10601
Machine Learning

Hierarchical clustering

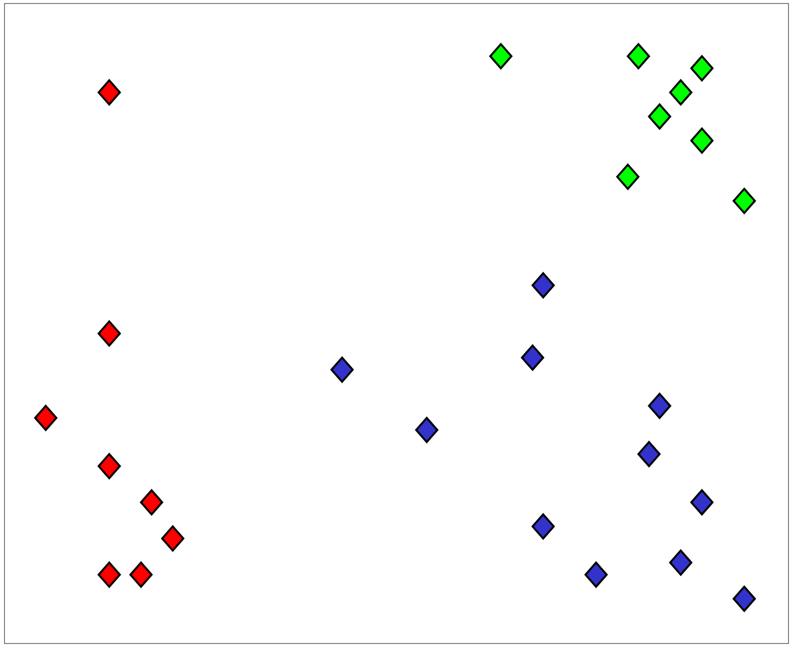
Reading: Bishop: 9-9.2

Second half: Overview

- Clustering
  - Hierarchical, semi-supervised learning
- Graphical models
  - Bayesian networks, HMMs, Reasoning under uncertainty
- Putting it together
  - Model / feature selection, Boosting, dimensionality reduction
- Advanced classification
  - SVM

What is Clustering?

- Organizing data into *clusters* such that there is
  - high intra-cluster similarity
  - low inter-cluster similarity

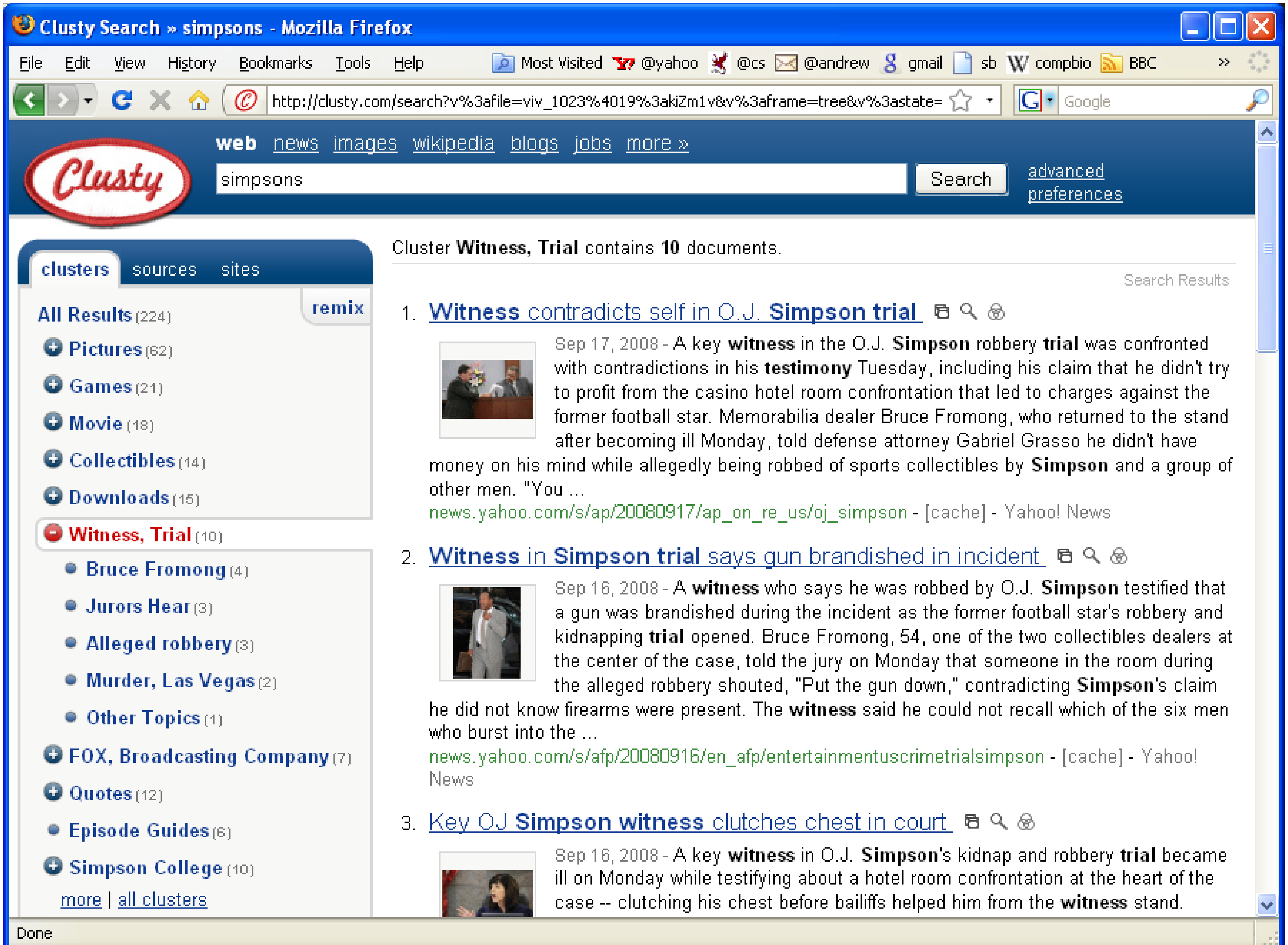


- Informally, finding natural groupings among objects.

- Why do we want to do that?

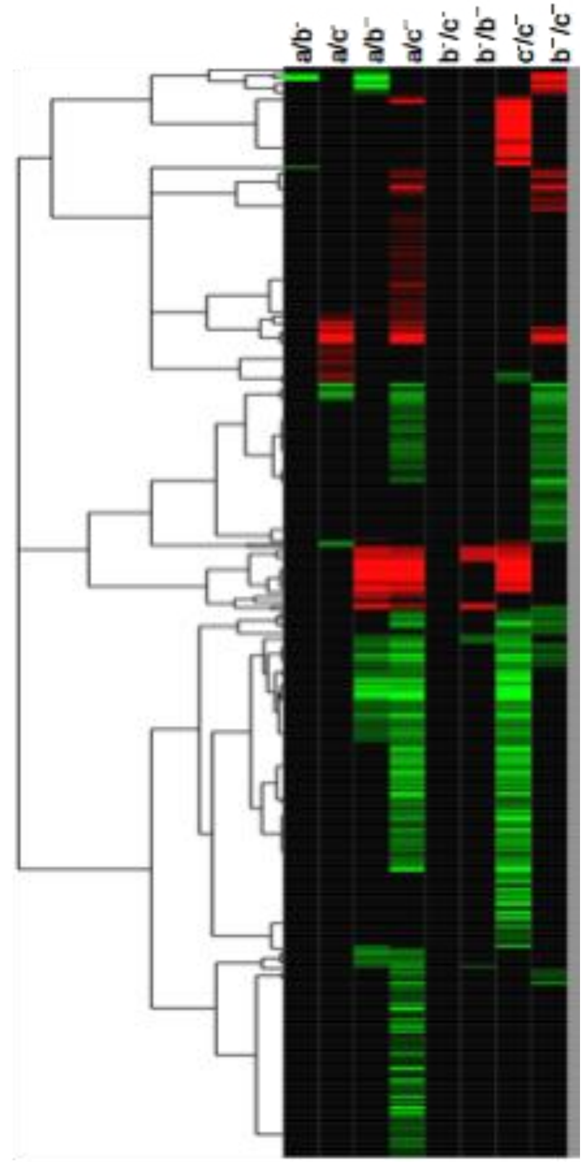
- Any REAL application?

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
  - The most cited (12,309) paper in PNAS!



Unsupervised learning

- 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

Outline

- Distance functions
- Hierarchical clustering
- Number of clusters

What is Similarity?

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

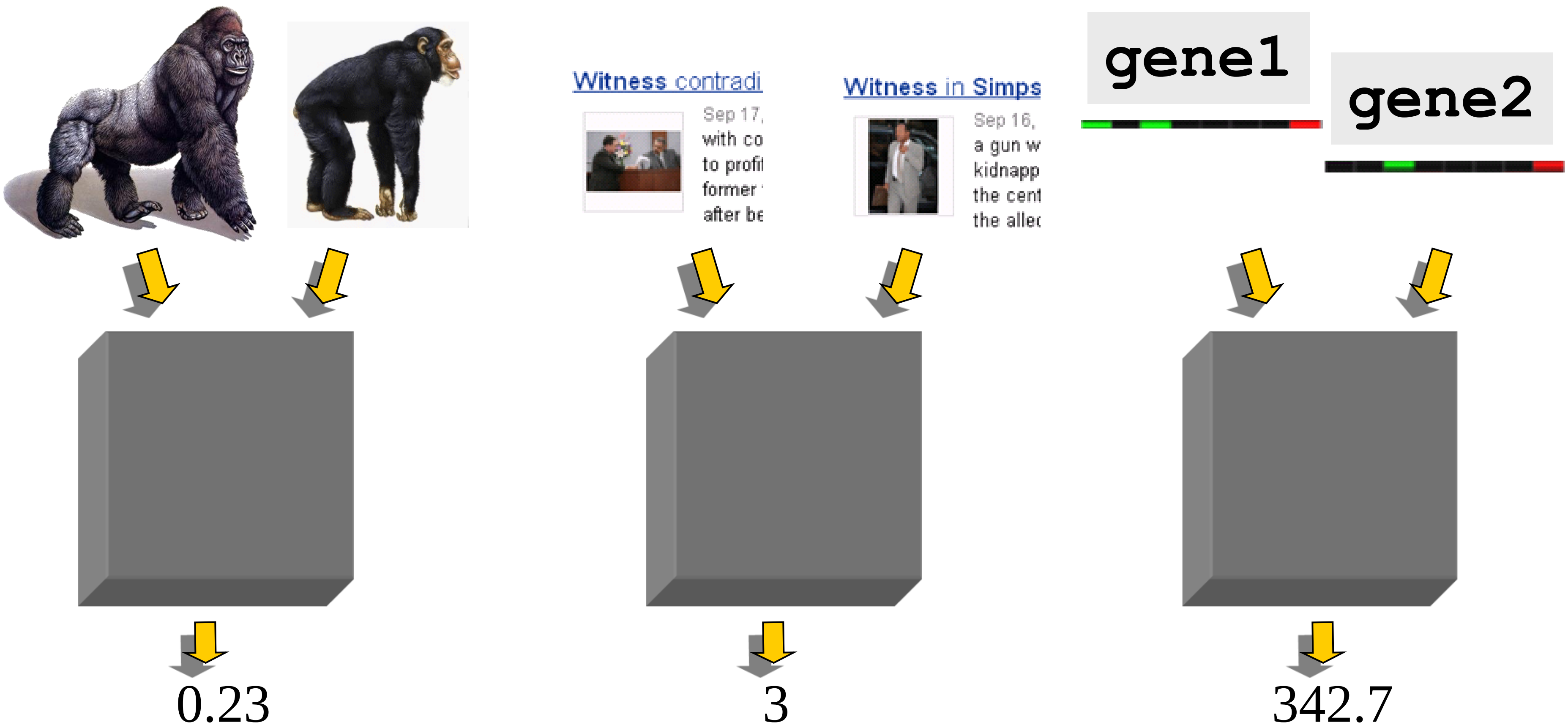
Webster's Dictionary

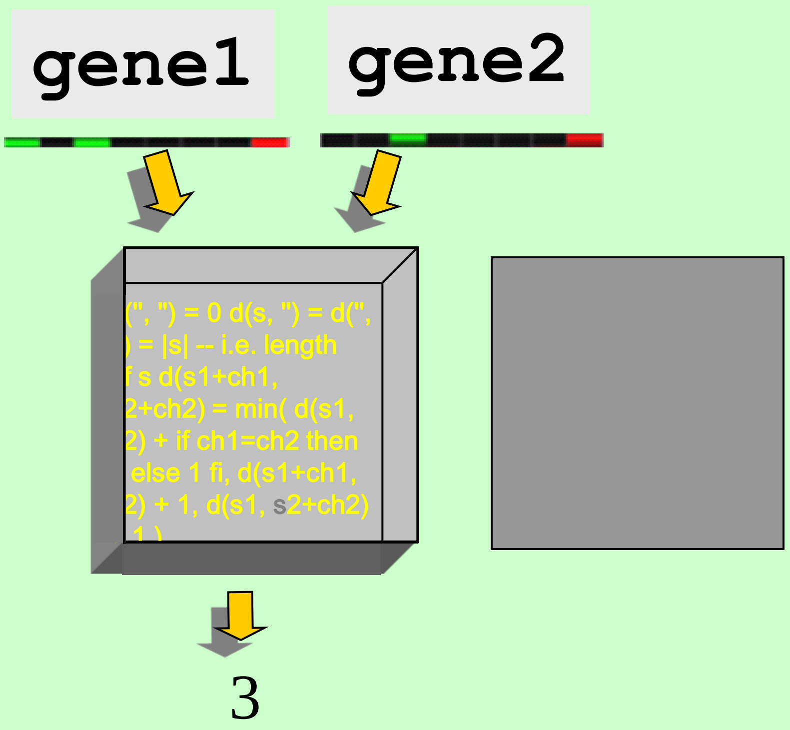
Similarity is hard to define, but… "*We know it when we see it*"

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)

A few examples:

- Euclidian distance

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

- Correlation coefficient

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

- Similarity rather than distance
- Can determine similar trends

Outline

- Distance measure
- Hierarchical clustering
- Number of clusters

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
- Interpretability and usability

Optional

- Incorporation of user-specified constraints

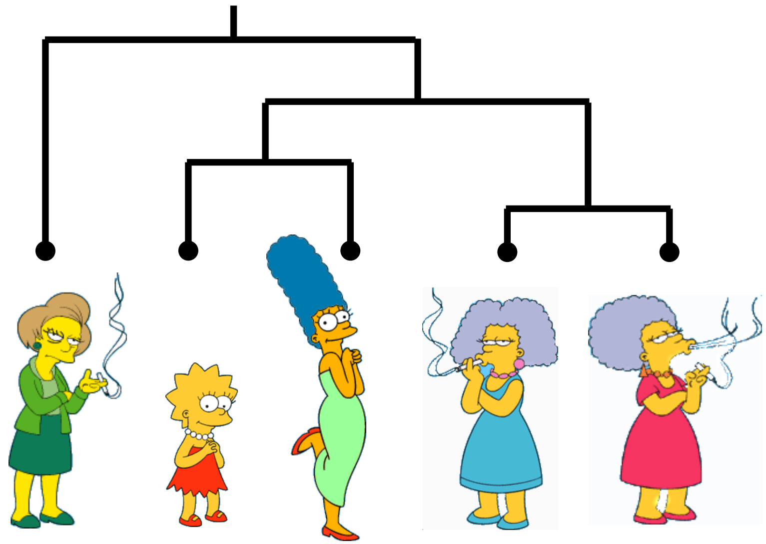
Two Types of Clustering

- **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)

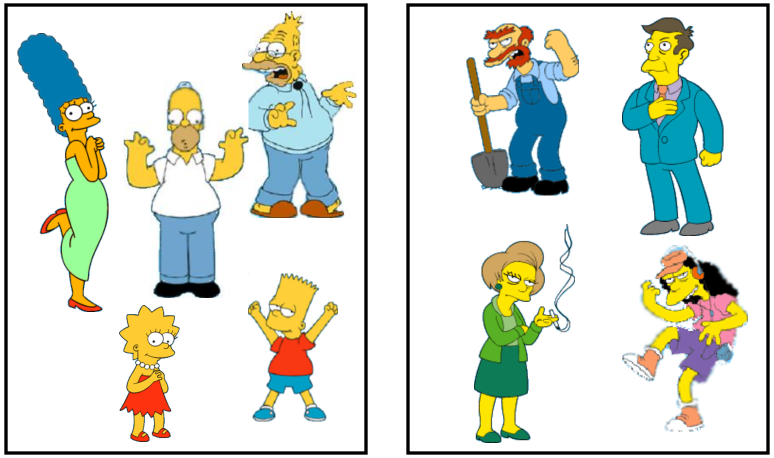
Bottom up or top down

Top down

Hierarchical



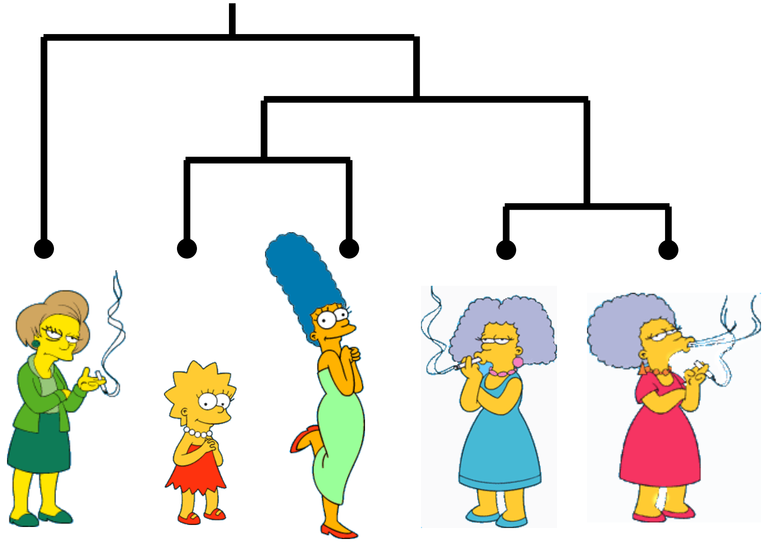
Partitional



(How-to) Hierarchical Clustering

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

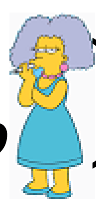
Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	…
10	34,459,425



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

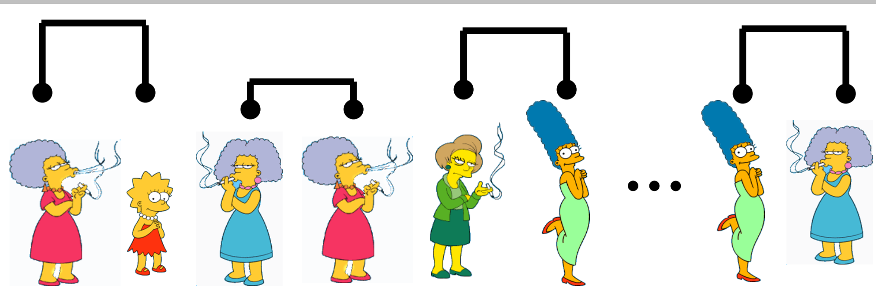
D(, ) = 1

					
	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…

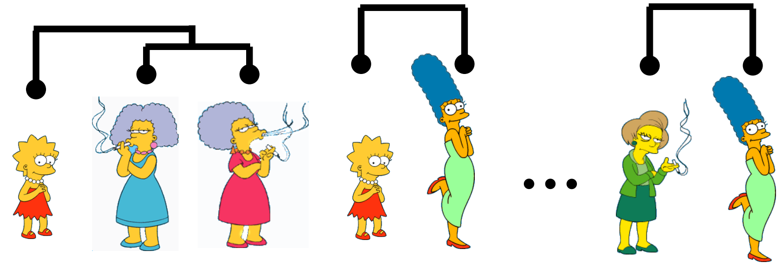


Choose the best

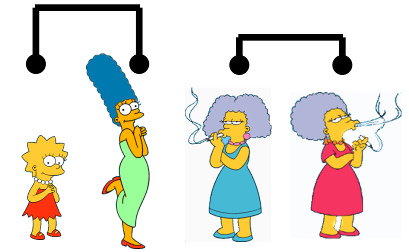
Bottom-Up (agglomerative):

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.

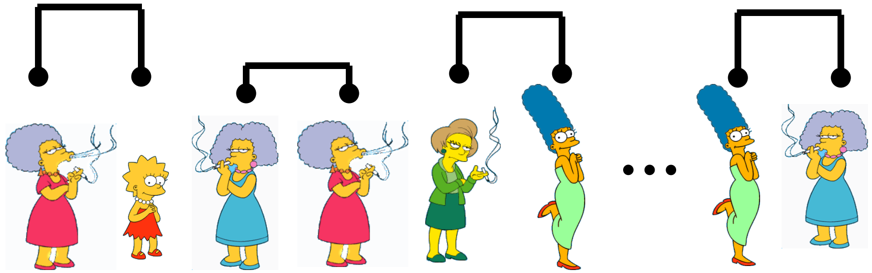
Consider all possible merges…



Choose the best



Consider all possible merges…

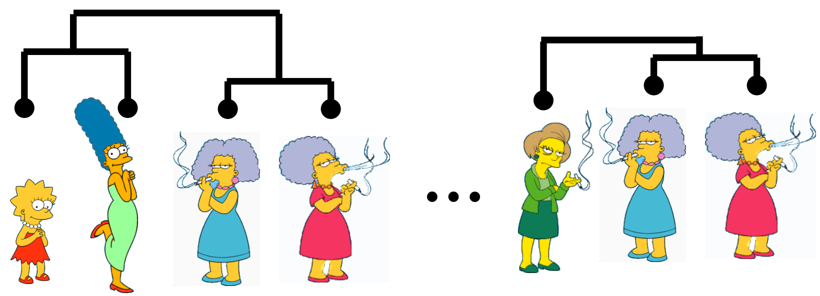


Choose the best

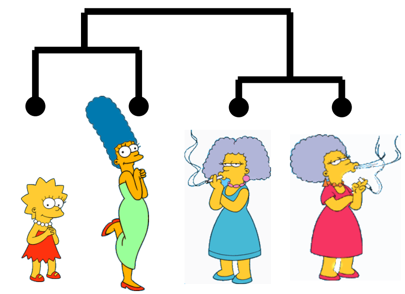
Bottom-Up (agglomerative):

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.

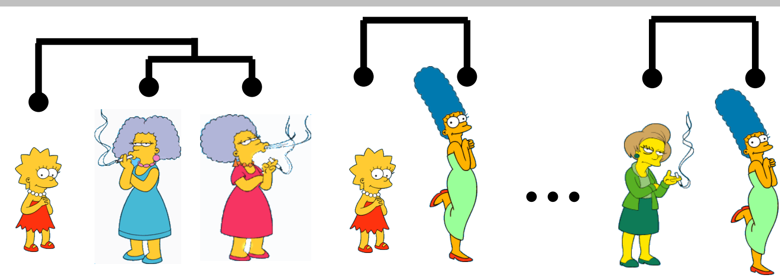
Consider all possible merges…



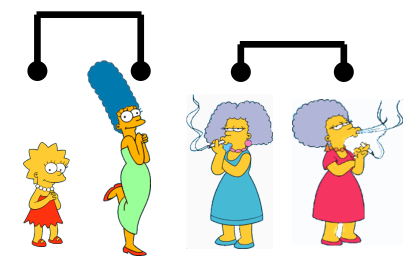
Choose the best



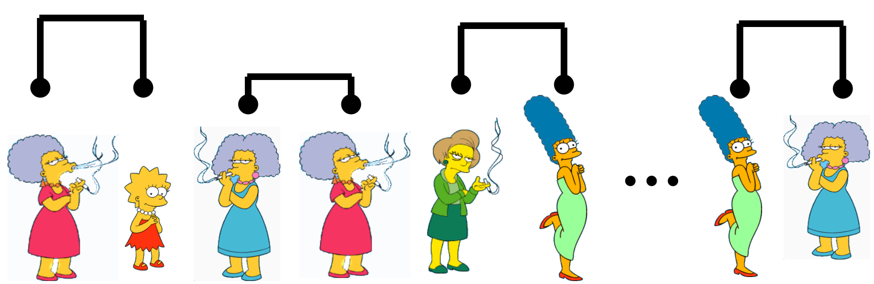
Consider all possible merges…



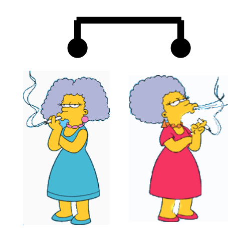
Choose the best



Consider all possible merges…

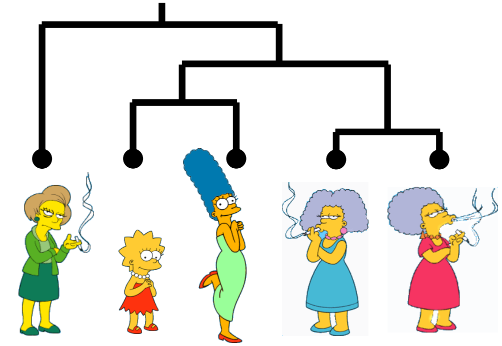


Choose the best

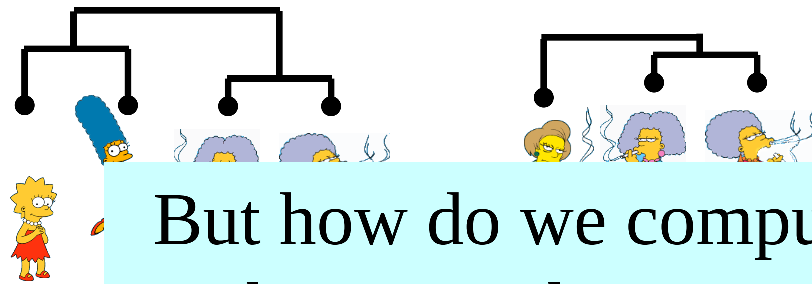


Bottom-Up (agglomerative):

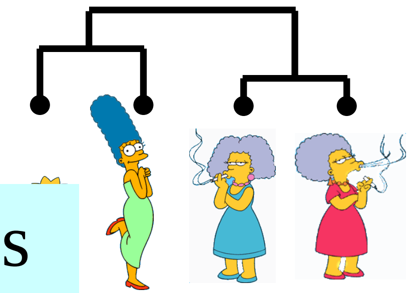
Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



Consider all possible merges…

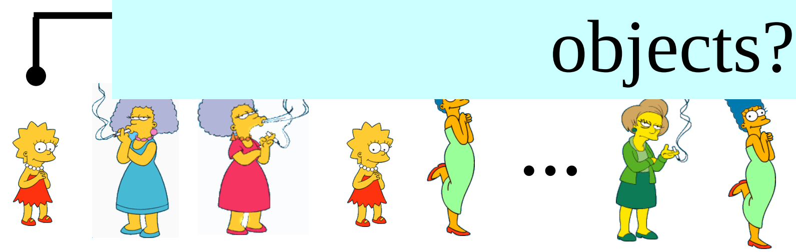


Choose

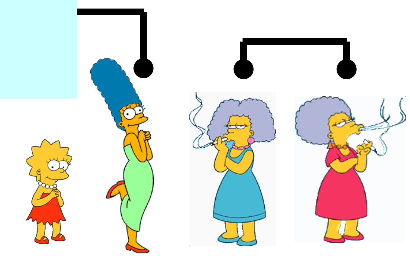


But how do we compute distances between clusters rather than objects?

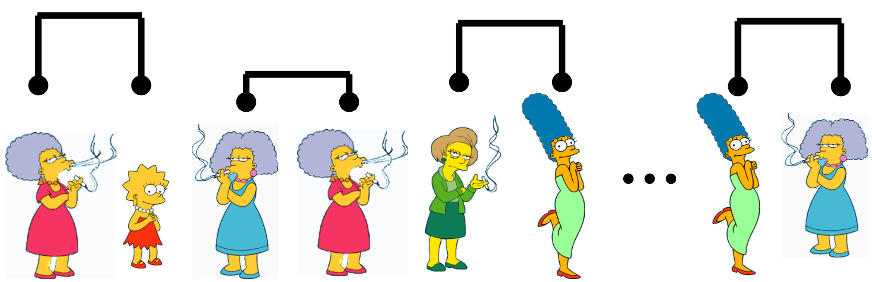
Consider all possible merges…



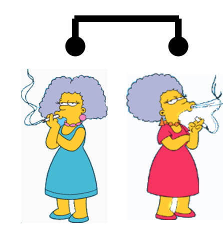
the best



Consider all possible merges…

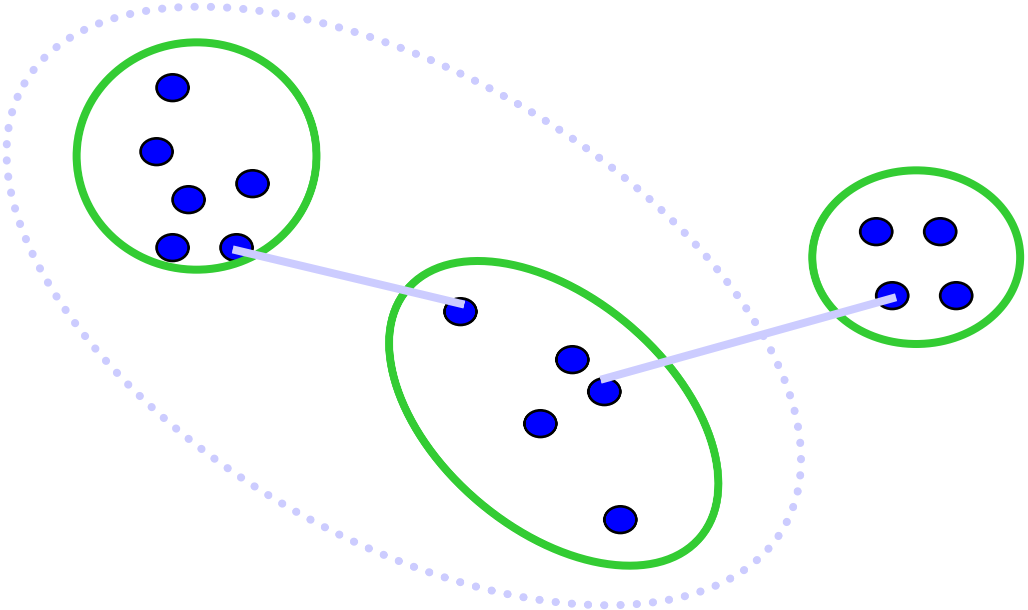


Choose the best



Computing distance between clusters: Single Link

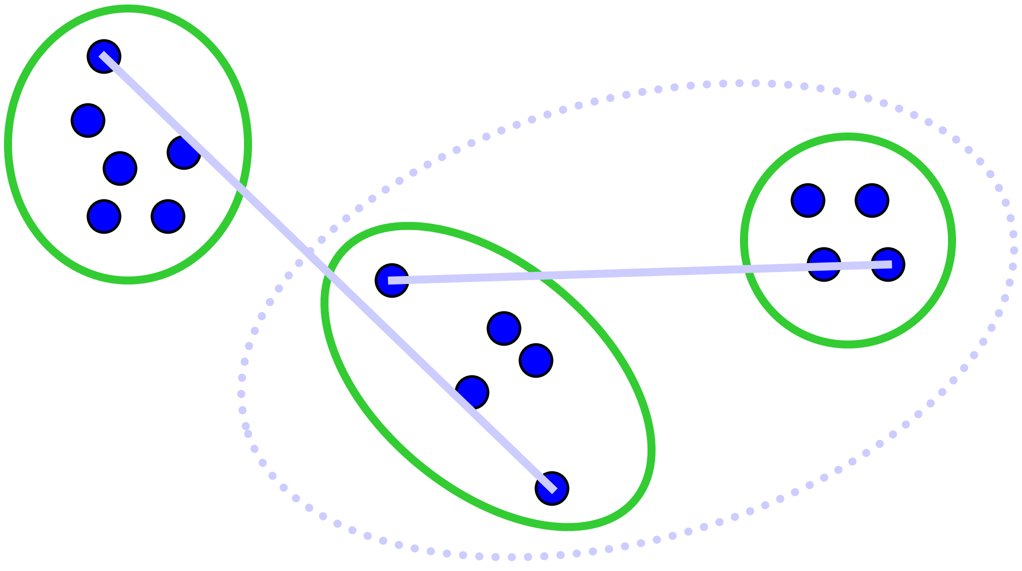
- cluster distance = distance of two closest members in each class



- Potentially long and skinny clusters

Computing distance between clusters: : Complete Link

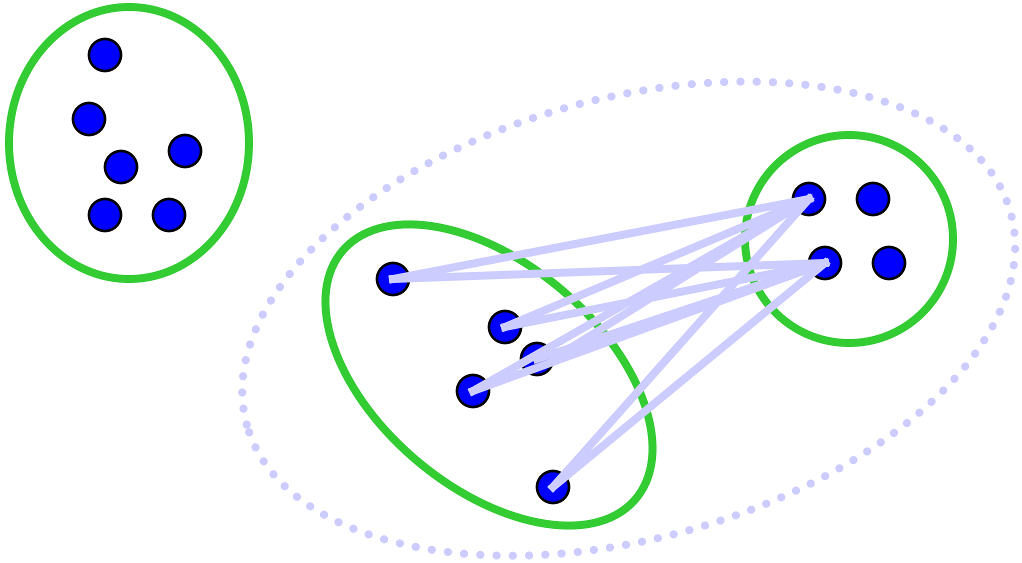
- cluster distance = distance of two farthest members



+ tight clusters

Computing distance between clusters: Average Link

- cluster distance = average distance of all pairs

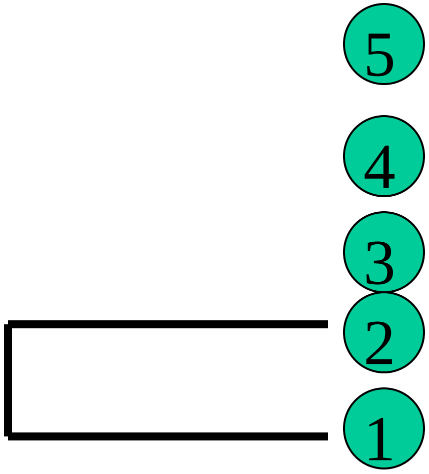


the most widely used measure

Robust against noise

Example: single link

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



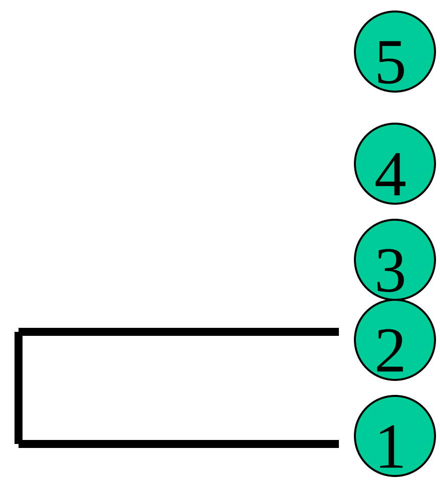
Example: single link

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

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

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

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

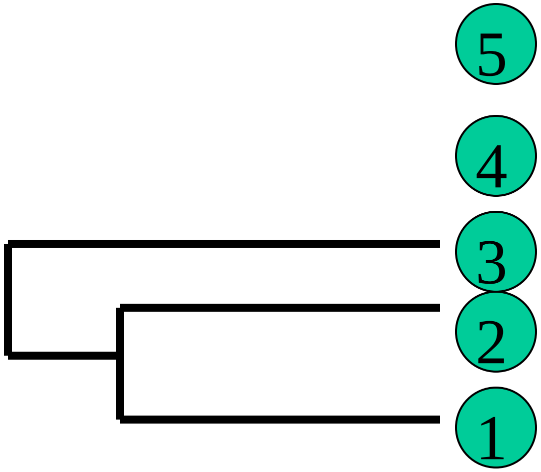


Example: single link

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

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

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



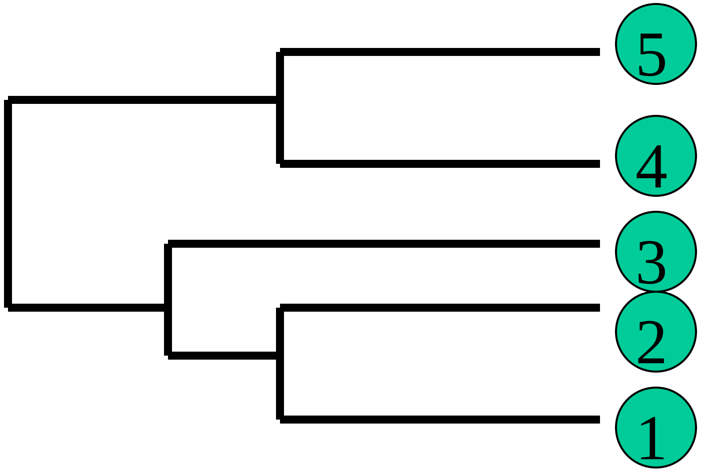
Example: single link

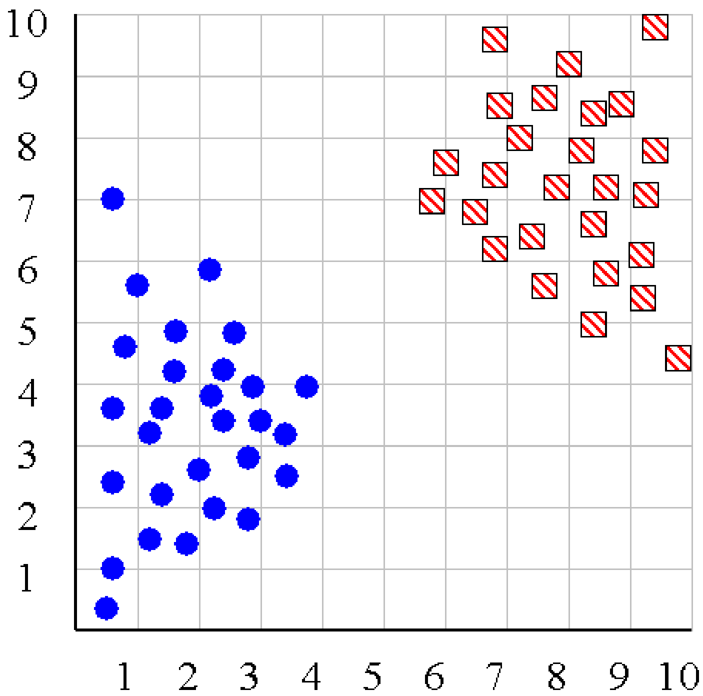
$$\begin{array}{cc} & \begin{array}{ccccc}1&2&3&4&5\end{array} \\ \begin{array}{c}1\\2\\3\\4\\5\end{array} & \begin{bmatrix}0&&&&\\2&0&&&\\6&3&0&&\\10&9&7&0&\\9&8&5&4&0\end{bmatrix}\end{array}$$

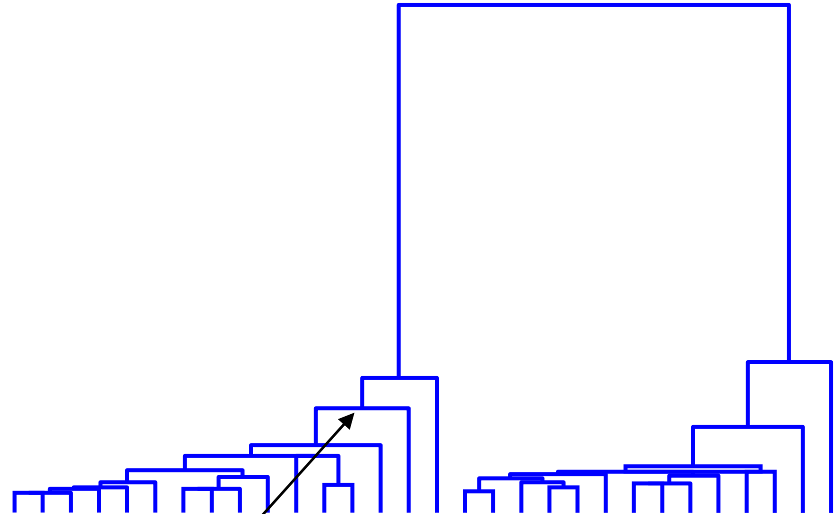
$$\begin{array}{cc} & \begin{array}{cccc}(1,2)&3&4&5\end{array} \\ \begin{array}{c}(1,2)\\3\\4\\5\end{array} & \begin{bmatrix}0&&&\\3&0&&\\9&7&0&\\8&5&4&0\end{bmatrix}\end{array}$$

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

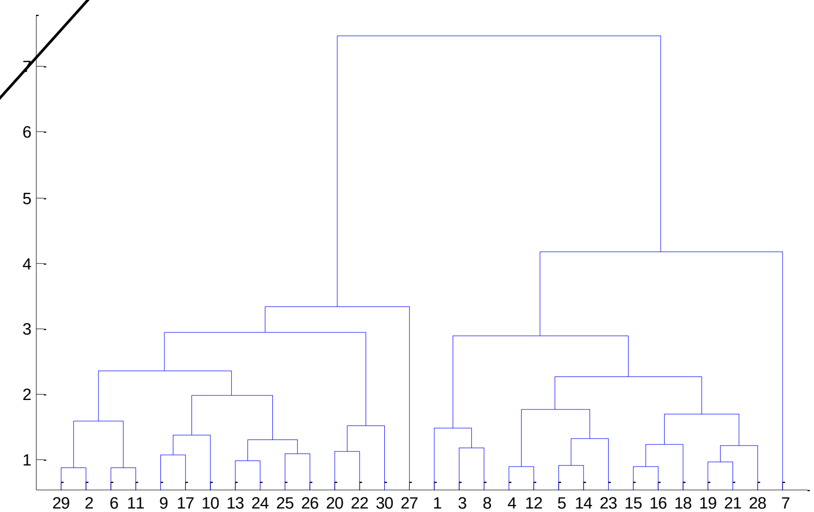
$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$







Single linkage



Average linkage

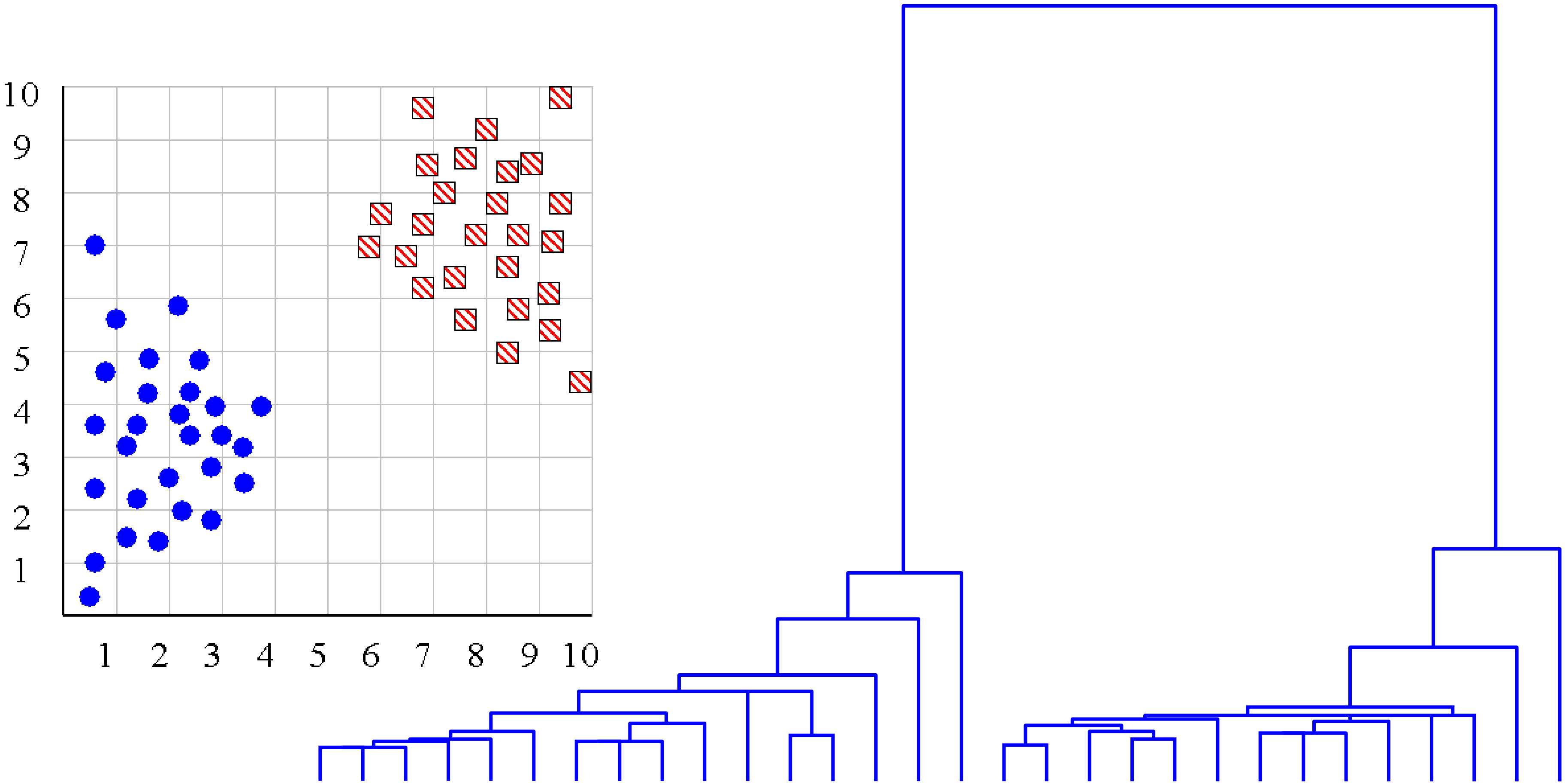
Height represents distance between objects / clusters

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
- They do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects.
- Like any heuristic search algorithms, local optima are a problem.
- Interpretation of results is (very) subjective.

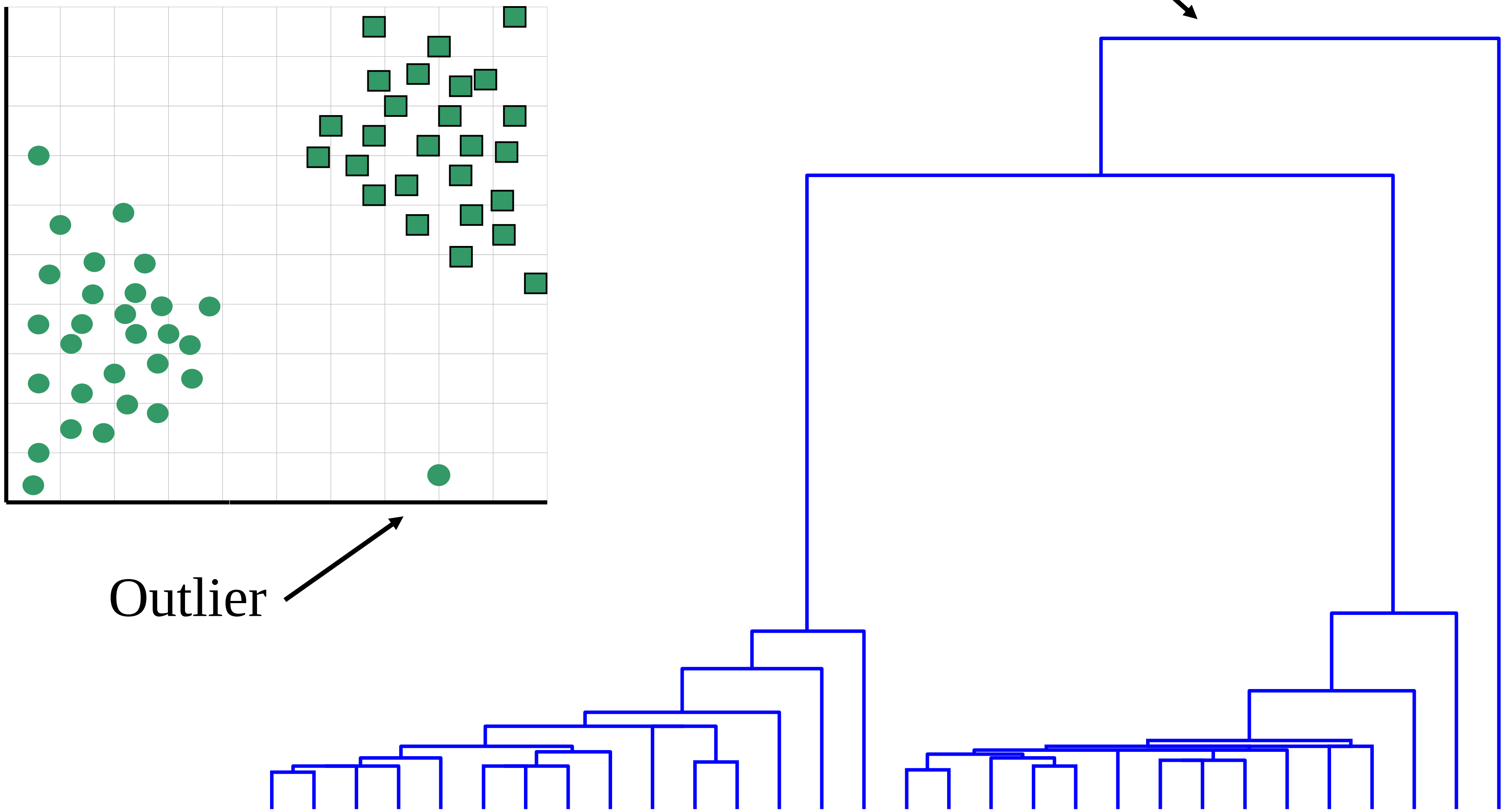
But what are the clusters?

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

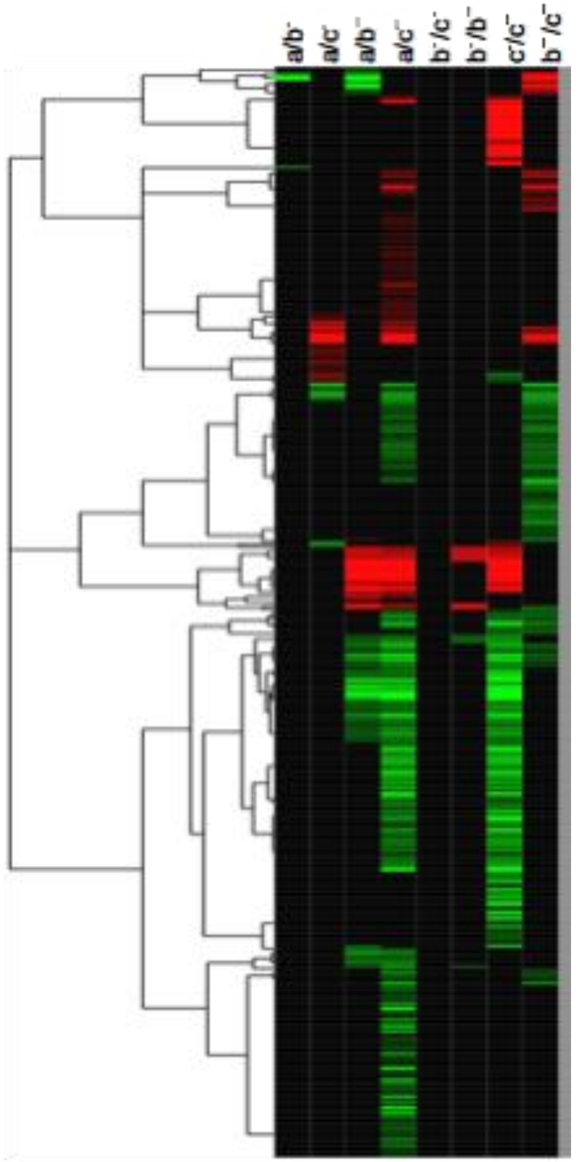


Outlier

Example: clustering genes

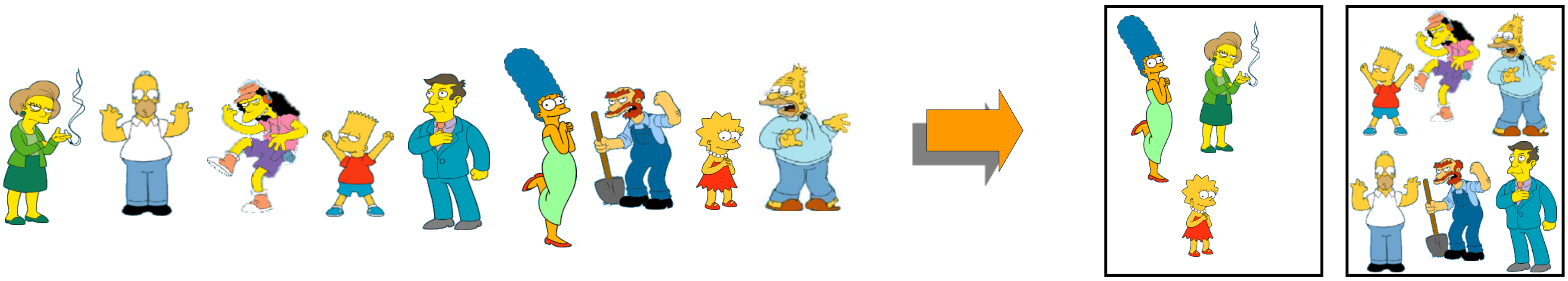
- Microarrays measures the activities of all genes in different conditions

- Clustering genes can help determine new functions for unknown genes



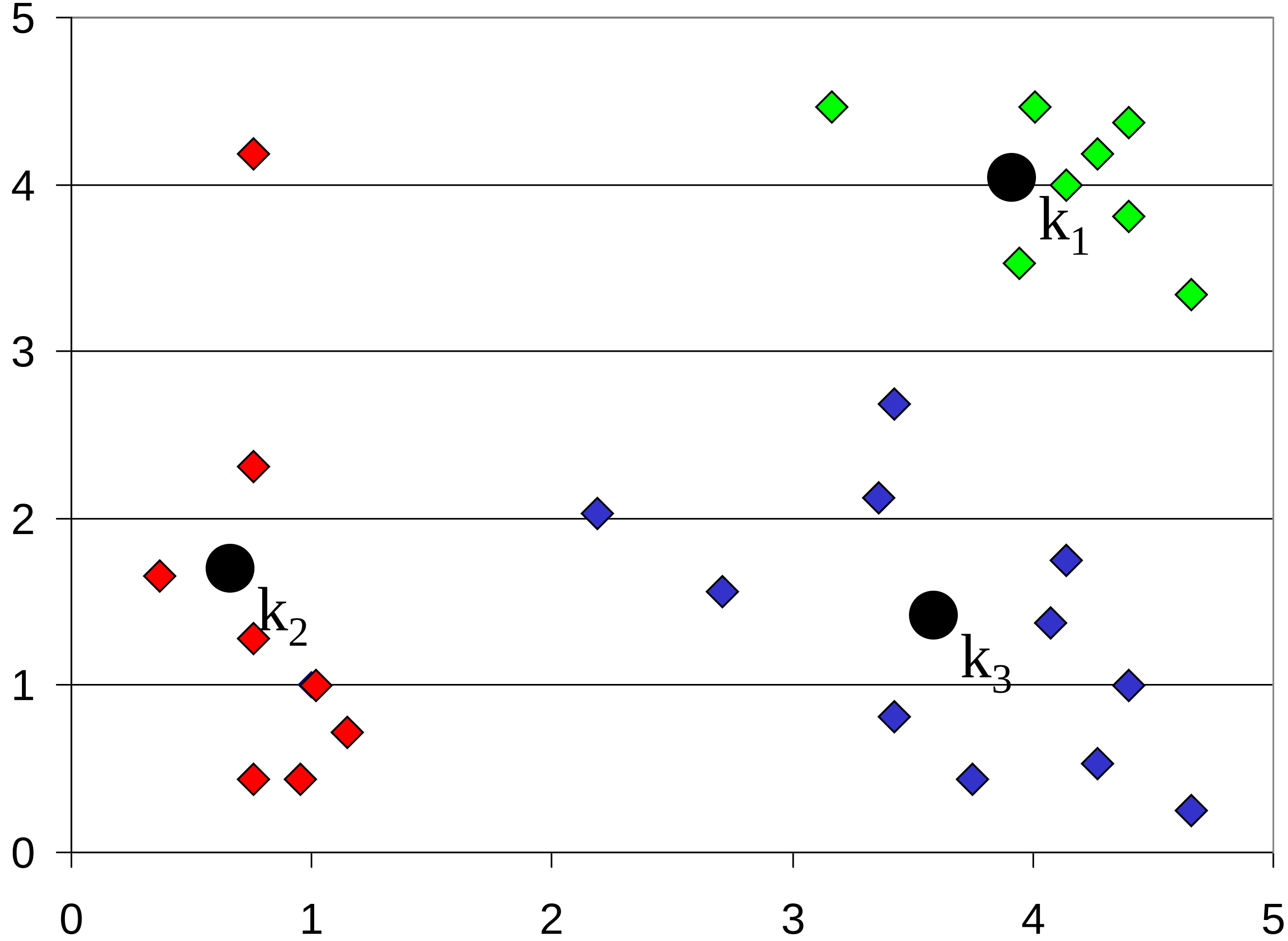
Partitional Clustering

- 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.

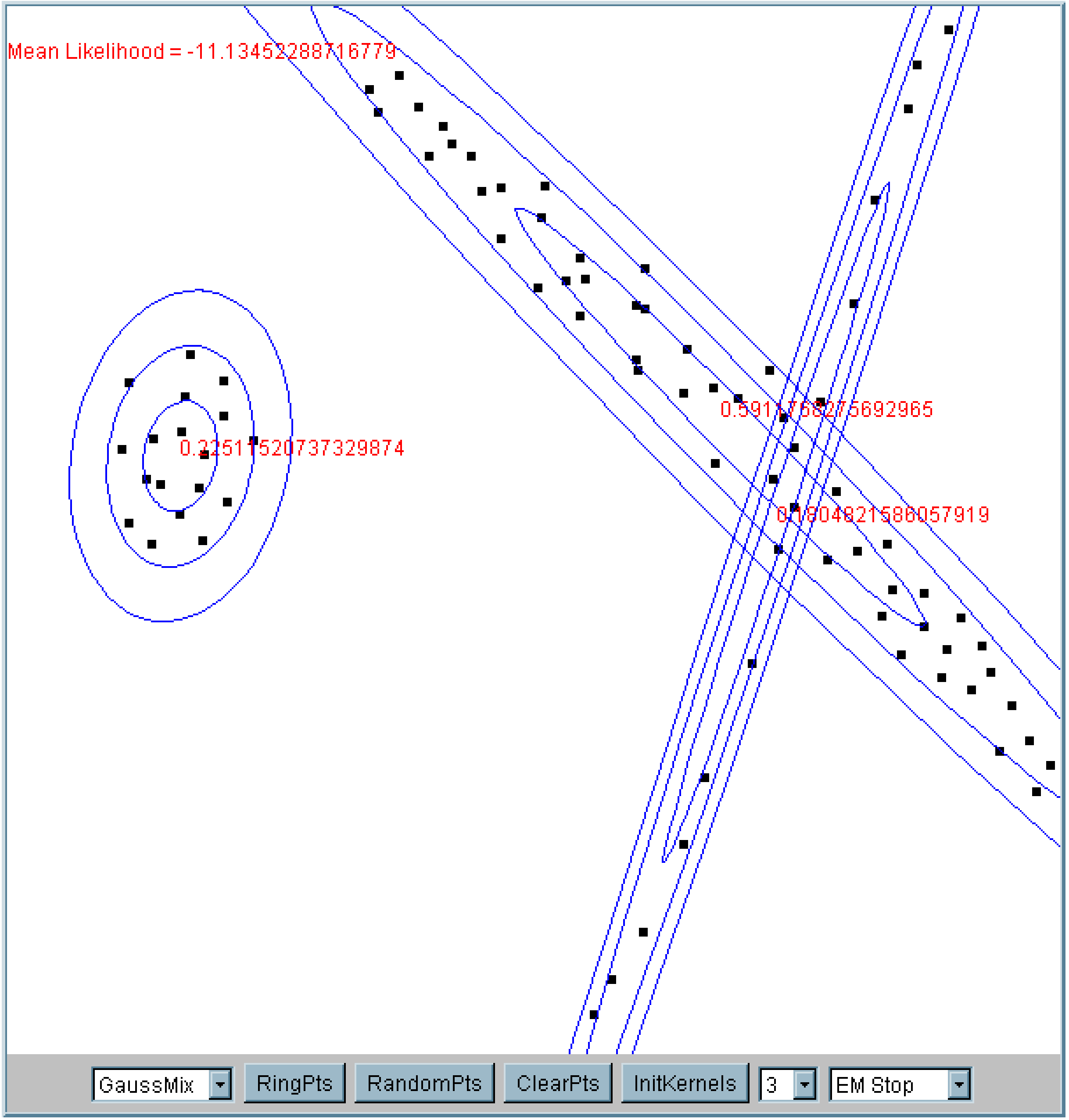


K-means Clustering: Finished!

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



Gaussian mixture clustering



Clustering methods: Comparison

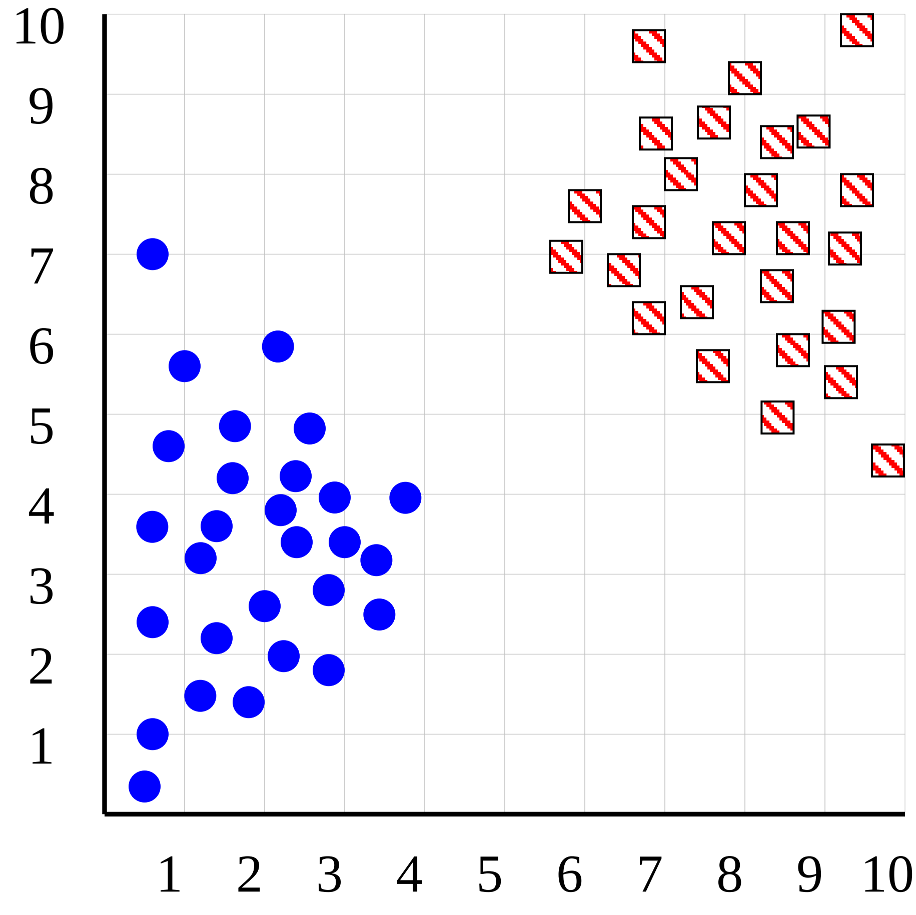
	Hierarchical	**K-means**	**GMM**
Running time	naively, O(N^3)	fastest (each iteration is linear)	fast (each iteration is linear)
Assumptions	requires a similarity / distance measure	strong assumptions	strongest assumptions
Input parameters	none	K (number of clusters)	K (number of clusters)
Clusters	subjective (only a tree is returned)	exactly K clusters	exactly K clusters

Outline

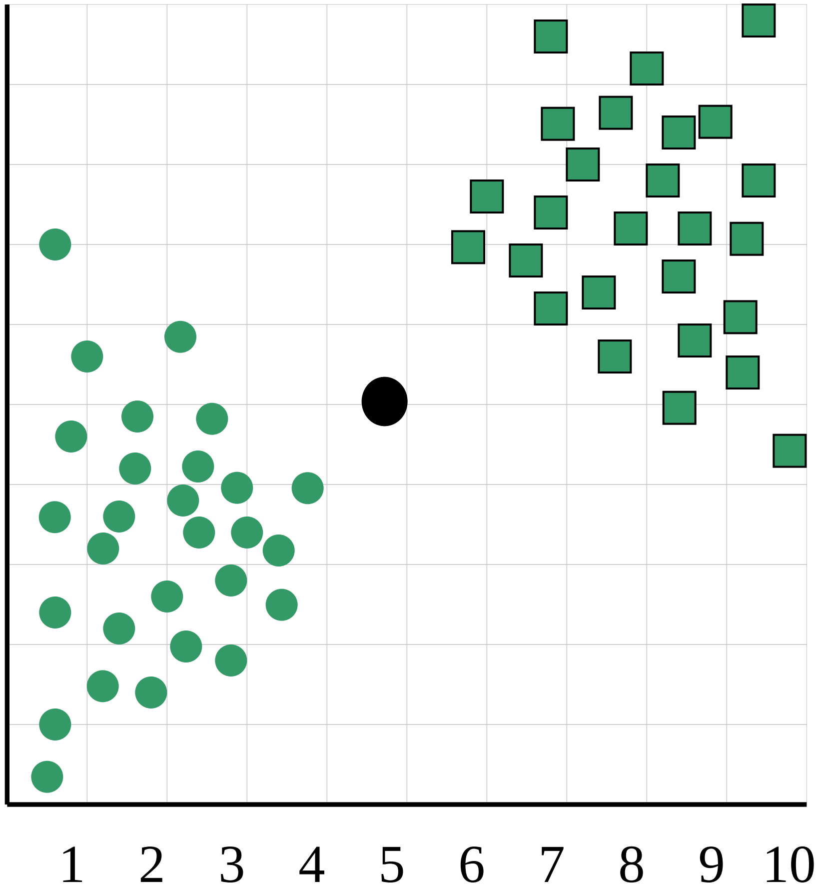
- Distance measure
- Hierarchical clustering
- Number of clusters

How can we tell the *right* number of clusters?

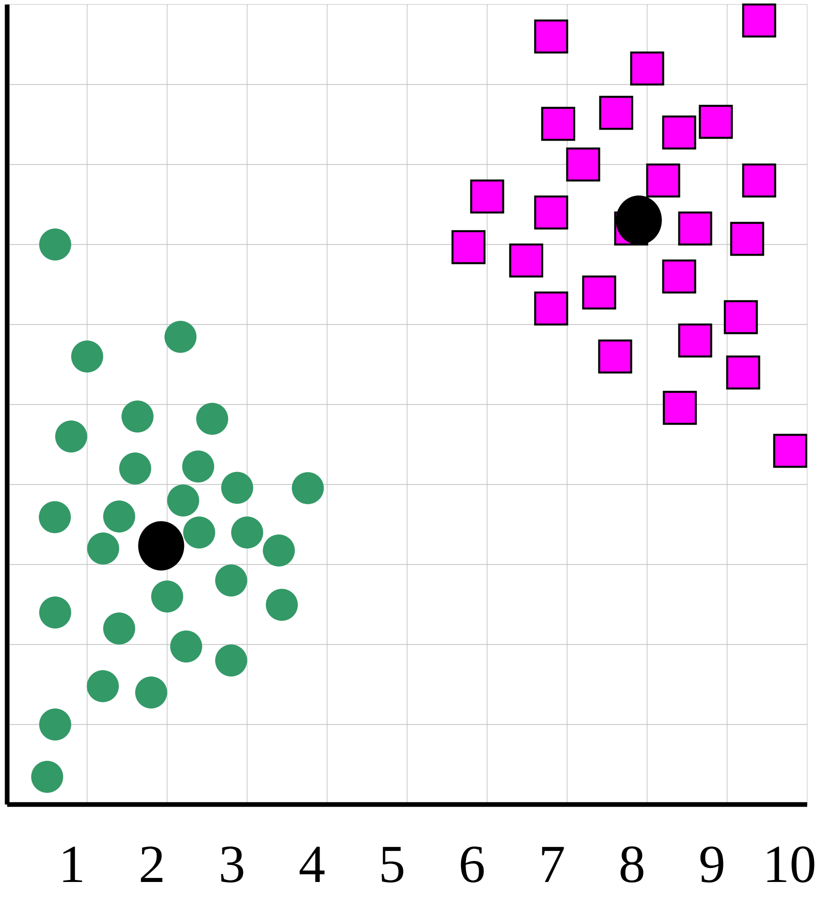
In general, this is a unsolved problem. However there are many approximate methods. In the next few slides we will see an example.



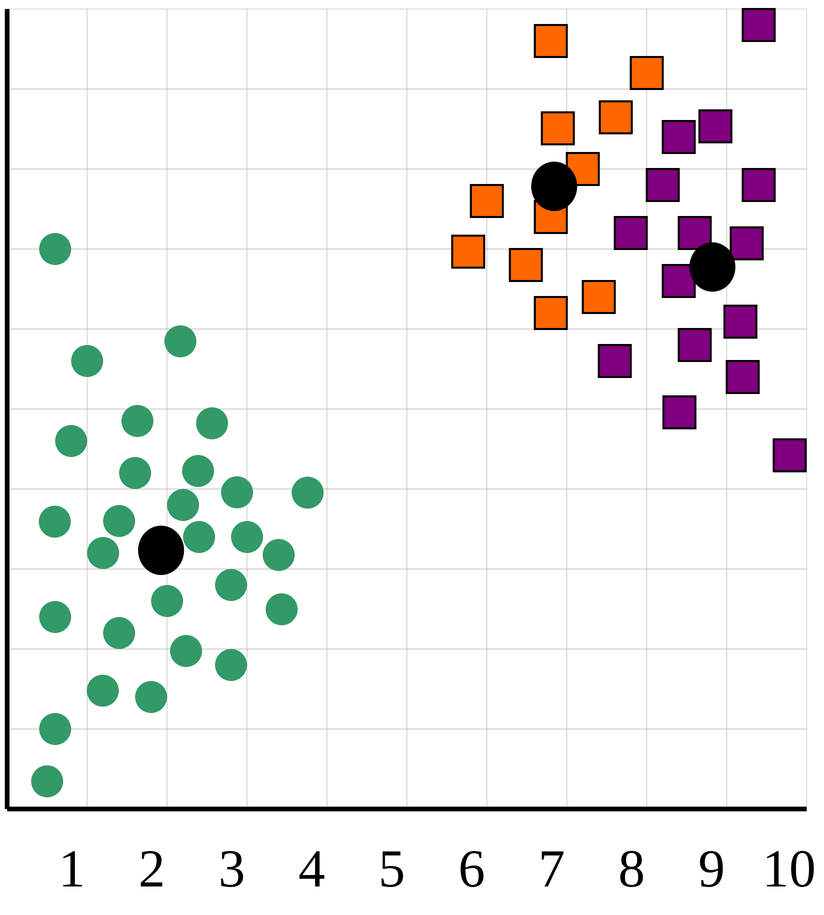
When k = 1, the objective function is 873.0



When k = 2, the objective function is 173.1

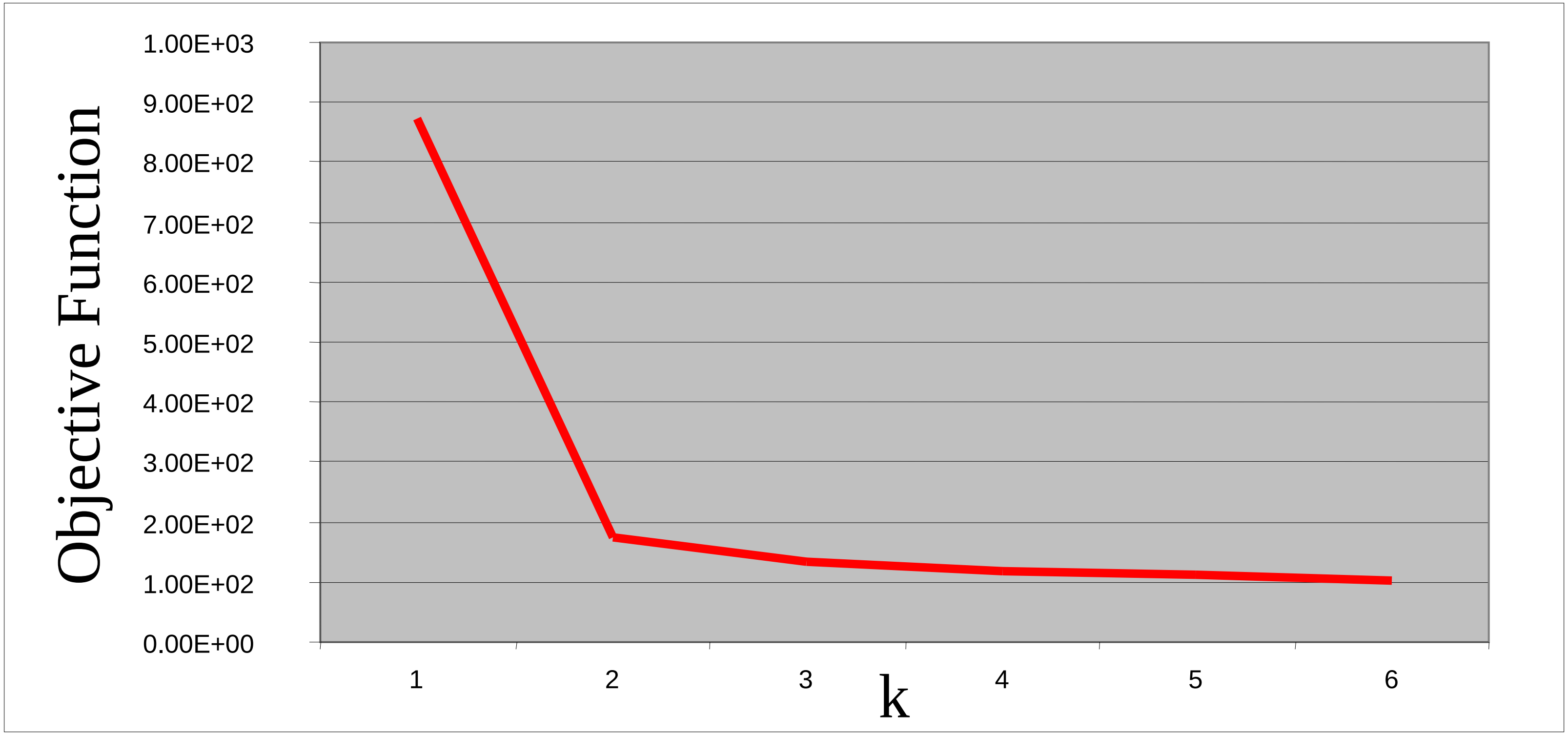


When k = 3, the objective function is 133.6



We can plot the objective function values for k equals 1 to 6…

The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.

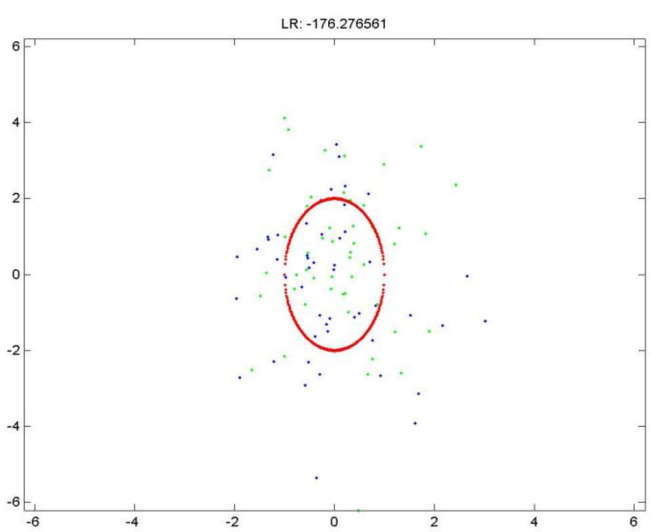


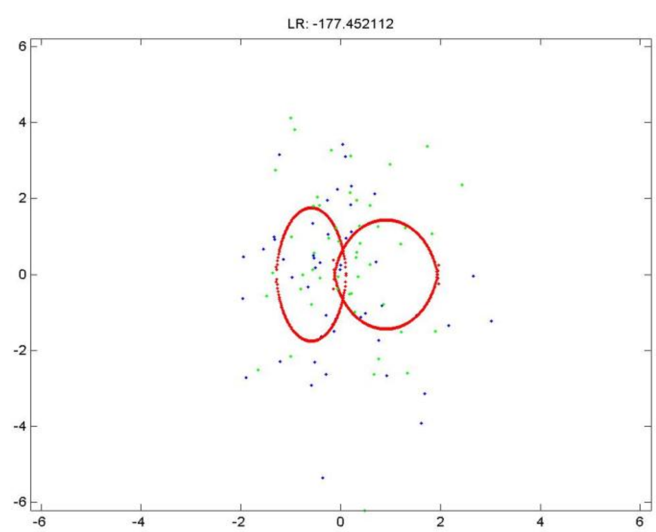
Note that the results are not always as clear cut as in this toy example

Cross validation

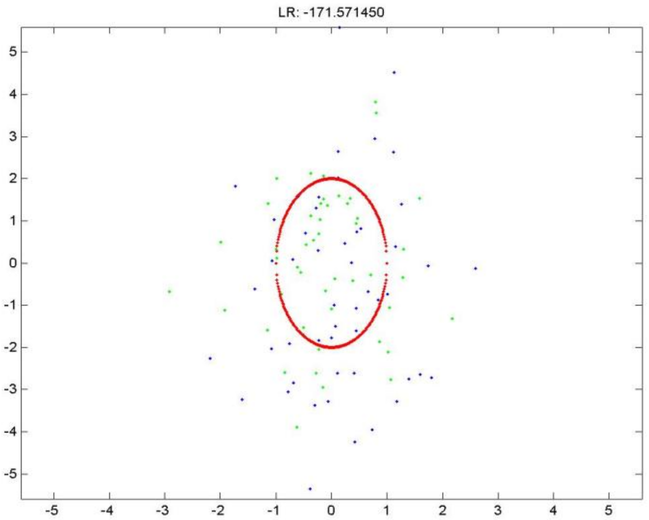
- We can also use cross validation to determine the correct number of classes
- Recall that GMMs is a generative model. We can compute the likelihood of the left out data to determine which model (number of clusters) is more accurate

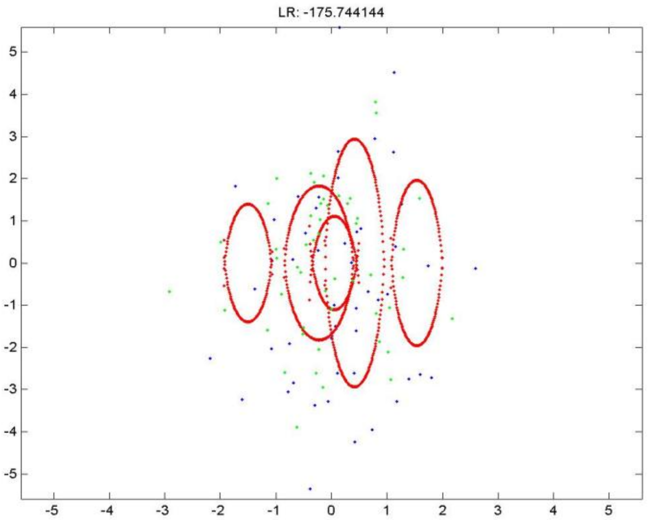
$$p(x_1 \cdots x_n \mid \theta) = \prod_{j=1}^{n} \left(\sum_{i=1}^{k} p(x_j \mid C = i) w_i \right)$$





Cross validation





What you should know

- Why is clustering useful
- What are the different types of clustering algorithms
- What are the assumptions we are making for each, and what can we get from them
- Unsolved issues: number of clusters, initialization, etc.