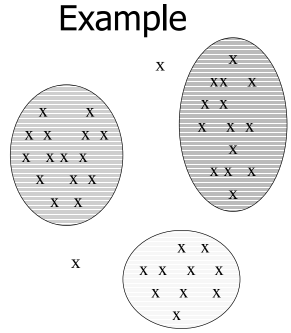
Clustering

The Problem of Clustering

- Given a set of points, with a notion of distance between points, group the points into some number of *clusters*, so that members of a cluster are in some sense as nearby as possible.

Example



Applications

- E-Business-related applications of clustering tend to involve very high-dimensional spaces.
  - The problem looks deceptively easy in a 2-dimensional, Euclidean space.

Example: Clustering CD's

- Intuitively, music divides into categories, and customers prefer one or a few categories.
  - But who's to say what the categories really are?
- Represent a CD by the customers who bought it.
- Similar CD's have similar sets of customers, and vice-versa.

The Space of CD's

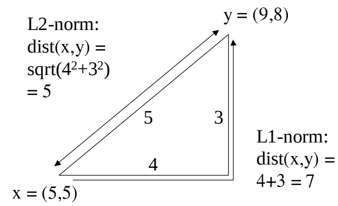
- Think of a space with one dimension for each customer.
  - Values 0 or 1 only in each dimension.
- A CD's point in this space is $(x_1, x_2, \ldots, x_k)$, where $x_i = 1$ iff the i th customer bought the CD.
  - Compare with the "correlated items" matrix: rows = customers; cols. = CD's.

Distance Measures

- Two kinds of spaces:
  - Euclidean: points have a location in space, and dist(x,y) = sqrt(sum of square of difference in each dimension).
    - Some alternatives, e.g. Manhattan distance = sum of magnitudes of differences.
  - Non-Euclidean: there is a distance measure giving dist(x,y), but no "point location."
    - Obeys triangle inequality: $d(x,y) \leq d(x,z)+d(z,y)$.
    - Also, $d(x,x) = 0$; $d(x,y) \geq 0$; $d(x,y) = d(y,x)$.

7

Examples of Euclidean Distances



8

Non-Euclidean Distances

- *Jaccard measure* for binary vectors = ratio of intersection (of components with 1) to union.
- *Cosine measure* = angle between vectors from the origin to the points in question.

9

Jaccard Measure

- Example: p_1 = 00111; p_2 = 10011.
  - Size of intersection = 2; union = 4, J.M. = 1/2.
- Need to make a distance function satisfying triangle inequality and other laws.
- $dist(p_1,p_2)$ = 1 - J.M. works.
  - dist(x,x) = 0, etc.

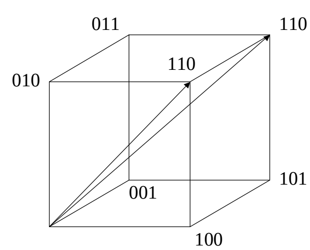
10

Cosine Measure

- Think of a point as a vector from the origin (0,0,…,0) to its location.
- Two points' vectors make an angle, whose cosine is the normalized dot-product of the vectors.
  - Example p_1 = 00111; p_2 = 10011.
  - $p_1.p_2 = 2$; $|p_1| = |p_2|$ = sqrt(3).
  - $\cos(\theta) = 2/3$.

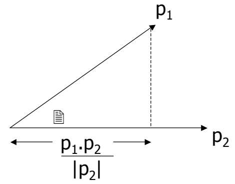
11

Example



12

Cosine-Measure Diagram



dist(p_1, p_2) = 🗎

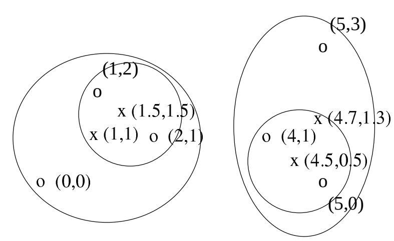
Methods of Clustering

- Hierarchical:
  - Initially, each point in cluster by itself.
  - Repeatedly combine the two "closest" clusters into one.
- Centroid-based:
  - Estimate number of clusters and their centroids.
  - Place points into closest cluster.

Hierarchical Clustering

- Key problem: as you build clusters, how do you represent the location of each cluster, to tell which pair of clusters is closest?
- Euclidean case: each cluster has a *centroid* = average of its points.
  - Measure intercluster distances by distances of centroids.

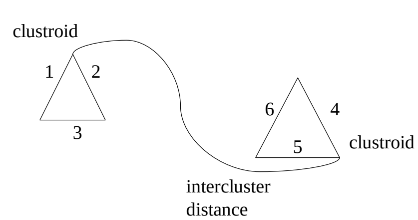
Example



And in the Non-Euclidean Case?

- The only "locations" we can talk about are the points themselves.
- Approach 1: Pick a point from a cluster to be the *clustroid* = point with minimum maximum distance to other points.
  - Treat clustroid as if it were centroid, when computing intercluster distances.

Example



Other Approaches

- Approach 2: let the intercluster distance be the minimum of the distances between any two pairs of points, one from each cluster.
- Approach 3: Pick a notion of "cohesion" of clusters, e.g., maximum distance from the clustroid.
  - Merge clusters whose combination is most cohesive.

19

k-Means

- Assumes Euclidean space.
- Starts by picking *k*, the number of clusters.
- Initialize clusters by picking one point per cluster.
  - For instance, pick one point at random, then *k* -1 other points, each as far away as possible from the previous points.

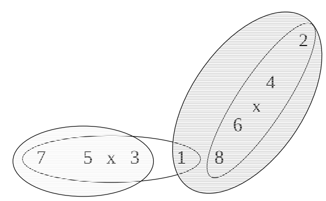
20

Populating Clusters

- For each point, place it in the cluster whose centroid it is nearest.
- After all points are assigned, fix the centroids of the *k* clusters.
- Reassign all points to their closest centroid.
  - Sometimes moves points between clusters.

21

Example



22

How Do We Deal With Big Data?

- Random-sample approaches.
  - E.g., CURE takes a sample, gets a rough outline of the clusters in main memory, then assigns points to the closest cluster.
- BFR (Bradley-Fayyad-Reina) is a *k*-means variant that compresses points near the center of clusters.
  - Also compresses groups of "outliers."

23