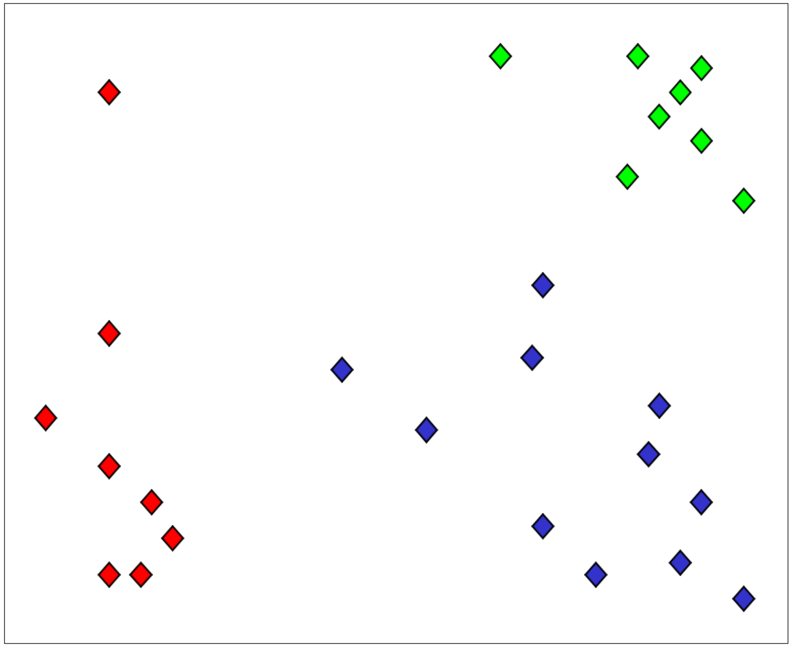
10701
Machine Learning

Clustering

What is Clustering?

- Organizing data into *clusters* such that there is
  - high intra-cluster similarity
  - low inter-cluster similarity
- Informally, finding natural groupings among objects.



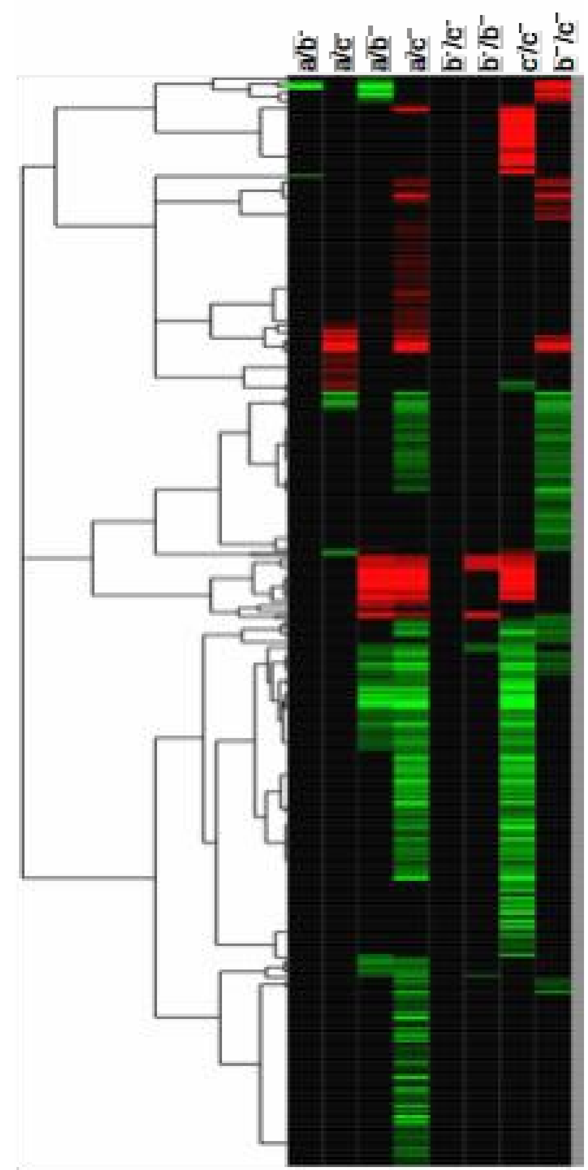
Example: clusty

Example: clustering genes

- Microarrays measures the activities of all genes in different conditions

- Clustering genes can help determine new functions for unknown genes

- An early “killer application” in this area



Why clustering?

- Organizing data into clusters provides information about the internal structure of the data
  - Ex. Clusty and clustering genes above
- Sometimes the partitioning is the goal
  - Ex. Image segmentation
- Knowledge discovery in data
  - Ex. Underlying rules, reoccurring patterns, topics, etc.

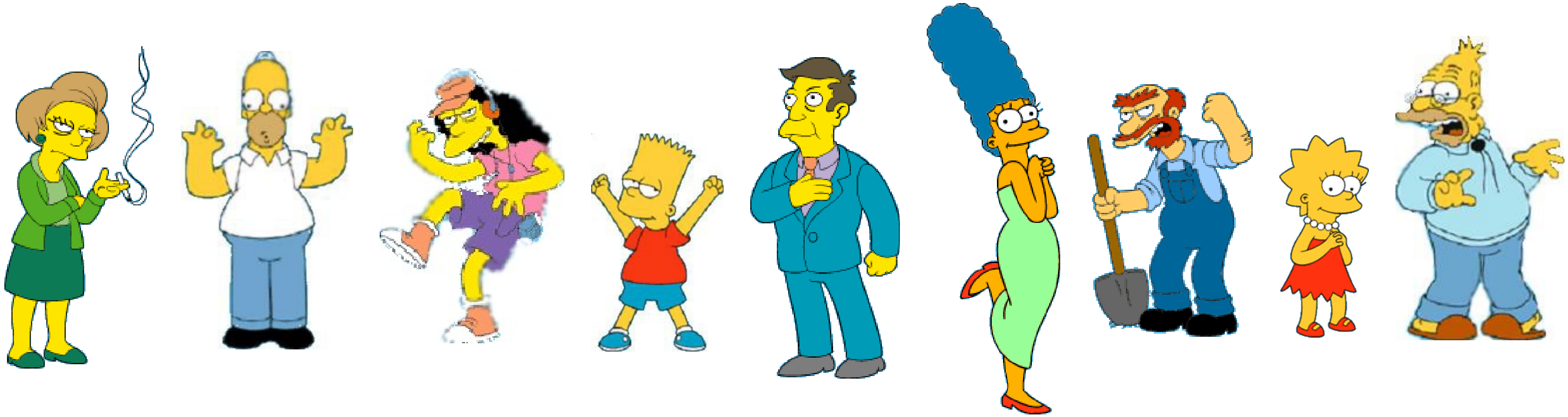
Unsupervised learning algorithms

- Clustering methods are unsupervised learning techniques

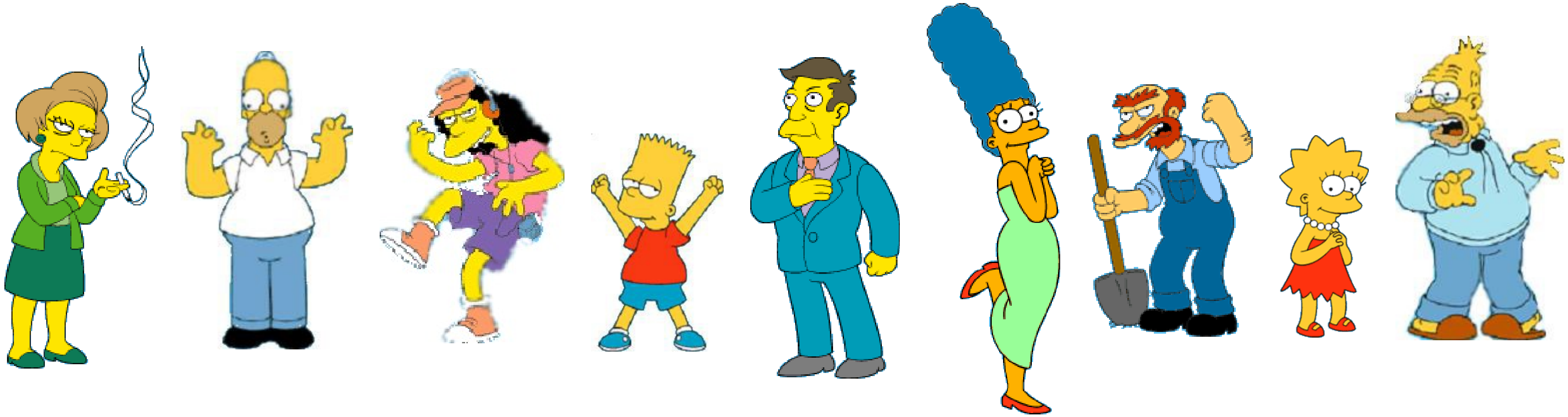
  - We do not have a teacher that provides examples with their labels

- We will also discuss dimensionality reduction, another unsupervised learning method later in the course

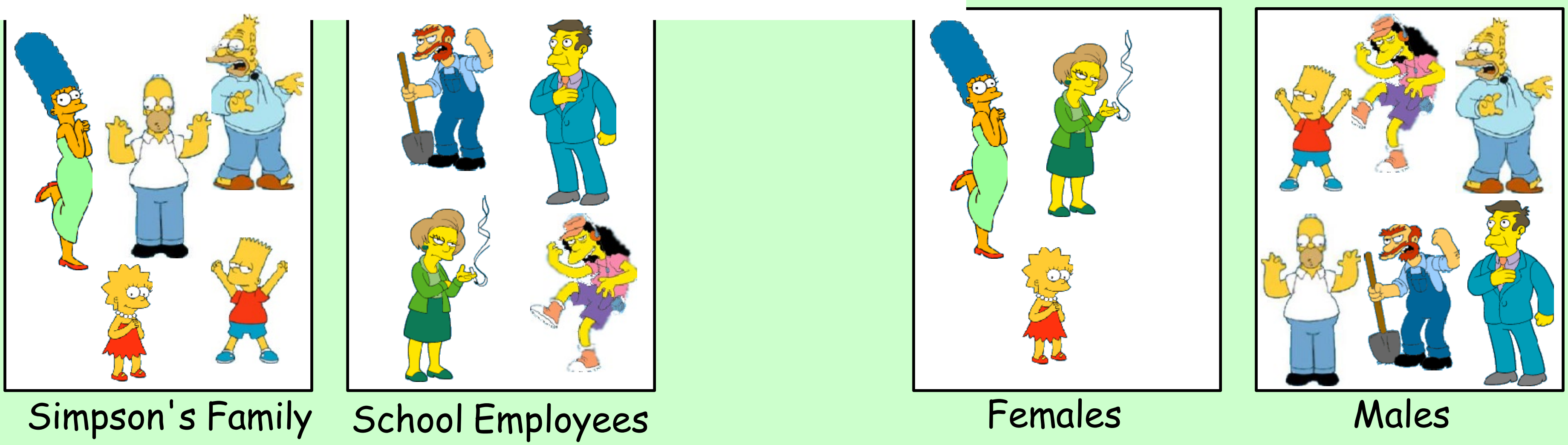
What is a natural grouping among these objects?



What is a natural grouping among these objects?



Clustering is subjective



What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features.

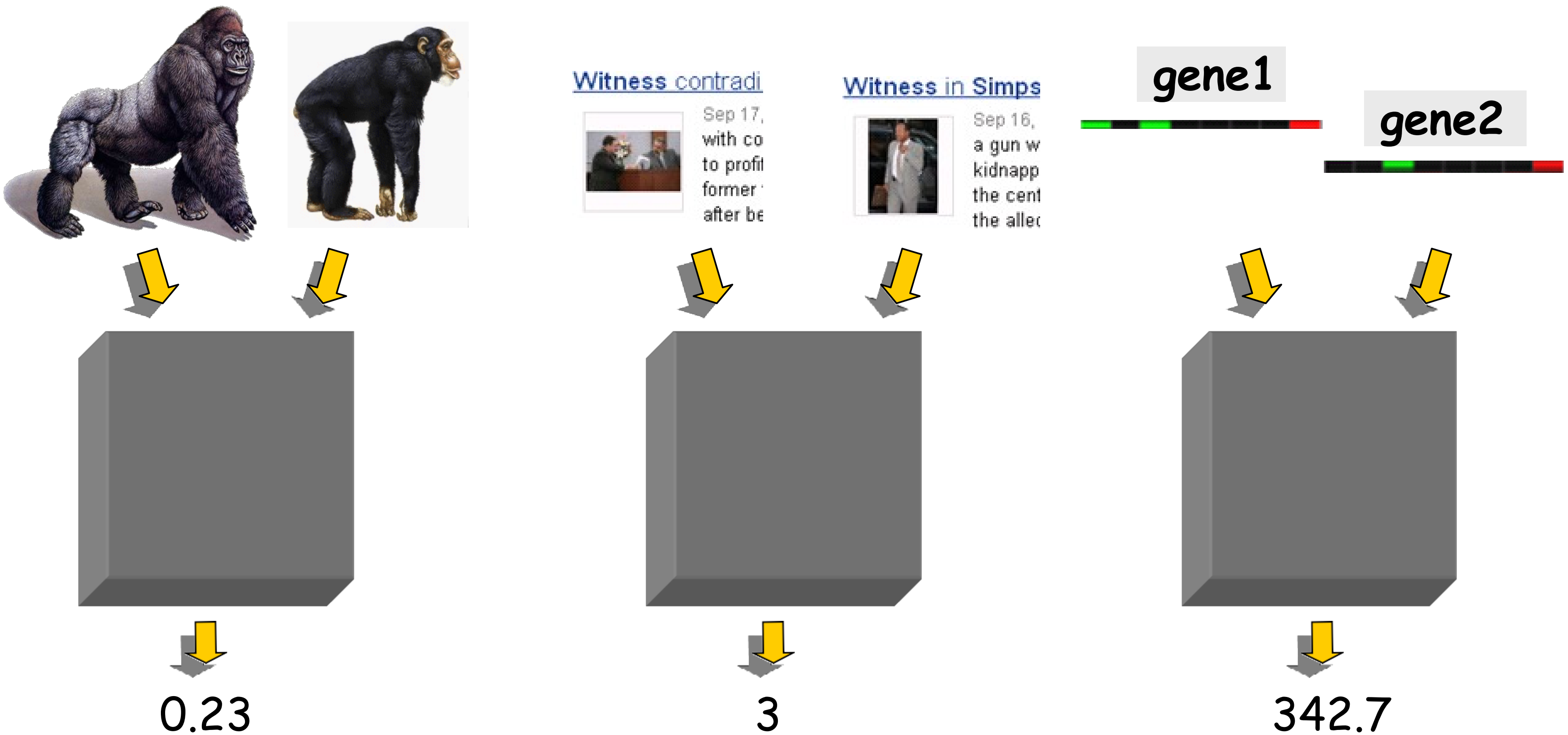
Webster's Dictionary

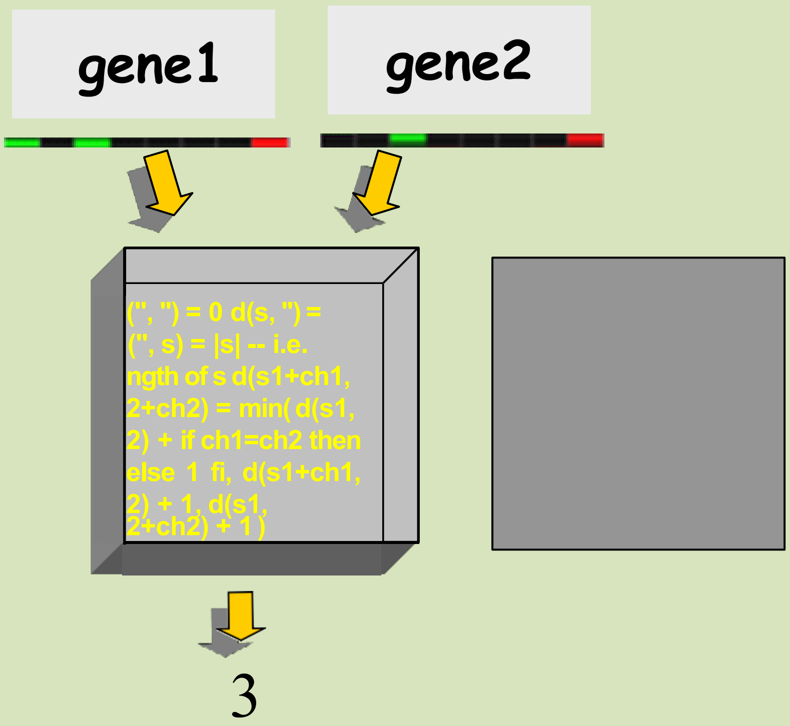
Similarity is hard to define, but… "*We know it when we see it*"

The real meaning of similarity is a philosophical question. We will take a more pragmatic approach.

Defining Distance Measures

Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $d(O_1,O_2)$.





Inside these black boxes:

some function on two variables

(might be simple or very complex)

Examples:

- Euclidean distance

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

- Similarity rather than distance
- Can determine similar trends

- Correlation coefficient

$$s(x,y) = \frac{\sum_i (x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$$

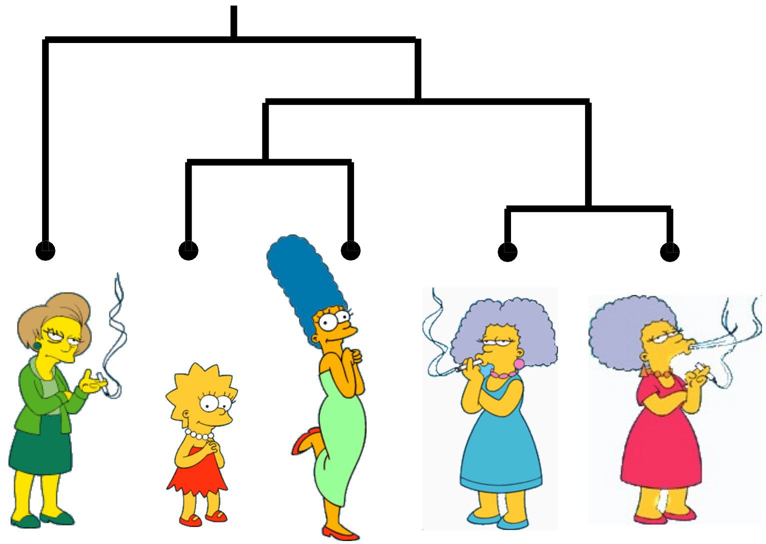
Desirable Properties of a Clustering Algorithm

- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Others: usable, interpretable

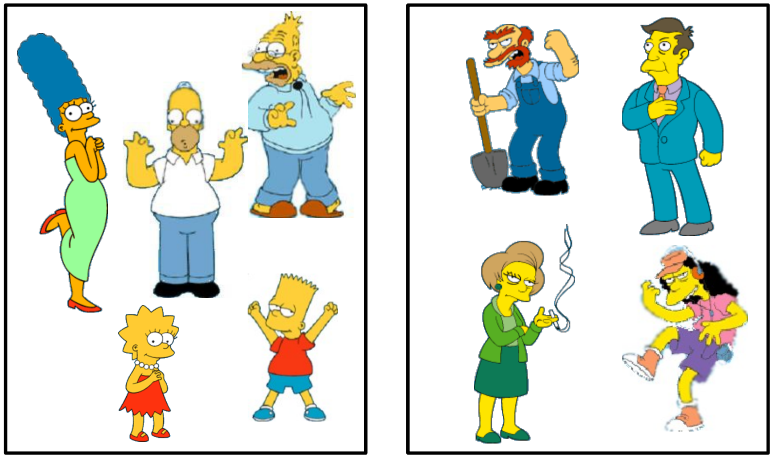
Two Types of Clustering

- **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion
- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion (focus of this class)

Hierarchical



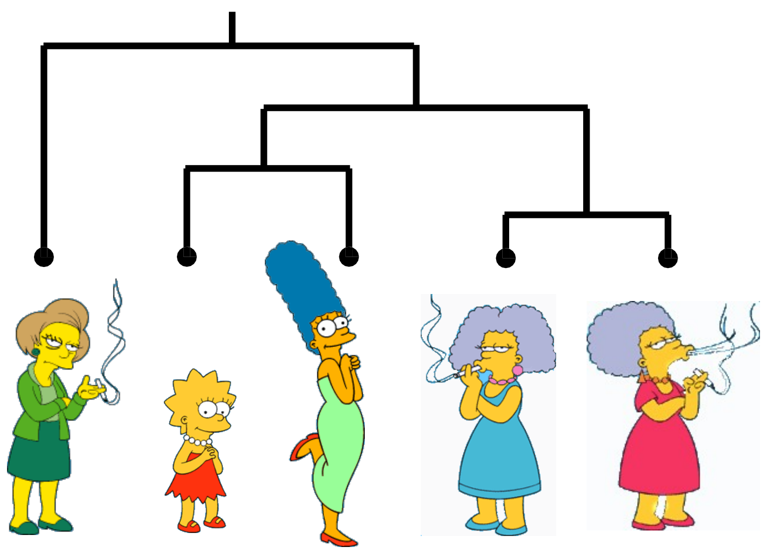
Partitional



How-to do Hierarchical Clustering

The number of dendrograms with n leafs = $(2n-3)!/[(2^{(n-2)})(n-2)!]$

Number of Leafs	Number Possible Dendrograms
2	1
3	3
4	15
5	105
…	…
10	34,459,425

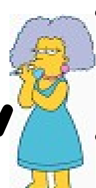


Bottom-Up (agglomerative):

Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

We begin with a distance matrix which contains the distances between every pair of objects in our database.

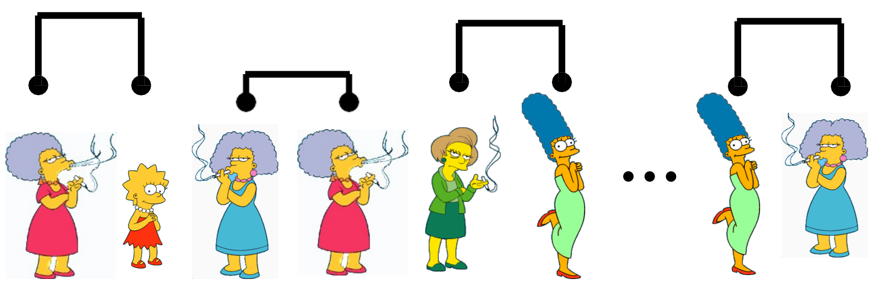
D(,) = 8

D(,) = 1

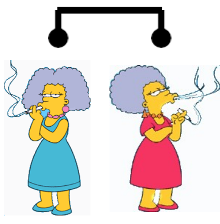
					
	0	8	8	7	7
		0	2	4	4
			0	3	3
				0	1
					0

Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

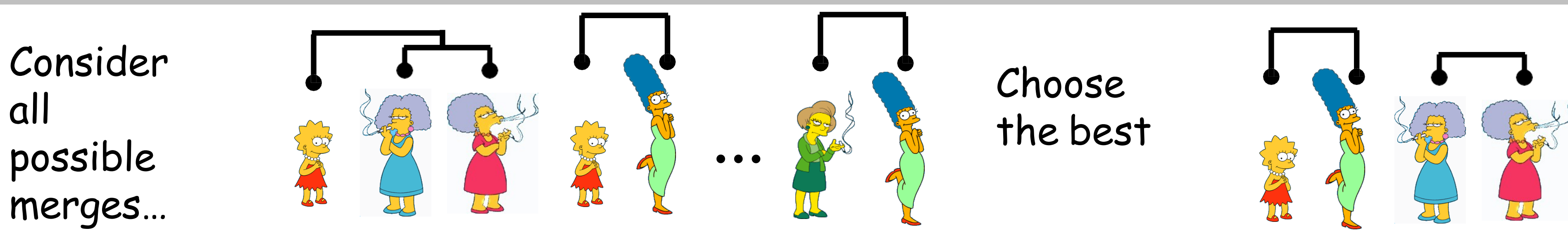
Consider all possible merges…

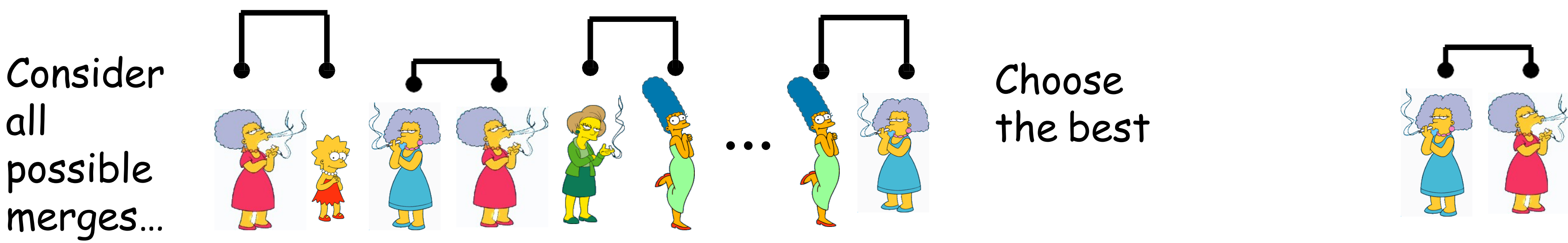


Choose the best



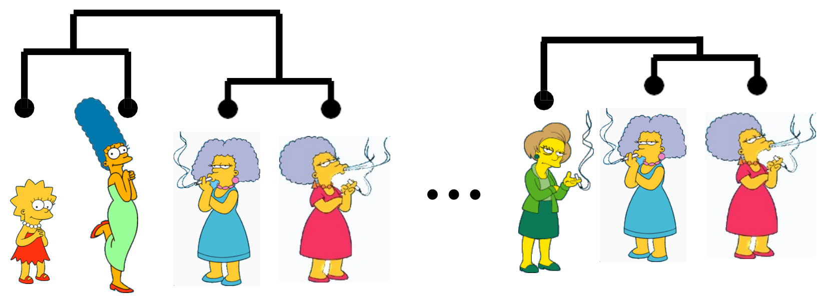
Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



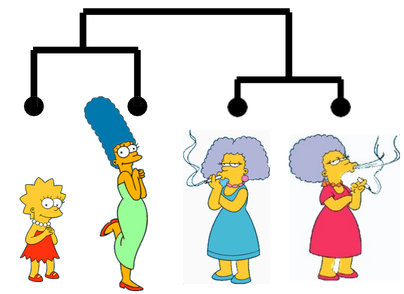


Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

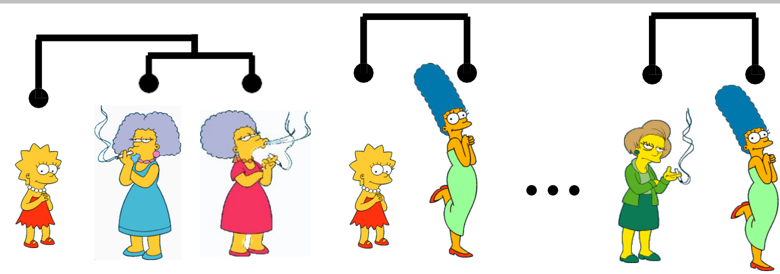
Consider all possible merges…



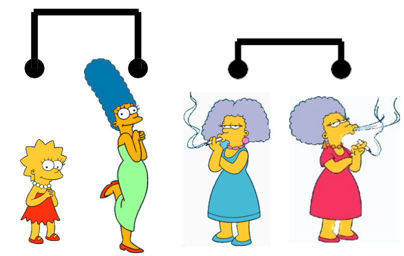
Choose the best



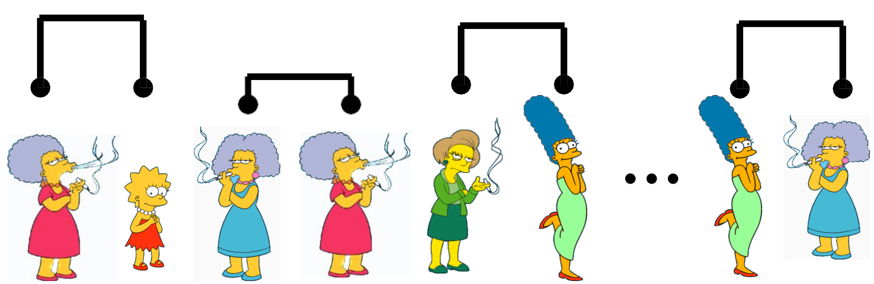
Consider all possible merges…



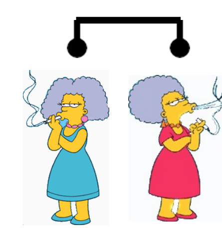
Choose the best



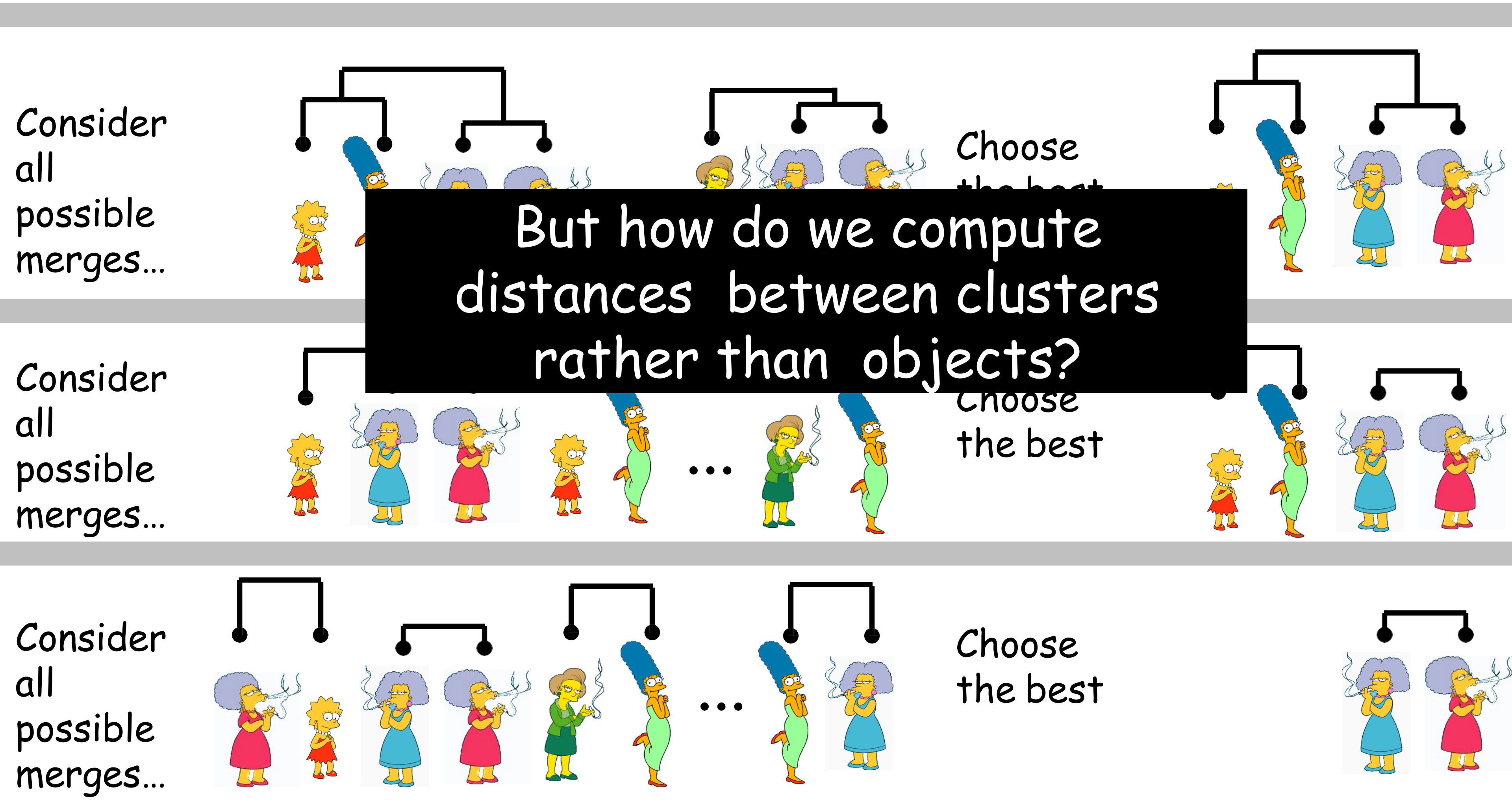
Consider all possible merges…



Choose the best

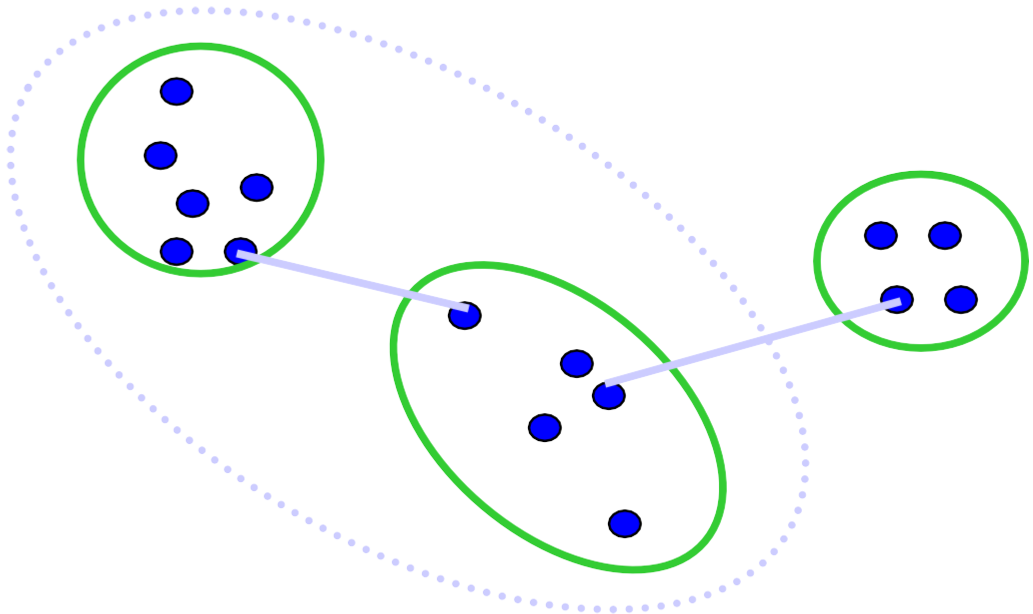


Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



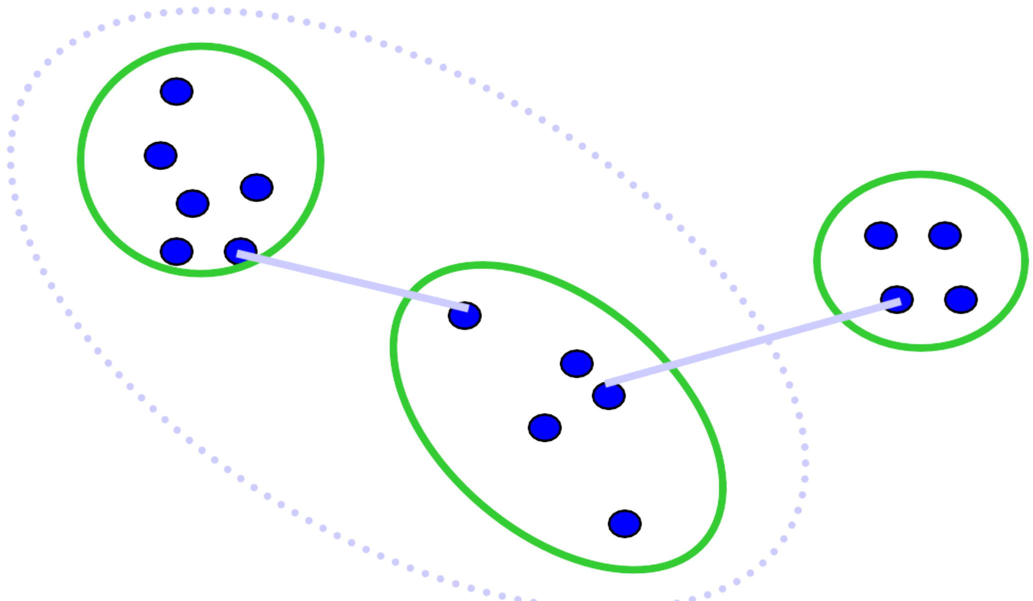
Computing distance between clusters: Single Linkage

- Cluster distance = distance of two closest members in each class



Computing distance between clusters: Single Linkage

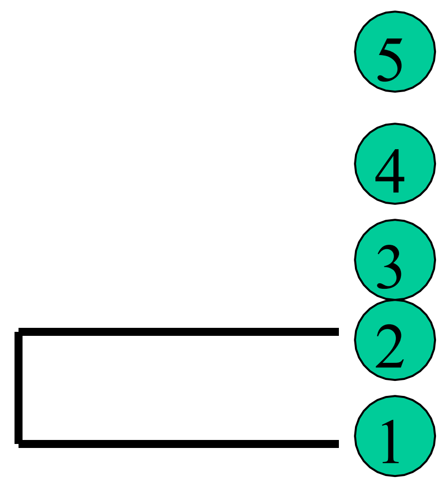
- Cluster distance = distance of two closest members in each class



- Potentially long and skinny clusters

Example: single linkage

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array}$$



Example: single linkage

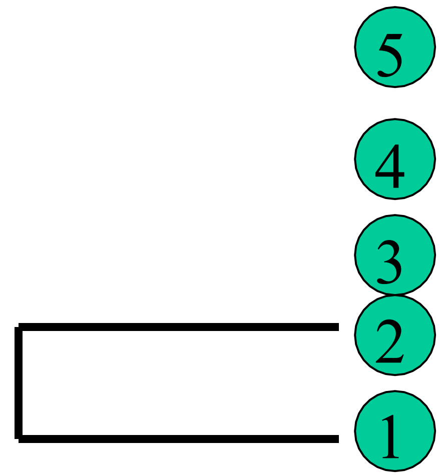
$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{cc} & \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{array}{c} (1,2) \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6,3\} = 3$$

$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10,9\} = 9$$

$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9,8\} = 8$$

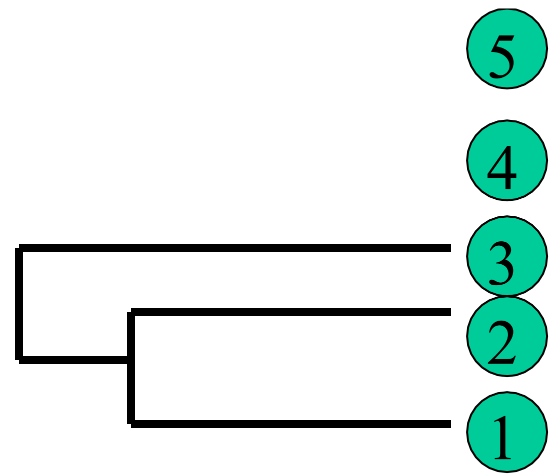


Example: single linkage

$$
\begin{array}{c c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix}
\end{array}
\quad
\begin{array}{c c}
 & \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\
\begin{array}{c} (1,2) \\ 3 \\ 4 \\ 5 \end{array} &
\begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix}
\end{array}
\quad
\begin{array}{c c}
 & \begin{array}{ccc} (1,2,3) & 4 & 5 \end{array} \\
\begin{array}{c} (1,2,3) \\ 4 \\ 5 \end{array} &
\begin{bmatrix} 0 & & \\ 7 & 0 & \\ 5 & 4 & 0 \end{bmatrix}
\end{array}
$$

$$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9,7\} = 7$$

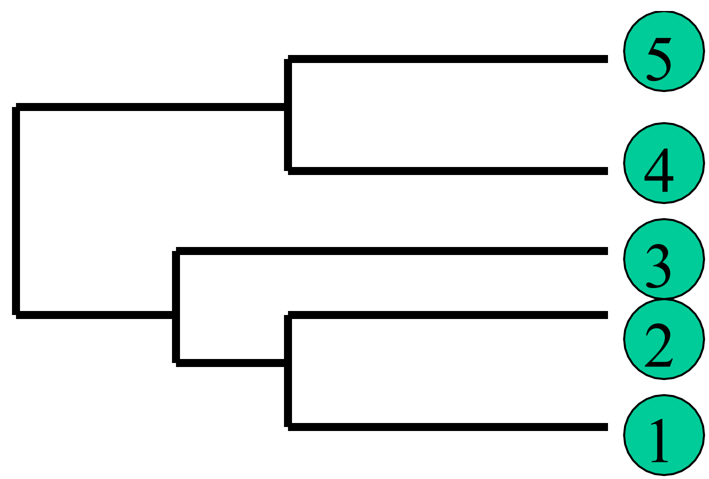
$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5$$



Example: single linkage

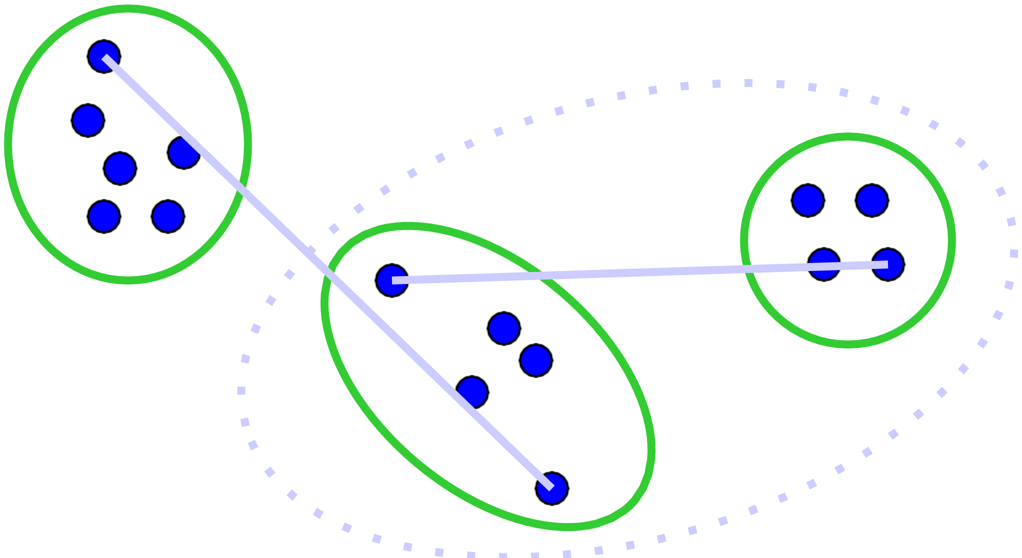
$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|cccc}
 & (1,2) & 3 & 4 & 5 \\
\hline
(1,2) & 0 & & & \\
3 & 3 & 0 & & \\
4 & 9 & 7 & 0 & \\
5 & 8 & 5 & 4 & 0
\end{array}
\Rightarrow
\begin{array}{c|ccc}
 & (1,2,3) & 4 & 5 \\
\hline
(1,2,3) & 0 & & \\
4 & 7 & 0 & \\
5 & 5 & 4 & 0
\end{array}
$$

$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$



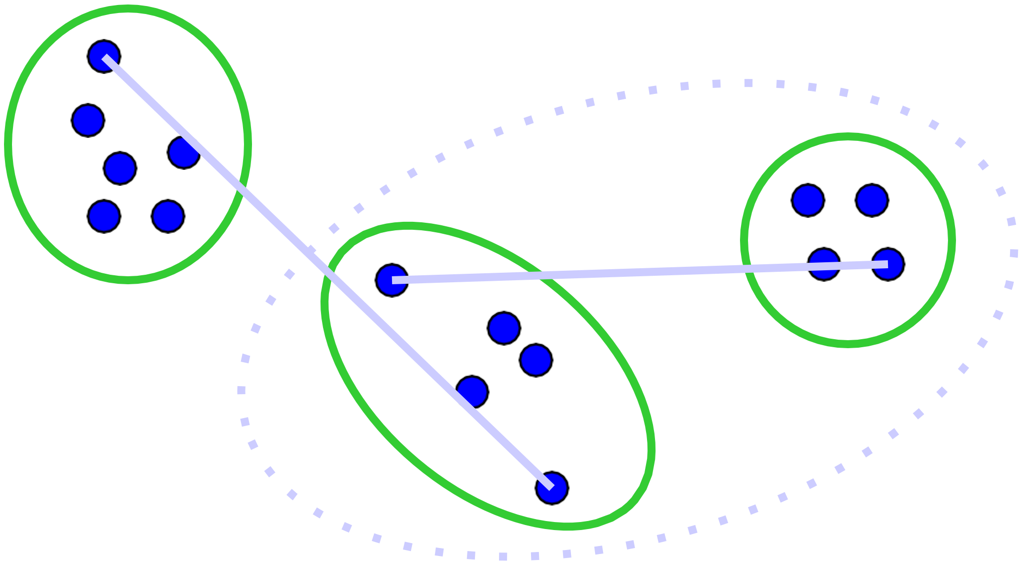
Computing distance between clusters: Complete Linkage

- Cluster distance = distance of two farthest members



Computing distance between clusters: Complete Linkage

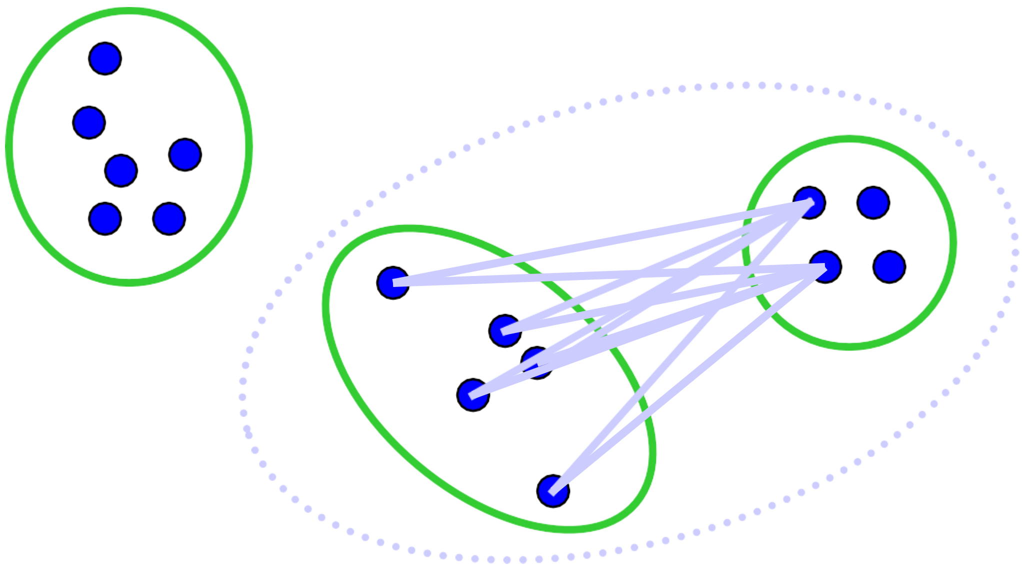
- Cluster distance = distance of two farthest members



+ Tight clusters

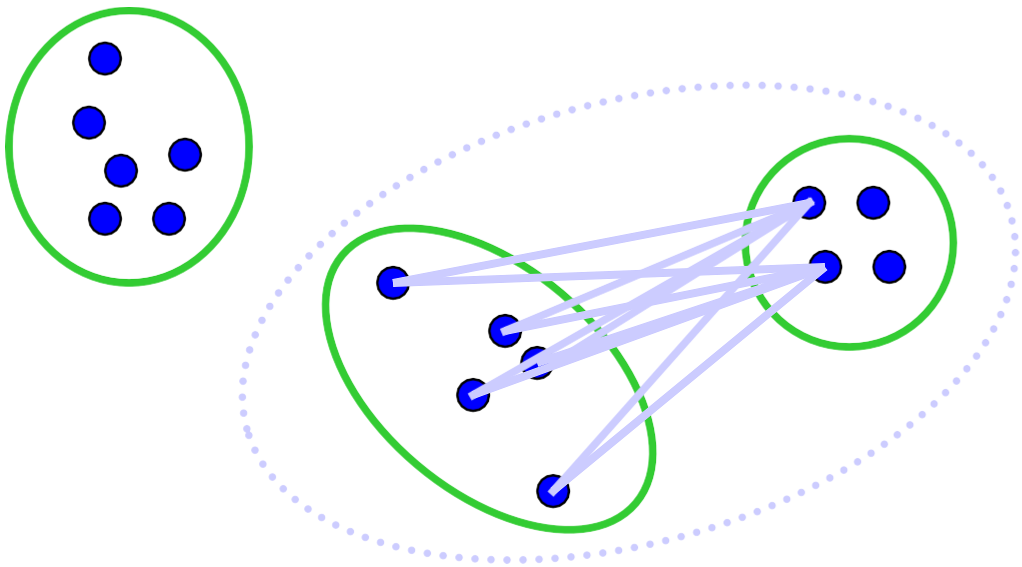
Computing distance between clusters: Average Linkage

- Cluster distance = average distance of all pairs



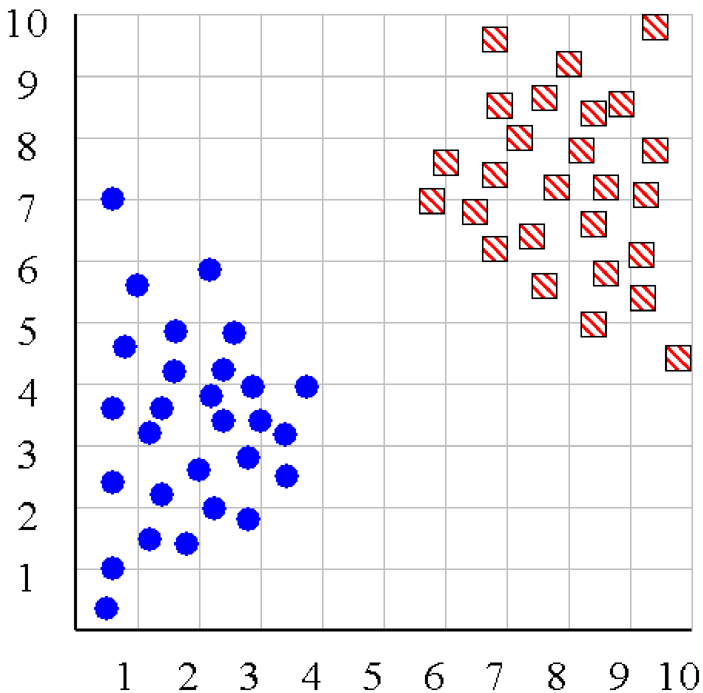
Computing distance between clusters: Average Linkage

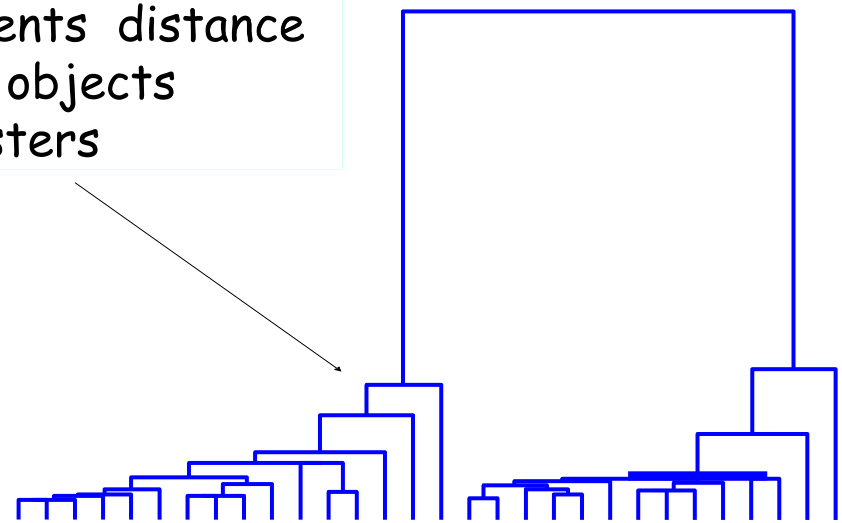
- Cluster distance = average distance of all pairs



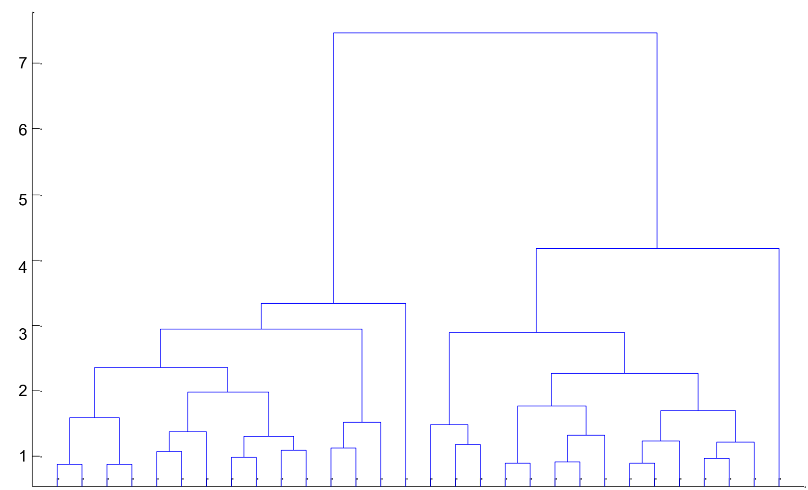
The most widely used measure; robust against noise

Height represents distance between objects / clusters





Single linkage



Average linkage

Summary of Hierarchal Clustering Methods

- No need to specify the number of clusters in advance.
- Hierarchical structure maps nicely onto human intuition for

some domains

- Time complexity O(n^2), where n is the number of total objects

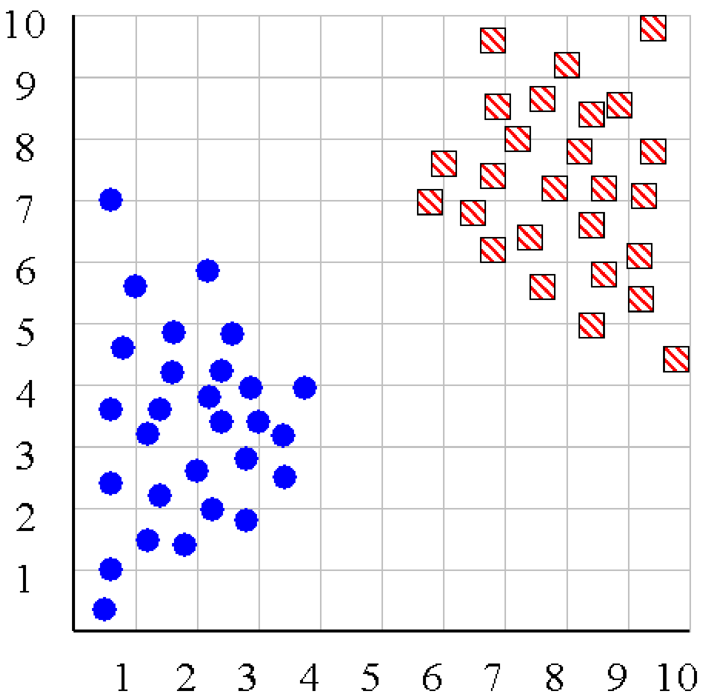
[depends on distance used]

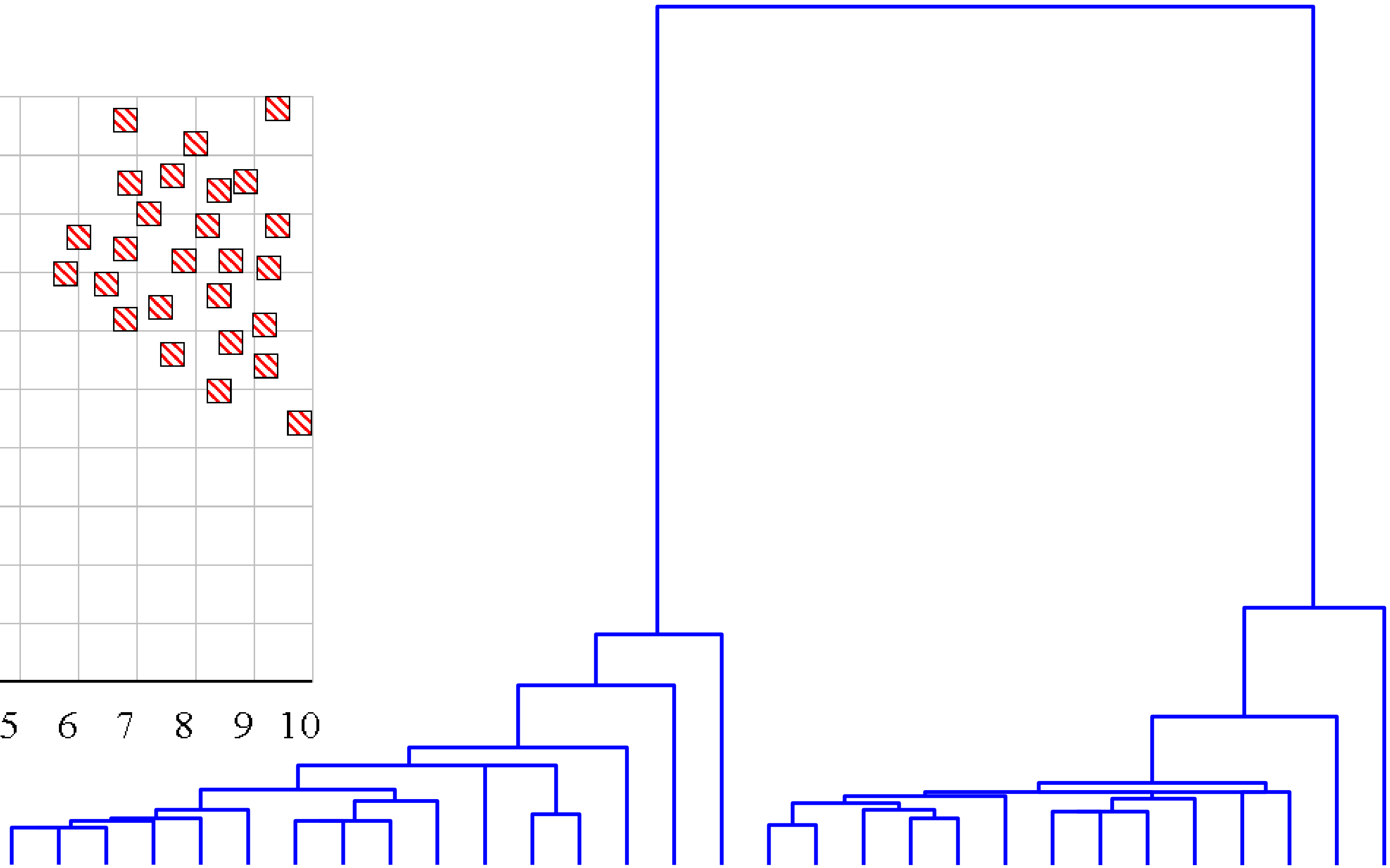
- Like any heuristic search algorithms, local optima are a problem.
- Interpretation of results is subjective.

But what are the clusters?

In some cases we can determine the “correct” number of clusters.

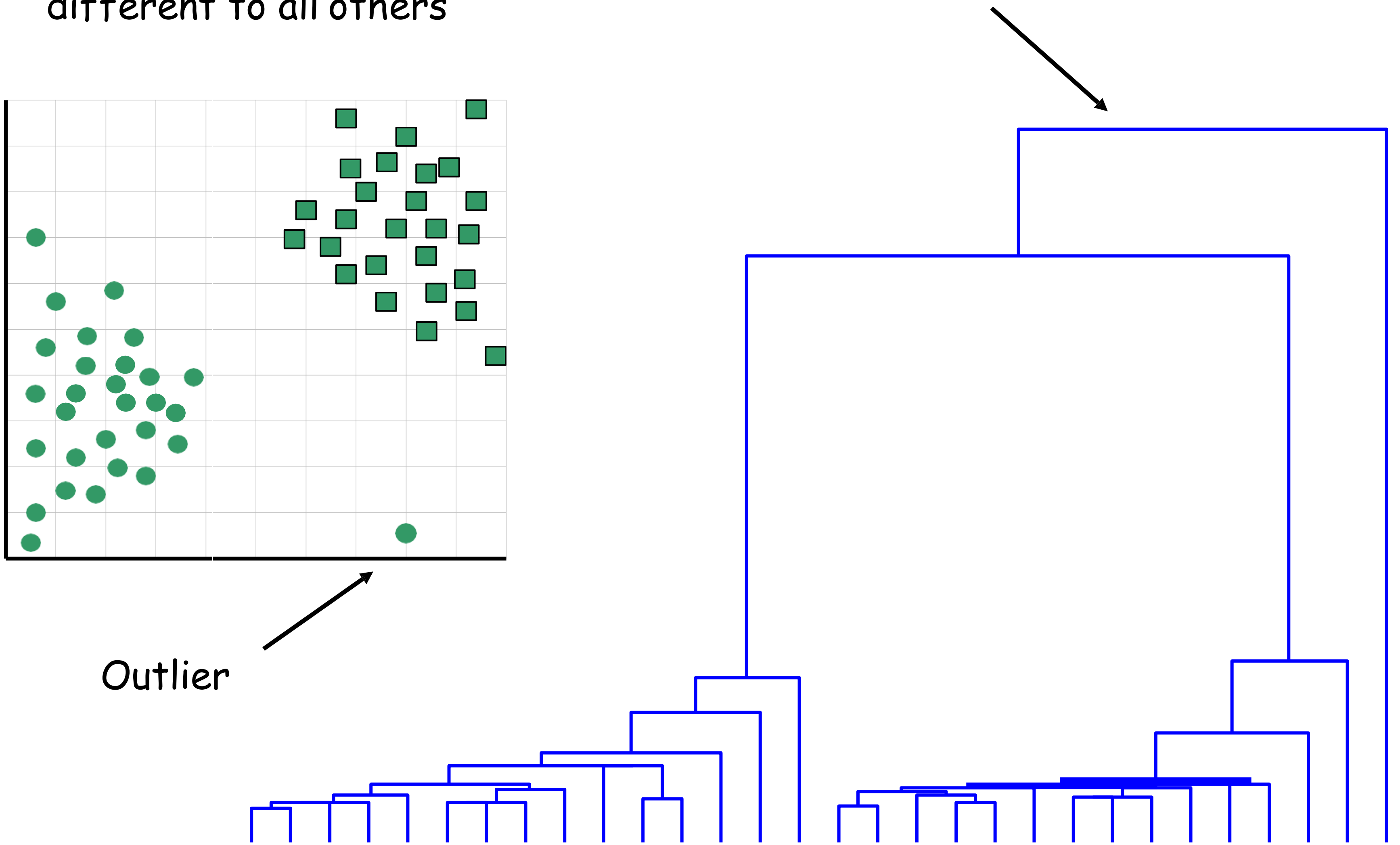
However, things are rarely this clear cut, unfortunately.





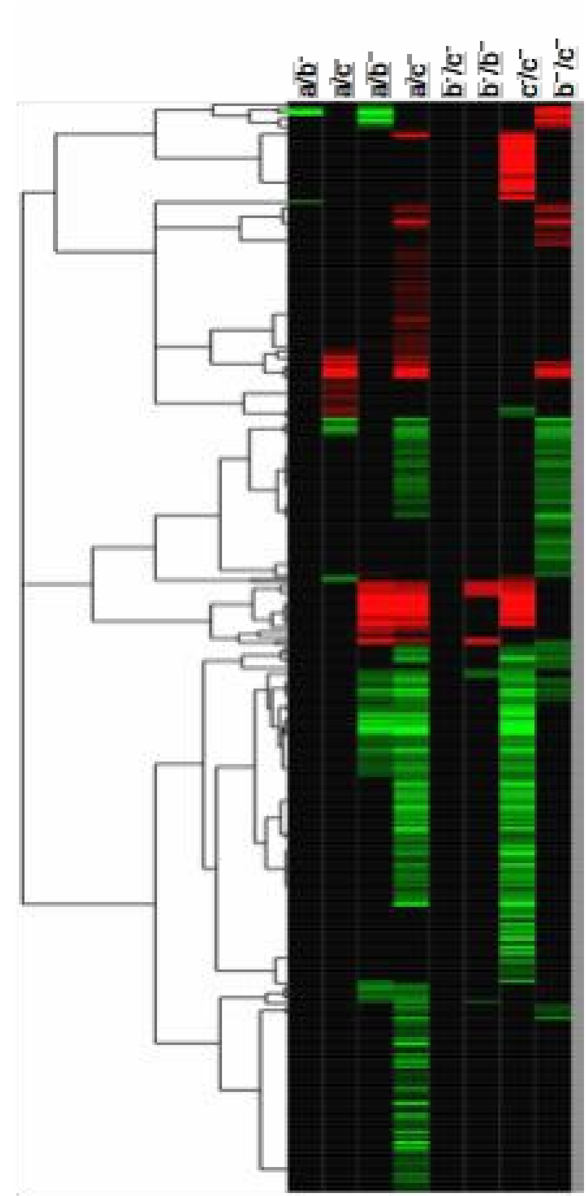
One potential use of a dendrogram is to detect outliers

The single isolated branch is suggestive of a data point that is very different to all others



Example: clustering genes

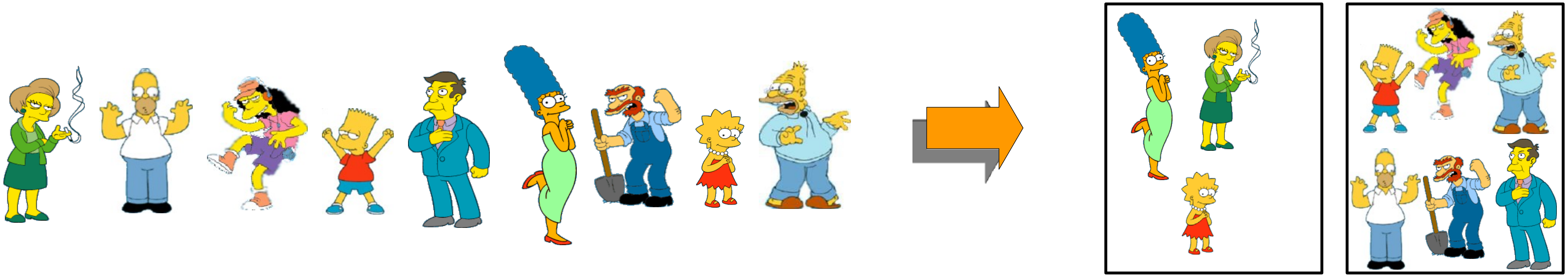
- Microarrays measures the activities of all genes in different conditions
- Clustering genes can help determine new functions for unknown genes



Partitional Clustering

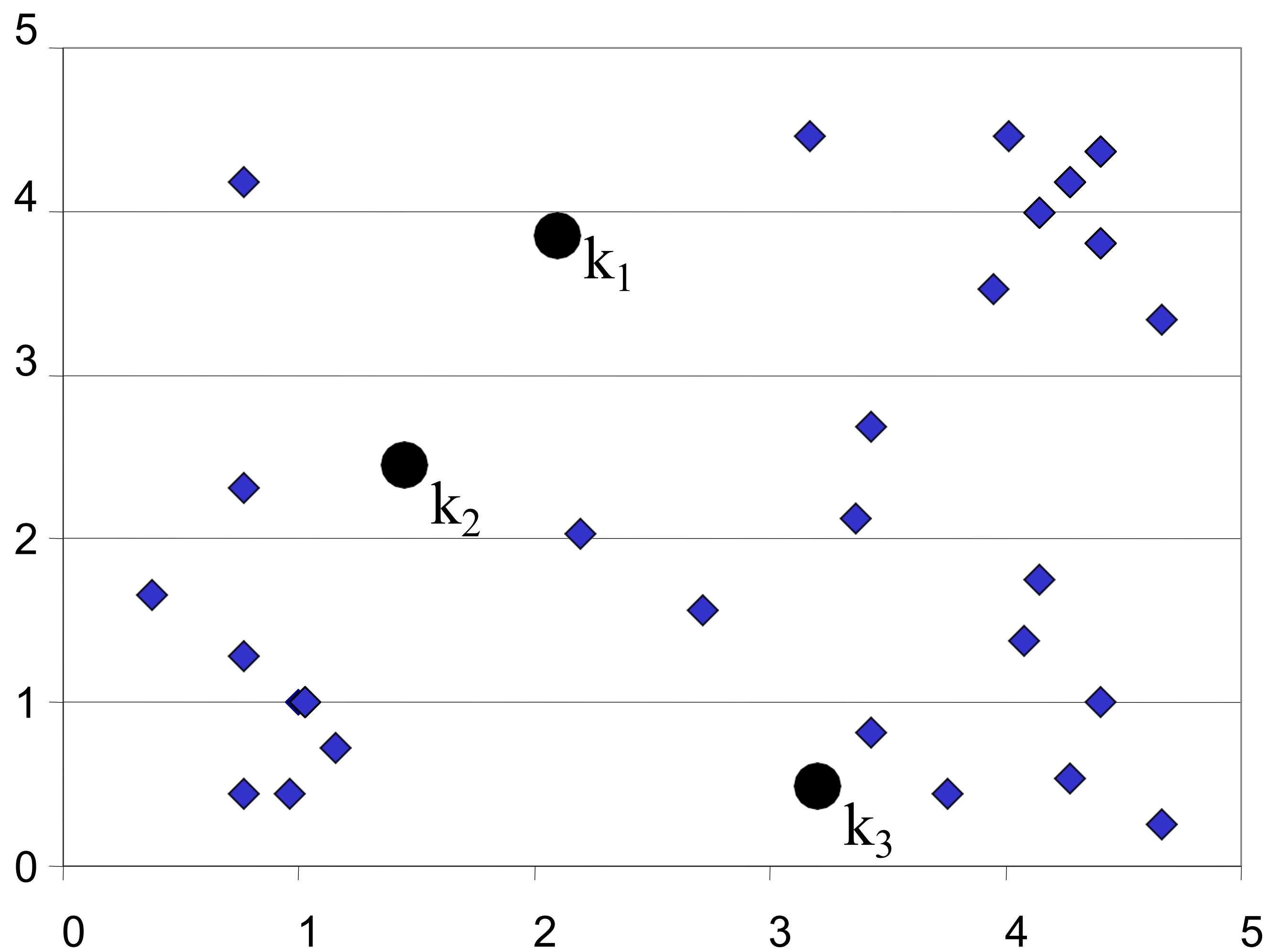
- Nonhierarchical, each instance is placed in exactly one of K non-overlapping clusters.

- Since the output is only one set of clusters the user has to specify the desired number of clusters K.



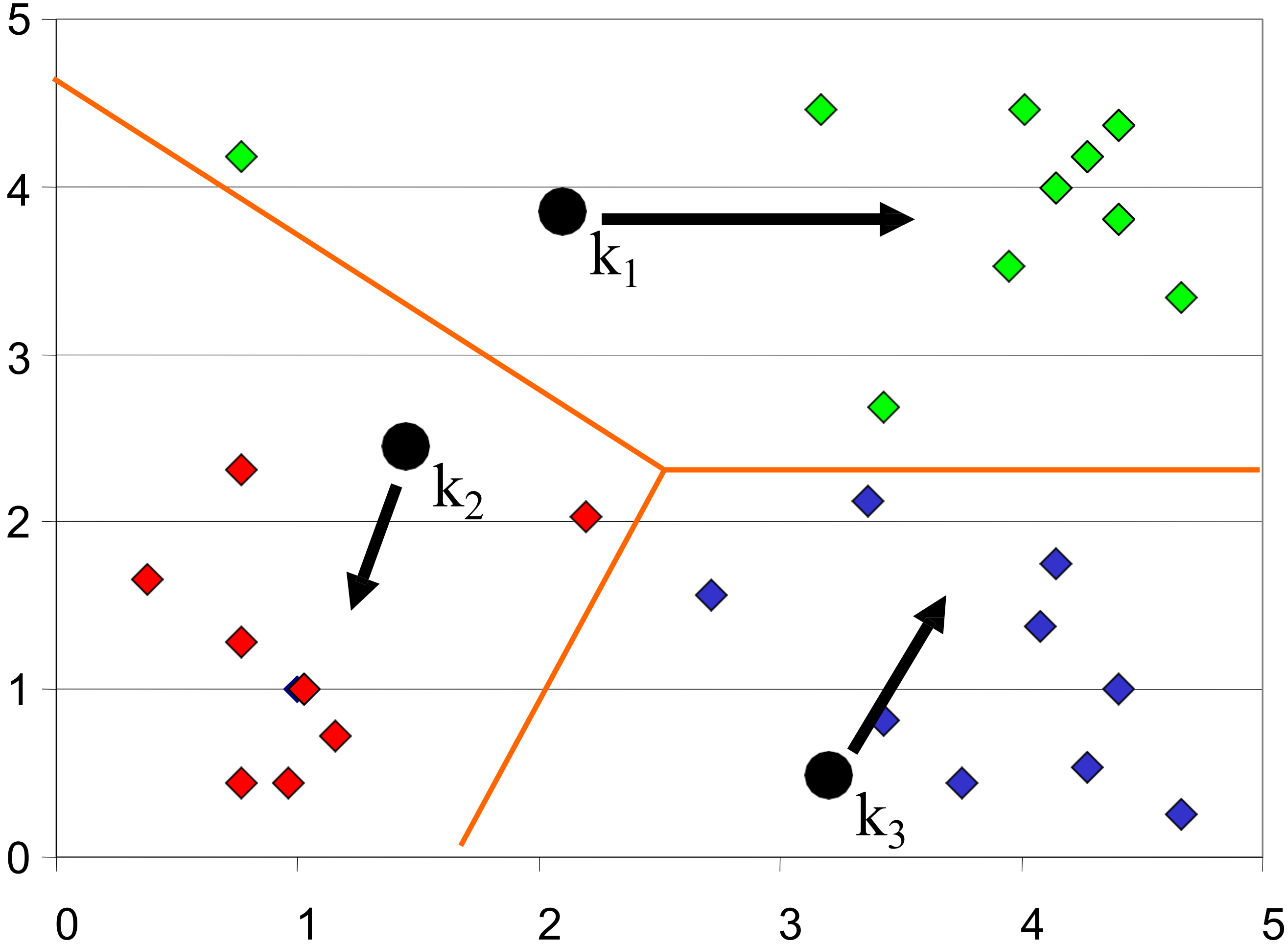
K-means Clustering

Decide *K*, and initialize *K* centers (randomly)



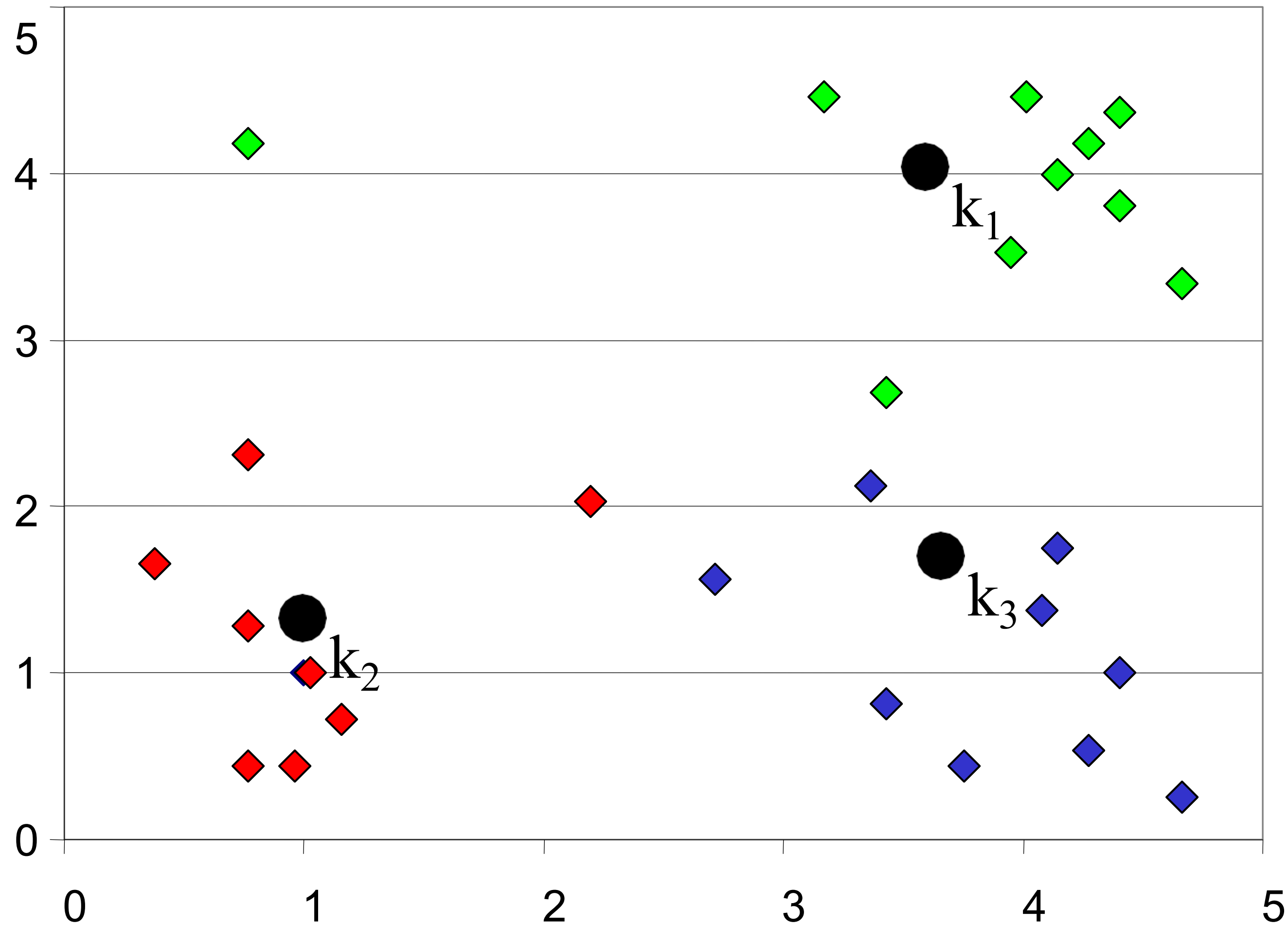
K-means Clustering

Assign all objects to the nearest center.
Move a center to the mean of its members.



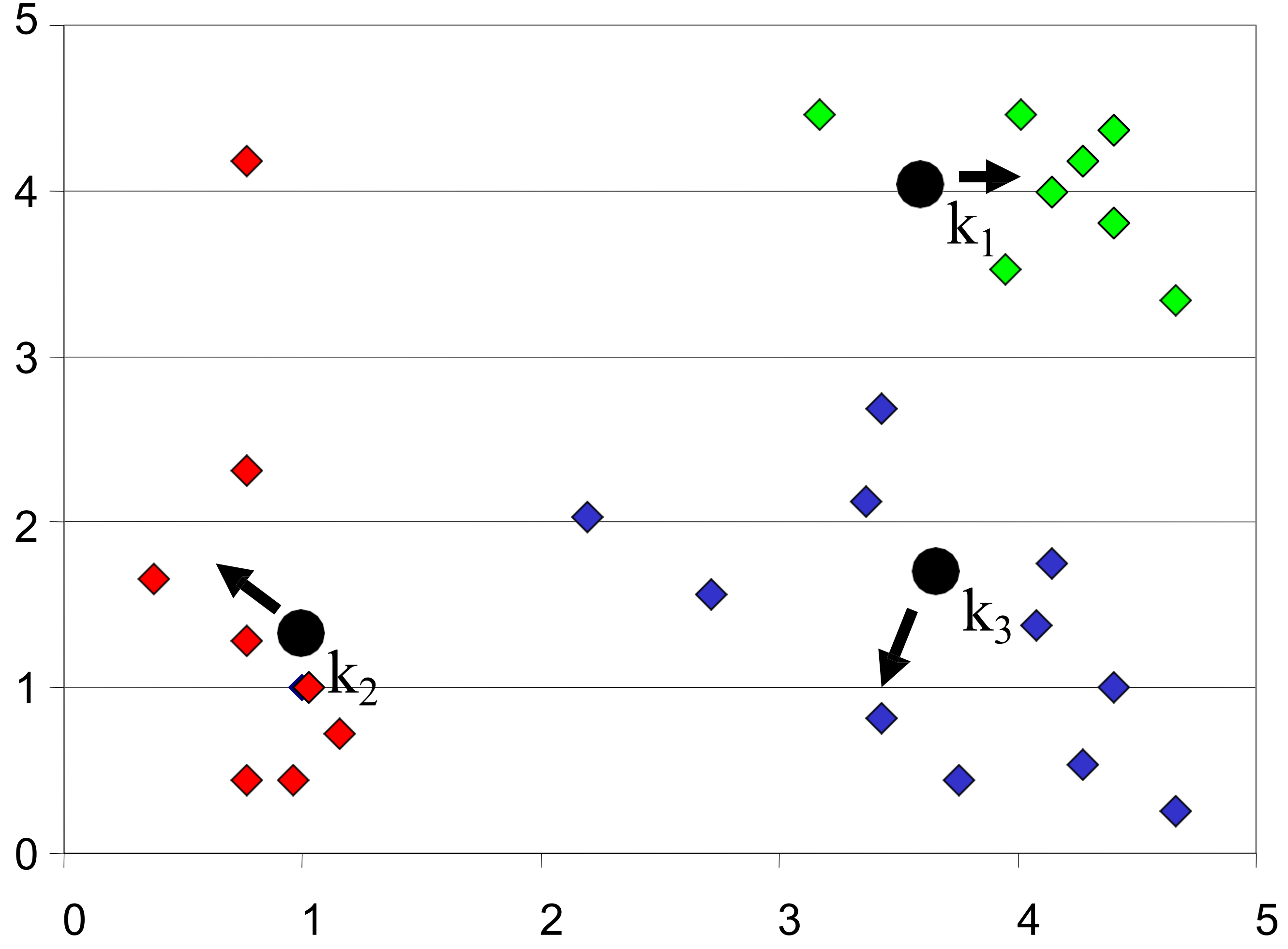
K-means Clustering

After moving centers, re-assign the objects…



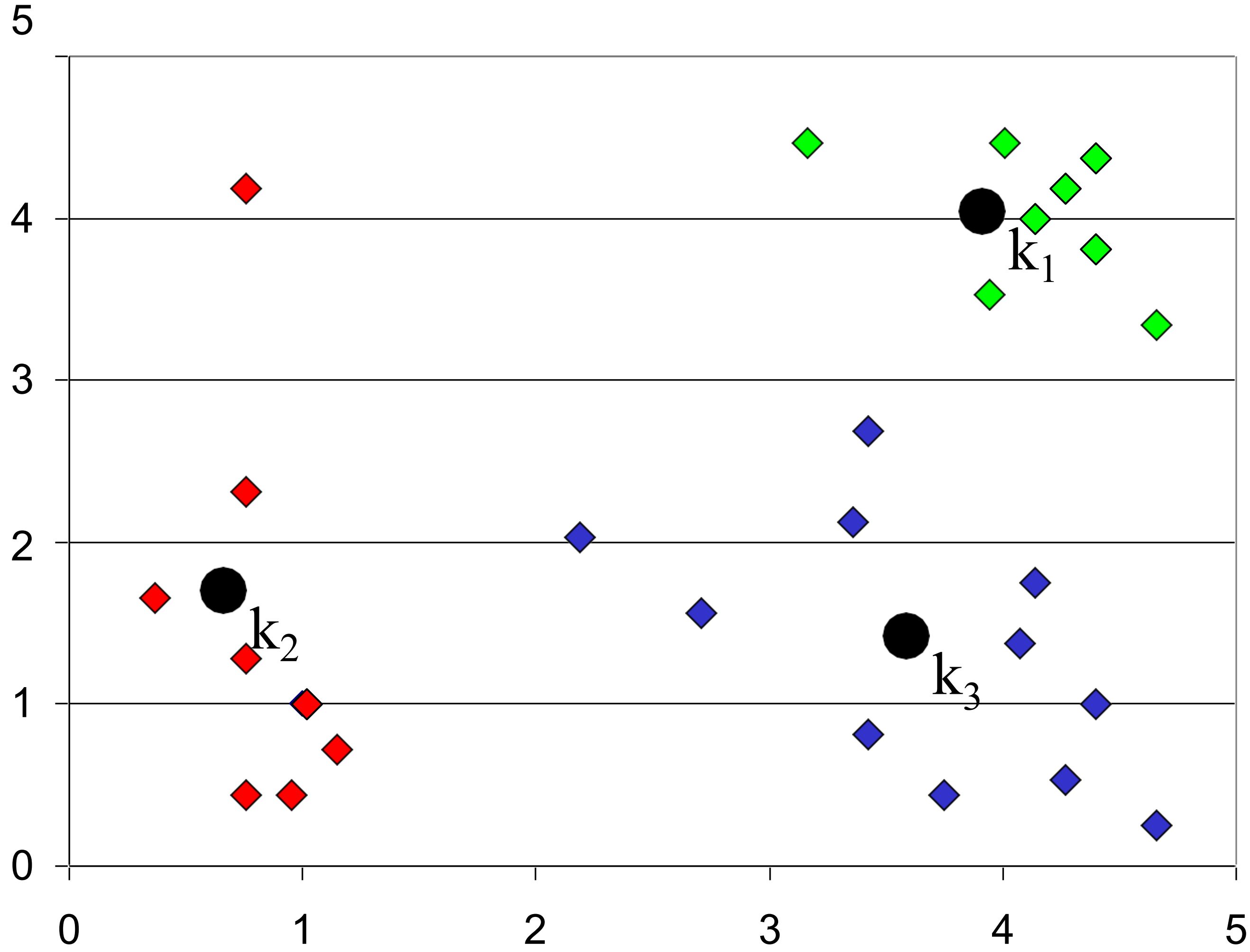
K-means Clustering

After moving centers, re-assign the objects to nearest centers.
Move a center to the mean of its new members.



K-means Clustering

Re-assign and move centers, until no objects changed membership.



k-means Algorithm

1. Decide on a value for *K*, the number of clusters.
2. Initialize the *K* cluster centers (randomly, if necessary).
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.
4. Re-estimate the *K* cluster centers, by assuming the memberships found above are correct.
5. Repeat 3 and 4 until none of the *N* objects changed membership in the last iteration.

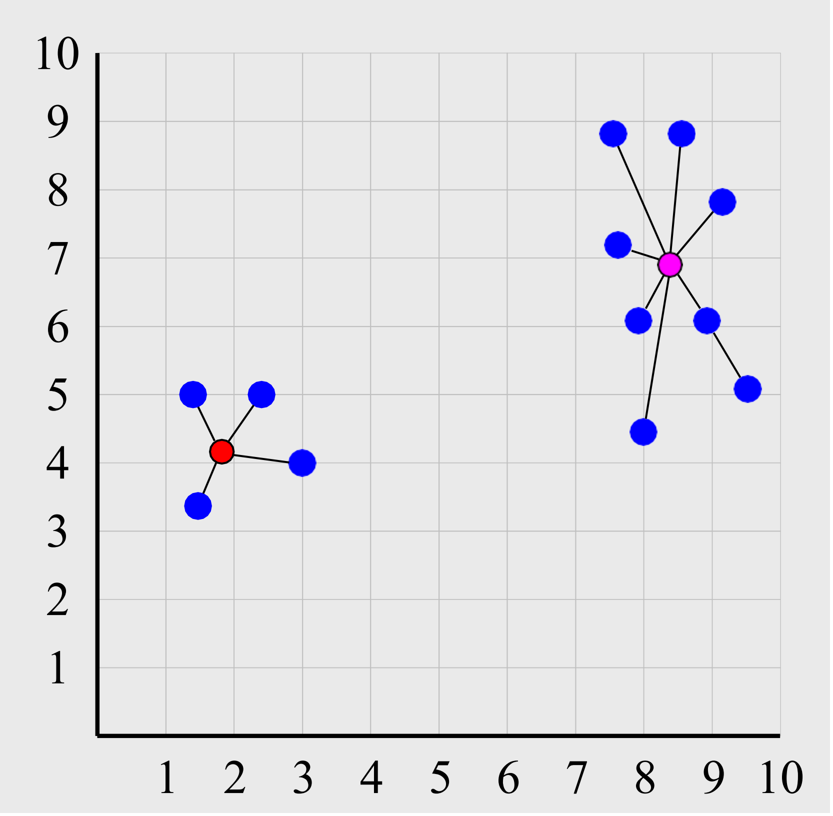
The K-means Objective

- What is a good partition?
- High intra-cluster similarity
- K-means optimizes
  - the average distance to members of the same cluster

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} \left\| x_{ki} - x_{kj} \right\|^2$$

  - which is twice the total distance to centers, also called squared error

$$se = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \left\| x_{ki} - \mu_k \right\|^2$$



Summary: *K-Means*

- Strength
  - Simple, easy to implement and debug
  - Intuitive objective function: optimizes intra-cluster similarity
  - *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.
- Weakness
  - Applicable only when *mean* is defined, what about categorical data?
  - Often terminates at a *local optimum*. Initialization is important.
  - Need to specify *K*, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*
- Summary
  - Assign members based on current centers
  - Re-estimate centers based on current assignment