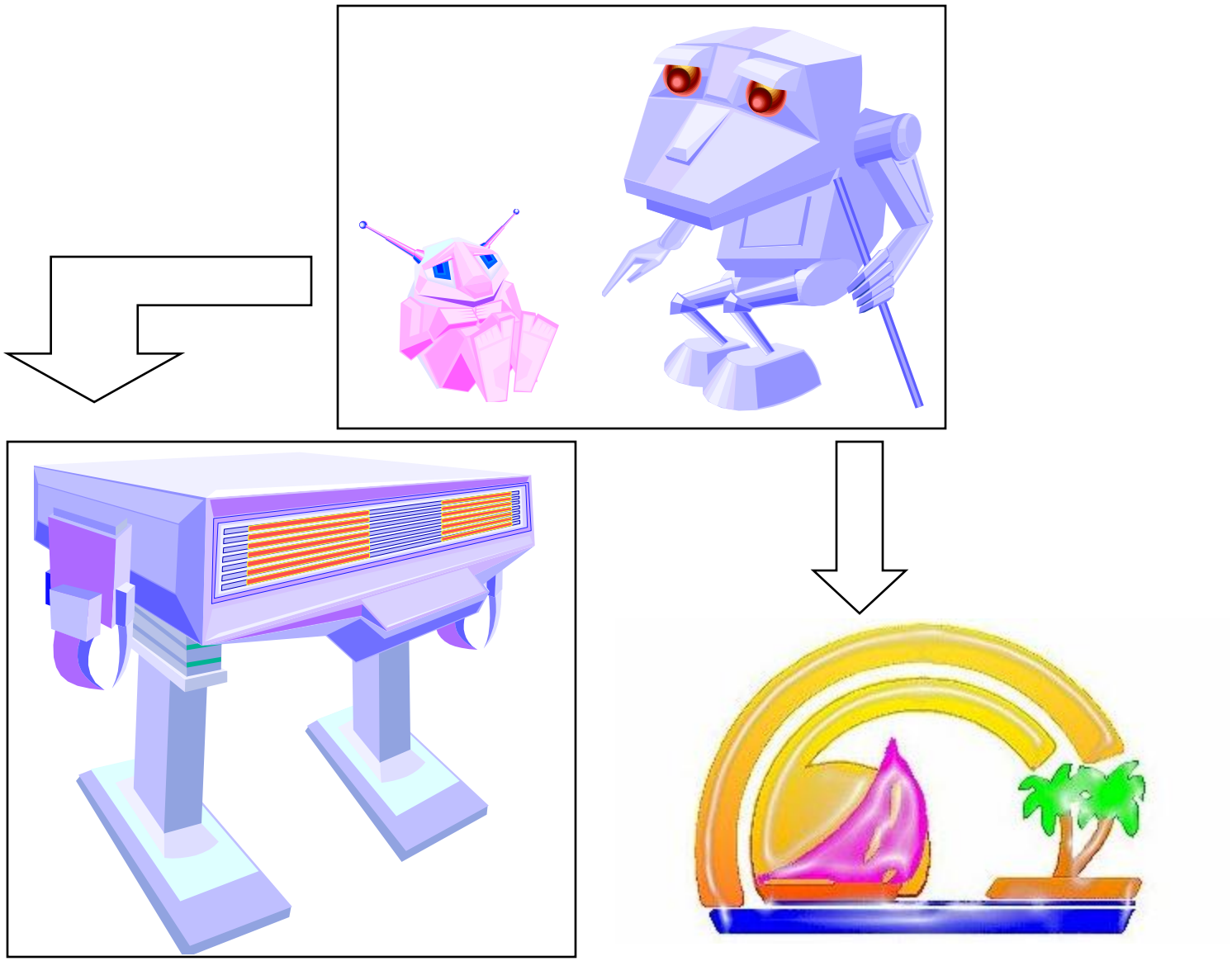
10c Machine Learning: Symbol-based

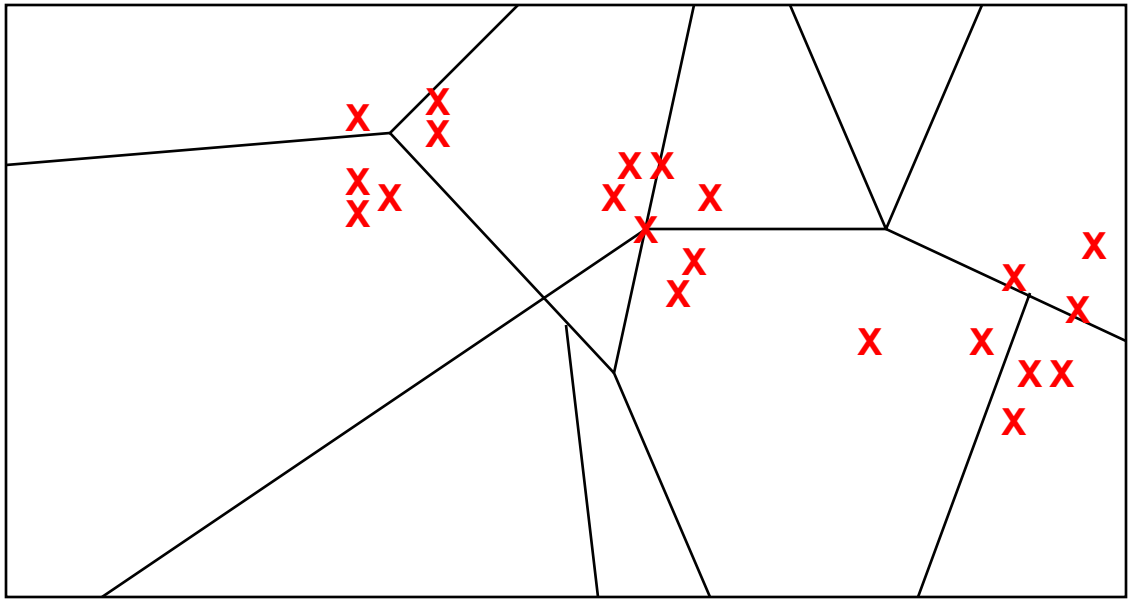
Additional references for the slides:
Jeffrey Ullman's clustering slides:
www-db.stanford.edu/~ullman/cs345-notes.html
Ernest Davis' clustering slides:
www.cs.nyu.edu/courses/fall02/G22.3033-008/index.htm

Unsupervised learning



Example: a cholera outbreak in London

Many years ago, during a cholera outbreak in London, a physician plotted the location of cases on a map. Properly visualized, the data indicated that cases clustered around certain intersections, where there were polluted wells, not only exposing the cause of cholera, but indicating what to do about the problem.



Conceptual Clustering

The clustering problem

Given

• a collection of unclassified objects, and

• a means for measuring the similarity of objects (*distance metric*),

find

• classes (clusters) of objects such that some standard of quality is met (e.g., maximize the similarity of objects in the same class.)

Essentially, it is an approach to *discover* a useful summary of the data.

Conceptual Clustering (cont'd)

Ideally, we would like to represent clusters and their semantic explanations. In other words, we would like to define clusters *extensionally* (i.e., by general rules) rather than *intensionally* (i.e., by enumeration).

For instance, compare

{ X | X teaches AI at MTU CS}, and

{ John Lowther, Nilufer Onder}

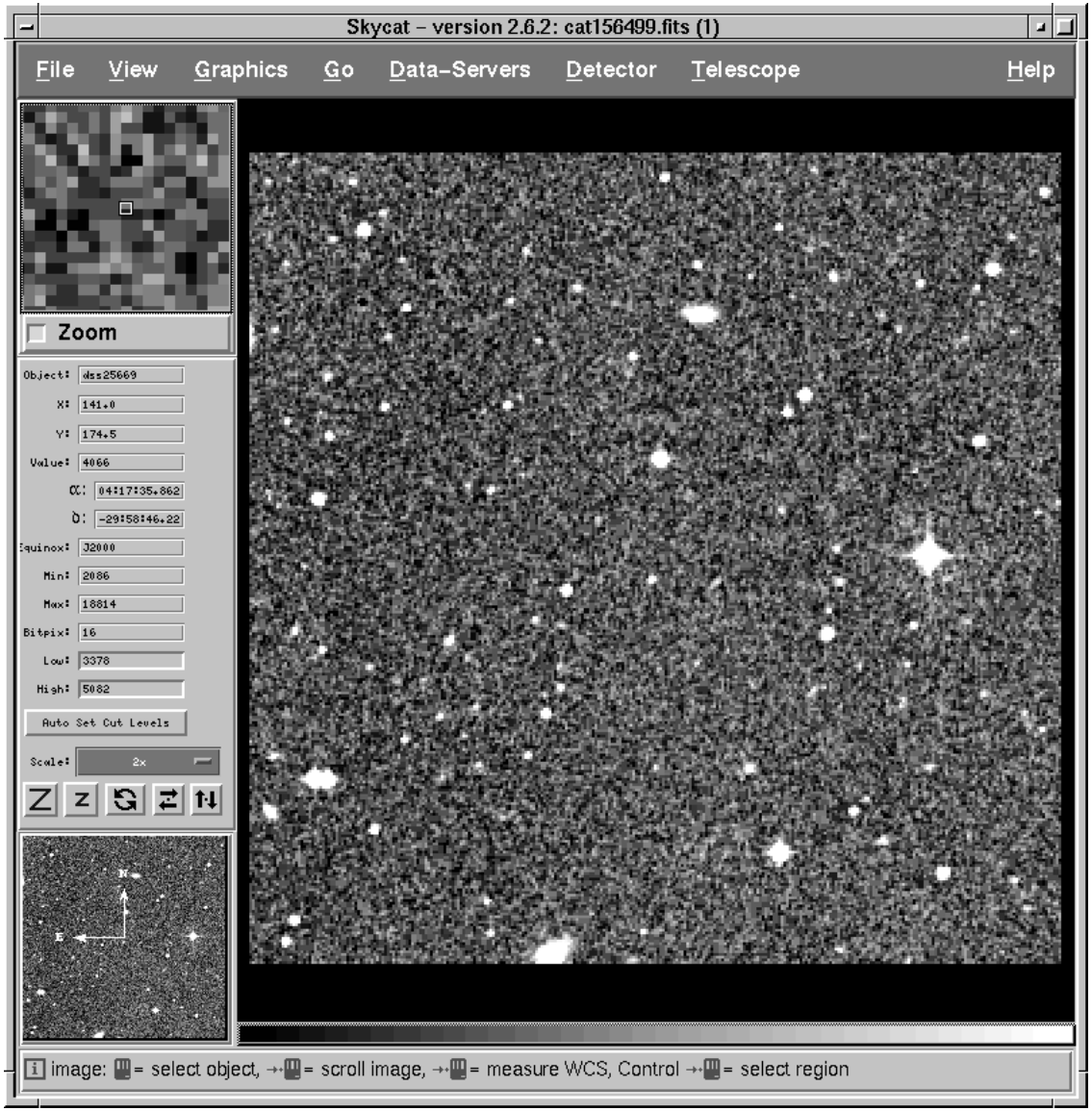
Curse of dimensionality

- **While clustering looks intuitive in 2 dimensions, many applications involve 10 or 10,000 dimensions**
- **High-dimensional spaces look different: the probability of random points being close drops quickly as the dimensionality grows**

Higher dimensional examples

- **Observation that customers who buy diapers are more likely to buy beer than average allowed supermarkets to place beer and diapers nearby, knowing many customers would walk between them. Placing potato chips between increased the sales of all three items.**

Skycat software



Skycat software (cont'd)

- **Skycat is a catalog of sky objects**
- **Objects are represented by their radiation in 9 dimensions (each dimension represents radiation in one band of the spectrum**
- **Skycat clustered 2 x 10^9 sky objects into similar objects e.g., stars, galaxies, quasars, etc.**
- **The Sloan Sky Survey is a newer, better version to catalog and cluster the entire visible universe. Clustering sky objects by their radiation levels in different bands allowed astronomers to distinguish between galaxies, nearby stars, and many other kinds of celestial objects.**

Clustering CDs

- **Intuition: music divides into categories and customers prefer a few categories**
- **But what are categories really?**
- **Represent a CD by the customers who bought it**
- **Similar CDs have similar sets of customers and vice versa**

The space of CDs

- Think of a space with one dimension for each customer
- Values in a dimension may be 0 or 1 only
- A CD's point in this space is $(x_1, x_2, \ldots, x_n)$, where $x_i = 1$ iff the i^{th} customer bought the CD
- Compare this with the correlated items matrix:

rows = customers
columns = CDs

Clustering documents

- Query "salsa" submitted to MetaCrawler returns 246 documents in 15 clusters, of which the top are:
  - Puerto Rico; Latin Music (8 docs)
  - Follow Up Post; York Salsa Dancers (20 docs)
  - music; entertainment; latin; artists (40 docs)
  - hot; food; chiles; sauces; condiments; companies (79 docs)
  - pepper; onion; tomatoes (41 docs)
- The clusters are: dance, recipe, clubs, sauces, buy, mexican, bands, natural, …

Clustering documents (cont'd)

- **Documents may be thought of as points in a high-dimensional space, where each dimension corresponds to one possible word.**
- **Clusters of documents in this space often correspond to groups of documents on the same topic, i.e., documents with similar sets of words may be about the same topic**
- **Represent a document by a vector $(x_1, x_2, \ldots, x_n)$, where $x_i = 1$ iff the i^{th} word (in some order) appears in the document**
- **n can be infinite**

Analyzing protein sequences

- Objects are sequences of {C, A, T, G}
- Distance between sequences is “edit distance,” the minimum number of inserts and deletes to turn one into the other
- Note that there is a “distance,” but no convenient space of points

Measuring distance

- To discuss, whether a set of points is close enough to be considered a cluster, we need a *distance measure* D(x,y) that tells how far points x and y are.
- The axioms for a distance measure D are:

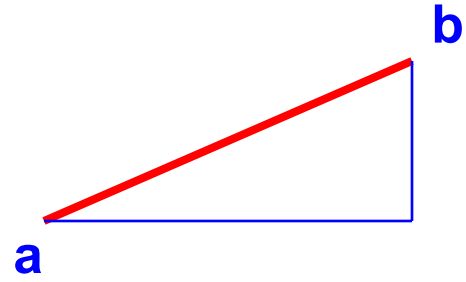
1. $D(x,x) = 0$	A point is distance 0 from itself
2. $D(x,y) = D(y,x)$	Distance is symmetric
3. $D(x,y) \leq D(x,z) + D(z,y)$	The triangle inequality
4. $D(x,y) \geq 0$	Distance is positive

K-dimensional Euclidean space

The distance between any two points, say $a = [a_1, a_2, \ldots, a_k]$ and $b = [b_1, b_2, \ldots, b_k]$ is given some manner such as:

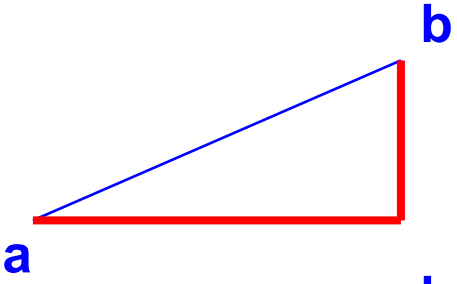
1. Common distance ("L_2 norm") :

$$\sqrt{\Sigma_{i=1}^{k} (a_i - b_i)^2}$$



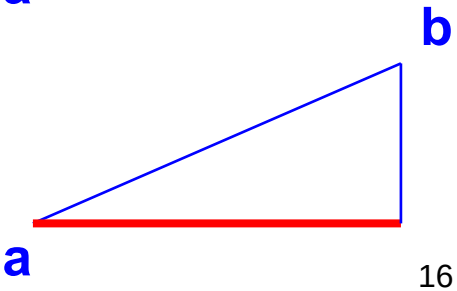
2. Manhattan distance ("L_1 norm"):

$$\Sigma_{i=1}^{k} |a_i - b_i|$$



3. Max of dimensions ("L∞ norm"):

$$\max_{i=1}^{k} |a_i - b_i|$$



Non-Euclidean spaces

Here are some examples where a distance measure without a Euclidean space makes sense.

- **Web pages: Roughly 10^8-dimensional space where each dimension corresponds to one word. Rather use vectors to deal with only the words actually present in documents a and b.**

- **Character strings, such as DNA sequences: Rather use a metric based on the LCS---Lowest Common Subsequence.**

- **Objects represented as sets of symbolic, rather than numeric, features: Rather base similarity on the proportion of features that they have in common.**

Non-Euclidean spaces (cont'd)

object1 = {small, red, rubber, ball}

object2 = {small, blue, rubber, ball}

object3 = {large, black, wooden, ball}

similarity(object1, object2) = 3 / 4

similarity(object1, object3) =
similarity(object2, object3) = 1/4

Note that it is possible to assign different weights to features.

Approaches to Clustering

Broadly specified, there are two classes of clustering algorithms:

1. *Centroid approaches*: We guess the *centroid* (central point) in each cluster, and assign points to the cluster of their nearest centroid.

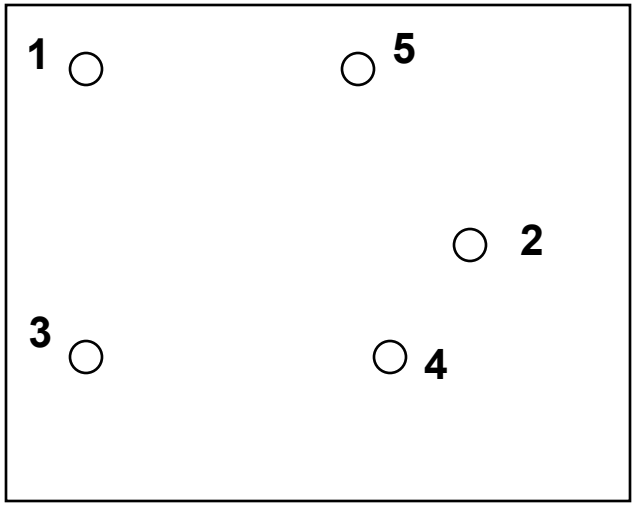
2. *Hierarchical approaches*: We begin assuming that each point is a cluster by itself. We repeatedly merge nearby clusters, using some measure of how close two clusters are (e.g., distance between their centroids), or how good a cluster the resulting group would be (e.g., the average distance of points in the cluster from the resulting centroid.)

The *k*-means algorithm

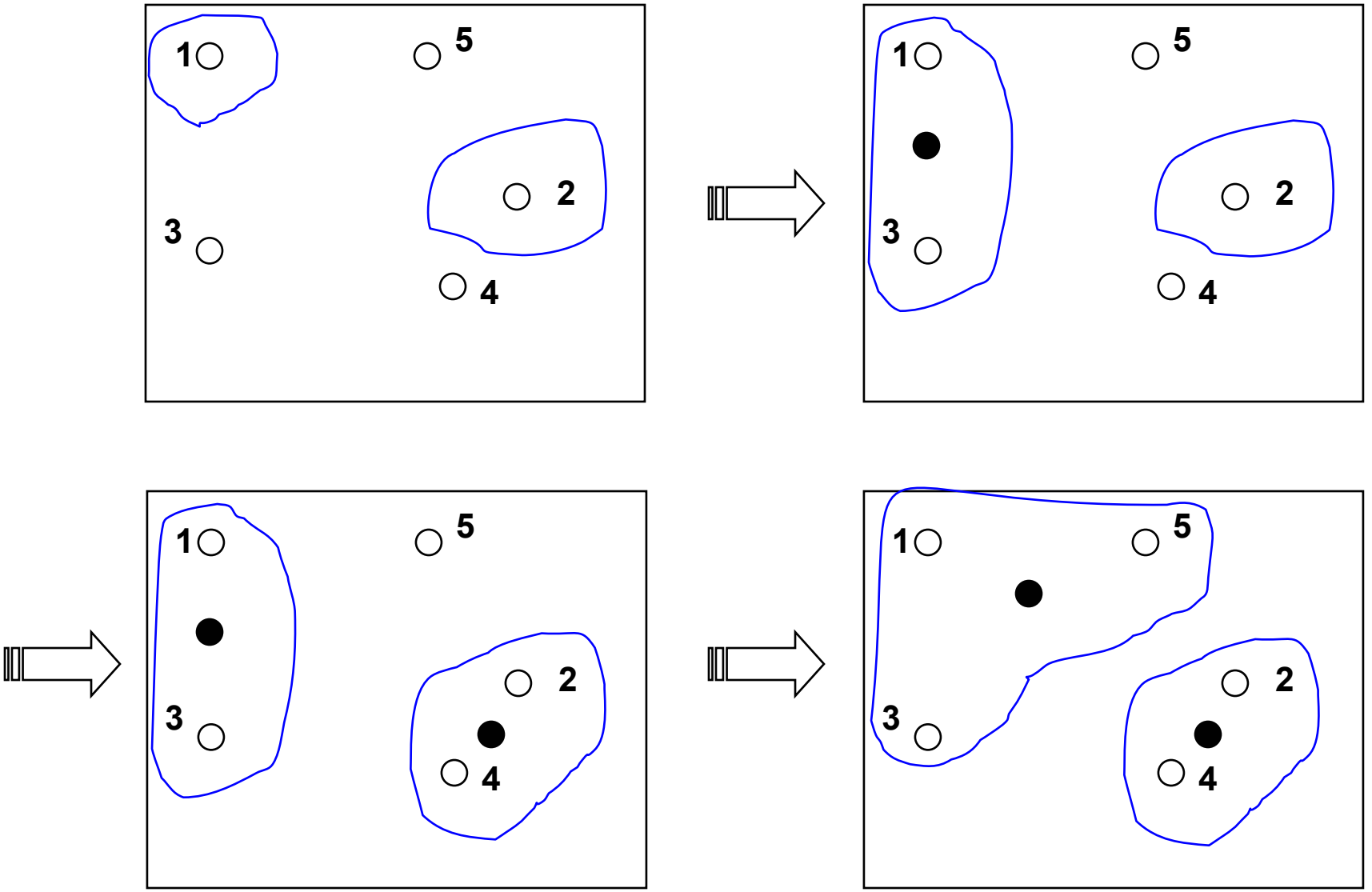
- Pick *k* cluster centroids.

- Assign points to clusters by picking the closest centroid to the point in question. As points are assigned to clusters, the centroid of the cluster may migrate.

Example: Suppose that *k* = 2 and we assign points 1, 2, 3, 4, 5, in that order. Outline circles represent points, filled circles represent centroids.



The *k*-means algorithm example (cont'd)



Issues

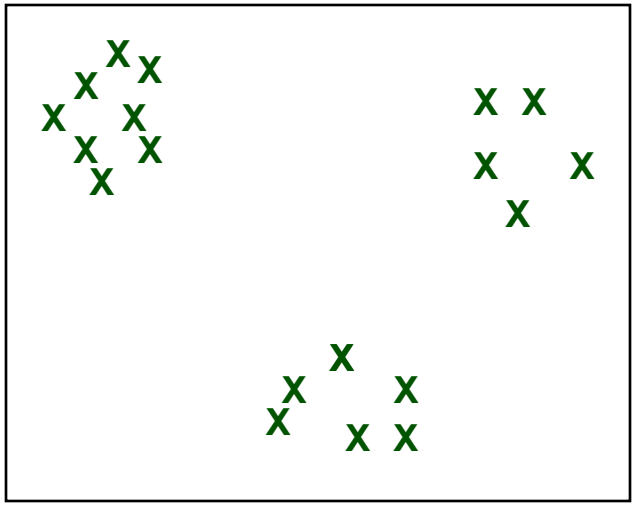
- How to initialize the *k* centroids?
  Pick points sufficiently far away from any other centroid, until there is *k*.

- As computation progresses, one can decide to split one cluster and merge two, to keep the total at *k*. A test for whether to do so might be to ask whether doing so reduces the average distance from points to their centroids.

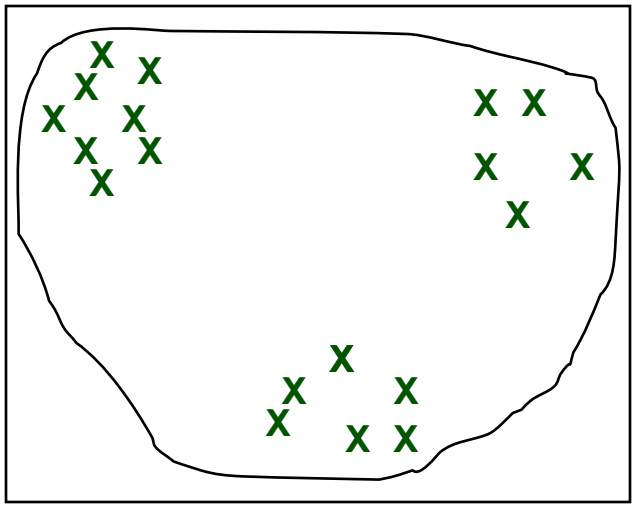
- Having located the centroids of *k* clusters, we can reassign all points, since some points that were assigned early may actually wind up closer to another centroid, as the centroids move about.

Issues (cont'd)

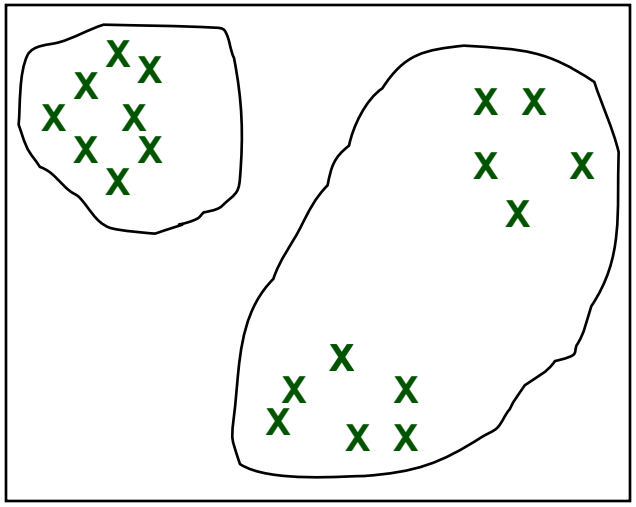
- **How to determine *k*?**
  One can try different values for *k* until the smallest *k* such that increasing *k* does not much decrease the average points of points to their centroids.



Determining k

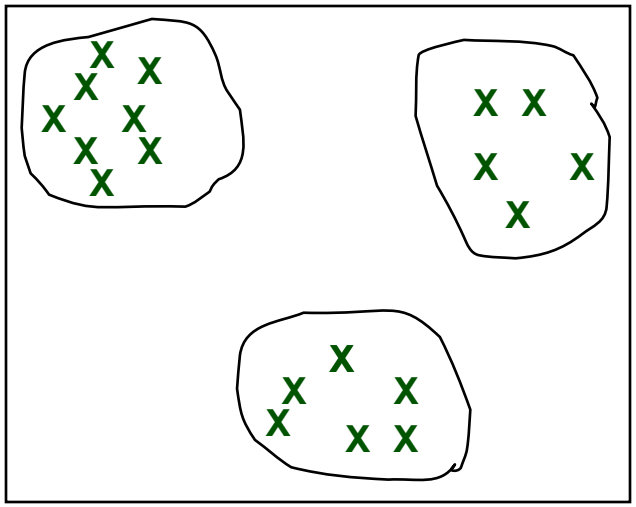


When *k* = 1, all the points are in one cluster, and the average distance to the centroid will be high.

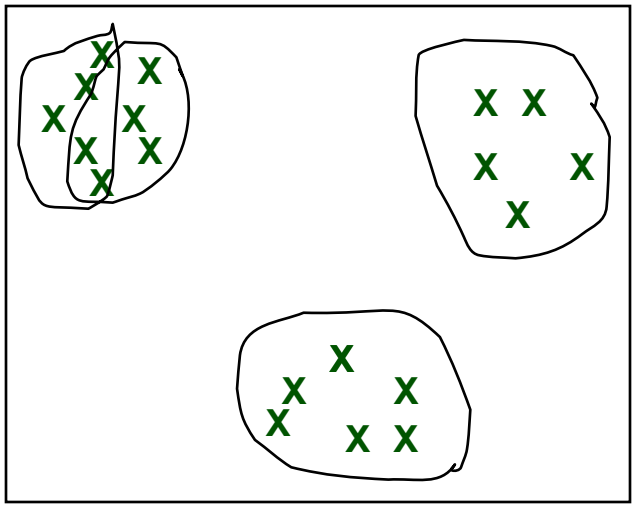


When *k* = 2, one of the clusters will be by itself and the other two will be forced into one cluster. The average distance of points to the centroid will shrink considerably.

Determining k (cont'd)

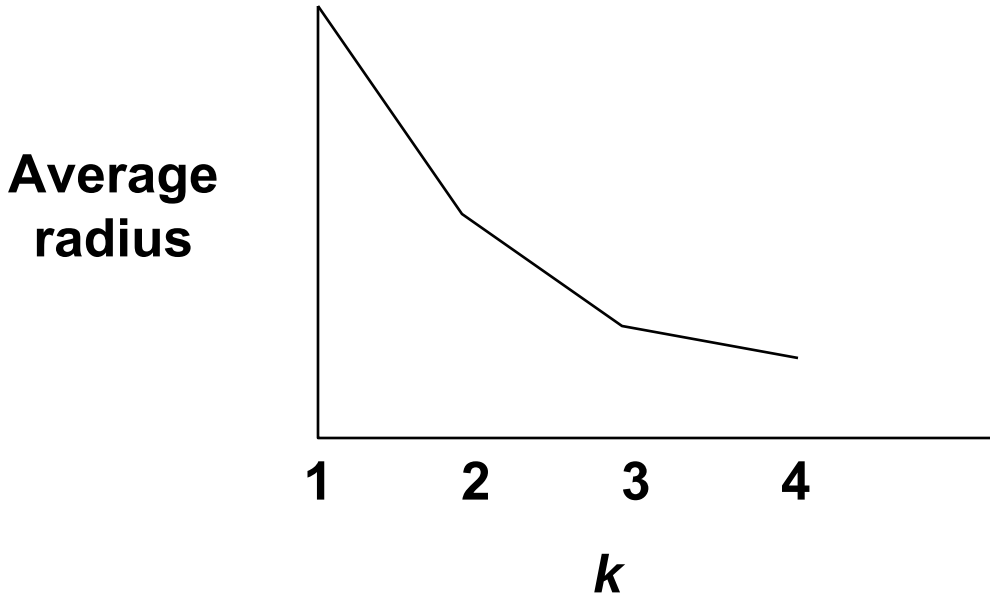


When *k* = 3, each of the apparent clusters should be a cluster by itself, and the average distance from the points to their centroids shrinks again.



When *k* = 4, then one of the true clusters will be artificially partitioned into two nearby clusters. The average distance to centroid will drop a bit, but not much.

Determining k (cont'd)



This failure to drop further suggests that k = 3 is right. This conclusion can be made even if the data is in so many dimensions that we cannot visualize the clusters.

The CLUSTER/2 algorithm

1. Select k seeds from the set of observed objects. This may be done randomly or according to some selection function.

2. For each seed, using that seed as a positive instance and all other seeds as negative instances, produce a maximally general definition that covers all of the positive and none of the negative instances (multiple classifications of non-seed objects are possible.)

The CLUSTER/2 algorithm (cont'd)

3. Classify all objects in the sample according to these descriptions. Replace each maximally specific description that covers all objects in the category (to decrease the likelihood that classes overlap on unseen objects.)

4. Adjust remaining overlapping definitions.

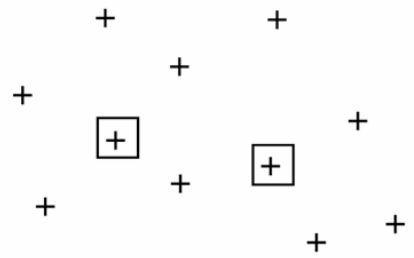
5. Using a distance metric, select an element closest to the center of each class.

6. Repeat steps 1-5 using the new central elements as seeds. Stop when clusters are satisfactory.

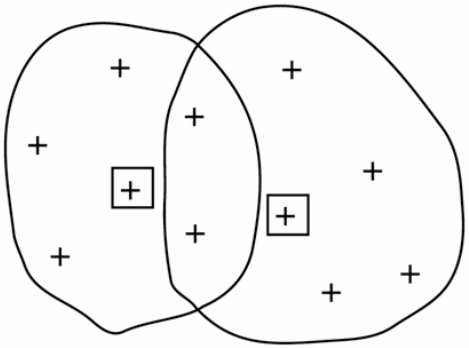
The CLUSTER/2 algorithm (cont'd)

7. If clusters are unsatisfactory and no improvement occurs over several iterations, select the new seeds closest to the edge of the cluster.

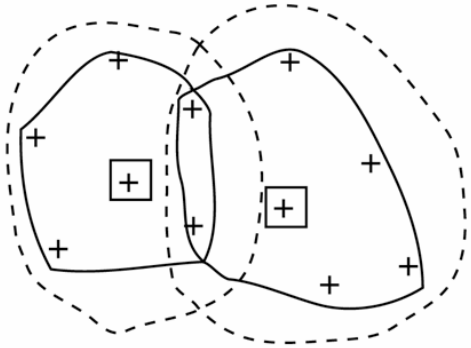
The steps of a CLUSTER/2 run



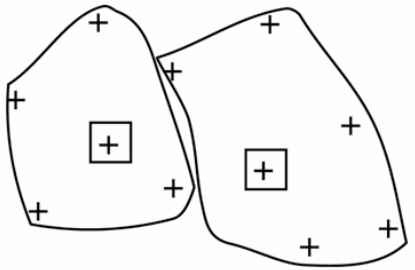
After selecting seeds (step 1).



After generating general descriptions (steps 2 and 3). Note that the categories overlap.



After specializing concept descriptions (step 4). There are still intersecting elements.



After eliminating duplicate elements (step 5).

A COBWEB clustering for four one-celled organisms (Gennari et al.,1989)

Note: we will skip the COBWEB algorithm

Category	C1	P(C1)=4/4
Feature	Value	p(vlc)
Tails	One	0.50
	Two	0.50
Color	Light	0.50
	Dark	0.50
Nuclei	One	0.25
	Two	0.50
	Three	0.25

Category	C2	P(C2)=1/4
Feature	Value	p(vlc)
Tails	One	1.0
	Two	0.0
Color	Light	1.0
	Dark	0.0
Nuclei	One	1.0
	Two	0.0
	Three	0.0

Category	C3	P(C3)=2/4
Feature	Value	p(vlc)
Tails	One	0.0
	Two	1.0
Color	Light	0.50
	Dark	0.50
Nuclei	One	0.0
	Two	1.0
	Three	0.0

Category	C4	P(C4)=1/4
Feature	Value	p(vlc)
Tails	One	1.0
	Two	0.0
Color	Light	0.0
	Dark	1.0
Nuclei	One	0.0
	Two	0.0
	Three	1.0

Category	C5	P(C5)=1/4
Feature	Value	p(vlc)
Tails	One	0.0
	Two	1.0
Color	Light	1.0
	Dark	0.0
Nuclei	One	0.0
	Two	1.0
	Three	0.0

Category	C6	P(C6)=1/4
Feature	Value	p(vlc)
Tails	One	0.0
	Two	1.0
Color	Light	0.0
	Dark	1.0
Nuclei	One	0.0
	Two	1.0
	Three	0.0

Related communities

- data mining (in databases, over the web)
- statistics
- clustering algorithms
- visualization
- databases

Clustering vs. classification

- ***Clustering*** **is when the clusters are not known**
- **If the system of clusters is known, and the problem is to place a new item into the proper cluster, this is *classification***

Cluster structure

- Hierarchical vs flat
- Overlap
  - Disjoint partitioning, e.g., partition congressmen by state
  - Multiple dimensions of partitioning, each disjoint, e.g., partition congressmen by state; by party; by House/Senate
  - Arbitrary overlap, e.g., partition bills by congressmen who voted for them
- Exhaustive vs. non-exhaustive
- Outliers: what to do?
- How many clusters? How large?

More on document clustering

- **Applications**
  - **Structuring search results**
  - **Suggesting related pages**
  - **Automatic directory construction / update**
  - **Finding near identical pages**
    - Finding mirror pages (e.g., for propagating updates)
    - Eliminate near-duplicates from results page
    - Plagiarism detection
    - Lost and found (find identical pages at different URLs at different times)
- **Problems**
  - **Polysemy, e.g., “bat,” “Washington,” “Banks”**
  - **Multiple aspects of a single topic**
  - **Ultimately amounts to general problem of information structuring**