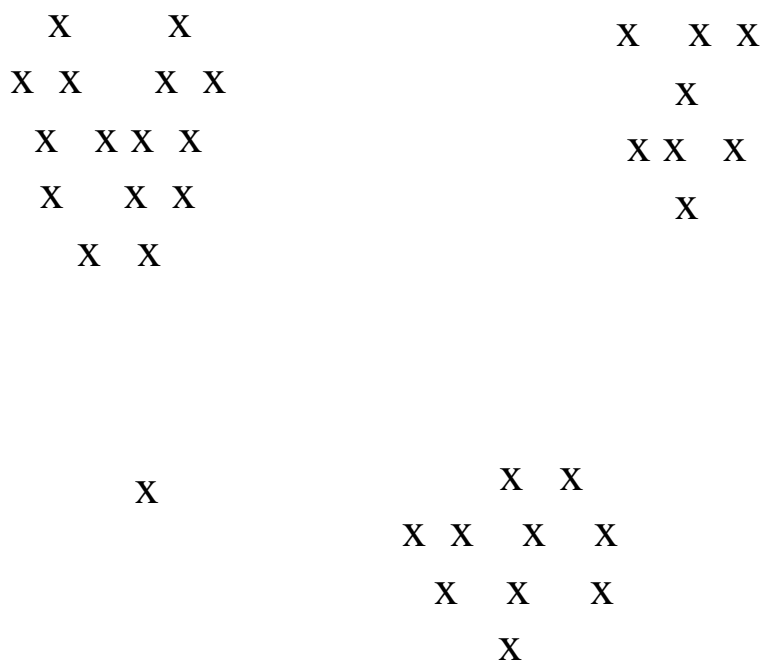
Clustering

Applications
Overview of methods

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University

The Problem

Given a cloud of data points we'd like to understand their structure

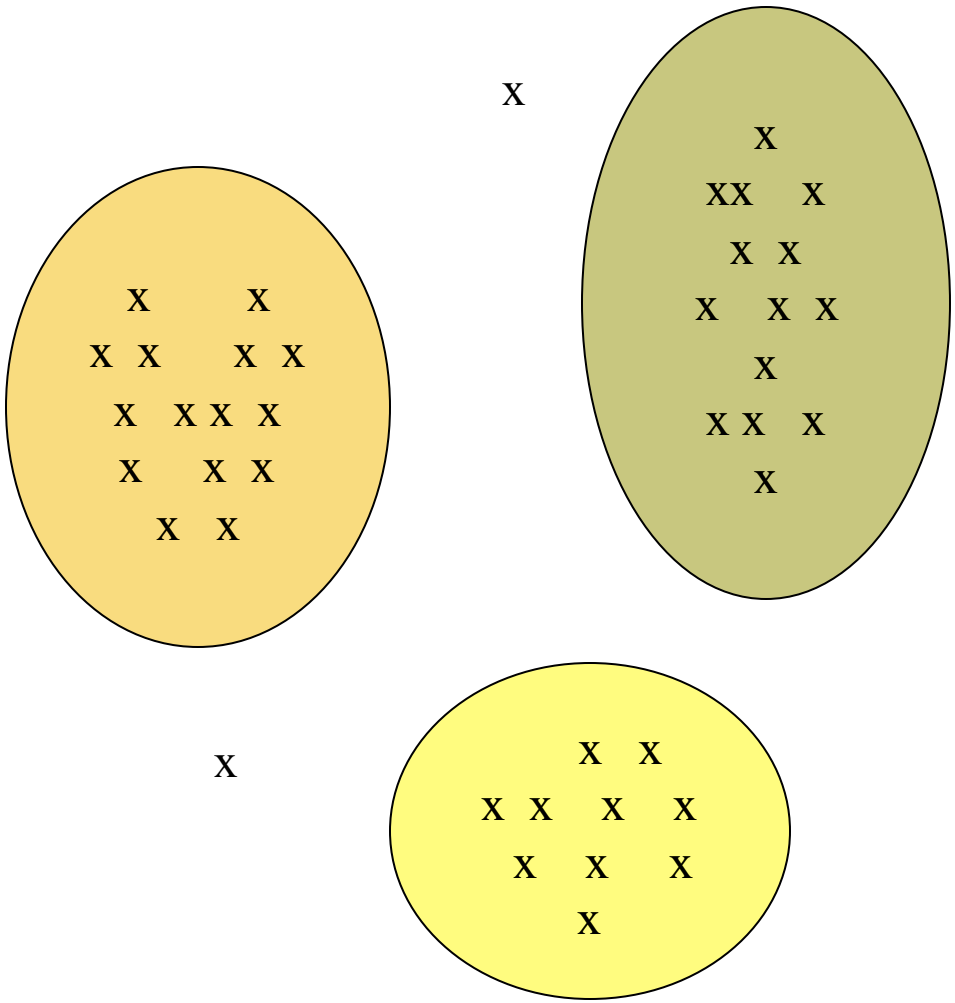


More formally:

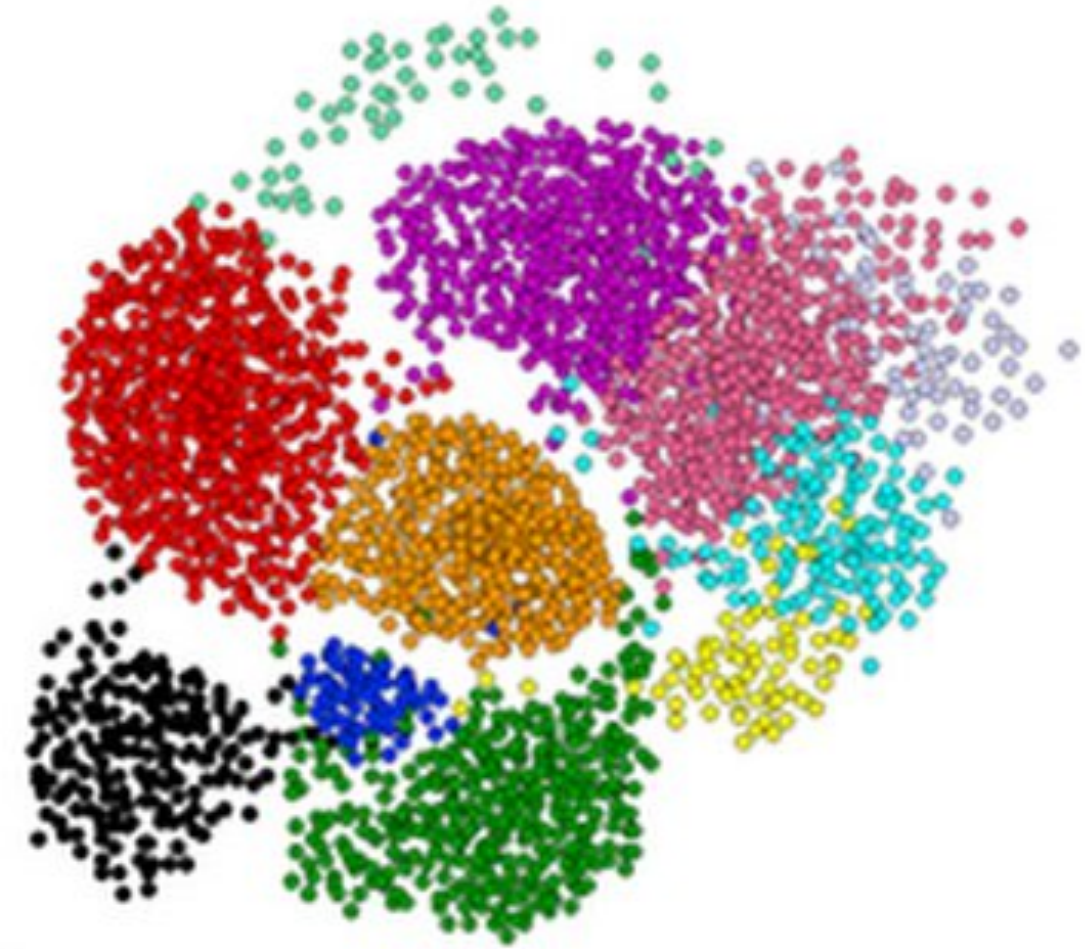
- Given a **set of points**, with a notion of **distance** between points, **group the points** into some number of ***clusters***, so that
  - Members of a cluster are close/similar to each other
  - Members of different clusters are dissimilar

- **Usually:**
  - Points are in a high-dimensional space
  - Similarity is defined using a distance measure
    - Euclidean, Cosine, Jaccard, edit distance, …

Example: Clusters



Clustering is a hard problem!



Why is it hard?

- Clustering in two dimensions looks easy
- Clustering small amounts of data looks easy
- And in most cases, looks are *not* deceiving

- Many applications involve not 2, but 10 or 10,000 dimensions
- **High-dimensional spaces look different:** Almost all pairs of points are at about the same distance

Clustering Sky Objects: SkyCat

- 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, stars, quasars, etc.
- Sloan Digital Sky Survey is a newer, better version of this

Example: Clustering CD's

- Intuitively: Music divides into categories, and customers prefer a few categories
  - But what are categories really?
- Represent a CD by a set of customers who bought it
- Similar CDs have similar sets of customers, and vice-versa

Example: Clustering Documents

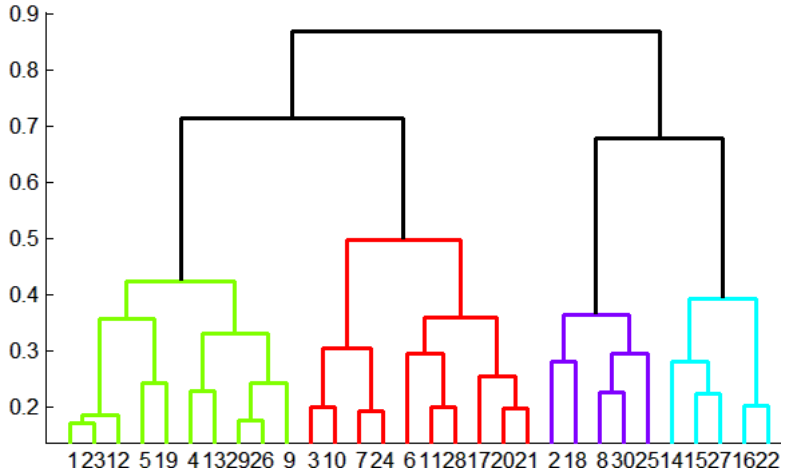
- Problem: Group together documents on the same topic
- Documents with similar sets of words may be about the same topic
- Dual formulation: a topic is a group of words that co-occur in many documents
  - Cluster words instead of documents

Cosine, Jaccard, Euclidean

- Different ways of representing documents or CDs lead to different distance measures
- Document = set of words
  - Jaccard distance
- Document = point in space of words
  - $(x_1, x_2, \ldots, x_N)$, where $x_i = 1$ iff word i appears in doc
  - Euclidean distance
- Document = vector in space of words
  - Vector from origin to $(x_1, x_2, \ldots, x_N)$
  - Cosine distance

Overview: Methods of Clustering

- **Hierarchical:**
  - **Agglomerative** (bottom up):
    - Initially, each point is a cluster
    - Repeatedly combine the two "nearest" clusters into one
  - **Divisive** (top down):
    - Start with one cluster and recursively split it



- **Point assignment:**
  - Maintain a set of clusters
  - Points belong to "nearest" cluster