Introduction to Data Science

GIRI NARASIMHAN, SCIS, FIU
Clustering
Clustering dogs using height & weight

Figure 7.1: Heights and weights of dogs taken from three varieties
Clustering dogs using height & weight

Figure 7.1: Heights and weights of dogs taken from three varieties
Clustering is the process of making clusters, which put similar things together into same cluster ...

And put dissimilar things into different clusters

Need a similarity function

Need a similarity distance function

Convenient to map items to points in space
Distance Functions

- Jaccard Distance
- Hamming Distance
- Euclidean Distance
- Cosine Distance
- Edit Distance
- ...

What is a distance function

- $D(x,y) \geq 0$
- $D(x,y) = D(y,x)$
- $D(x,y) \leq D(x,z) + D(z,y)$
Clustering Strategies

- Hierarchical or Agglomerative
  - Bottom-up

- Partitioning methods
  - Top-down

- Density-based

- Cluster-based

- Iterative methods
Curse of Dimensionality

- N points in d-dimensional unit (hyper)sphere
  - If d = 1, then average distance = 1/3
  - As d gets larger, what is the average distance? Distribution of distances?
    - # of nearby points for any given point vanishes. So, clustering does not work well
    - # of points at max distance (~sqrt(d)) also vanishes. Real range actually very small
  - Angle ABC given 3 points approaches 90
    - Denominator grows linearly with d
    - Expected cos = 0 since equal points expected in all 4 quadrants

\[
\frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \sqrt{\sum_{i=1}^{d} y_i^2}}
\]
Hierarchical Clustering
Hierarchical Clustering

- Starts with each item in different clusters
- Bottom up
- In each iteration
  - Two clusters are identified and merged into one
- Items are combined as the algorithm progresses

**Questions:**
- How are clusters represented
- How to decide which ones to merge
- What is the stopping condition

**Typical algorithm:** find smallest distance between nodes of different clusters
Hierarchical Clustering
Output of Clustering: Dendrogram
Measures for a cluster

- Radius: largest distance from a centroid
- Diameter: largest distance between some pair of points in cluster
- Density: # of points per unit volume
- Volume: some power of radius or diameter
- Tightness, separation, …
- **Good cluster**: when diameter of each cluster is much larger than its nearest cluster or nearest point outside cluster
Stopping condition for clustering

- Cluster radius or diameter crosses a threshold
- Cluster density drops below a certain threshold
- Ratio of diameter to distance to nearest cluster drops below a certain threshold
K-Means Clustering
Start

Example from Andrew Moore's tutorial on Clustering.
Example from Andrew Moore's tutorial on Clustering.
Example from Andrew Moore's tutorial on Clustering.
Example from Andrew Moore’s tutorial on Clustering.
K-Means Clustering [McQueen ’67]

Repeat

- Start with randomly chosen cluster centers
- Assign points to give greatest increase in score
- Recompute cluster centers
- Reassign points

until (no changes)

Try the applet at: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletH.html
How to find K for K-means?

Average Diameter

Correct value of $k$

Number of Clusters
Comparisons

- Hierarchical clustering
  - Number of clusters not preset.
  - Complete hierarchy of clusters
  - Not very robust, not very efficient.

- K-Means
  - Need definition of a mean. Categorical data?
  - Can be sensitive to initial cluster centers; Stopping condition unclear
  - More efficient and often finds optimum clustering.
Implementing Clustering
A catalog of 2 billion “sky objects” represents objects by their radiation in 7 dimensions (frequency bands).

**Problem:** cluster into similar objects, e.g., galaxies, nearby stars, quasars, etc.

**Sloan Sky Survey** is a newer, better version.
Curse of Dimensionality

- Assume random points within a bounding box, e.g., values between 0 and 1 in each dimension.
- In 2 dimensions: a variety of distances between 0 and 1.41.
- In 10,000 dimensions, the difference in any one dimension is distributed as a triangle.
How to find K for K-means?
BFR Algorithm

- **BFR (Bradley-Fayyad-Reina)** – variant of K-means for very large (disk-resident) data sets.
- Assumes that clusters are normally distributed around a centroid in Euclidean space.
  - SDs in different dimensions may vary
Points read “chunk” at a time.

Most points from previous chunks summarized by simple statistics.

First load handled by some sensible approach:

1. Take small random sample and cluster optimally.
2. Take sample; pick random point, & $k - 1$ more points incrementally, each as far from previously points as possible.
1. **Discard set**: points close enough to a centroid to be summarized.

2. **Compression set**: groups of points that are close together but not close to any centroid. They are summarized, but not assigned to a cluster.

3. **Retained set**: isolated points.
Points in the retained set

Compressed sets

A cluster. Its points are in the discard set.

Its centroid
BFR: How to summarize?

- **Discard Set & Compression Set**: N, SUM, SUMSQ
- 2d + 1 values
- Average easy to compute
  - SUM/N
- SD not too hard to compute
  - VARIANCE = (SUMSQ/N) – (SUM/N)^2
BFR: Processing

- Maintain N, SUM, SUMSQ for clusters
- Policies for merging compressed sets needed and for merging a point in a cluster
- Last chunk handled differently
  - Merge all compressed sets
  - Merge all retained sets into nearest clusters
- BFR suggests **Mahalanobis Distance**
Mahalanobis Distance

- Normalized Euclidean distance from centroid.
- For point \((x_1, \ldots, x_k)\) and centroid \((c_1, \ldots, c_k)\):
  1. Normalize in each dimension: 
     \[
     y_i = \frac{x_i - c_i}{\sigma_i}
     \]
  2. Take sum of the squares of the \(y_i\) 's.
  3. Take the square root.
- For Gaussian clusters, \(~65\%\) of points within SD dist
GRPGF

Algorithm
GRPGF Algorithm

- Works for non-Euclidean distances
- Efficient, but approximate
- Works well for high dimensional data
  - Exploits orthogonality property for high dim data
- Rules for splitting and merging clusters
Clustering for Streams

- BDMO (authors, B. Babcock, M. Datar, R. Motwani, & L. O’Callaghan)
- Points of stream partitioned into, and summarized by, buckets with sizes equal to powers of two. Size of bucket is number of points it represents.
- Sizes of buckets obey restriction that <= two of each size. Sizes are required to form a sequence -- each size twice previous size, e.g., 3,6,12,24,...
- Bucket sizes restrained to be nondecreasing as we go back in time. As in Section 4.6, we can conclude that there will be O(log N) buckets.
- Rules for initializing, merging and splitting buckets