Data Mining and Analysis

Clustering

CSCE 676 :: Fall 2019
Texas A&M University
Department of Computer Science & Engineering
Prof. James Caverlee

High Dimensional Data

Given a cloud of data points, we want to understand their structure

Clustering

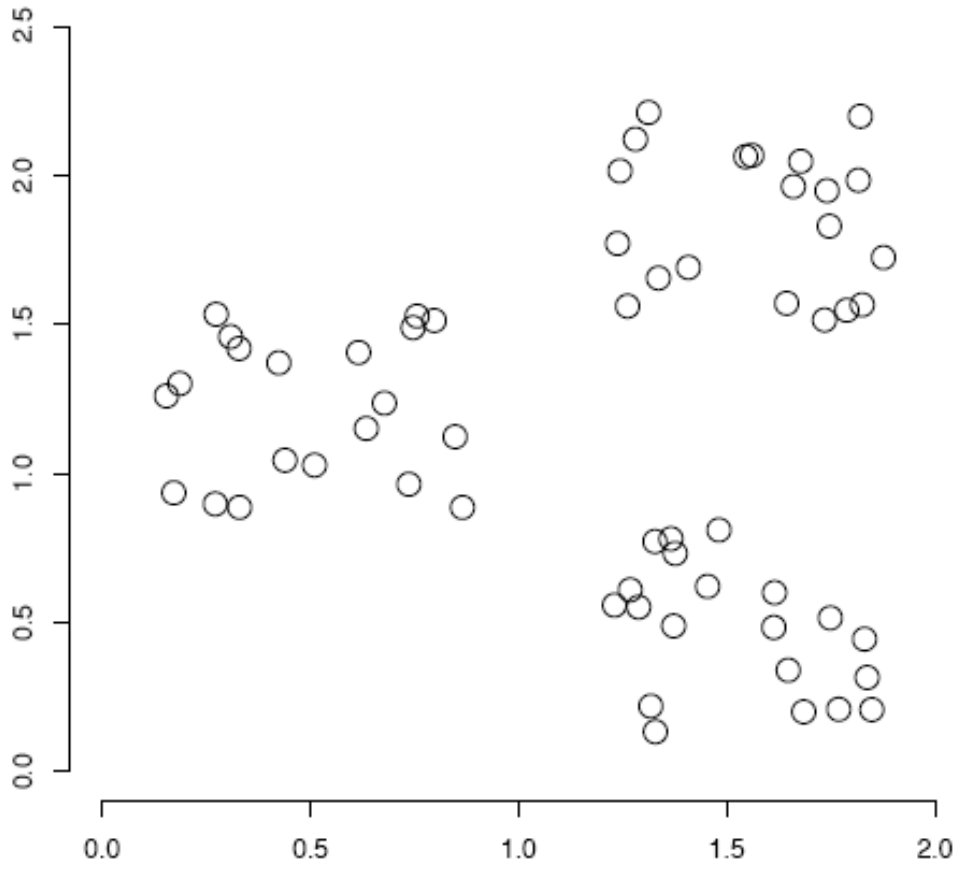
Given a set of points, with a notion of distance between points, group the points into some number of clusters, so that

- Members of a cluster are 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, ...

Data set with clear cluster structure



How would you design an algorithm for finding these three clusters?

Clustering vs. Classification

Clustering: unsupervised learning

Classification: supervised learning

Classification: Classes are human-defined and input to the learning algorithm.

Clustering: Clusters are inferred from the data without human input.

- However, there are many ways of influencing the outcome of clustering: number of clusters, similarity measure, representation of documents, . . .

Clustering: Examples

Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species

Information retrieval: document clustering

Land use: Identification of areas of similar land use in an earth observation database

Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs

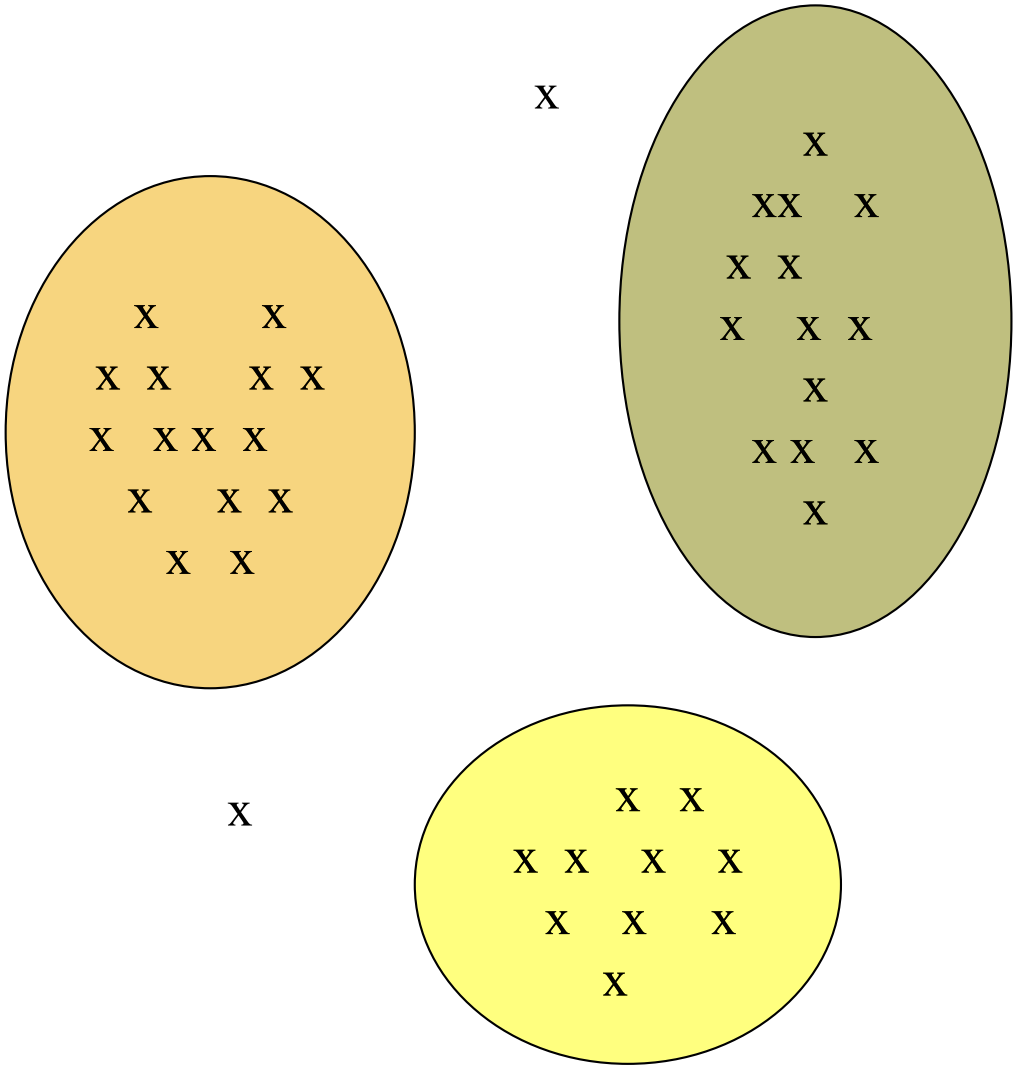
City-planning: Identifying groups of houses according to their house type, value, and geographical location

Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

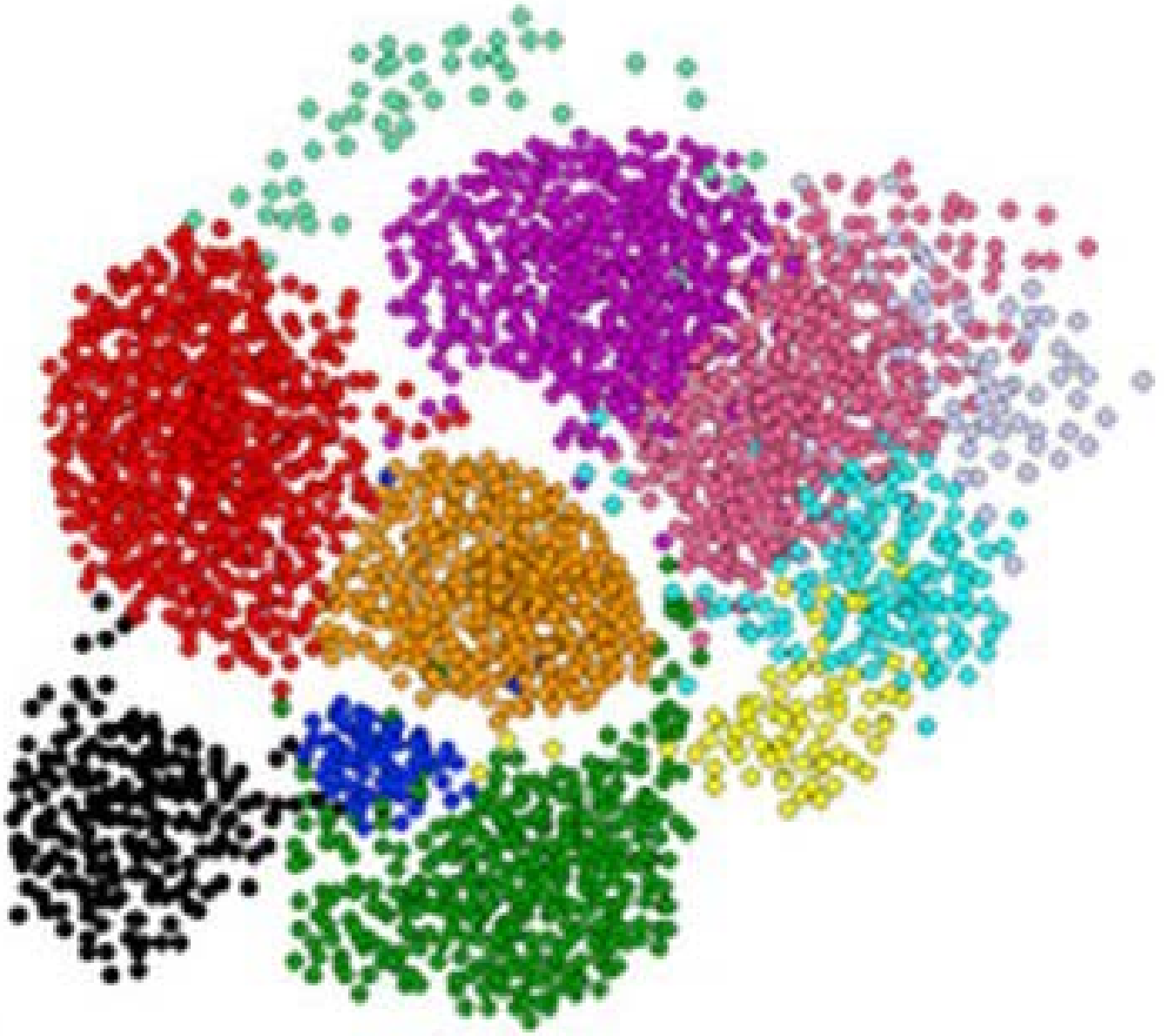
Climate: understanding earth climate, find patterns of atmospheric and ocean

Economic Science: market research

Example Clusters



Clustering is Hard!



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

Typical applications

- As a stand-alone tool to get insight into data distribution
- As a preprocessing step for other algorithms

Applications of Cluster Analysis

- Data reduction
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
  - Compression: Image processing: vector quantization
- Hypothesis generation and testing
- Prediction based on groups
  - Cluster & find characteristics/patterns for each group
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection: Outliers are often viewed as those “far away” from any cluster

Example: Clustering Songs

Intuitively: Music divides into categories, and customers prefer a few categories

But what are categories really?

Represent a song by a set of customers who liked it / listened to it

Similar songs have similar sets of likers/listeners, and vice-versa

Goal: Find clusters of similar songs

Challenge

To cluster songs:

How do we define the problem?

How do we tackle it?

Hint: Represent a song by a set of customers who liked it

Space of all songs:

- Think of a space with one dim. for each customer
- Values in a dimension may be 0 or 1 only
- A song is a point in this space is (x1, x2,..., xk), where xi = 1 iff the i th customer liked the song

Compare with boolean matrix: rows = customers; cols. = songs

For Amazon, the dimension is tens of millions

Task: Find clusters of similar songs

k-means (in one slide!)

Input is k (the number of clusters), data points in Euclidean space

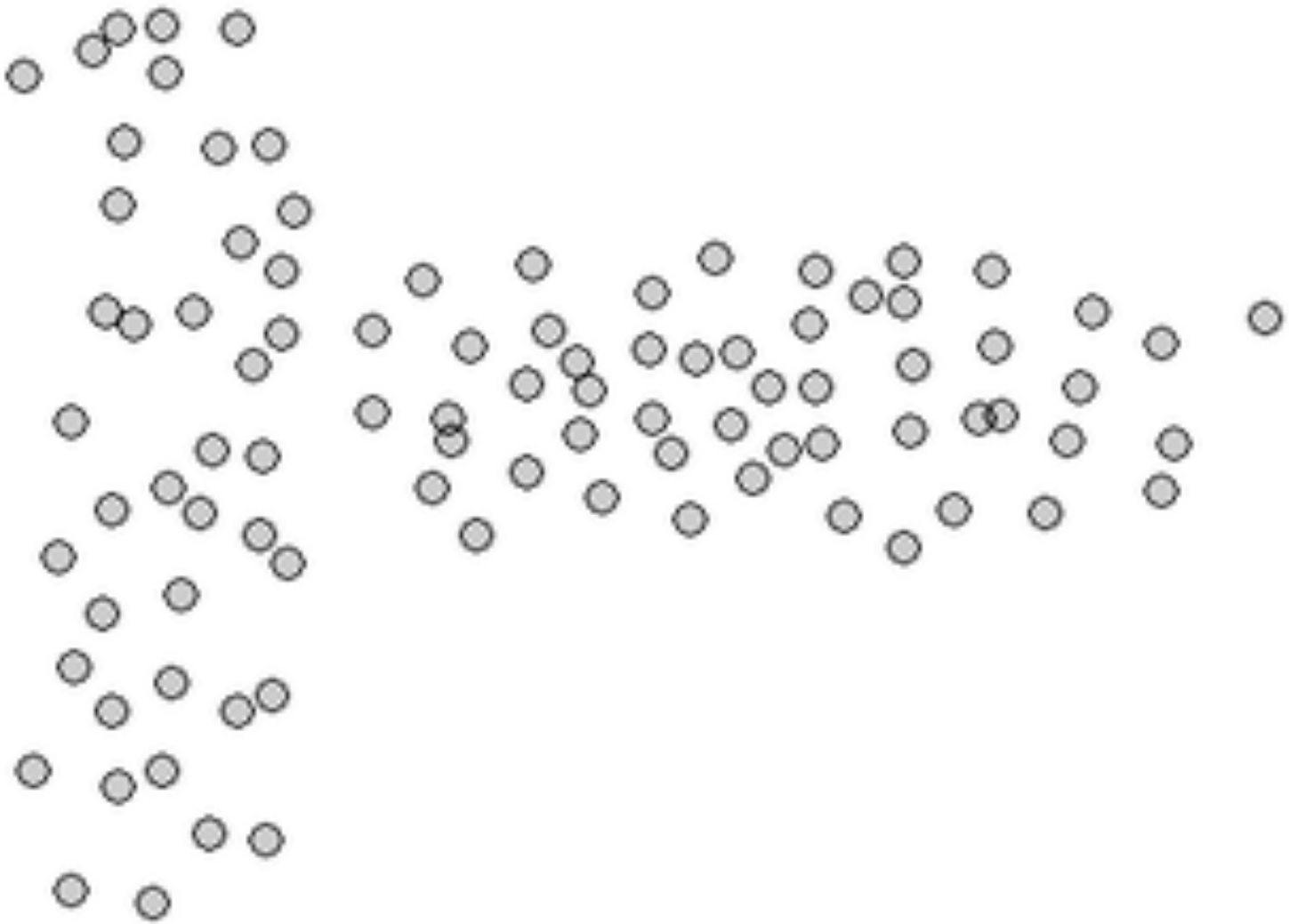
0. Initialize clusters by picking one point per cluster

Loop:

1. Place each point in the cluster whose current centroid is nearest

2. Find the new centroid for each cluster

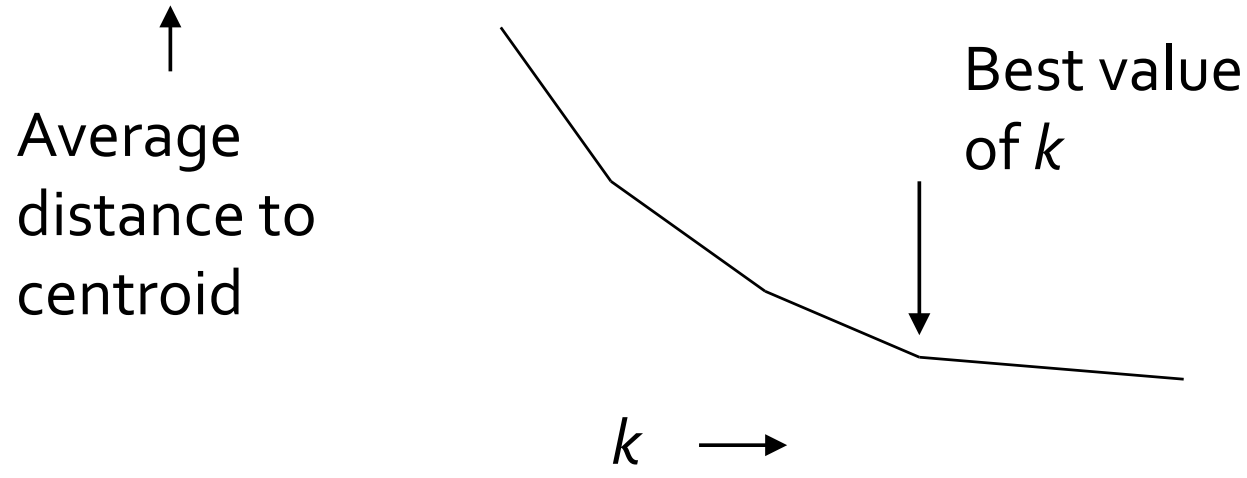
Example



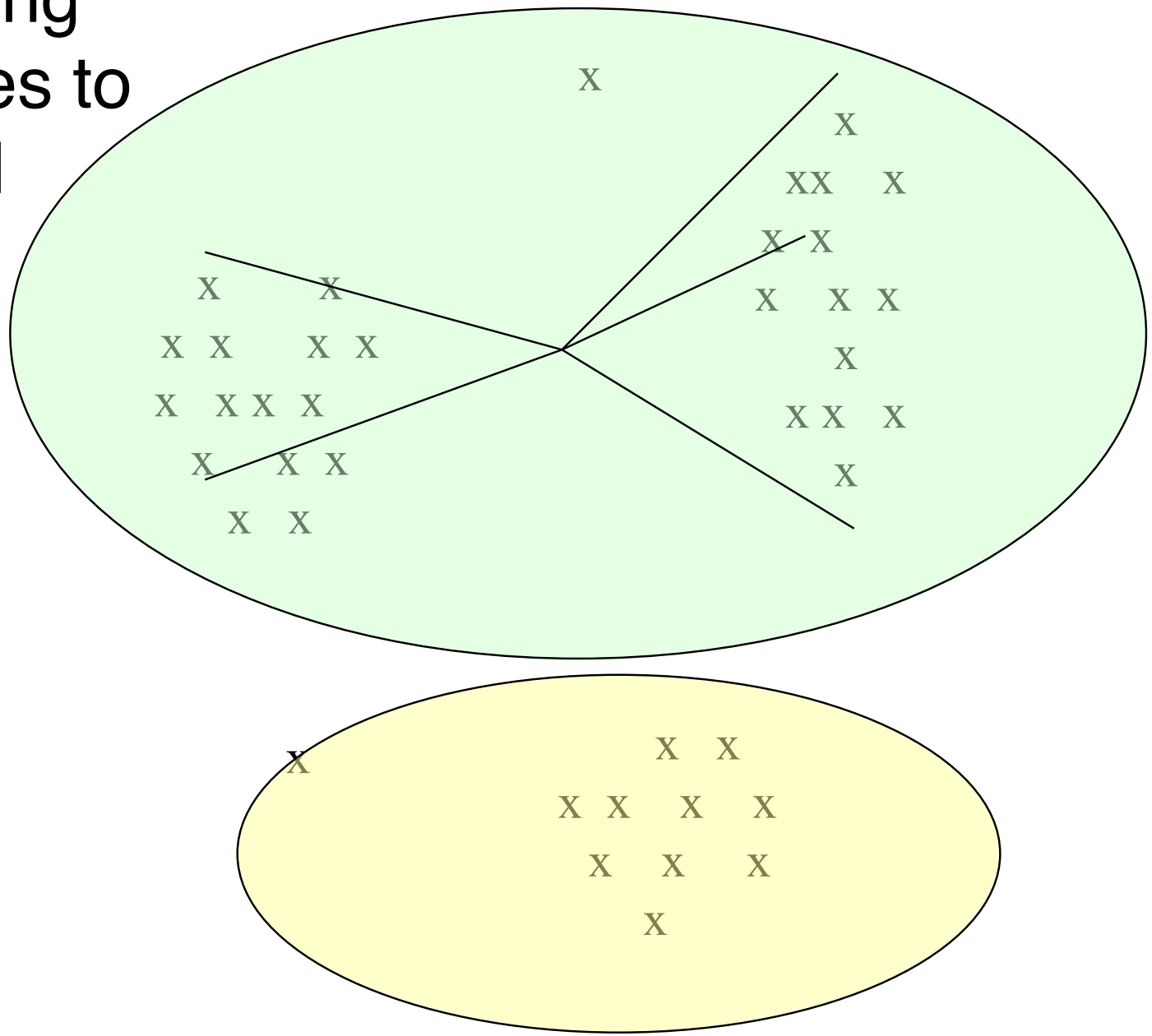
Picking k

Try different k, looking at the change in average distance to centroid as k increases

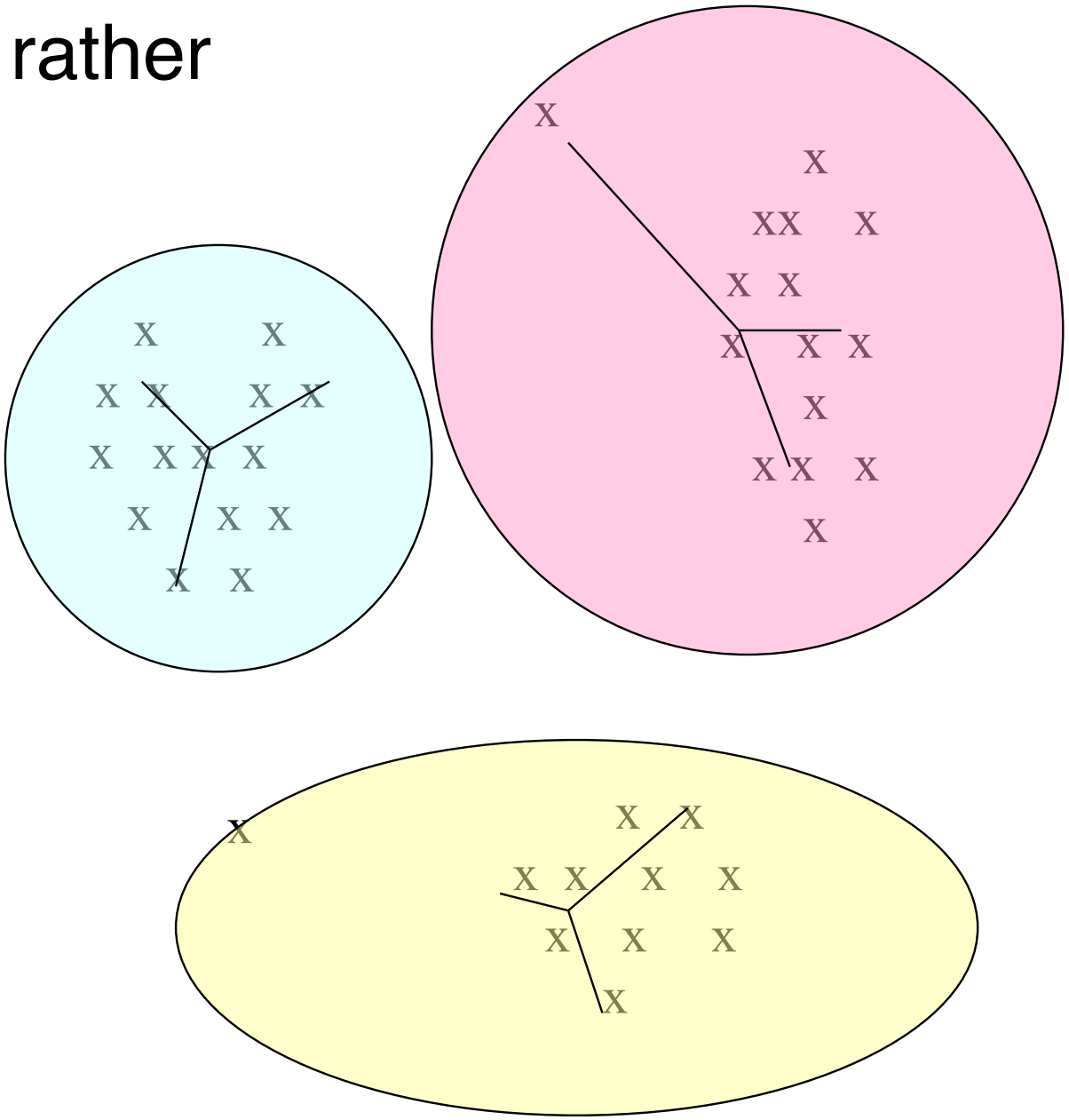
Average falls rapidly until right k, then changes little



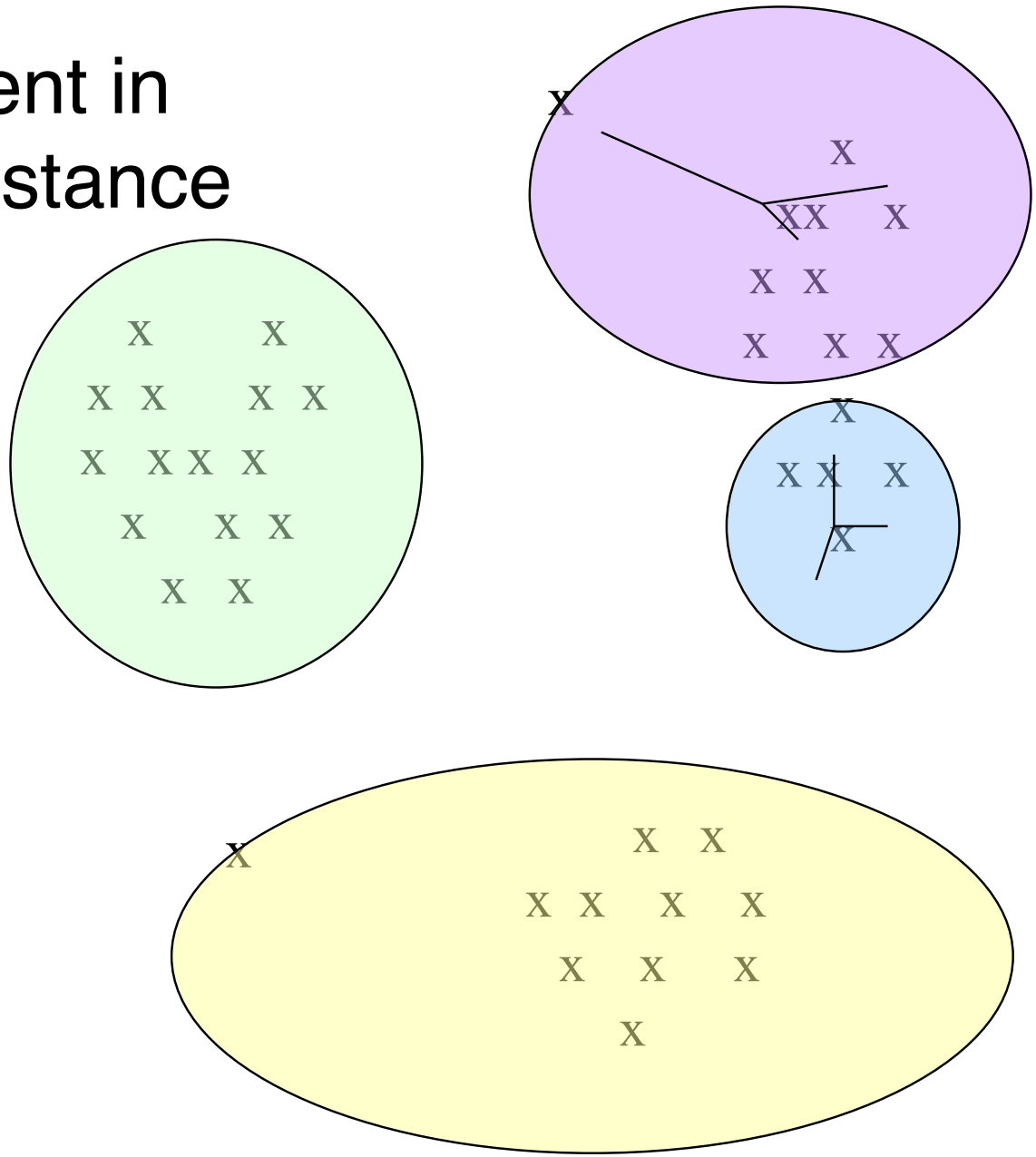
Too few clusters
Many long distances to centroid



Just right
Distances rather
short



Too many
Little
improvement in
average distance



k-medoids aka PAM

instead of finding a centroid for each cluster, use one of the points (the medoid) as the cluster “center”

PAM = partitioning around medoids

Basic Steps: Clustering

Feature selection

- Select info concerning the task of interest
- Minimal information redundancy

Proximity measure

- Similarity of two feature vectors

Clustering criterion

- Expressed via a cost function or some rules

Clustering algorithms

- Choice of algorithms

Validation of the results

- Validation test (also, clustering tendency test)

Interpretation of the results

- Integration with applications

Major Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

Major Clustering Approaches

Model-based:

- A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
- Typical methods: EM, SOM, COBWEB

Frequent pattern-based:

- Based on the analysis of frequent patterns
- Typical methods: p-Cluster

User-guided or constraint-based:

- Clustering by considering user-specified or application-specific constraints
- Typical methods: COD (obstacles), constrained clustering

Link-based clustering:

- Objects are often linked together in various ways
- Massive links can be used to cluster objects: SimRank, LinkClus

scikit-learn: lots of options