Machine Learning

Dr. Baldassano

chrisb@princeton.edu

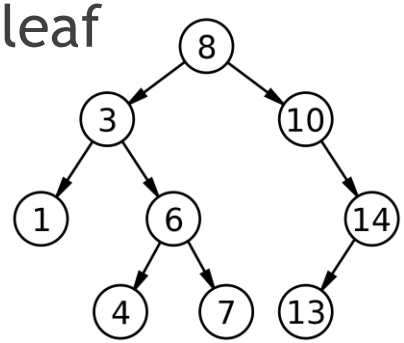
Yu's Elite Education

Last week recap: BST

- Maintains a sorted set of numbers in a tree
- BST property: every node is greater than all nodes in left subtree, less than all in right
- O(log N) time to find, insert, or remove
- In-order traversal gives sorted list

Last week's assignment: Maximum depth of BST

- Calculate the maximum depth of a BST - the longest path from the root to a leaf
- For example, max depth = 3



- This is often an important thing to keep track of – if max depth is too high, BST is turning into a list

Learning from data

- All the algorithms we've discussed so far are entirely designed by humans
- E.g. hashing – we pick a hash based on what we think will work well on some data
- Machine learning: create a model of the world that is partly *learned* from examples

Types of machine learning

- **Unsupervised** learning
  - Given a whole bunch of datapoints, learn something about their structure
  - Example: given a Facebook friends network, predict who might want to become friends
- **Supervised** learning
  - Given a bunch of *labeled* datapoints, learn to predict labels from data
  - Example: given lots of images labeled as “cat” or “dog”, learn to predict whether a new image contains a cat or a dog

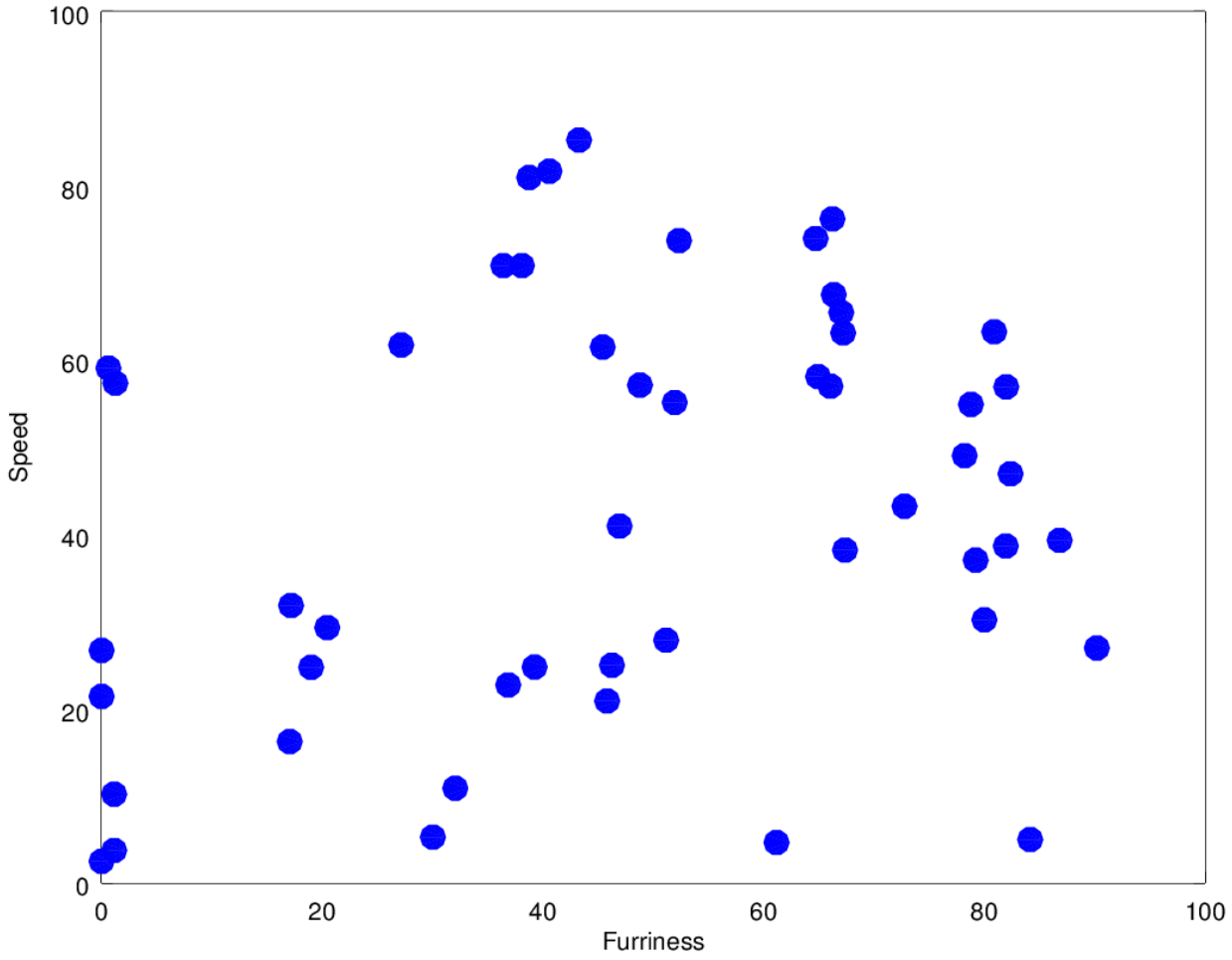
Unsupervised learning: setup

- We have examples of a bunch of items
  - Movies, plants, people, products, pictures…
- The information that we have about each item are called “features”
- Can think of this data as a matrix of items and features

Features example: Animals

	Size	Furriness	Domesticated	Speed
Antelope	39	89	8	71
Grizzly Bear	87	82	11	47
Killer Whale	91	1	15	57
Beaver	7	46	13	25
Dalmation	39	27	72	62
Persian Cat	6	90	73	27
Horse	71	41	59	82
German Shepherd	55	66	69	57
Blue Whale	86	0	5	21
Siamese Cat	2	73	84	43

Feature space



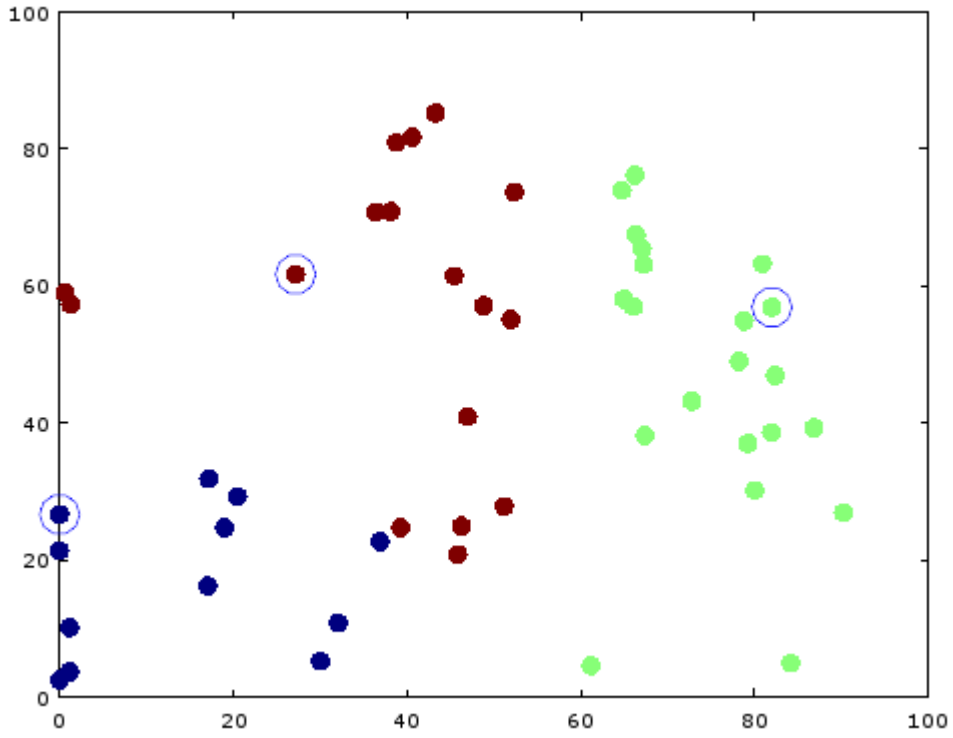
Clustering

- Most unsupervised learning is called “clustering” – we are given a bunch of items, and want to find some structure
  - Given movies, group movies into genres
  - Given patients, group into disease subtypes
  - Given customers, group into segments
- In this example: can we find clusters of animals?

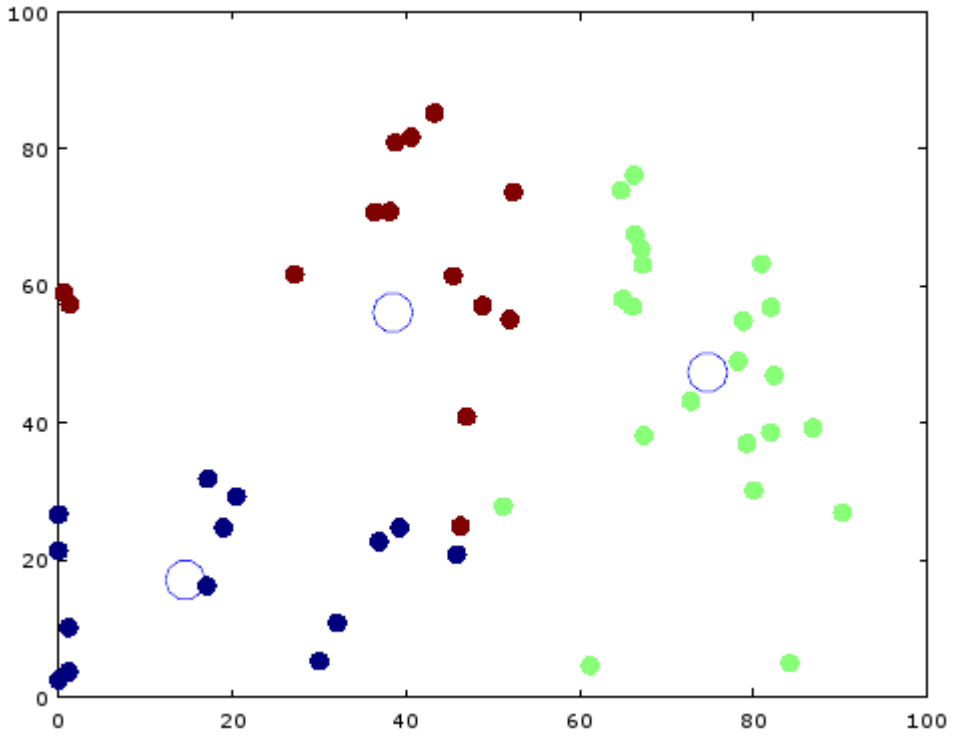
K-means clustering

- A common clustering algorithm
- Randomly pick k animals as cluster centers
- Repeat until convergence:
  - Assign animals to the closest “center”
  - Re-compute the centers to be at the center of their cluster

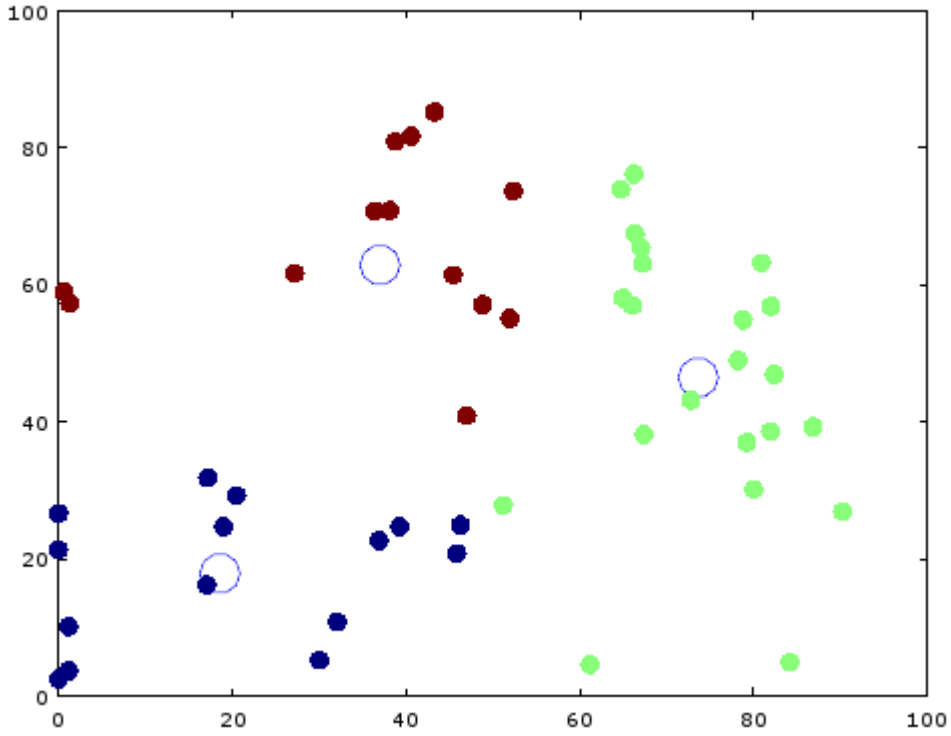
K-means: Step 1



K-means: Step 2



K-means: Step 3



Animal results

- Our clustering suggests there are three kinds of animals:
  - Unfurry, slow: Mole, hippo, elephant
  - Very furry, medium speed: Persian cat, skunk, tiger, hamster
  - Medium furry, very fast: Antelope, horse, weasel, mouse

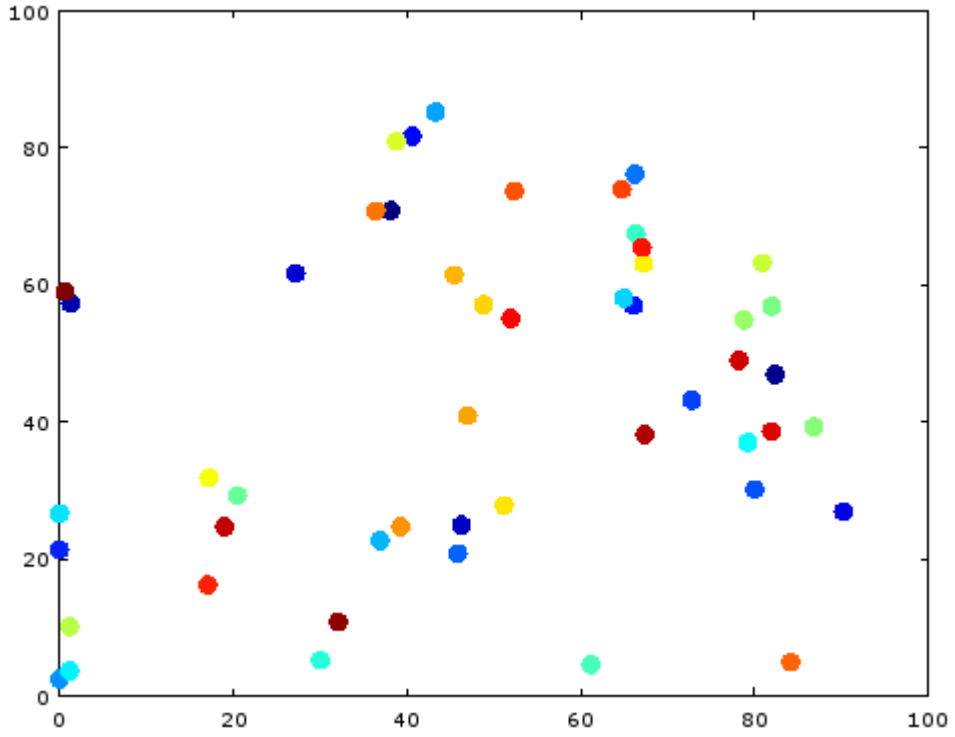
Analysis of k-means

- What is the big-O complexity of k-means?
- Each iteration:
  - Closest-center: O(N*K)
  - Update-center: O(N/K * K) = O(N)
- Number of iterations less obvious, but generally a small constant value
- Time dominated by closest-center operation
  - Can be sped up with approximations, like locality-sensitive hashing

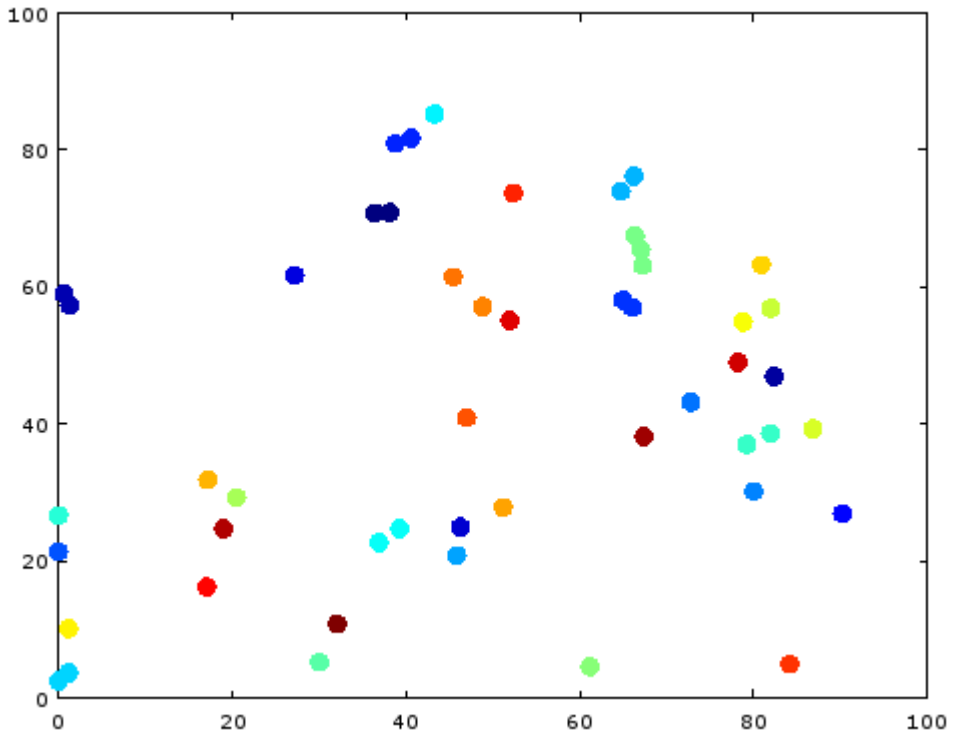
Hierarchical clustering

- An alternative clustering method
- Start with everything as its own cluster
- Iteratively merge together the two clusters that are “closest”
- Stop when there are only K clusters left

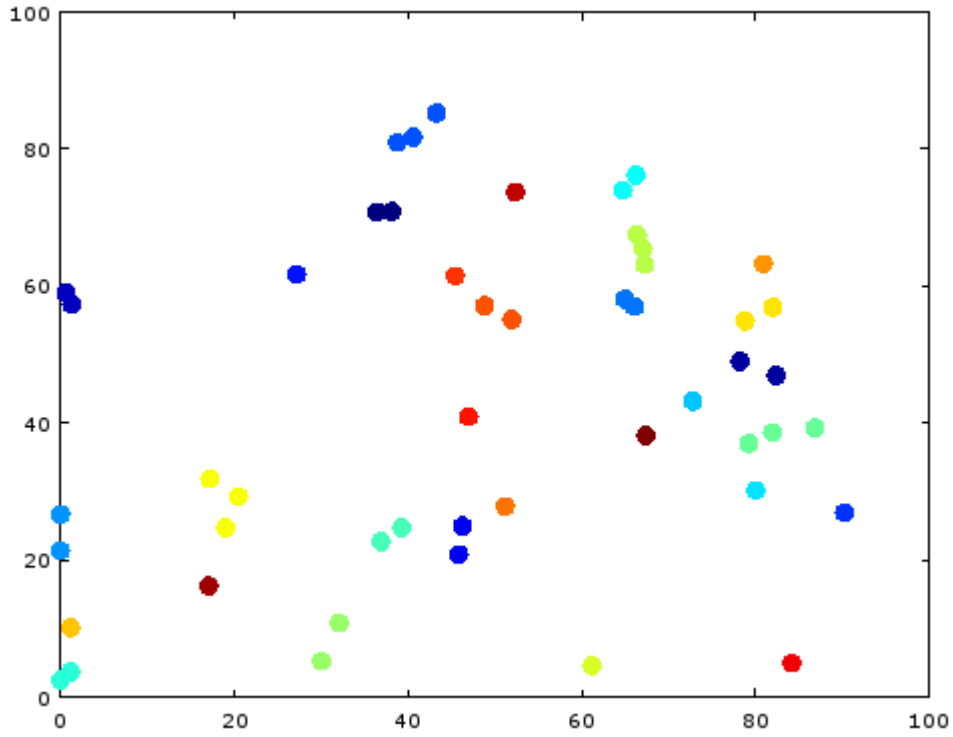
Hierarchical clustering: Step 1



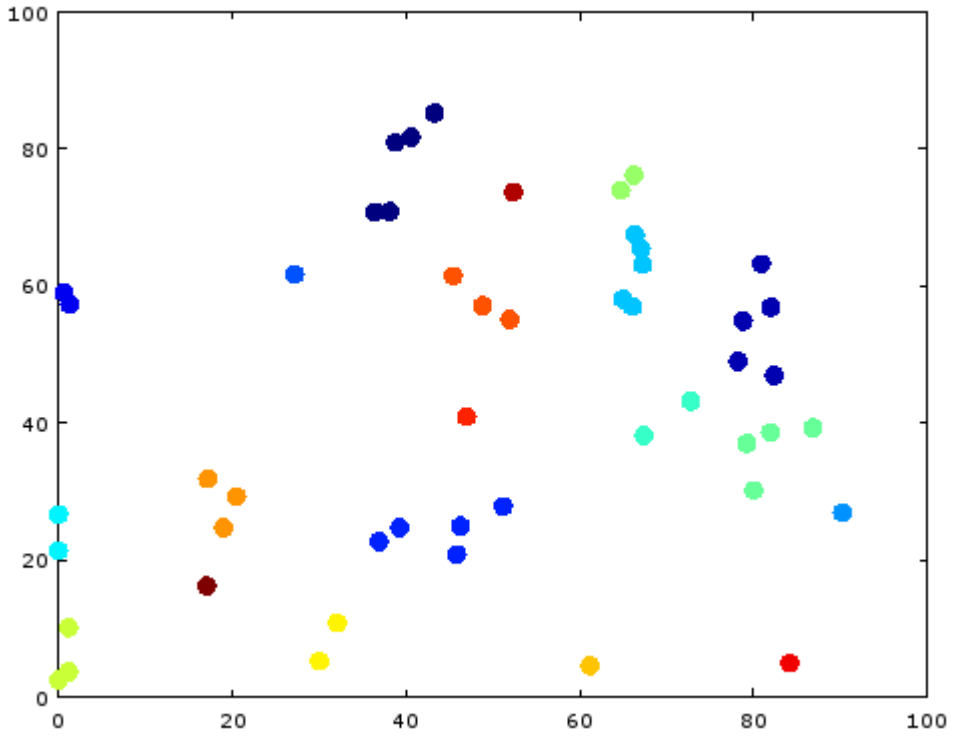
Hierarchical clustering: Step 11



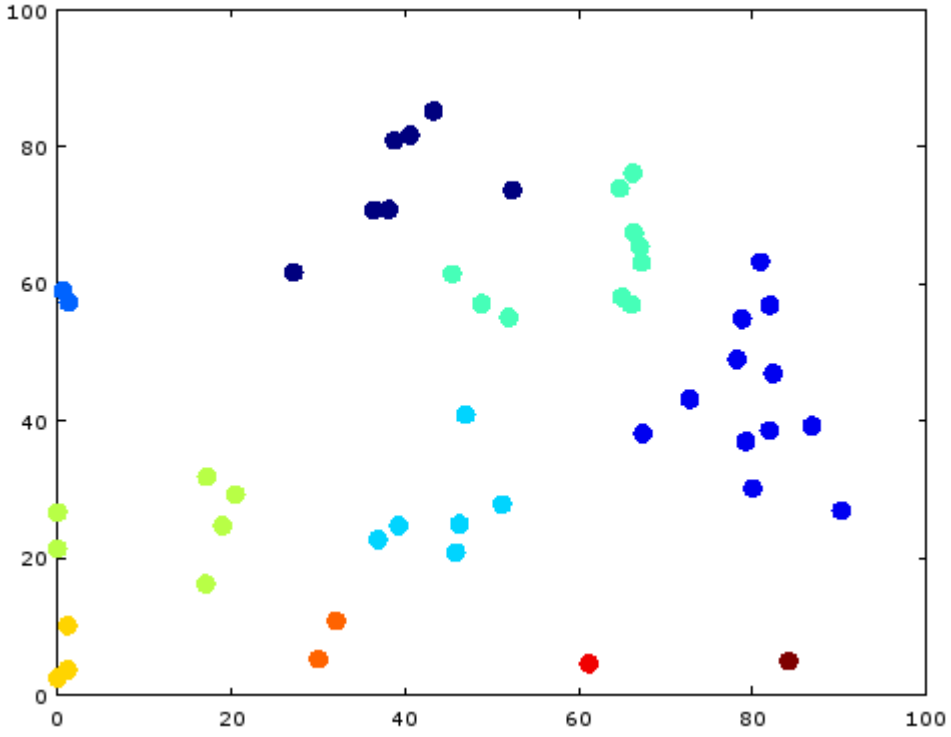
Hierarchical clustering: Step 21



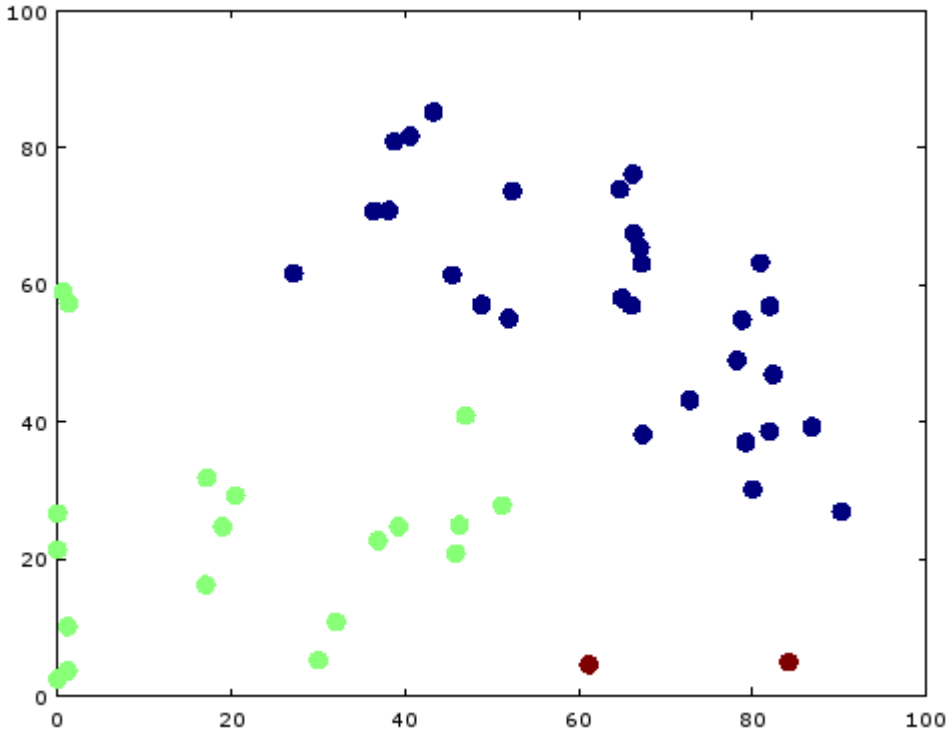
Hierarchical clustering: Step 31



Hierarchical clustering: Step 41



Hierarchical clustering: Step 48



Analysis of hierarchical clustering

- # of steps?
  - N-K
- Each step, compare every pair of clusters
  - O(N) clusters, $O(N^2)$ comparisons
- Overall $O(N^2(N-K)) = O(N^3)$ for small K

Summary: Unsupervised clustering

- Goal is to find structure in unlabeled datapoints
- K-means: pick cluster centers randomly, iterate until clusters are stable
- Hierarchical clustering: merge clusters together based on some closeness criterion

Supervised learning

- More popular kind of machine learning
- Every data item has both features and a label
- Goal is to predict labels given features
  - Category labels: classification
    - Email spam prediction
  - Continuous labels: regression
    - Predict stock market tomorrow given information about today

Titanic data set

Passenger	Class	Gender	Survived
1	3	M	N
2	1	F	Y
3	3	F	Y
4	1	F	Y
5	3	M	N
6	3	M	N
7	1	M	N
8	3	F	N
9	3	F	Y
10	2	F	Y

Supervised Machine Learning Methods

- Many, many different algorithms for learning relationship between features and label
- Pick a model based on:
  - Amount of training data - Deep neural networks can give great performance, but only for very large number of examples
  - Interpretability – do we want to be able to understand the model, or do we just want the best predictions possible?
  - Batch vs. online – do we want to be able to easily update the model with new examples, or is this a one-time training?

Decision trees

- Old-fashioned method for classification
- Want to learn a flowchart for prediction:

Building decision tree

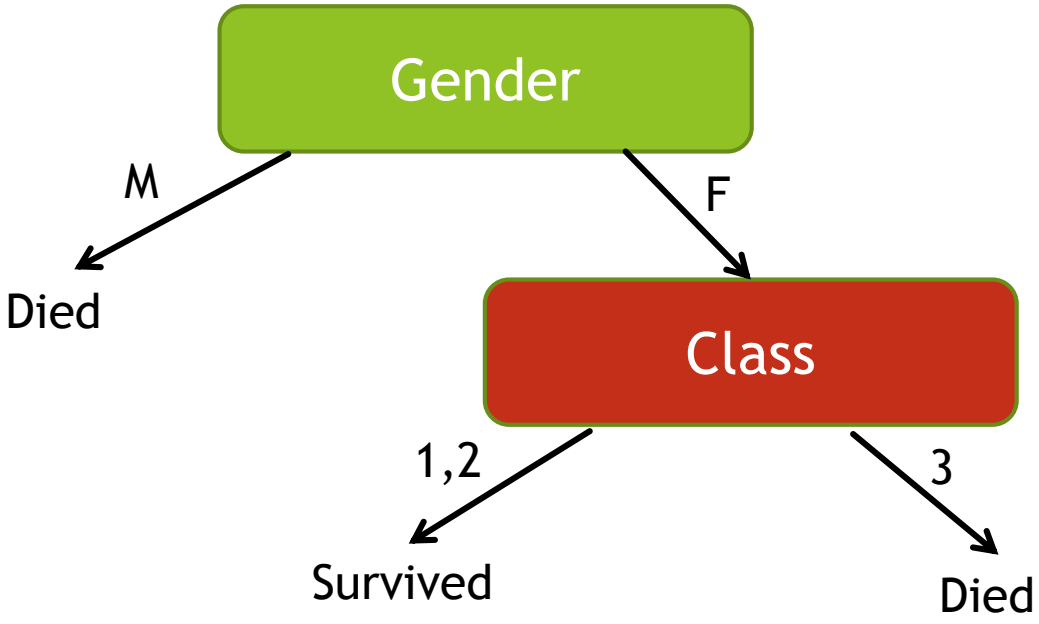
- Want each branch to be as informative as possible
- For Titanic data, want each branch to be mostly survivors or mostly deaths
- First branch options:
  - Class:

	1	2	3
% Survived	63	47	24

  - Gender:

	M	F
% Survived	19	74

Final decision tree

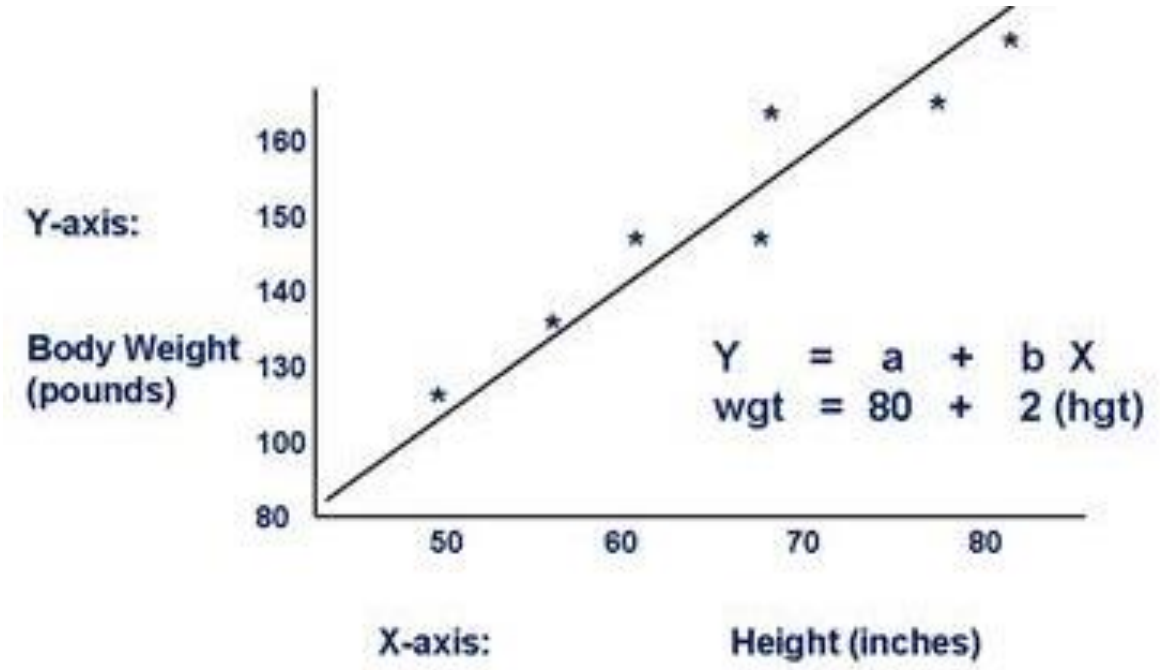


Analysis of decision trees

- For D features, have D choices at first split
  - Takes O(N) to evaluate each choice, so O(D*N)
- Second split: have (D-1) choices for each of 2 splits, O(2*D*(N/2)) = O(D*N)
- Have D total splits, so overall $O(D^2N)$

Linear Regression

- Simplest model for predicting a continuous label



Linear Regression

- Assume that predictor is of the form

  label = a*feat1 + b*feat2 + c*feat3 + … + const
- Pick coefficients a,b,c… using training data
- Turns out that we can calculate these with an equation:

$$\Theta = (X^T X)^{-1} X^T y$$

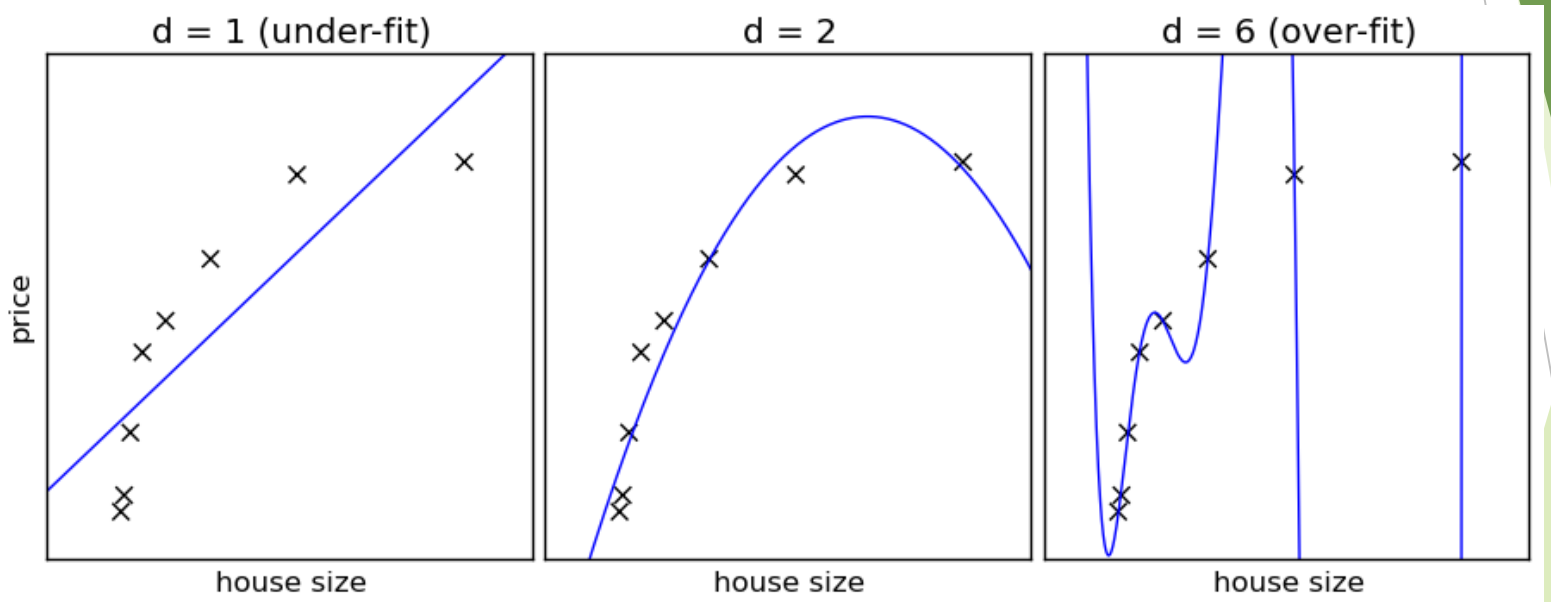
coeff — data matrix — label

- Complexity $O(D^2(N+D))$

Grading our model

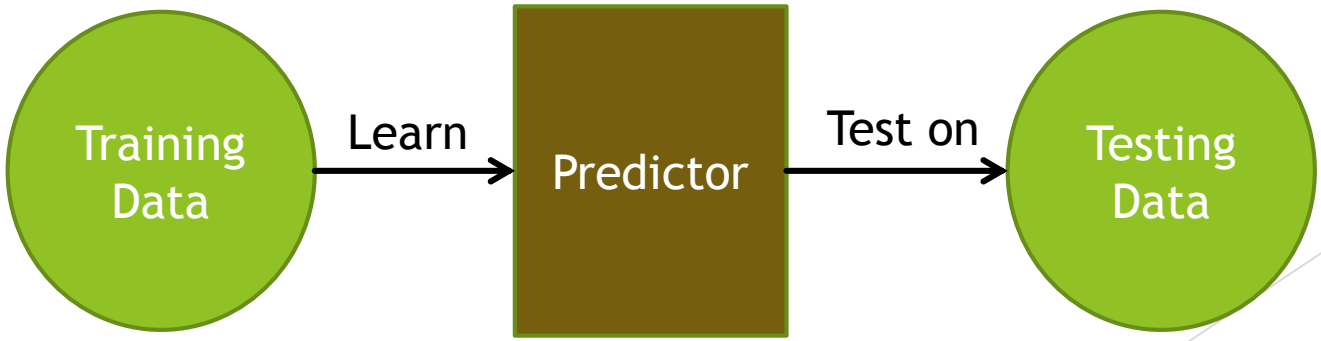
- How do we know if our model is good?
- Might just measure the fit of our model to the data – how well did we predict the labels in our training data?

Overfitting



Overfitting

- We can almost always get perfect accuracy on our training data if we want ("overfitting"), but it may not work well on new data!
- We always measure prediction performance on a *new* dataset, called a test dataset



Visualization

- http://www.r2d3.us/visual-intro-to-machine-learning-part-1/
- Three-D regression: http://www.miabella-llc.com/demo.html

Homework: Titanic ages

- Download www.chrisbaldassano.com/class/titanic.txt which gives age of each passenger and whether they survived (1) or died(0)
- Generate the best (one-layer) decision tree on this data that gives the highest accuracy
- E.g. if age < 20 guess survived, else guess died
- I can get ~62% accuracy