CSE 4/587
Data Intensive Computing

Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Day 05
Algorithms and Models Part 2

Announcements and Feedback

- Attendance: More information coming, will be announced on Piazza
- Read Ch 11, 12, 14, 15 in Data Science from Scratch as needed

Recap from Last Class

- Data cleaning and munging
  - Prepare data for analysis
  - Fix formatting issues (ie: trim whitespace, convert string to number, etc)
  - Check for outliers and/or missing data
- Models and Algorithms
  - Statistical models attempt to model the real-world process which generated the data
  - ML algorithms attempt to make accurate predictions based on the observed data

Recap from Last Class

- Linear regression
  - Attempt to model the relationship between two (or more) variables
  - $y = B_0 + B_1x + e$
  - B are the coefficients we are solving for, e is our error term, or *noise*
  - The B terms determine the *trend* (or the direction of the line)
  - The noise term determines the amount of *variation* (or how close our observed data is to the line)
  - We can also add more predictor variables, or try non-linear relationships

Recap from Last Class

- Linear regression
  - Attempt to model the relationship between two (or more) variables
  - $y = B_0 + B_1x + e$
  - B are the coefficients we are solving for, e is our error term, or *noise*
  - The B terms determine the *trend* (or the direction of the line)
  - The noise term determines the amount of *variation* (or how close our observed data is to the line)
  - We can also add more predictor variables, or try non-linear relationships

Complexity can be estimated as $O(np^2)$ but $p << n$

Two More Algorithms: k-means and K-NN

- Last time we learned Linear Regression, a model that can be useful for predicting outcomes based on a set of independent variables
- Today we are going to look at algorithms for clustering and classification
  - k-NN: classify/label data points by training on already labeled data
  - k-means: determine clusters of points in a dataset

Machine Learning Algorithm Classification

- Machine Learning algorithms can be divided into 3 categories:
  - Supervised: We know the "right answer", fit a model to that knowledge
  - Unsupervised: We don't know the answer, want the algorithm to find it
  - Semi-Supervised: We know some, but there is more to learn
- Linear regression is an example of a supervised algorithm. We have a training set of predictors and the observed outcomes, and we fit a model based on that knowledge.

Machine Learning Algorithm Classification

- Machine Learning algorithms can be divided into 3 categories:
  - Supervised: We know the "right answer", fit a model to that knowledge
  - Unsupervised: We don't know the answer, want the algorithm to find it
  - Semi-Supervised: We know some, but there is more to learn
- Linear regression is an example of a supervised algorithm. We have a training set of predictors and the observed outcomes, and we fit a model based on that knowledge.

Now we'll learn about k-NN, another supervised algorithm

k-Nearest Neighbors (k-NN)

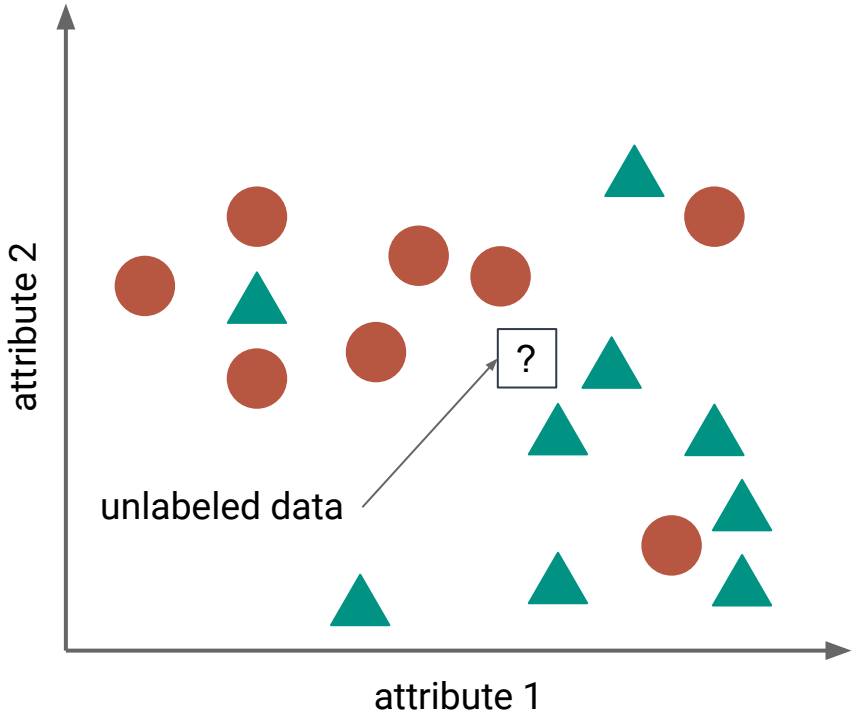
- Algorithm used to classify or label objects/data points
  - You start with some already labeled data points
  - Goal is to be able to automatically label a new set of unlabeled points
- Examples could be: "Good" or "Bad" credit score, political affiliation, star rating of a restaurant, at risk for illness, etc.
- Would linear regression work for this?
  - ...maybe, but it depends on what you are doing
  - Not all data can be easily mapped to continuous scale

k-Nearest Neighbors (k-NN)

- Intuition: For a given unlabeled element, look at just the k *most similar* elements in the labeled dataset based on various attributes, and choose the label that most of those elements have
  - ie: Look at movies with similar runtime, budget, genre, actors, awards to label a movie as good or bad
  - ie: Look at people with similar height, weight, age, gender, to determine if a person is at risk or not for a certain disease

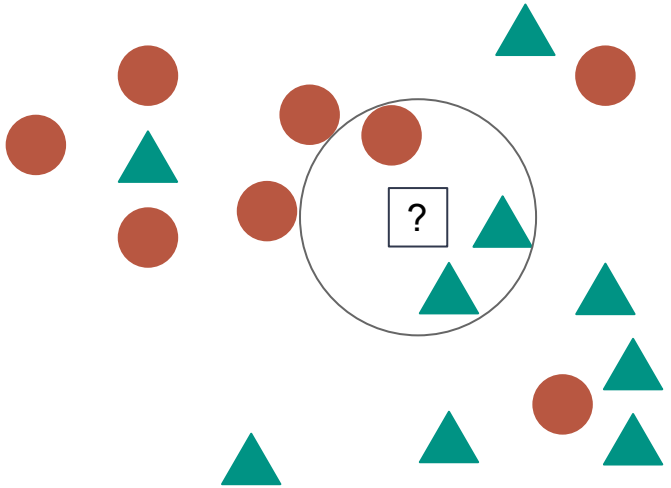
A Simple Example

- For the example to the left, we have a number of data points labeled as either red circles, or green triangles
- How do we label the new unknown data point?
- Depends on the value of k



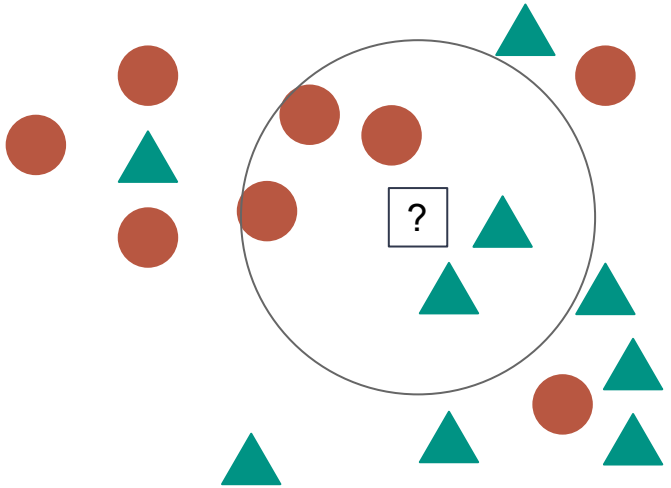
A Simple Example

- If k = 3:
  - Green triangles have 2 votes
  - Red circles have 1 vote
  - The new point will be labeled green triangle



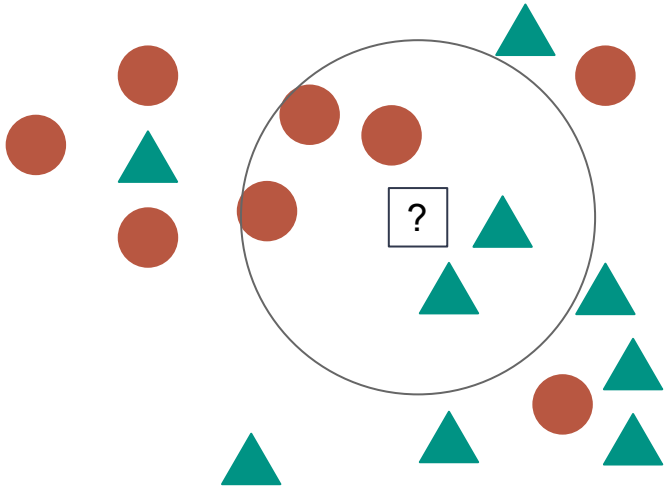
A Simple Example

- If k = 5:
  - Green triangles have 2 votes
  - Red circles have 3 votes
  - The new point will be labeled red circle



A Simple Example

- In order to apply our intuition to real data we need to:
  - Determine how we measure *closeness*
  - Determine a good value for k



The Basic Process

1. Decide on your *similarity* metric
2. Split the labeled set into training and test data
3. Pick an evaluation metric (similar to R^2 and p-values for linear reg)
4. Run with a few different values of k, check against evaluation metric
5. Select k with the best evaluation metric
6. Run on unlabeled data

Distance Metrics

- This varies a lot based on context
  - Numerical values (ie salary, height, age, etc) are "easy" (sort of)
  - What about more abstract attributes
    - Social networks
    - Text based data
    - Movie genre

Numerical Distance and Scale

- If our data is numerical in nature, there are a number of known ways to define "distance" between two things
  - Euclidian, Cosine, Manhattan, Mahalanobis, etc
- What about scale?
  - Consider clustering people based on salary and SAT scores:
    - The distance between ($30,000, 1400) and ($100,000, 1450) is dominated by the salary difference
    - Rescaling data, ie (30, 1400) and (100, 1450) balances the effect of each parameter…but is that necessarily the goal?

Numerical Distance and Scale

- If our data is numerical in nature, there are a number of known ways to define "distance" between two things
  - Euclidian, Cosine, Manhattan, Mahalanobis, etc
- What about scale?
  - Consider clustering people based on salary and SAT scores:
    - The distance between ($30,000, 1400) and ($100,000, 1450) is dominated by the salary difference
    - Rescaling data, ie (30, 1400) and (100, 1450) balances the effect of each parameter…but is that necessarily the goal?

How you scale your data can have a significant impact on outcome, and therefore is also part of your model!

Non-Numerical Data

- Certain distance metrics can deal with non-numerical data
  - ie Jaccard Distance, Hamming Distance
- Many times, however, you will have to define your own
  - Consider movie genre, how "far" apart are two genres?
    - Could define the same genre as 0 apart and different genres as x apart. x can be chosen based on the scale of other numeric attributes
      - This choice is now also a parameter to your model
    - Some other way to define closeness? For example, are sci-fi and fantasy closer than sci-fi and documentary?

Evaluation Metrics

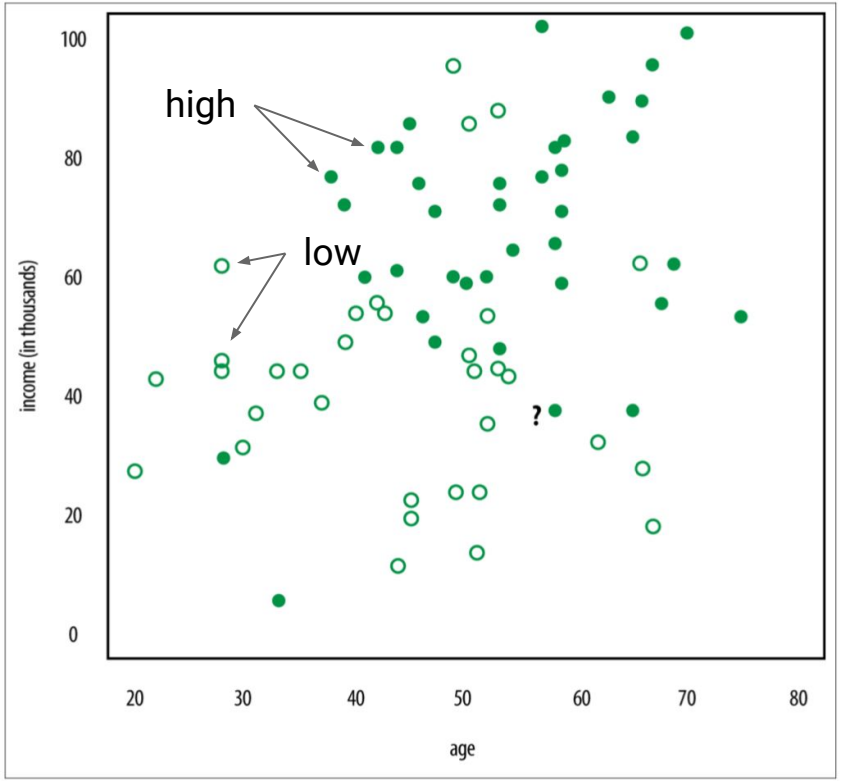
- How do you measure the effectiveness of your model?
  - Accuracy? (number of items correctly categorized)
- Accuracy seems like an obvious choice, but that's not always true…
  - See DSfS Ch 11
- Are all misclassifications created equal? Does a false-positive carry more weight than a false-negative?
  - Precision: how accurate our positive predictions are
  - Recall: what fraction of positive results did our model identify

Finding k

- Now that you have your model setup and know how you will evaluate, you can run the algorithm for different values of k
  - For each item in your *test* set, assume you don't know its label
  - Find its k-nearest neighbors in the training set to determine its label by majority vote
  - After labeling everything in the test set, evaluate effectiveness with your chosen evaluation metric
- Select k which yielded the best results based on your chosen metric

Doing Data Science Ch. 3 Example

- Dataset tracking age, income (in thousands), and "high" or "low" credit
- k=5 had the lowest misclassification rate
- ? is a 57 year old with $37k income
- Model predicts "low" credit



k-means Clustering

- Unsupervised algorithm to find "clusters" in data
  - We don't know or assume anything about the data
- Goal is to "segment" or "cluster" data
  - For example, your data is users, you want to divide them into groups of "similar" users. Why?
    - Serve different ads/provide different experiences
    - Further modeling may differ based on groups

The Algorithm

1. Choose the number of clusters
2. Initialize centroids to some value
    a. Could be via some special algorithm, or could be random
3. Then repeat the following steps…
    a. Reassign all points to the closest centroid
    b. Recalculate the centroids position based on this assignment
4. …until there is no change in centroid values or points stop switching

Interactive Example: Visualizing K-Means Clustering

The Theory Behind the Algorithm

- k-means searches for the minimum *sum of squares* assignment
- In order to converge consider the following two conditions:
  - Re-assigning points reduces the sum of squares
  - Re-computing centroids reduces the sum of squares
- Since both steps reduce the sum of squares, does it converge?
- There are only a finite number of ways to assign the finite number of points to each centroid, so the algorithm must converge
  - It will find a *local* minimum…

Some Issues with k-means

- How do we choose the number of clusters?
  - For 2D data, we may be able to intuit via inspection…but higher dimensional data gets tricky fast
- It will not necessarily find the global minimum
  - Doing so is an NP-Hard problem
- How do you interpret results? What do the clusters represent?
  - Sometimes it makes sense, sometimes it does not

A Small Parting Example

- Consider just the ages of a group of users:

{ 23, 25, 24, 23, 21, 31, 32, 30, 31, 30, 37, 35, 38, 37, 39, 42, 43, 45, 43, 45 }

A Small Parting Example

- Consider just the ages of a group of users:

{ 23, 25, 24, 23, 21, 31, 32, 30, 31, 30, 37, 35, 38, 37, 39, 42, 43, 45, 43, 45 }

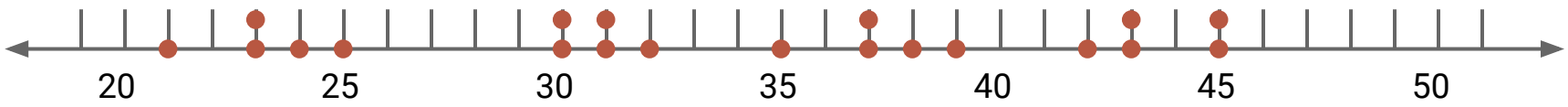
- Let's sort it quick (just for our benefit)

{ 21, 23, 23, 24, 25, 30, 30, 31, 31, 32, 35, 37, 37, 38, 39, 42, 43, 43, 45, 45 }

- Assuming 3 groupings, how might we as humans segment the data?

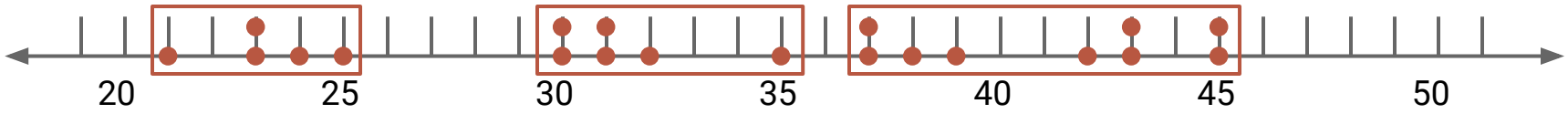
A Small Parting Example

- As humans, it might seem natural to segment by 10s: 20 year olds, 30 year olds, and 40 year olds
- Clustering with k-means gives slightly different results



A Small Parting Example

- As humans, it might seem natural to segment by 10s: 20 year olds, 30 year olds, and 40 year olds
- Clustering with k-means gives slightly different results
  - Is this useful?
  - What can we do with this information?
  - What if we use a different number for k?



Summary

- Learned 3 different algorithms (linear regression, k-NN, k-means)
  - Different use cases: prediction vs labeling vs clustering
  - All come with their own set of assumptions and parameters
    - Learning to deal with these is part of the "art" of being a good data scientist
    - Be aware of how your choices play a role in the results of your analysis

Factors to Consider

- What are the basic variables in your problem?
  - What do they represent?
  - What is their scale?
  - How are they related? (based on intuition and observation)
  - What are your predictors?
- What are the underlying processes?
  - Are you attempting to capture them in a model?
- **What do you want to know? (may require a domain expert)**