

Clustering

1

Informal goal

- Given set of objects and measure of similarity between them, group similar objects together
- What mean by “similar”?
- What is good grouping?
- Computation time / quality tradeoff

2

General types of clustering

- “Soft” versus “hard” clustering
 - **Hard**: partition the objects
 - each object in exactly one partition
 - **Soft**: assign degree to which object in cluster
 - view as probability or score
- **flat** versus **hierarchical** clustering
 - hierarchical = clusters within clusters

3

Applications:

Many

- biology
- astronomy
- computer aided design of circuits
- information organization
- marketing
- ...

4

Clustering in information search and analysis

- Group information objects
 - ⇒ **discover topics**
 - ? **other groupings desirable**
- Clustering versus classifying
 - classifying: have **pre-determined classes** with example members
 - clustering:
 - get groups of similar objects
 - added problem of labeling clusters by topic
 - e.g. common terms within cluster of docs.

5

Example applications in search

- **Query evaluation**: cluster pruning (§7.1.6)
 - cluster all documents
 - choose representative for each cluster
 - evaluate query w.r.t. cluster reps.
 - evaluate query for docs in cluster(s) having most similar cluster rep.(s)
- **Results presentation**: labeled clusters
 - cluster only query results
 - e.g. Yippy.com (metasearch)

hard / soft? flat / hier?

6

Issues

- What, if any, **attributes** represent **items** for clustering purposes?
- What is **measure of similarity** between **items**?
 - General objects and matrix of pairwise similarities
 - Objects with specific properties that allow other specifications of measure
 - Most common:
 - Objects are **d-dimensional vectors**
 - » Euclidean distance
 - » cosine similarity
- What is **measure of similarity** between **clusters**?

7

Issues continued

- Cluster **goals**?
 - Number of clusters?
 - flat or hierarchical clustering?
 - cohesiveness of clusters?
- How **evaluate cluster** results?
 - relates to measure of closeness between clusters
- **Efficiency** of clustering **algorithms**
 - large data sets => external storage
- Maintain clusters in **dynamic setting**?
- Clustering **methods**? - **MANY!**

8

Quality of clustering

- In applications, quality of clustering depends on how **well solves problem at hand**
- Algorithm uses **measure of quality** that can be **optimized**, but that may or may not do a good job of capturing application needs.
- Underlying **graph-theoretic problems** usually **NP-complete**
 - e.g. graph partitioning
- Usually algorithm **not finding optimal clustering**

9

General types of clustering methods

- **constructive**: decide in what cluster each object belongs and don't change
 - often faster
- **iterative improvement**: start with a clustering and move objects around to see if can improve clustering
 - often slower but better

10

Vector model: K- means algorithm

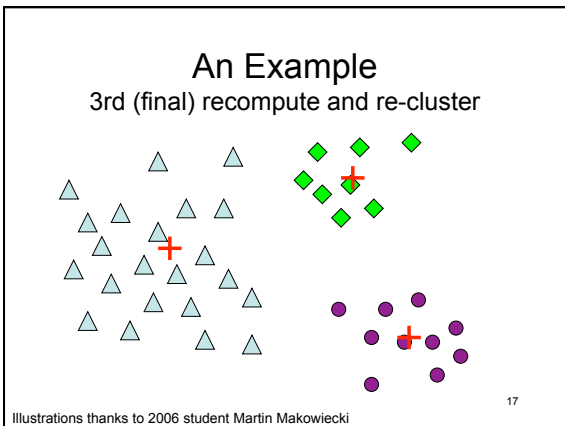
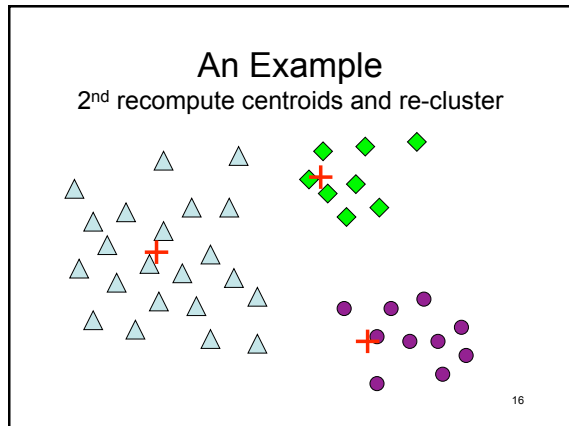
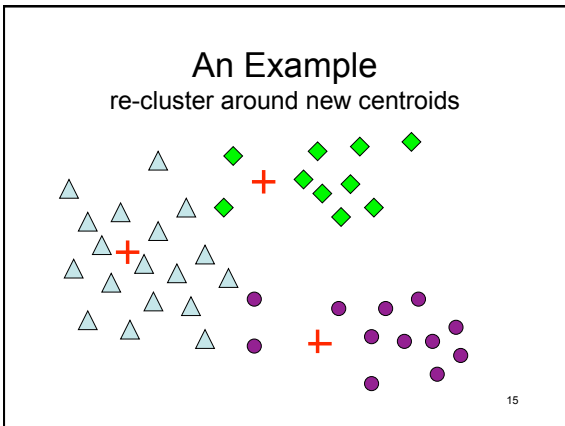
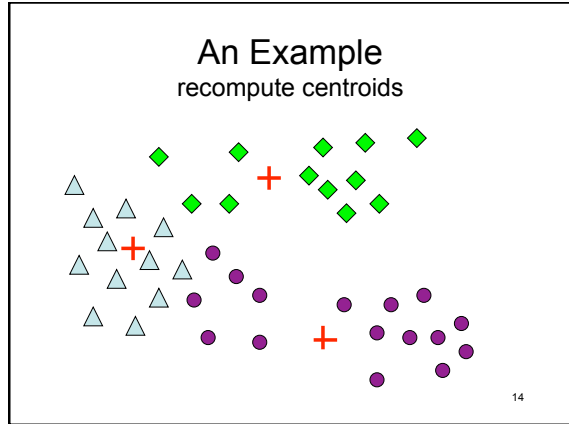
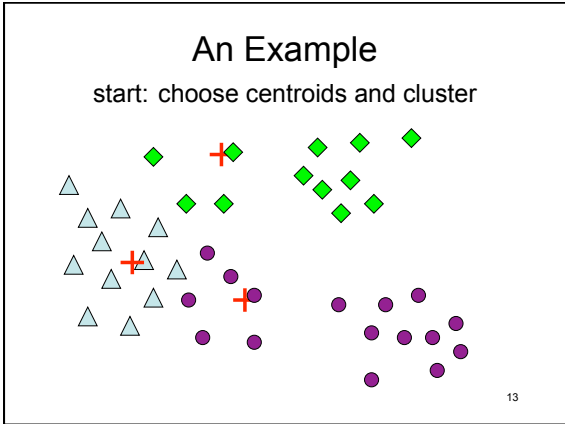
- Well known, well used
- Flat clustering
- Number of clusters picked ahead of time
- Iterative improvement
- Uses notion of **centroid**
- Typically uses Euclidean distance

11

K-means overview

- **Choose k** points among set to be clustered
 - Call them **k centroids**
 - not required to be in set to be clustered
- For **each point** not selected, assign it to its **closest centroid**
 - Assignments give initial clustering
- Until "happy" do:
 - **Recompute centroids** of clusters:
 - centroid of set of vectors $\{v_i | 1 \leq i \leq n\} = 1/n * \sum_{i=1}^n v_i$**
 - New centroids may not be points of original set
 - **Reassign all points** to closest centroid
 - Updates clusters

12



Illustrations thanks to 2006 student Martin Makowiecki

Details for K-means

- Need definition of **centroid**
 $c_i = 1/|C_i| \sum_{x \in C_i} x$ for i^{th} cluster C_i containing objects x
 notion of **sum of objects** ?
- Need definition of **distance to (similarity to) centroid**
- Typically vector model with Euclidean distance
- minimizing sum of squared distances of each point to its centroid = **Residual Sum of Squares**

$$RSS = \sum_{i=1}^K \sum_{x \in C_i} dist(c_i, x)^2$$

18

K-means performance

- Can prove RSS decreases with each iteration, so **converge**
- Can achieve **local optimum**
 - No change in centroids
- Running time depends on how demanding stopping criteria
- Works well in practice
 - speed
 - quality

19

Time Complexity of K-means

- Let t_{dist} be the time to calculate the distance between two objects
- Each **iteration** time complexity:
 - $O(K \cdot n \cdot t_{\text{dist}})$
 - n = number of objects
- Bound number of **iterations** I giving
 - $O(I \cdot K \cdot n \cdot t_{\text{dist}})$
- for **m -dimensional vectors**:
 - $O(I \cdot K \cdot n \cdot m)$
 - m large and centroids not sparse

20

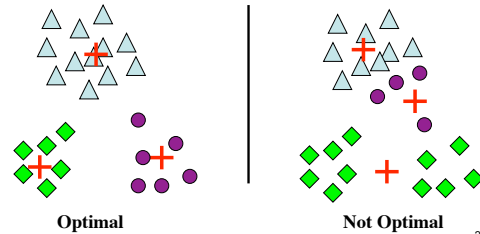
Space Complexity of K-means

- Store points and centroids
 - vector model: $O((n + K)m)$
- External algorithm versus internal?
 - store k centroids in memory
 - run through points each iteration

21

Choosing Initial Centroids

- Bad initialization leads to poor results



22

Choosing Initial Centroids

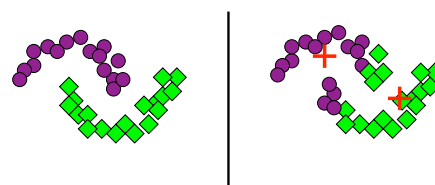
Many people spent much time examining how to choose seeds

- Random
 - Fast and easy, but often poor results
- Run random multiple times, take best
 - Slower, and still no guarantee of results
- Pre-conditioning
 - remove outliers
- Choose seeds algorithmically
 - run hierarchical clustering on sample points and use resulting centroids
 - Works well on small samples and for few initial centroids

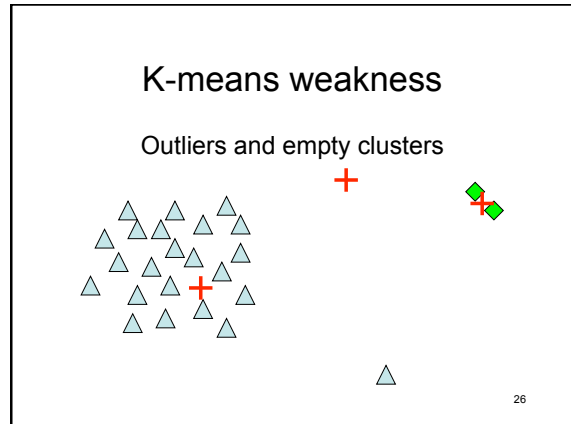
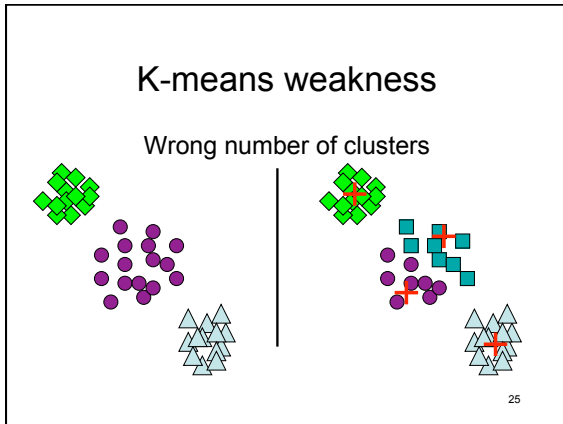
23

K-means weakness

Non-globular clusters



24



- ### Real cases tend to be harder
- Different attributes of the feature vector have vastly different sizes
 - size of star versus color
 - Can weight different features
 - how weight greatly affects outcome
 - Difficulties can be overcome
- 27