CMSC828G **Principles of Data Mining** Lecture #9

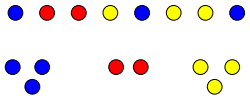
- Today's Reading:
  - HMS, chapter 9
- Today's Lecture:
  - Descriptive Modeling
  - Clustering Algorithms
- Upcoming:
  - hw2 available on web page this evening
  - project proposals due 3/12

Descriptive Models

- model presents the main features of the data, a global summary of the data
  - cluster analysis
  - density estimation

Cluster Analysis

- decomposing or partitioning a data set into groups so that
  - the points in one group are similar to each other
  - and are as different as possible from the points in other groups



This isn't really clustering, it's just binning the objects...

General Applications of Clustering

- Pattern Recognition
- Spatial Data Analysis
  - create thematic maps in GIS by clustering feature spaces
  - detect spatial clusters and explain them in spatial data mining
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns

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 a high average 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
- The appropriate clustering will depend on goals:
  - minimize delivery time ⇒ cluster by location
  - others?

Clustering

- decomposing or partitioning a data set into groups so that
  - the points in one group are **similar** to each other
  - and 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
- Measurement:
  - distance computed from features
  - we saw a number of different ways of doing this in ch. 2

Clustering

- **Huge** body of work
- (aka unsupervised learning, segmentation, …)
- One of the major difficulties is in evaluating the success of a method
- validity depends on goals
- if goal is to find 'interesting' clusters, this is rather difficult to quantify
- however, for our probabilistic methods, we will present some tools for validating our models

Choosing an Algorithm

- As we will see, different algorithms will result in clusters of different 'shapes'
- The appropriate shape will depend on the application and should be consider when choosing an algorithm
- match method to objectives

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

Partition-based Clustering Algorithms

- Given set of n data points D={x(1), …, 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 $\approx n^k$
- exhaustive typically finding the optimal solution is intractable
- Resort to iterative improvement

Score Function

- Score function:
  - clusters compact $\Rightarrow$ minimize within cluster distance, wc(C)
  - clusters should be far apart $\Rightarrow$ 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

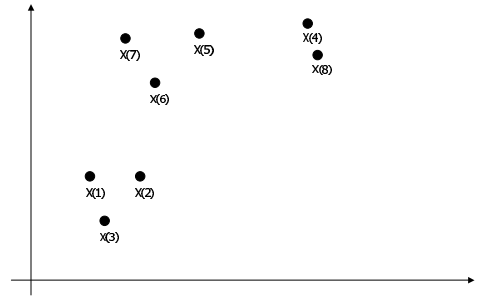
$$bc(C) = \sum_{1 \le j < k \le K} d(c_j, c_k)^2$$

- Score(C,D) = f(wc(C), bc(C))

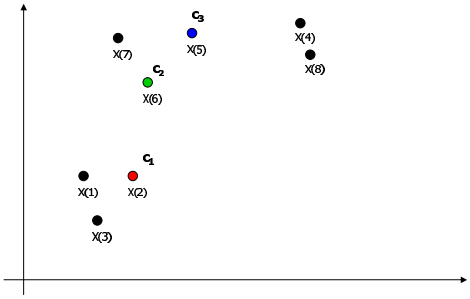
K-means

- Idea:
  - Start with randomly chosen cluster centers
  - Assign points to give greatest increase in score
  - Recompute cluster centers
  - Reassign points
  - Repeat until no changes

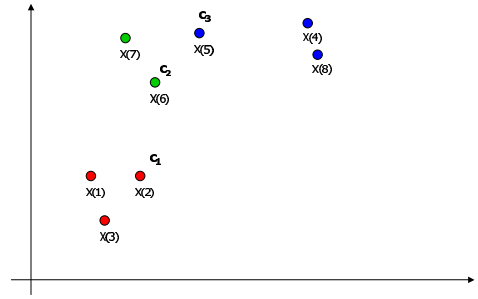
K-means example



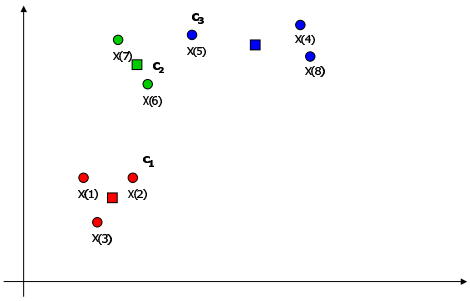
K-means example



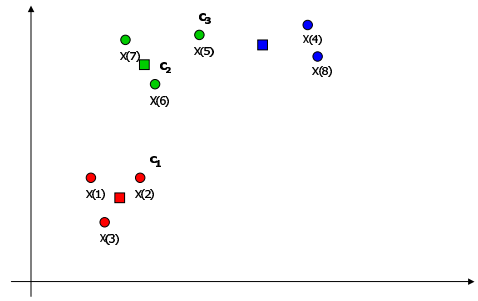
K-means example



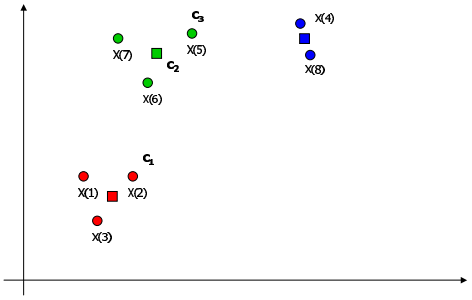
K-means example



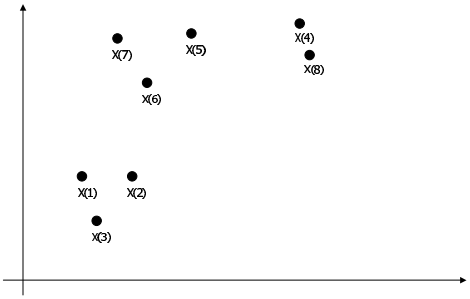
K-means example



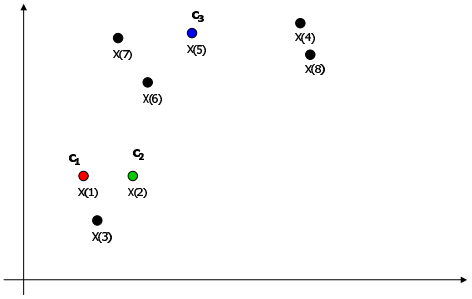
K-means example



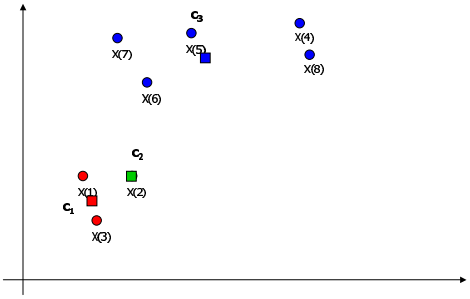
K-means example #2



K-means example #2



K-means example #2



Demos

k-means applet
another demo
image example

Complexity

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

Algorithm Variations

- recompute centroid as soon as a point is reassigned
- allow merge and split of clusters
- methods for improving solution accuracy?
- in 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'

Binary Variables

- A contingency table for binary data

		Object *j*		
		1	0	*sum*
Object *i*	1	*a*	*b*	*a+b*
	0	*c*	*d*	*c+d*
	sum	*a+c*	*b+d*	*p*

- Simple matching coefficient (invariant, if the binary variable is *symmetric*): $d(i,j) = \frac{b+c}{a+b+c+d}$
- Jaccard coefficient (noninvariant if the binary variable is *asymmetric*): $d(i,j) = \frac{b+c}{a+b+c}$

Dissimilarity between Binary Variables

- Example

Name	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	Y	N	P	N	N	N
Mary	Y	N	P	N	P	N
Jim	Y	P	N	N	N	N

  - attributes are asymmetric binary
  - let the values Y and P be set to 1, and the value N be set to 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

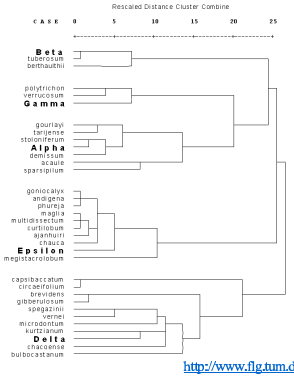
$$d(i,j) = \frac{p-m}{p}$$

- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

Hierarchical Clustering

- rather than deciding the number of clusters K at the start, build a hierarchy of nested clusters
- either gradually
  - merge points (agglomerative)
  - divide superdusters (divisive)
- result of either approach can be shown as a dendogram which depicts the sequence of merges or splits

Dendogram



http://www.flg.tum.de/plopz/mm/mt/dendro.htm

Agglomerative Methods

- based on measures of distance between clusters

```
for i = 1 to n
    let Ci = {x(i)}
while there is more than one cluster left do
    let Ci and Cj be the pair of clusters with minimum
      D(Ci, Cj)
    Ci = Ci ∪ Cj
    remove Cj
end
```

- time complexity?
- space complexity?

Measuring Distances between Clusters

- single link/nearest neighbor method:
  $D(C_i, C_j) = \min\{d(x,y) \mid x \in C_i, y \in C_j\}$
- complete link/furthest neighbor method:
  $D(C_i, C_j) = \max\{d(x,y) \mid x \in C_i, y \in C_j\}$
- average link:
  $D(C_i, C_j) = \text{avg}\{d(x,y) \mid x \in C_i, y \in C_j\}$
- centroid measure:
  $D(C_i, C_j) = d(c_i, c_j)$ where c_i and c_j are centroids
- Ward's measure: difference between total within cluster sum of squares for the two clusters separately and the sum of squares error in the merged cluster

Divisive Methods

- Begin with a single cluster, consisting of all the data points
- split into components
- ultimately ends with a partition in which each cluster has a single point
- **monolithic** methods split cluster using one variable at a time
- **polythetic** methods make splits based on all of the variables together; difficulty comes in how to choose potential splits
- in general, divisive methods are less widely used than agglomerative methods

Demos

- http://www.cs.mcgill.ca/~papou/#applet
- http://www.biology.ualberta.ca/jbrzusto/cluster.php#ClusterCalc

Next Time

- Reading:
  - HMS, chapter 9 cont.

References

- *Principles of Data Mining*, Hand, Mannila, Smyth. MIT Press, 2001.
- *Data Mining*, Jiawei Han and Micheline Kamber. Morgan Kaufmann, 2001. slides: http://www.cs.sfu.ca/~han/dmbook
- http://www.flg.tum.de/pbpz/mm/mt/dendro.htm