Clustering Algorithms: K-means

Last time: K-means

- Need notion of centroid
  \[ c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \]
  for \( i \)th cluster \( C_i \) containing objects \( x \)
  - notion of sum of objects?
- Need notion of distance to / similarity to centroid
- Typically (we used) vector model with Euclidean distance

• minimizing \( \text{RSS} = \sum_{i=1}^{K} \sum_{x \in C_i} \text{dist}(c_i, x)^2 \)
  residual sum of squares
Illustrations thanks to 2006 student Martin Makowiecki

An Example
Time Complexity of K-means

- Let $t_{dist}$ be the time to calculate the distance between two objects.
- Each iteration time complexity:
  \[ O(Kn t_{dist}) \]
  
  $K =$ number of clusters (centroids)
  
  $n =$ number of objects

- Bound number of iterations $I$ giving
  \[ O(I K n t_{dist}) \]

- for m-dimensional vectors:
  \[ O(I K n m) \]
  
  $m$ large and centroids not sparse
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

Choosing Initial Centroids

• Bad initialization leads to poor results
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

K-means weakness

Different sized clusters
K-means weakness

Clusters of different densities

Non-globular clusters
K-means weakness

Wrong number of clusters

K-means weakness

Outliers and empty clusters
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