Introduction to Machine Learning Part 1

Yingyu Liang

yliang@cs.wisc.edu

Computer Sciences Department

University of Wisconsin, Madison

[Based on slides from Jerry Zhu]

Read Chapter 1 of this book:

Xiaojin Zhu and Andrew B. Goldberg.

[Introduction to Semi-Supervised Learning](http://www.morganclaypool.com/doi/abs/10.2200/S00196ED1V01Y200906AIM006).

http://www.morganclaypool.com/doi/abs/10.2200/S00196ED1V01Y200906AIM006

Morgan & Claypool Publishers, 2009.

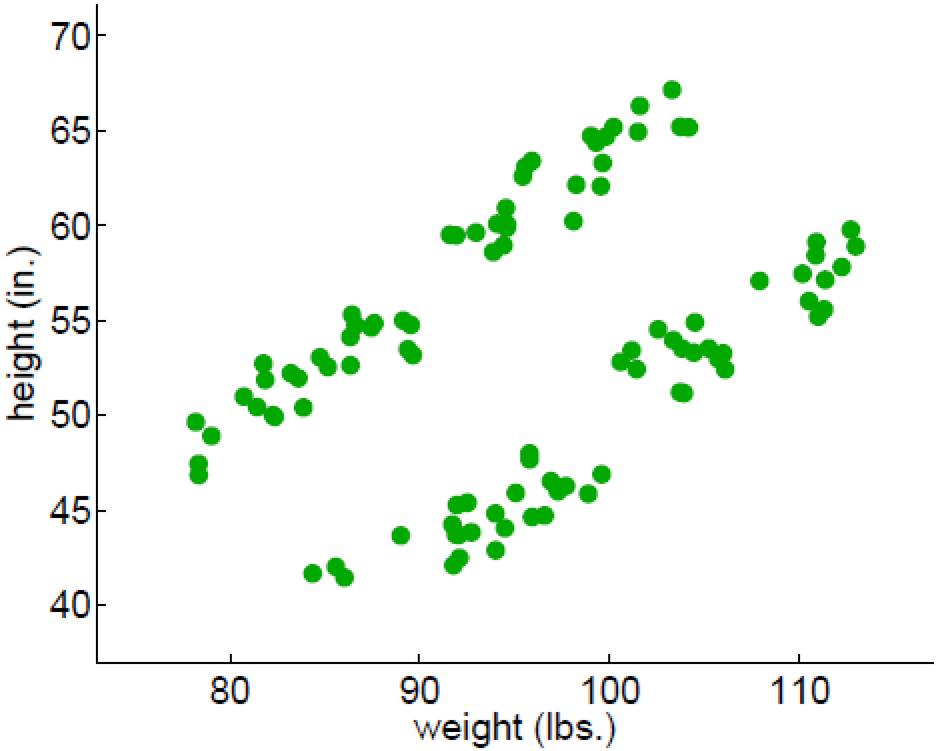
(download from UW computers)

Outline

- Representing “things”
  - Feature vector
  - Training sample
- Unsupervised learning
  - Clustering
- Supervised learning
  - Classification
  - Regression

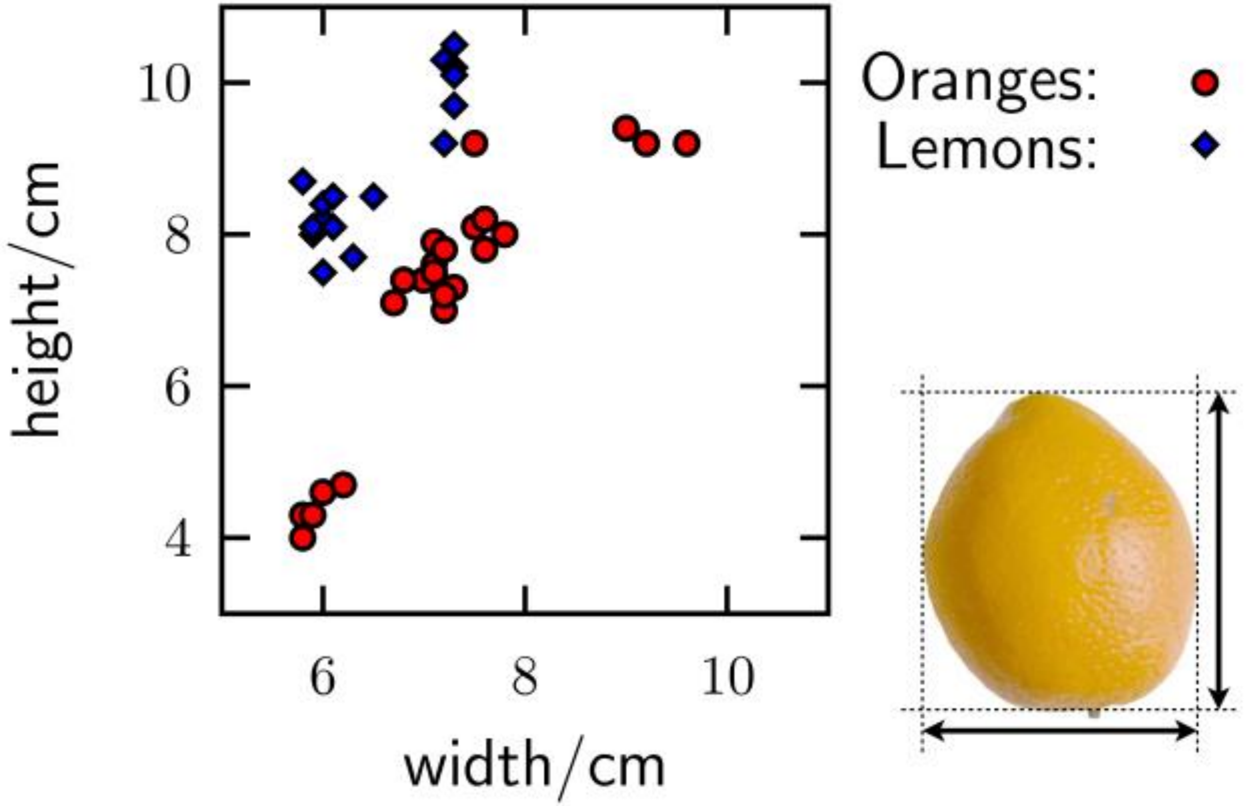
Little green men

- The weight and height of 100 little green men



- What can you learn from this data?

A less alien example



- From Iain Murray http://homepages.inf.ed.ac.uk/imurray2/

Representing "things" in machine learning

- An instance x represents a specific object ("thing")
- x often represented by a D-dimensional feature vector $x = (x_1, \ldots, x_D) \in R^D$
- Each dimension is called a feature. Continuous or discrete.
- x is a dot in the D-dimensional feature space
- Abstraction of object. Ignores any other aspects (two men having the same weight, height will be identical)

Feature representation example

- Text document
  - Vocabulary of size D (~100,000): “aardvark … zulu”
- “bag of word”: counts of each vocabulary entry
  - To marry my true love → (3531:1 13788:1 19676:1)
  - I wish that I find my soulmate this year → (3819:1 13448:1 19450:1 20514:1)
- Often remove stopwords: the, of, at, in, …
- Special “out-of-vocabulary” (OOV) entry catches all unknown words

More feature representations

- Image
  - Color histogram
- Software
  - Execution profile: the number of times each line is executed
- Bank account
  - Credit rating, balance, #deposits in last day, week, month, year, #withdrawals …
- You and me
  - Medical test1, test2, test3, …

Training sample

- A *training sample is a collection of instances* $\mathbf{x}_1, \ldots, \mathbf{x}_n$*, which is the input to the learning process.*
- $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iD})$
- Assume these instances are sampled independently from an unknown (population) distribution $P(x)$
- We denote this by $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} P(x)$, where i.i.d. stands for independent and identically distributed.

Training sample

- A training sample is the “experience” given to a learning algorithm
- What the algorithm can learn from it varies
- We introduce two basic learning paradigms:
  - *unsupervised learning*
  - *supervised learning*

No teacher.

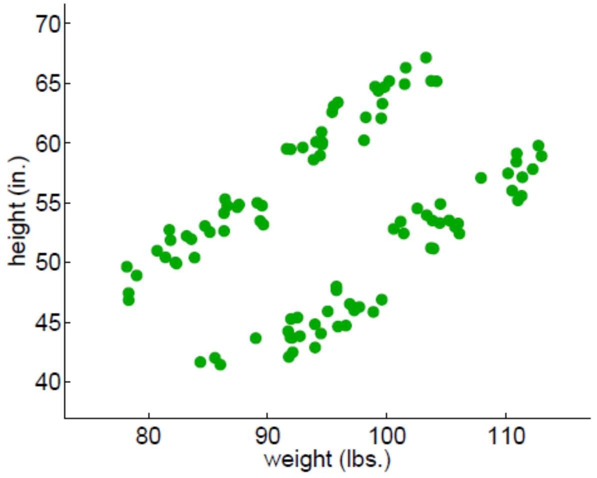
UNSUPERVISED LEARNING

Unsupervised learning

- Training sample $\mathbf{x}_1, \ldots, \boldsymbol{x}_n$, that's it
- No teacher providing supervision as to how individual instances should be handled
- Common tasks:
  - clustering, separate the *n* instances into groups
  - novelty detection, find instances that are very different from the rest
  - dimensionality reduction, represent each instance with a lower dimensional feature vector while maintaining key characteristics of the training samples

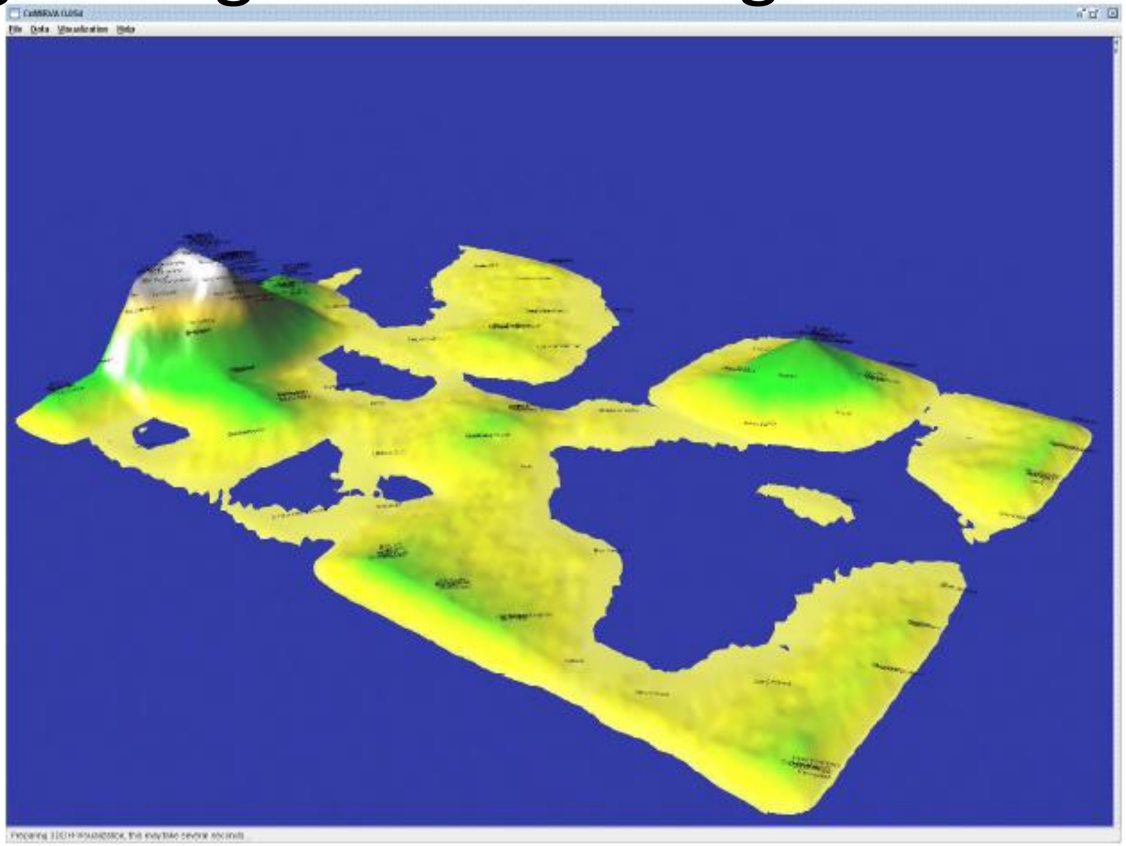
Clustering

- Group training sample into *k* clusters
- How many clusters do you see?
- Many clustering algorithms
  - HAC
  - k-means
  - …



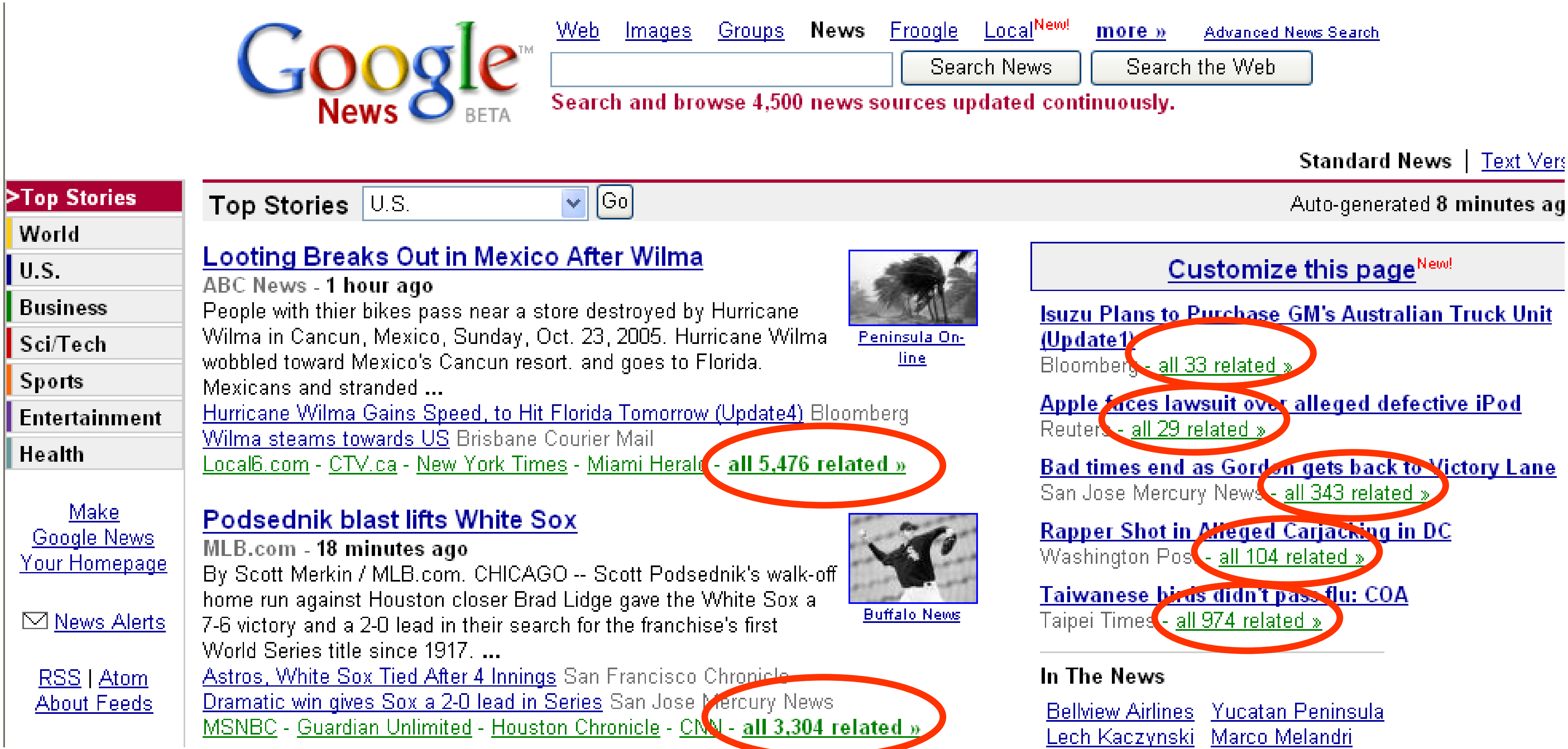
Example 1: music island

- Organizing and visualizing music collection



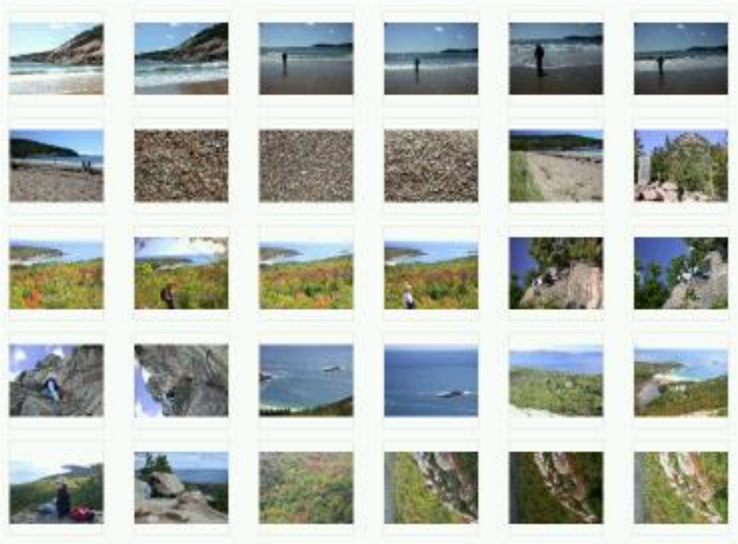
CoMIRVA http://www.cp.jku.at/comirva/

Example 2: Google News



Example 3: your digital photo collection

- You probably have >1000 digital photos, ‘neatly’ stored in various folders…
- After this class you’ll be about to organize them better
  - Simplest idea: cluster them using image creation time (EXIF tag)
  - More complicated: extract image features



Two most frequently used methods

- Many clustering algorithms. We'll look at the two most frequently used ones:
  - Hierarchical clustering

    Where we build a binary tree over the dataset
  - K-means clustering

    Where we specify the desired number of clusters, and use an iterative algorithm to find them

Hierarchical clustering

- Very popular clustering algorithm
- Input:
  - A dataset $x_1, \ldots, x_n$, each point is a numerical feature vector
  - Does NOT need the number of clusters

Hierarchical Agglomerative Clustering

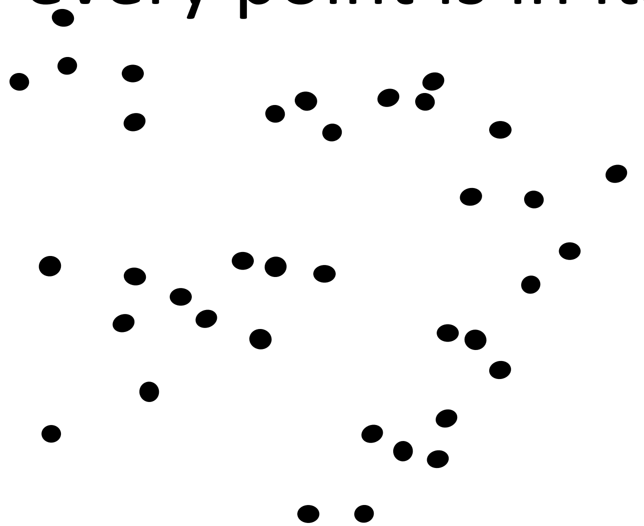
Input: *a training sample* $\{\mathbf{x}_i\}_{i=1}^{n}$; *a distance function* $d()$.
1. Initially, place each instance in its own cluster (called a singleton cluster).
2. ***while*** (*number of clusters* > 1) ***do:***
3. Find the closest cluster pair A, B, *i.e., they minimize* $d(A, B)$.
4. Merge A, B *to form a new cluster.*
Output: *a binary tree showing how clusters are gradually merged from singletons to a root cluster, which contains the whole training sample.*

- Euclidean (L2) distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{s=1}^{D} (x_{is} - x_{js})^2}.$$

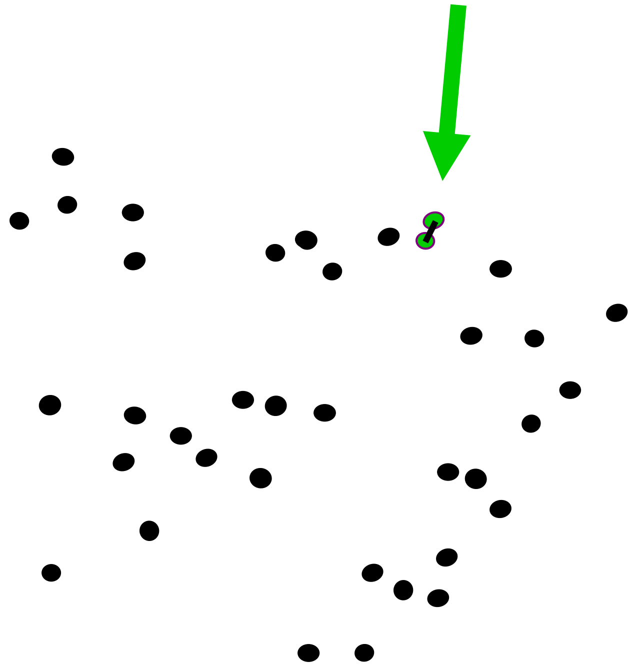
Hierarchical clustering

- Initially every point is in its own cluster



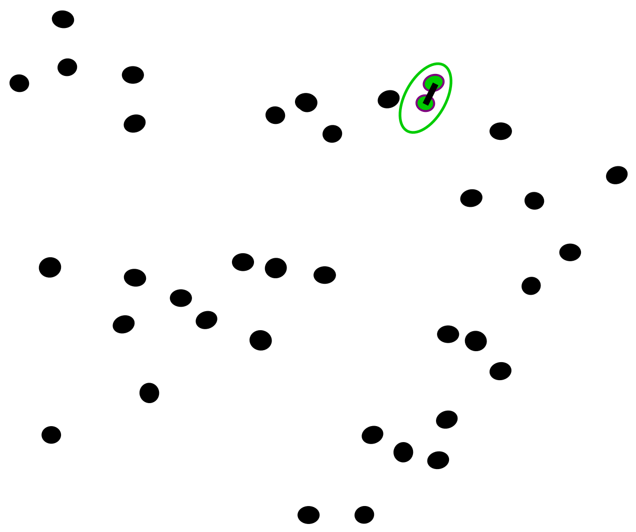
Hierarchical clustering

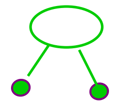
- Find the pair of clusters that are the closest



Hierarchical clustering

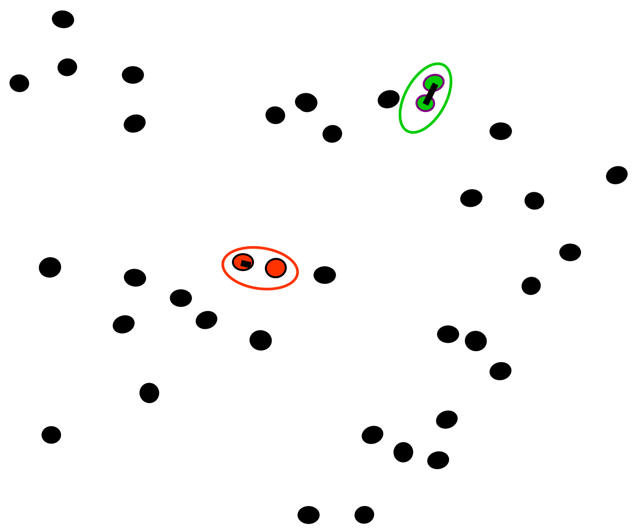
- Merge the two into a single cluster





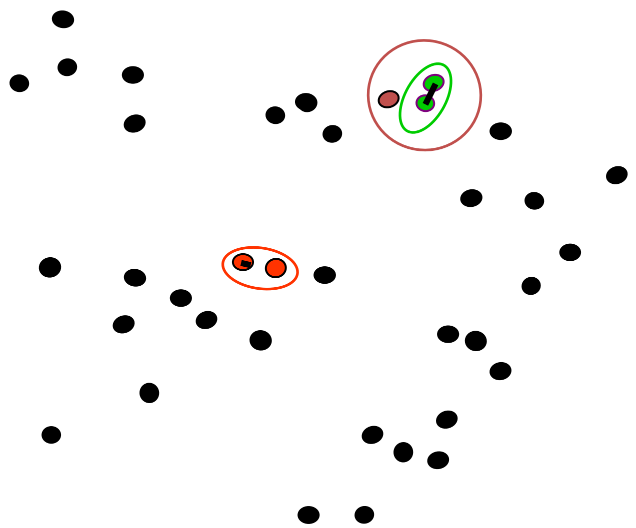
Hierarchical clustering

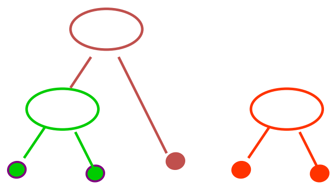
- Repeat…



Hierarchical clustering

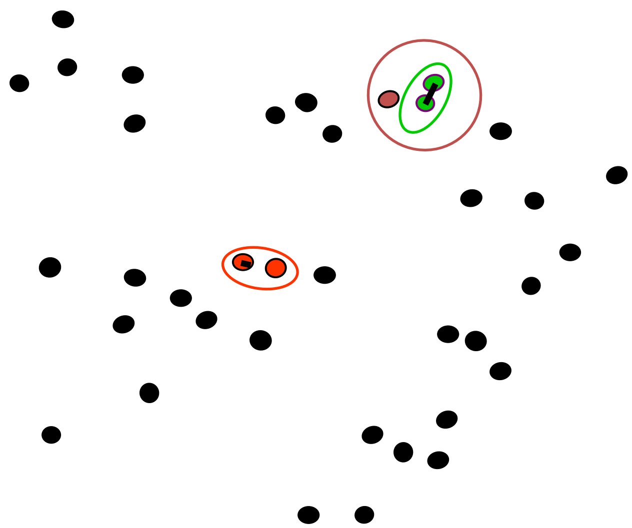
- Repeat…

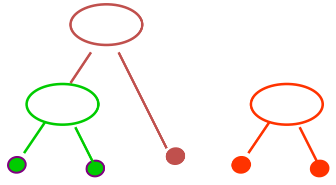




Hierarchical clustering

- Repeat…until the whole dataset is one giant cluster
- You get a binary tree (not shown here)





Hierarchical clustering

- How do you measure the closeness between two clusters?

Hierarchical clustering

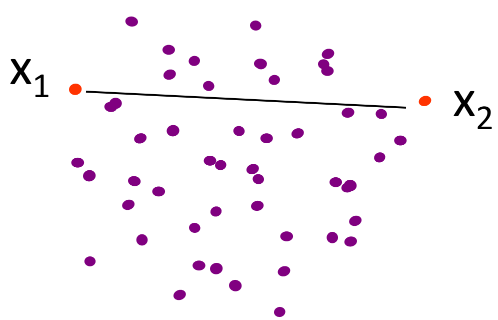
- How do you measure the closeness between two clusters? At least three ways:
  - Single-linkage: the shortest distance from any member of one cluster to any member of the other cluster. Formula?
  - Complete-linkage: the greatest distance from any member of one cluster to any member of the other cluster
  - Average-linkage: you guess it!

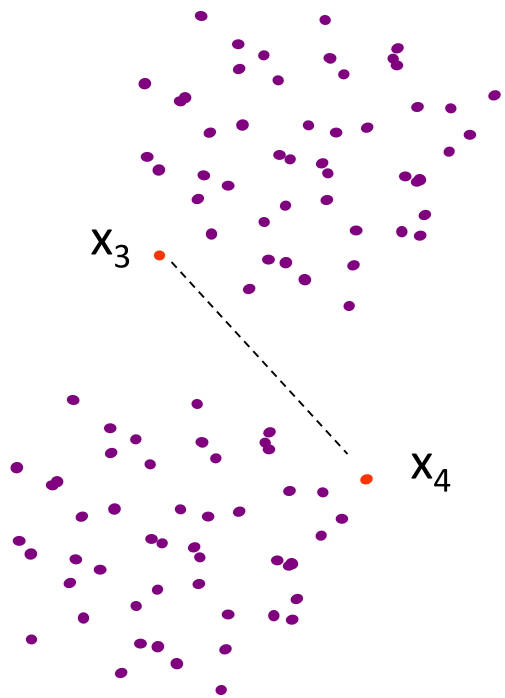
Hierarchical clustering

- The binary tree you get is often called a dendrogram, or taxonomy, or a hierarchy of data points
- The tree can be cut at various levels to produce different numbers of clusters: if you want k clusters, just cut the (k-1) longest links
- Sometimes the hierarchy itself is more interesting than the clusters
- However there is not much theoretical justification to it…

Advance topics

- **Constrained clustering**: What if an expert looks at the data, and tells you
  - "I think x1 and x2 **must** be in the same cluster" (must-links)
  - "I think x3 and x4 **cannot** be in the same cluster" (cannot-links)





Advance topics

- This is clustering with supervised information (must-links and cannot-links). We can
  - Change the clustering algorithm to fit constraints
  - Or , learn a better distance measure
- See the book

Constrained Clustering: Advances in Algorithms, Theory, and Applications

Editors: Sugato Basu, Ian Davidson, and Kiri Wagstaff

http://www.wkiri.com/conscluster/