Clustering

CPS270
Ron Parr

material from: Lise Getoor, Andrew Moore, Tom Dietterich, Sebastian Thrun, Rich Maclin

Unsupervised Learning

- Supervised learning: Data <x1, x2, … xn, y>
- Unsupervised Learning: Data <x1, x2, … xn>

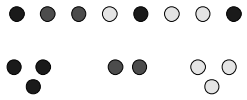
- So, what's the big deal?
- Isn't y just another feature?
- No explicit performance objective
  - Bad news: Problem not necessarily well defined without further assumptions
  - Good news: Results can be useful for more than predicting y

Model Learning

- Produce a global summary of the data
- Not an exact copy
- Assume data are sampled from a larger set that has some easily summarized properties
  - Cluster analysis: What things should be grouped together?
  - Density estimation: How are things distributed in space?

Cluster Analysis

- Decomposition or partition of data into groups where
  - the points in one group are similar to each other
  - and are as different as possible from the points in other groups



Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with similar claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

Example

- Households:
  location, income, number of children, rent/own, crime rate, number of cars

- Appropriate clustering may depend on use:
  - Goal to minimize delivery time ⇒ cluster by location
  - others?
  - (Suggests problem is ill defined)

Clustering

- Decomposition or partition of data into groups so that
  - Points in one group are **similar** to each other
  - Are as **different** as possible from the points in other groups
- Measure of **distance** is fundamental
- Explicit representation:
  - D(x(i),x(j)) for each x
  - Only feasible for small domains
- Implicit representation by measurement:
  - Distance computed from features
  - We've already seen a number of different ways of doing this

Clustering

- **Huge** body of work
- (aka unsupervised learning, segmentation, …)
- Major difficulty: Measuring success
- Evaluation depends on goals
- If goal is to find 'interesting' clusters, this is rather difficult to quantify
- However, for some probabilistic methods, there are tools for validating our models

Families of Clustering Algorithms

- Partition-based methods
  - e.g., K-means
- Hierarchical clustering
  - e.g., hierarchical agglomerative clustering
- Probabilistic model-based clustering
  - e.g., mixture models
- Graph-based Methods
  - e.g., spectral methods

Partition-based Clustering Algorithms

- Given set of n data points $D=\{x^{(1)}, \ldots, x^{(n)}\}$ partition data into k clusters $C = \{C_1, \ldots, C_k\}$ such that each x(i) is assigned to a unique C_j and Score(C,D) is minimized/maximized
- Combinatorial optimization: searching for allocation of n objects into k classes that maximizes score function
- Number of possible allocations $\approx k^n$
- Exhaustive search is intractable
- Resort to iterative improvement

Possible Scoring Functions

- Score function:
  - clusters compact ⇒ minimize within cluster distance, wc(C)
  - clusters should be far apart ⇒ maximize distance between clusters, bc(C)
- Given a clustering C, assign cluster centers, c_k
  - if points belong to space where means make sense, we can use the centroid of the points in the cluster:

$$c_k = \frac{1}{n_k} \sum_{x \in C_k} x$$

- wc(C) – sum-of-squares within cluster distance

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x \in C_k} d(x, c_k)$$

- bc(C) – distance between clusters

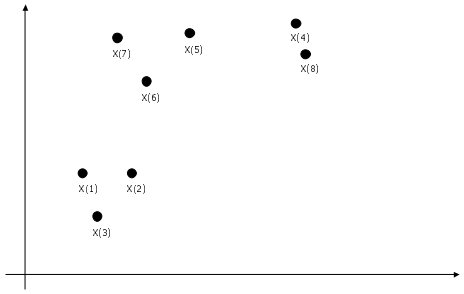
$$bc(C) = \sum_{1 \le j < k \le K} d(c_j, c_k)$$

- Score(C,D) – f(wc(C), bc(C))

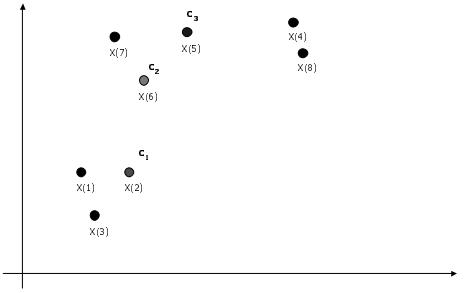
K-means

- Start with randomly chosen cluster centers
- Assign points to closest cluster
- Recompute cluster centers
- Reassign points
- Repeat until no changes

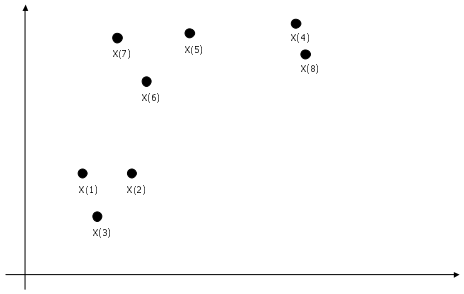
K-means example



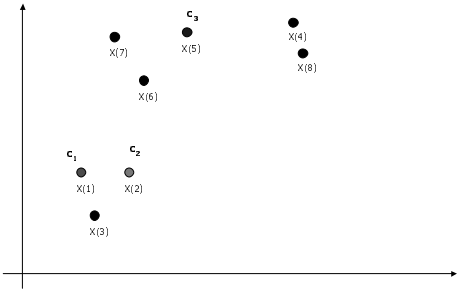
K-means example



K-means example #2



K-means example #2



Complexity

- Does algorithm terminate?
- Does algorithm converge to optimal clustering?
- Time complexity one iteration?

Understanding k-Means

- Models data as coming from spherical Gaussians centered at cluster centers
- log P(data) ~ sum of squared distances

$$P(x_i \in c_j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - c_j)^2}{2\sigma^2}\right)$$

$$P(data) = \prod_i P(x_i \in c_{clustering(i)})$$

$$\log(P(data)) = \alpha \sum_i (x_i - c_{clustering(i)})^2$$

Understanding k-Means

- Each step of k-Means increases log(P(data))
  - Reassigning
  - Recomputing means
- Fixed number of assignments and monotonic score implies convergence

Algorithm Variations

- Recompute centroid as soon as a point is reassigned
- Allow merge and split of clusters
- Methods for improving solution accuracy?
- Many cases where means do not make sense
  - k-mediods – use one of the data points as center
  - categorical data
- What if data set is too large for algorithm to be tractable?
  - compress data by replacing groups of objects by 'condensed representation'
  - Sub-sample data