i.e. the elements within a cluster should be as similar as possible
 - maximum *inter-cluster* – “center-to-center” distance, i.e. the cluster centers should be as far apart as possible

K-MEANS CLUSTERING

K-Means Clustering

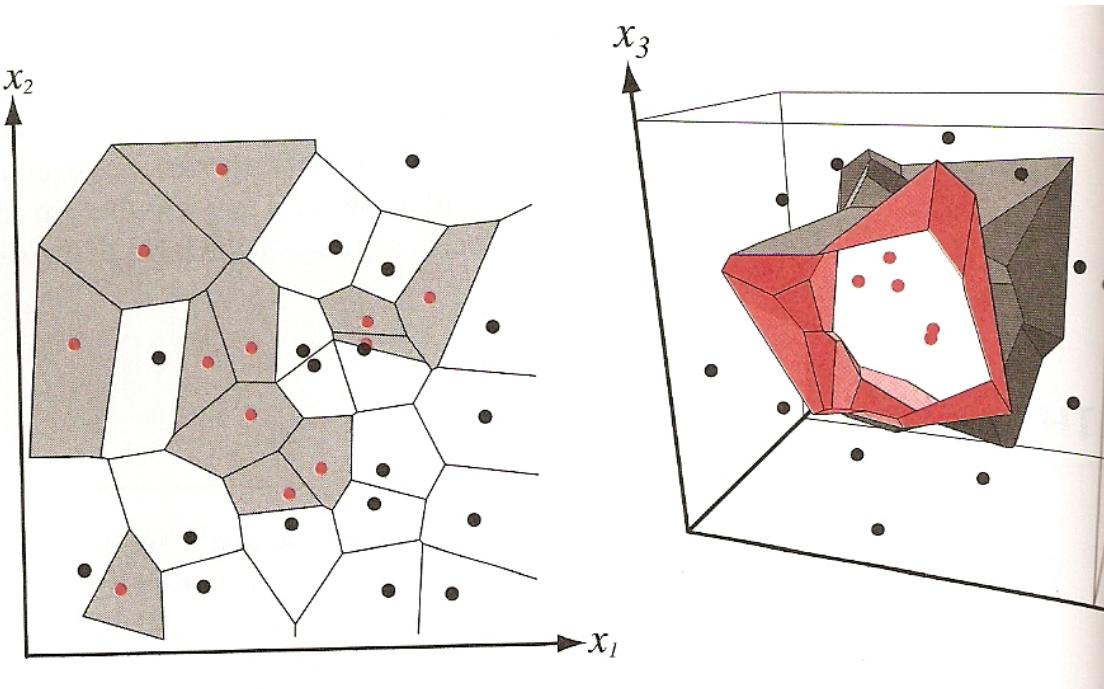
- To perform K-means clustering, one must first specify the desired number of clusters K
- Then the K-means algorithm will assign each observation to exactly one of the K clusters



How does K-Means work?

- We would like to *partition* that data set into K clusters C_1, \dots, C_K
- Each observation belongs to one of the K clusters
- K-Means results in a Voronoi Tessellation of the input space in \mathbb{R}^n

- A tessellation is a tiling/segmenting of the input space
- Each segment/region is a Voronoi Cell and indicates which part of the input space “belongs” to which cluster center



K-Means Clustering Algorithm - Book

- Initial Step: Randomly assign each observation to one of K clusters such that

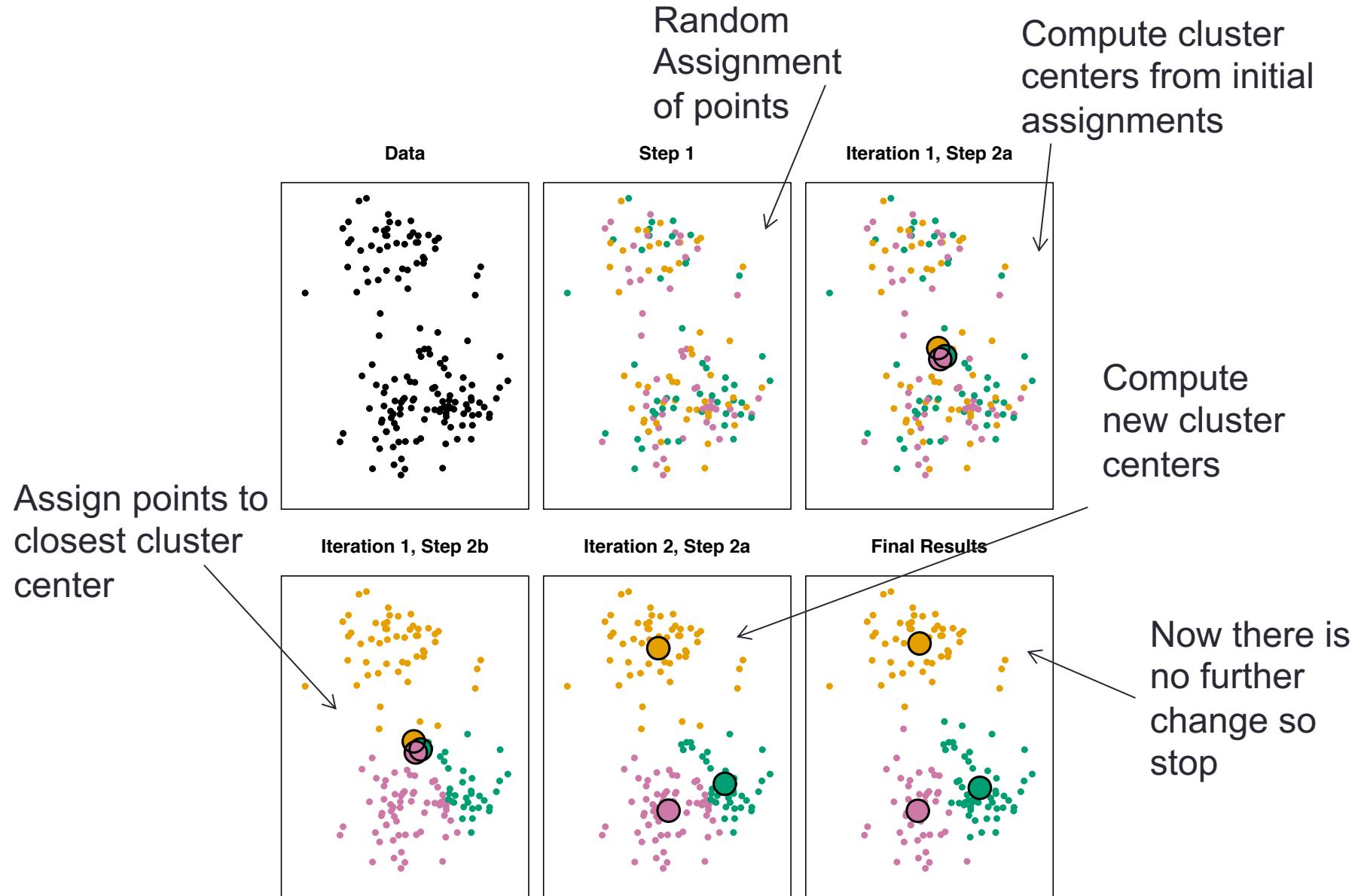
$$\forall i \in Obs, \exists k \in Clusters \mid i \in C_k \quad (\text{every observation belongs to a cluster})$$

$$\forall l, m \in Clusters, l \neq m \rightarrow C_l \cap C_m = \emptyset \quad (\text{clusters are mutually exclusive})$$

- Iterate until the cluster assignments stop changing:
 - For each of the K clusters, compute the cluster centroid. The k^{th} cluster centroid is the mean of the observations assigned to the k^{th} cluster
- $$\text{Centroid}_{k,j} = \frac{\sum X_{i,j}}{|C_k|}$$
- (cluster centroid is mean of each feature for the observations belonging to it)
- Assign each observation to the cluster whose centroid is closest (where “closest” is defined using Euclidean distance).

$$k_i = \operatorname{argmin}_k \|x_i - \text{Centroid}_k\|^2$$

K-Means Algorithm - Visualized



K-Means Clustering Algorithm

Alternative Initialization

- Alternative Initial Step:
Randomly select k starting centroids
- Iterate until the cluster centroids each change very little:
 - Assign each observation to the cluster whose centroid is closest (where “closest” is defined using Euclidean distance).

$$k_i = \operatorname{argmin}_k \|x_i - \text{Centroid}_k\|^2$$

- For each of the K clusters, update the cluster centroid. The k^{th} cluster centroid is the mean of the observations assigned to the k^{th} cluster

$$\forall k, \forall i \in C_k, \forall j \in p, \text{Centroid}_{k,j} = \frac{\sum_{i=1}^n X_{i,j}}{|C_k|}$$

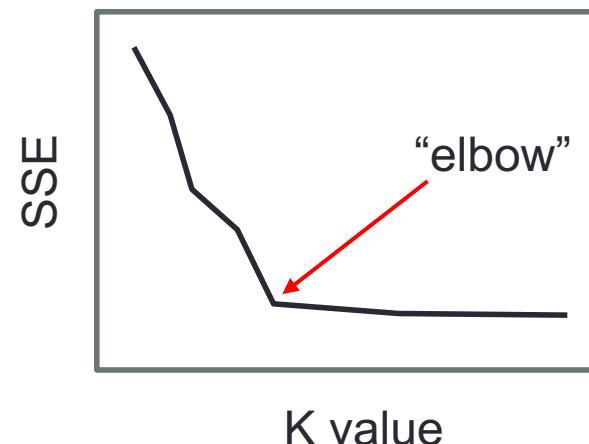
(cluster centroid is mean of each feature for the observations belonging to it)

K-Means Considerations: K

- K-Means Achieves the property:
$$\underset{C_1, \dots, C_k}{\text{minimize}} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{i,j} - x_{i',j})^2 \right\}$$
- In K -means clustering, we must specify K for the number of clusters we desire
 - If we know how many clusters we want, then we can select K
 - What if we don't know? How do we determine a "best" K ?
 - Elbow method – compute SSE for several K values and look for the "elbow":

$$SSE = \sum_{k=1}^K \sum_{x \in C_k} (x - C_k)^2$$

- Other Potential (Automatic) Solutions:
 - χ -Means (Bayesian Information Criteria)
 - G-Means (Anderson-Darling)
 - PG-Means (Kolmogorov-Smirnov test)



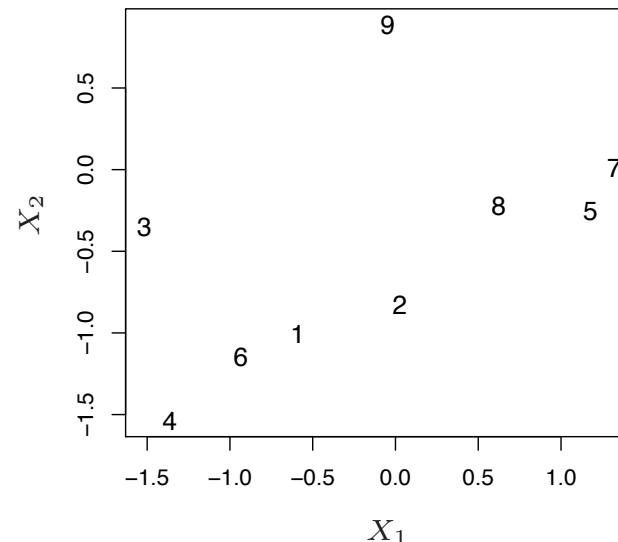
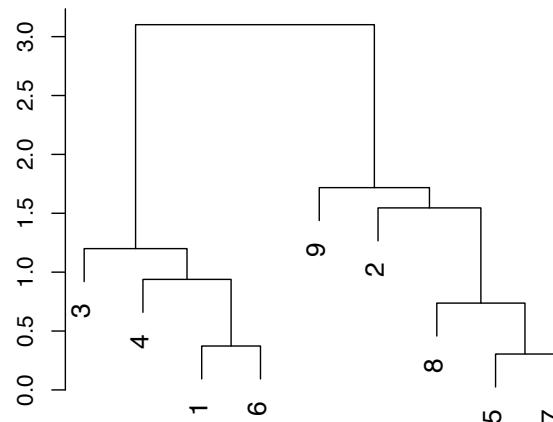
HIERARCHICAL CLUSTERING

Hierarchical Clustering

- K-Means clustering requires choosing the number of clusters.
- If we don't want to do that, an alternative is to use Hierarchical Clustering
- Hierarchical Clustering has an added advantage that it produces a tree based representation of the observations, called a Dendogram

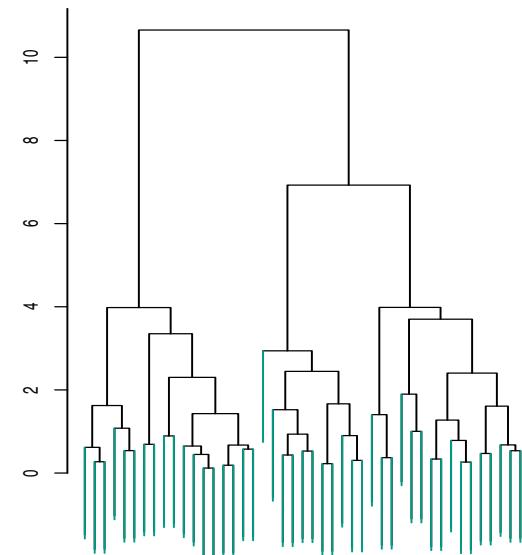
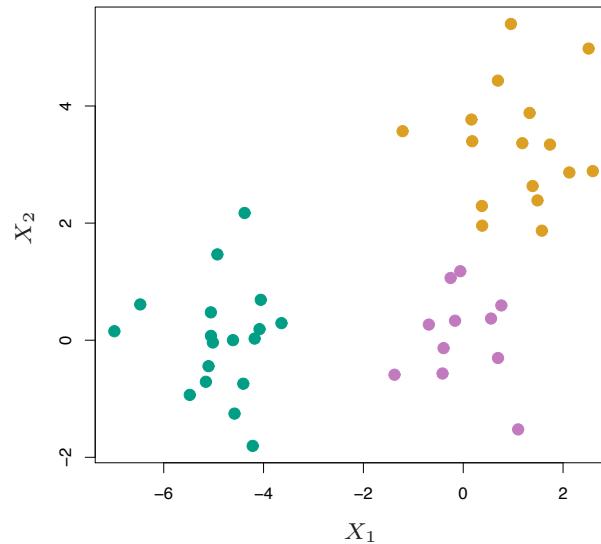
Dendograms

- First join closest points (5 and 7)
- Height of fusing/merging (on vertical axis) indicates how similar the points are
- After the points are fused they are treated as a single observation and the algorithm continues



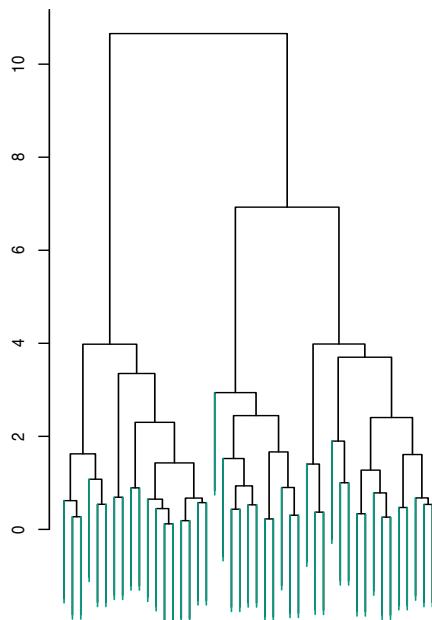
Interpretation

- Each “leaf” of the dendrogram represents one of the 45 observations
- At the bottom of the dendrogram, each observation is a distinct leaf. However, as we move up the tree, some leaves begin to fuse. These correspond to observations that are similar to each other.
- As we move higher up the tree, an increasing number of observations have fused. The earlier (lower in the tree) two observations fuse, the more similar they are to each other.
- Observations that fuse later are quite different

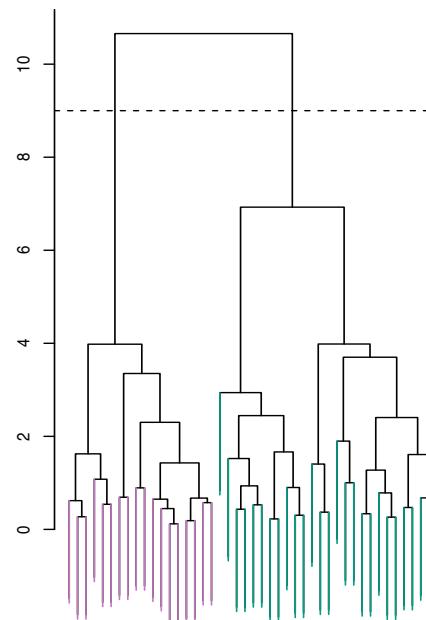


Choosing Clusters

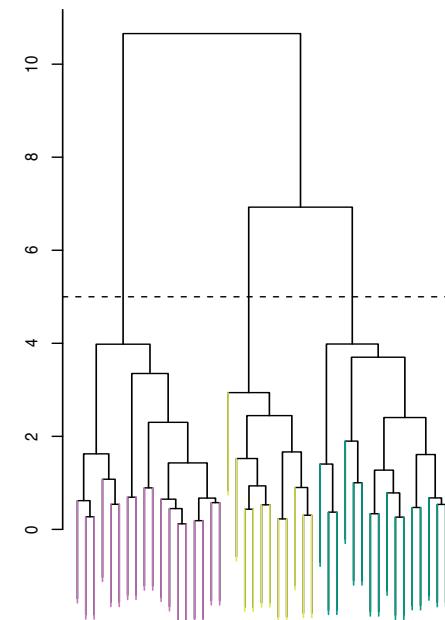
- To choose clusters we draw lines across the dendrogram
- We can form any number of clusters depending on where we draw the break point.



One Cluster



Two Clusters



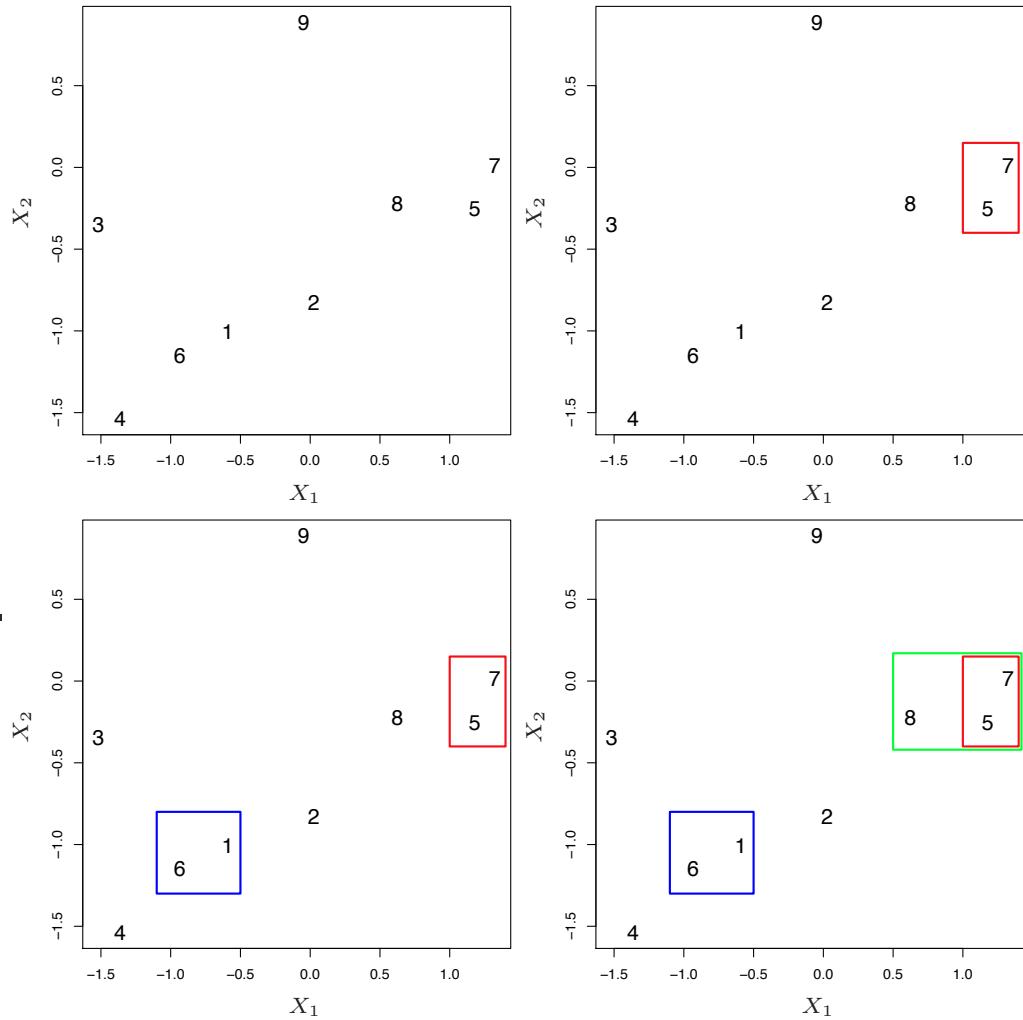
Three Clusters

Algorithm (Agglomerative Approach)

- The dendrogram is produced as follows:
 - Start with each point as a separate cluster (n clusters)
 - Calculate a measure of dissimilarity between all points/clusters
 - Fuse two clusters that are most similar so that there are now $n-1$ clusters
 - Fuse next two most similar clusters so there are now $n-2$ clusters
 - Continue until there is only 1 cluster

An Example

- Start with 9 clusters
- Fuse 5 and 7
- Fuse 6 and 1
- Fuse the (5,7) cluster with 8.
- Continue until all observations are fused.

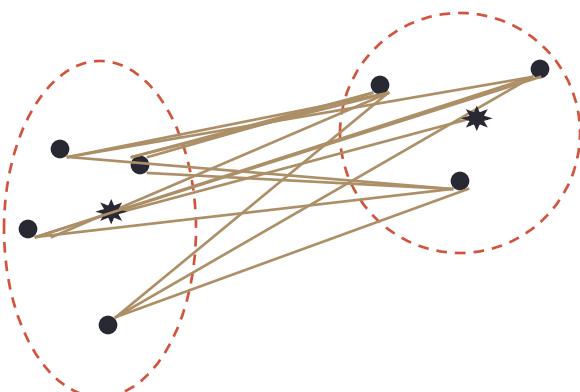


How do we define dissimilarity?

- Implementing hierarchical clustering involves one obvious issue
- How do we define the dissimilarity, or linkage, between the fused (5,7) cluster and 8?
- There are four options:
 - Complete Linkage
 - Single Linkage
 - Average Linkage
 - Centroid Linkage

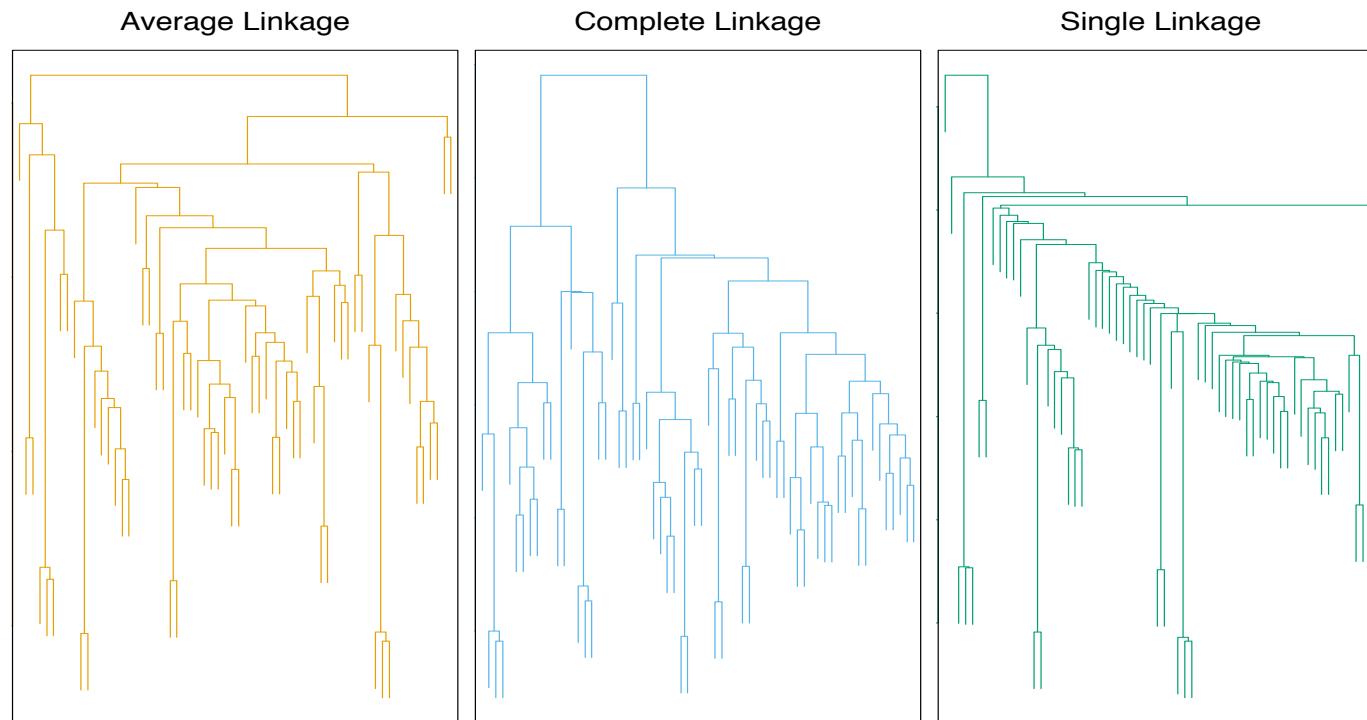
Linkage Methods: Distance Between Clusters

- **Complete Linkage:** Largest distance between observations
- **Single Linkage:** Smallest distance between observations
- **Average Linkage:** Average distance between observations
- **Centroid:** distance between centroids of the observations



Linkage Can be Important

- Here we have three clustering results for the same data
- The only difference is the linkage method but the results are very different
- Complete and average linkage tend to yield evenly sized clusters whereas single linkage tends to yield extended clusters to which single leaves are fused one by one.



Exercise

- Suppose that we have 5 observations, for which we compute a similarity (distance) matrix as follows:

	A	B	C	D	E
A	0				
B	9	0			
C	3	7	0		
D	6	5	9	0	
E	11	10	2	8	0

- On the basis of the similarity matrix, sketch the dendrogram that results from hierarchically clustering these 5 observations using complete linkage.

FINAL THOUGHTS

Practical Issues in Clustering

- In order to perform clustering, some decisions must be made:
 - Should the features first be normalized? i.e. Have the variables centered to have a mean of zero and standard deviation of one.
 - In case of hierarchical clustering:
 - What dissimilarity measure should be used?
 - What type of linkage should be used?
 - Where should we cut the dendrogram in order to obtain clusters?
 - In case of K-means clustering:
 - How many clusters should we look for the data?
- In practice, we try several different choices, and look for the one with the most useful or interpretable solution.

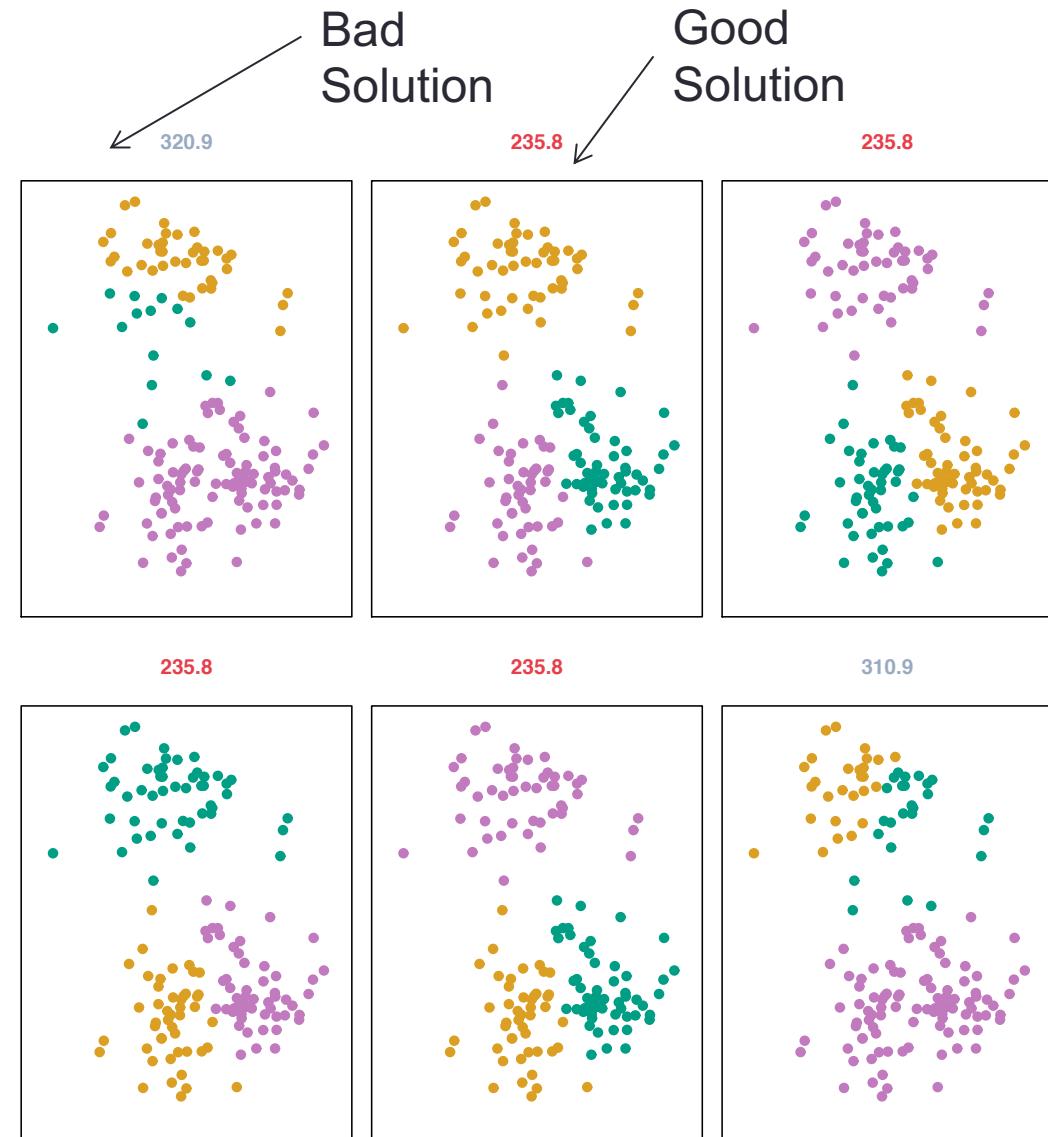
Using the results of clustering

- Most importantly, one must be careful about how the results of a clustering analysis are reported
- These results should not be taken as the absolute truth about a data set
- Rather, they should constitute a starting point for the developments of a scientific hypothesis and further study, preferably on independent data

Backup Slides

K-Means Considerations: Local Optimums

- The K-means algorithm can get stuck in “local optimums” and not find the best solution
- Hence, it is important to run the algorithm multiple times with random starting points to find a good solution



K-Means Considerations: Data

- Identification of potential cluster shape – non-convex shapes will perform poorly
 - Alternatives: CRYSTAL, DBSCAN,...
- Choice of similarity function
 - Continuous
 - Euclidean: $d(x_i, x_j) = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}$
 - Normalize disparate axis dimensions x [0..100], y [0..1]
 - Mahalanobis: $d_M(\vec{x}_i, \vec{x}_j) = (\vec{x}_i - \vec{x}_j) \Sigma^{-1} (\vec{x}_i - \vec{x}_j)^T$
 - Discrete Binary - Jaccard index
 - Discrete - Dice/Czehanovsky-Sorensen measure
 - Mixed:
 - Gower similarity
 - Podani (Gower extended with ordinals)
 - Discrete and Mixed: Where is the cluster center?
 - k-Mediods

K-Medioids

- 1) **Initialize:** Randomly assign $\{C_1 \dots C_K\}$ to K samples from $\{x_1 \dots x_N\}$
- 2) **Assignment:** Assign each x_n to one of the $k=\{1 \dots K\}$ cluster centers $\{C_1 \dots C_K\}$ (distance based on similarity measure)
- 3) **Update:** For the given cluster assignment, update each C_k to the x_n in each cluster k that minimizes the error
- 4) **Repeat** until converged



Subsampling/ Vector Quantization

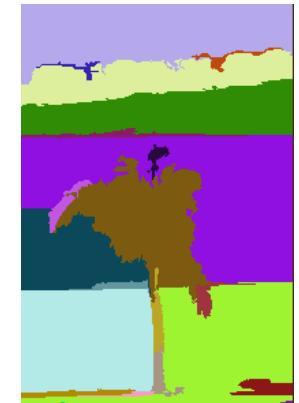


Image Summarization

Overview of Computational Complexities

- Use Big-Oh notation
 - Upper bounds on computational complexity
 - Performance bounds is based on:
 - M -Number of samples
 - L -Number of iterations
 - K -The number of clusters
 - n -Dimensionality of the data
- Performance Bounds:
 - K-Means – $O(nMKL)$
 - Soft K-Means – $O(nMKL)$
 - K-Mediods – $O(nLKM_k^2)$

How good is our clustering?

- Evaluation without class labels (recall inter and intra-cluster optimizations)
 - Homogeneity
 - Separation
 - Silhouette Width
 - Davies Bouldin index
 - Dunn index - ratio between the minimal inter-cluster (center to center) distance to maximal intra-cluster (farthest point in cluster to farthest point in cluster) distance
- Evaluation with class labels: purity, F-measure, Rand index, Adjusted Rand index, Jaccard index, Fowlkes-Mallows index