Clustering: Hierarchical Clustering and K-Means Clustering

Machine Learning 10-601B

Seyoung Kim

Many of these slides are derived from William Cohen, Ziv Bar-Joseph, Eric Xing. Thanks!

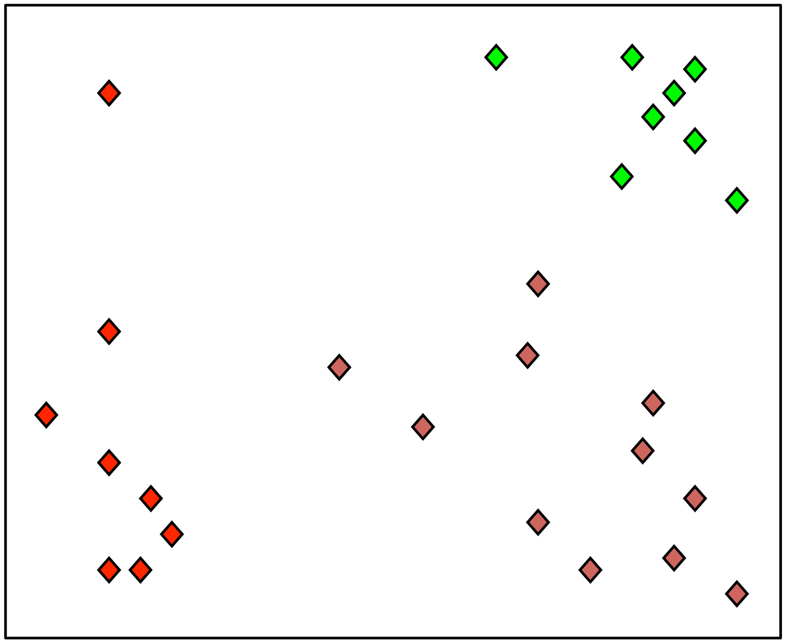
Two Classes of Learning Problems

- Supervised learning = learning from labeled data, where class labels (in classification) or output values (regression) are given
  - Train data: (X, Y) for inputs X and labels Y

- Unsupervised learning = learning from unlabeled, unannotated data
  - Train data: X for unlabeled data
  - we do not have a teacher that provides examples with their labels

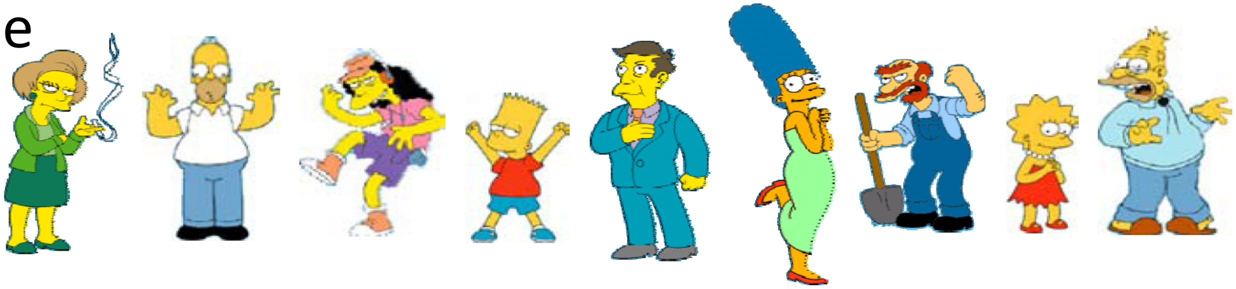
What is Clustering?

- 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?



Examples

- People



- Images

- Language

Piotr Pyotr Petros Pietro Pedro Pierre Piero Peter Peder Peka Peadar

- species

Unsupervised learning

- Clustering methods
  - Non-probabilistic method
    - Hierarchical clustering
    - K means algorithm
  - Probabilistic method
    - Mixture model

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

What is Similarity?

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

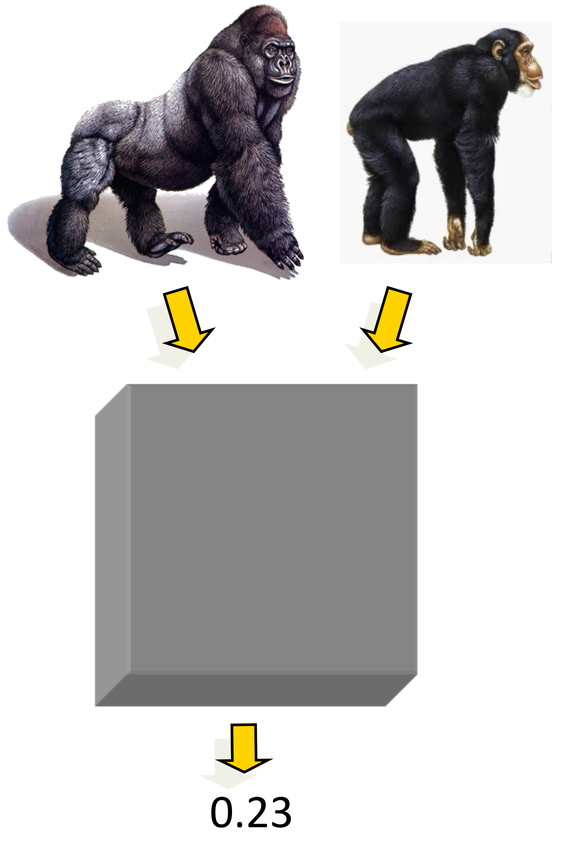
Webster's Dictionary

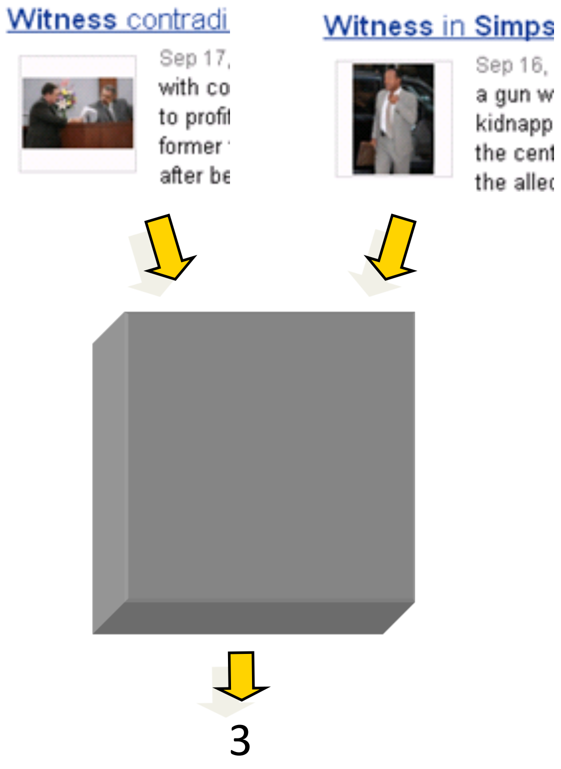
Similarity is hard to define, but… *"We know it when we see it"*

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 $D(O_1,O_2)$

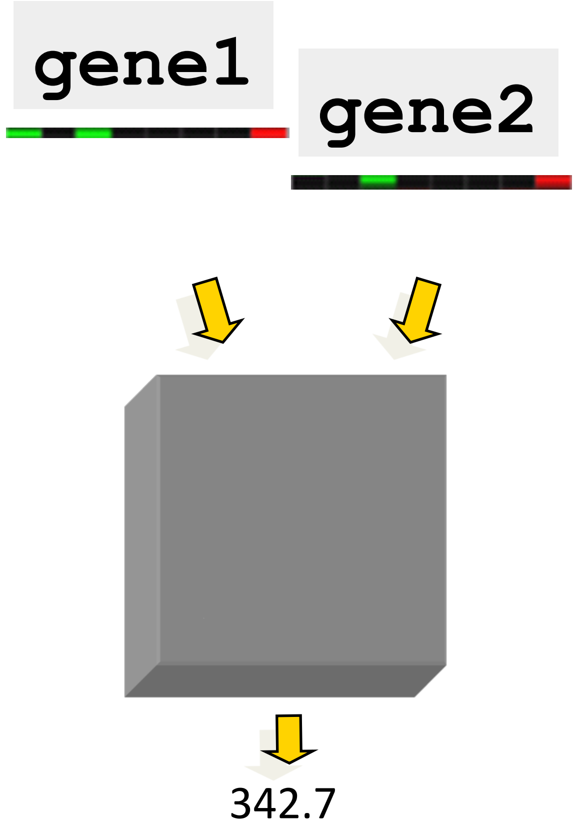



Witness contradi

Sep 17,
with co
to profi
former
after be

Witness in Simps

Sep 16,
a gun w
kidnapp
the cent
the allec



What properties should a distance measure have?

- D(A,B) = D(B,A) *Symmetry*

- D(A,A) = 0 *Constancy of Self-Similarity*

- D(A,B) = 0 IIf A= B *Positivity Separation*

- D(A,B) ≤ D(A,C) + D(B,C) *Triangular Inequality*

Intuitions behind desirable distance measure properties

- $D(A,B) = D(B,A)$ *Symmetry*
  - *Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"*

- $D(A,A) = 0$ *Constancy of Self-Similarity*
  - *Otherwise you could claim "Alex looks more like Bob, than Bob does"*

- $D(A,B) = 0$ IIf A= B *Positivity Separation*
  - *Otherwise there are objects in your world that are different, but you cannot tell apart.*

- $D(A,B) \leq D(A,C) + D(B,C)$ *Triangular Inequality*
  - *Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"*

Distance Measures

- Suppose two object *x* and *y* both have *p* features

$$x = (x_1, x_2, \cdots, x_p)$$

$$y = (y_1, y_2, \cdots, y_p)$$

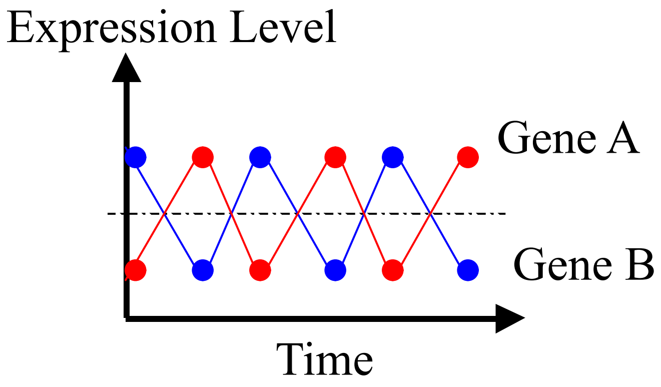
- Euclidean distance

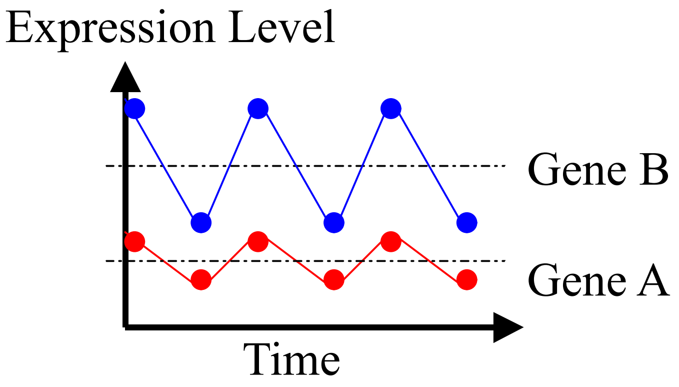
$$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} | x_i - y_i |^2}$$

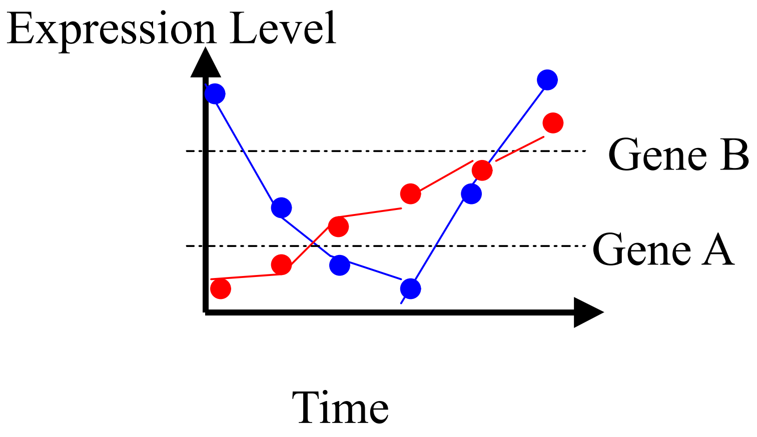
- Correlation coefficient

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

Similarity Measures: Correlation Coefficient





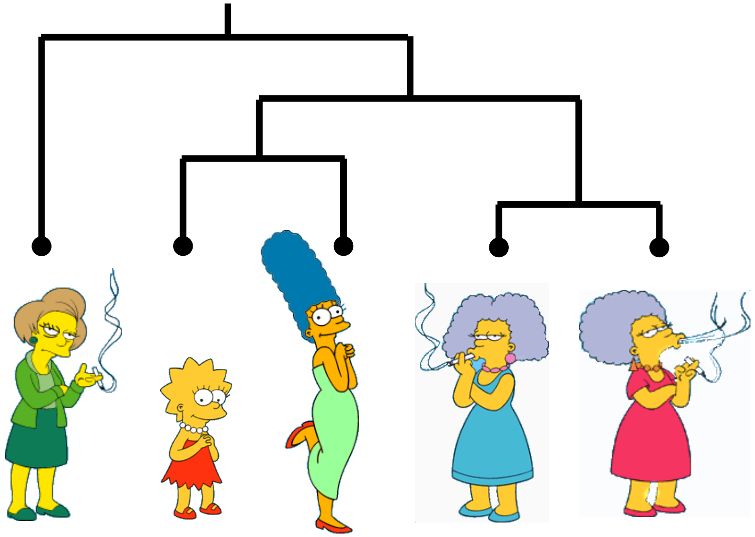


Two Types of Clustering

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

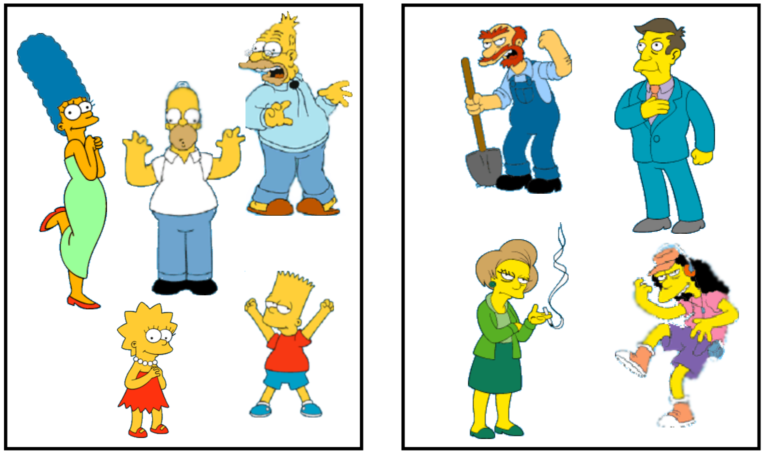
Bottom up or top down

Hierarchical



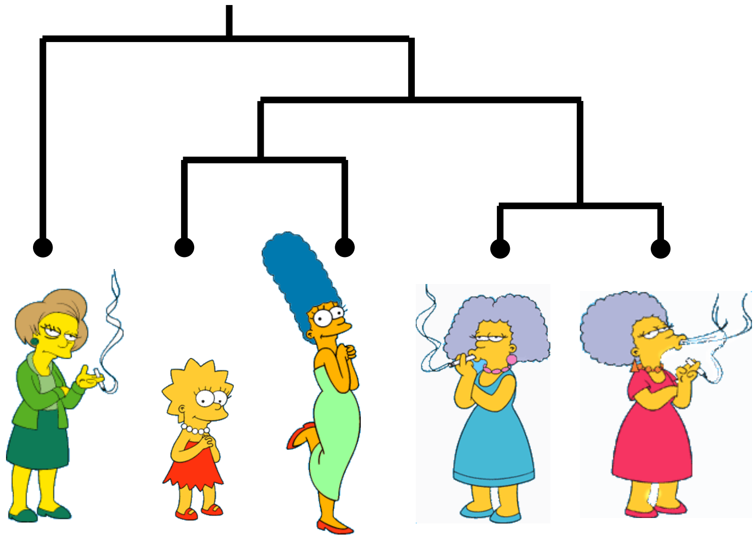
Top down

Partitional

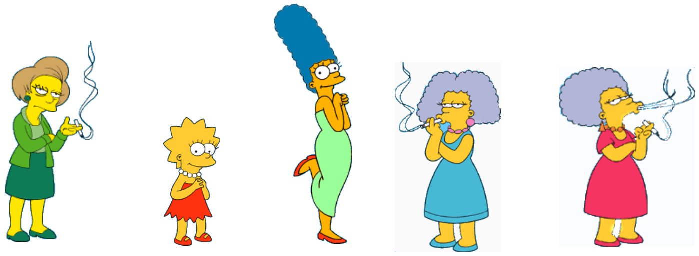


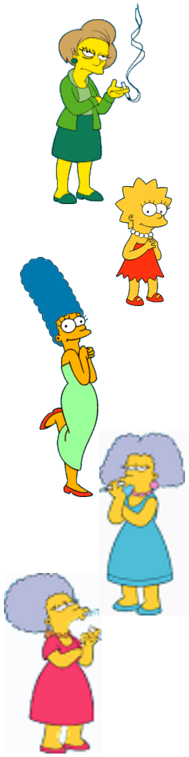
(How-to) Hierarchical Clustering

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.

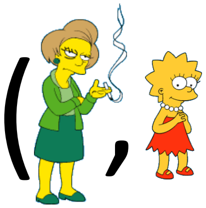


We begin with a distance matrix which contains the distances between every pair of objects in our database.





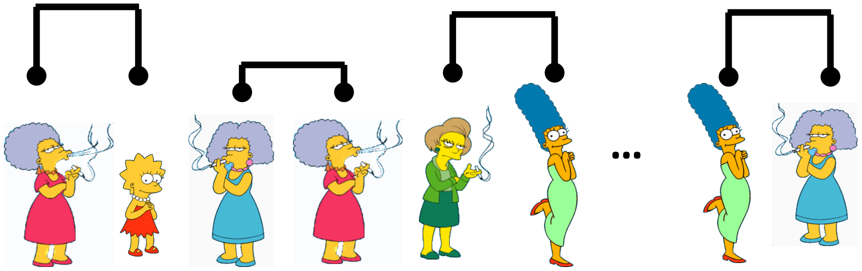
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

D() = 8

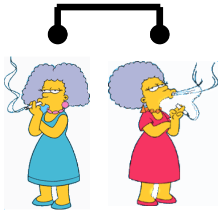
D( ) = 1

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…

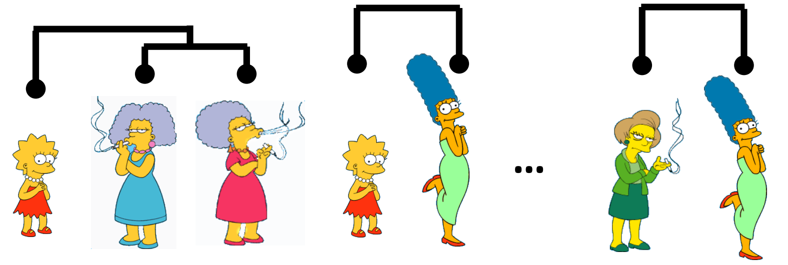


Choose the best

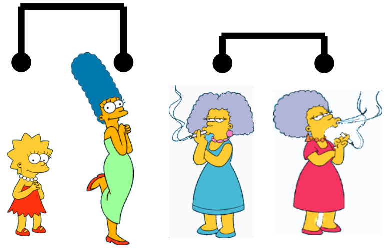


Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

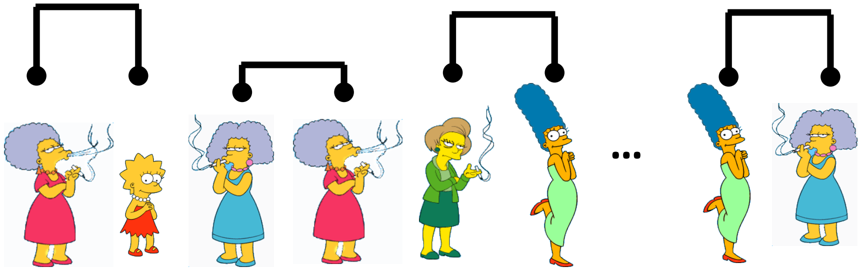
Consider all possible merges…



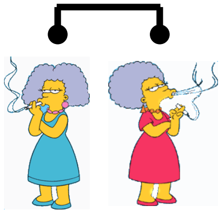
Choose the best



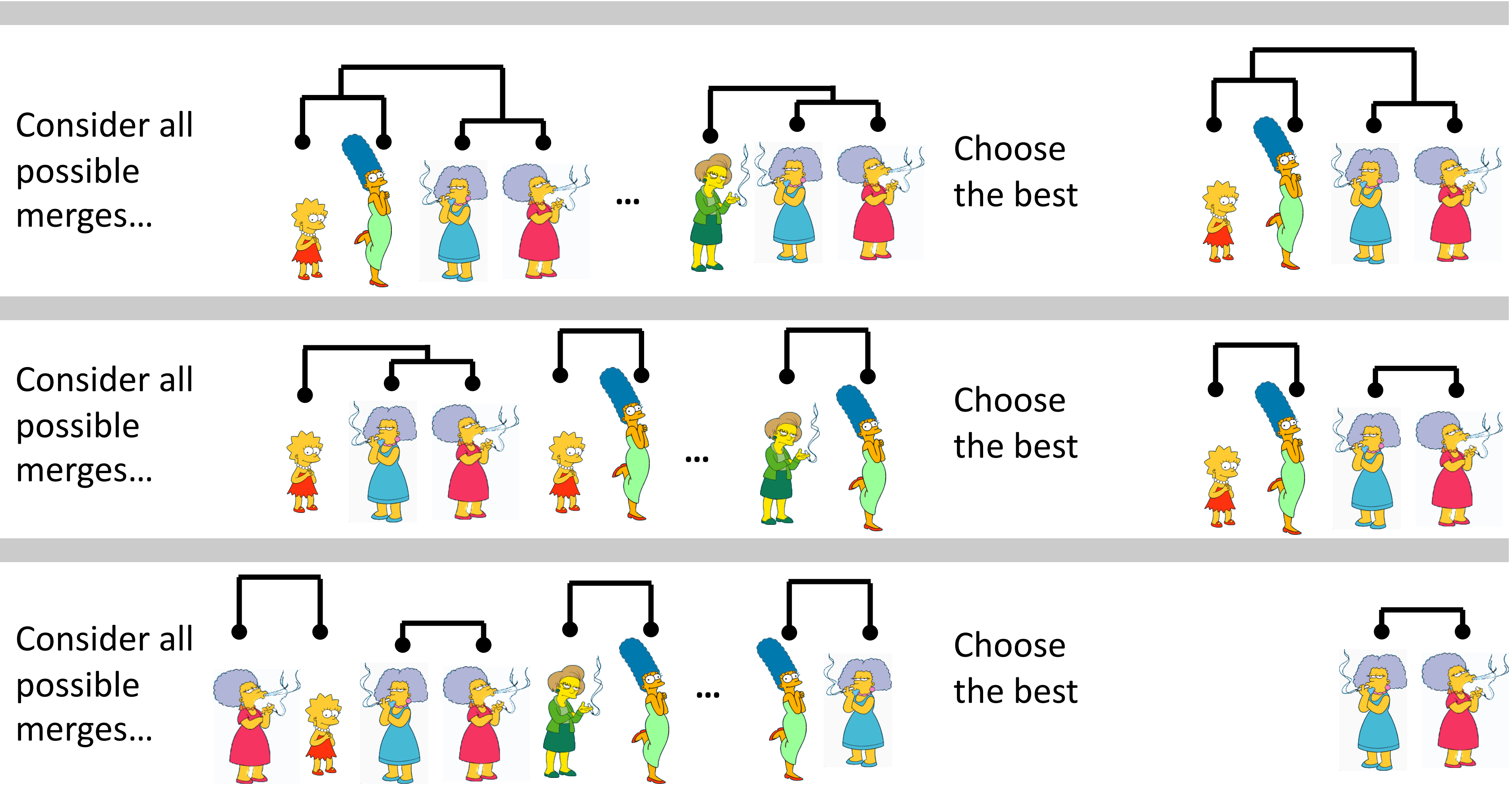
Consider all possible merges…



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…

…

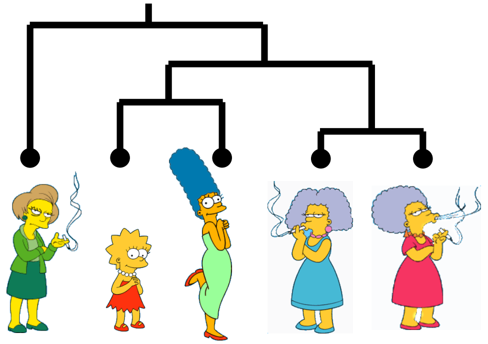
Choose the best

Consider all possible merges…

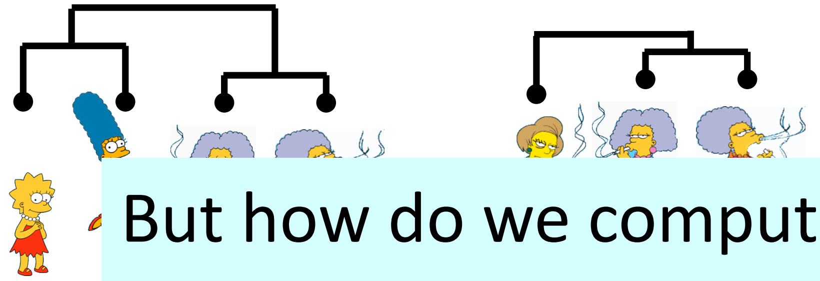
…

Choose the best

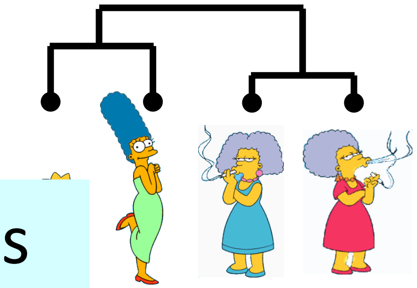
Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



Consider all possible merges…

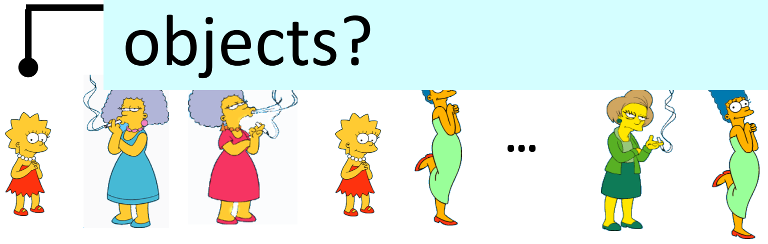


Choose

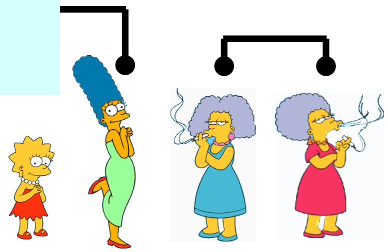


But how do we compute distances between clusters rather than objects?

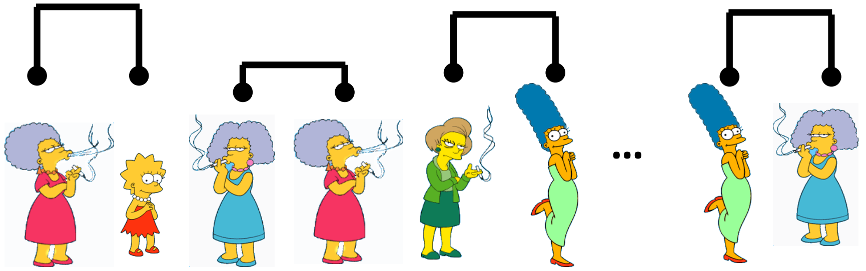
Consider all possible merges…



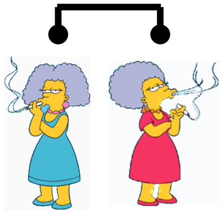
the best



Consider all possible merges…

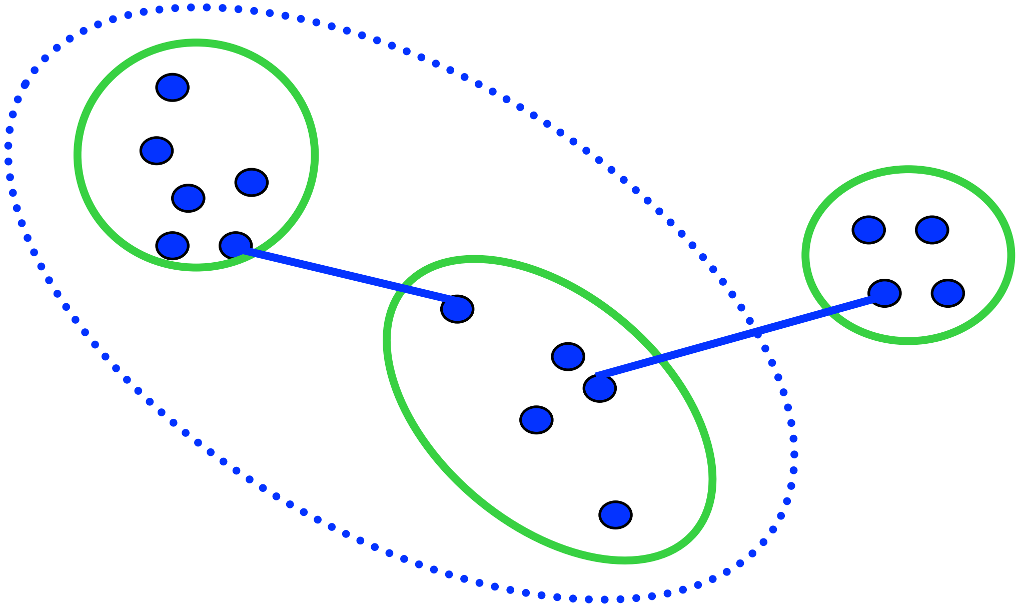


Choose the best



Computing distance between clusters: Single Link

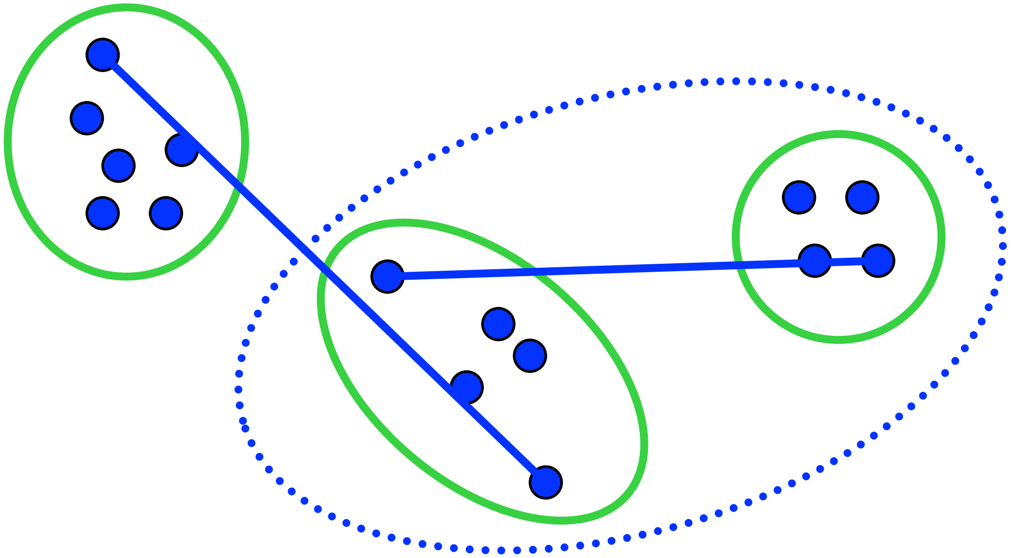
- cluster distance = distance of two closest members in each class



- Potentially long and skinny clusters

Computing distance between clusters: Complete Link

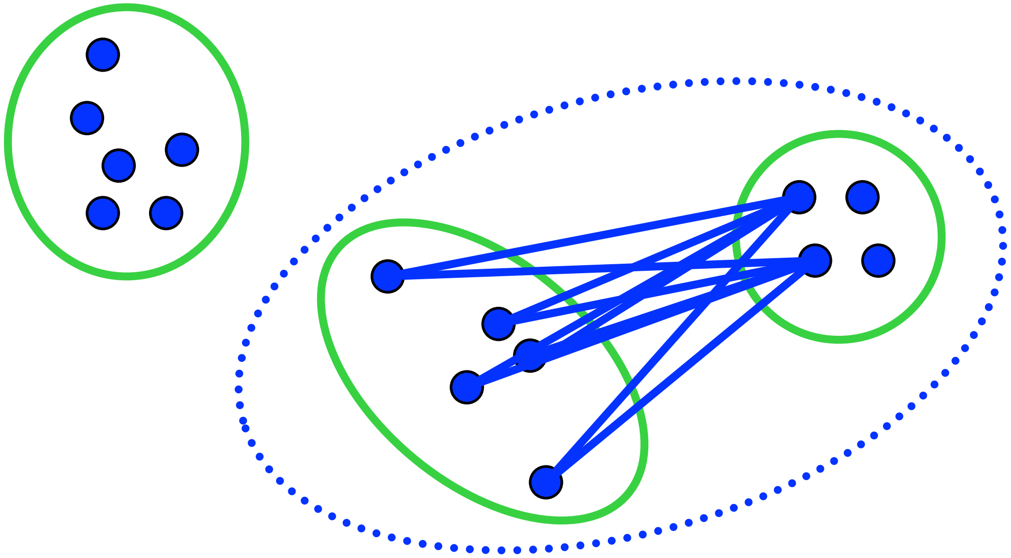
- cluster distance = distance of two farthest members



+ tight clusters

Computing distance between clusters: Average Link

- cluster distance = average distance of all pairs

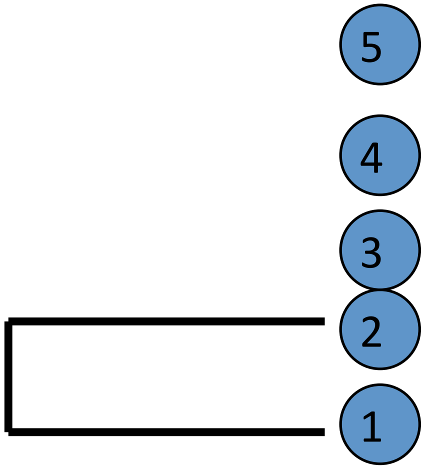


the most widely used measure

Robust against noise

Example: single link

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



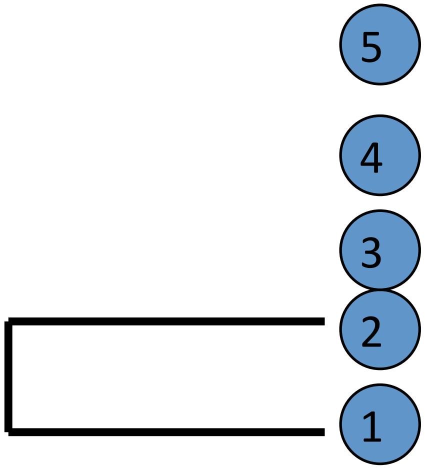
Example: single link

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

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

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

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

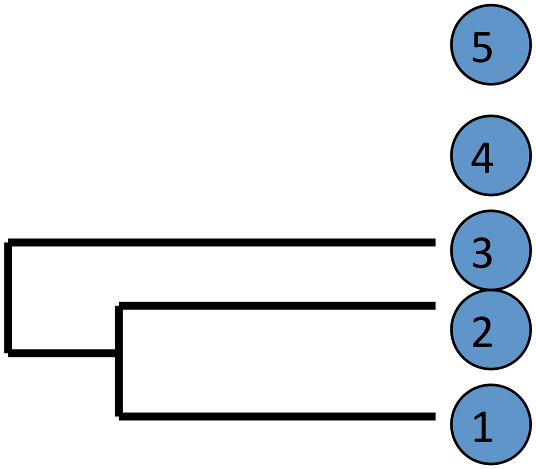


Example: single link

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

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

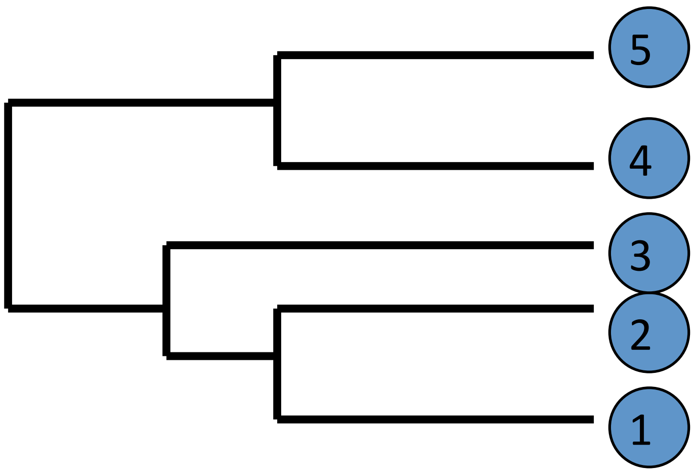
$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5$$

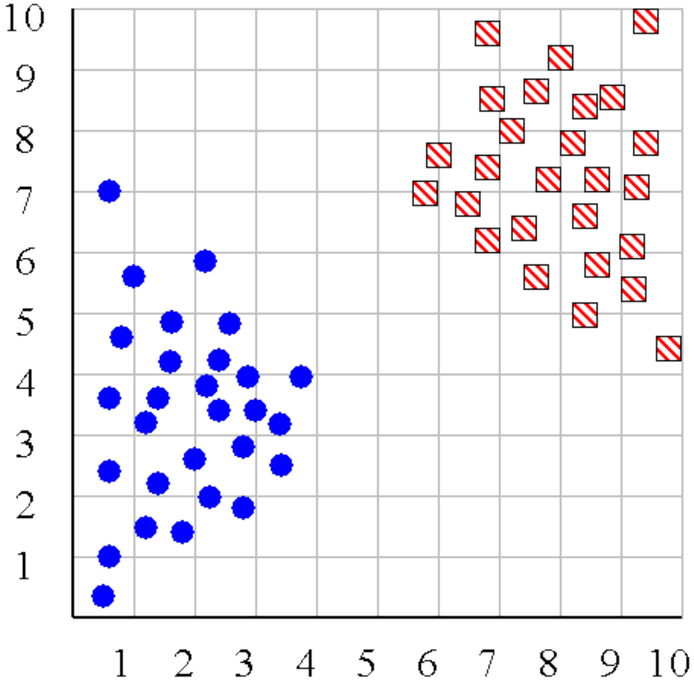


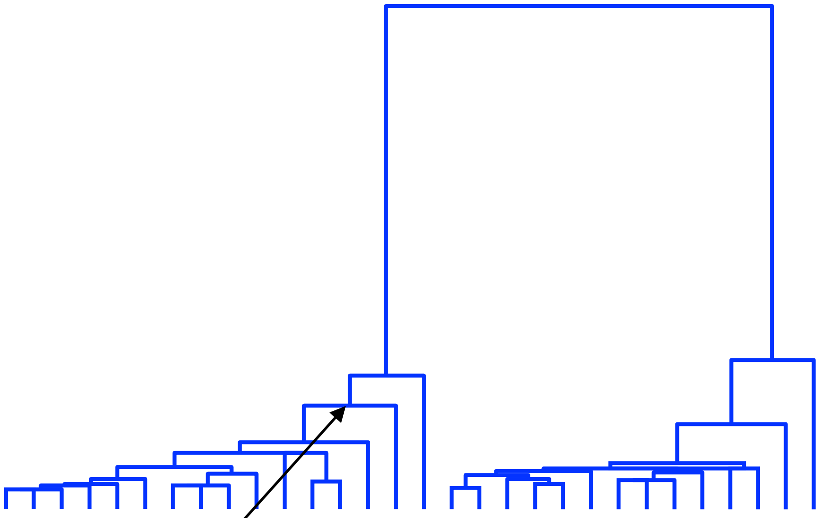
Example: single link

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

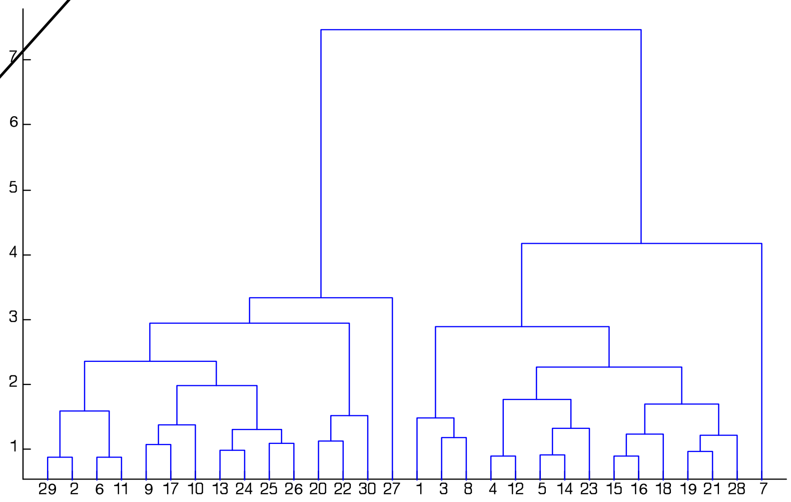
$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$







Single linkage



Average linkage

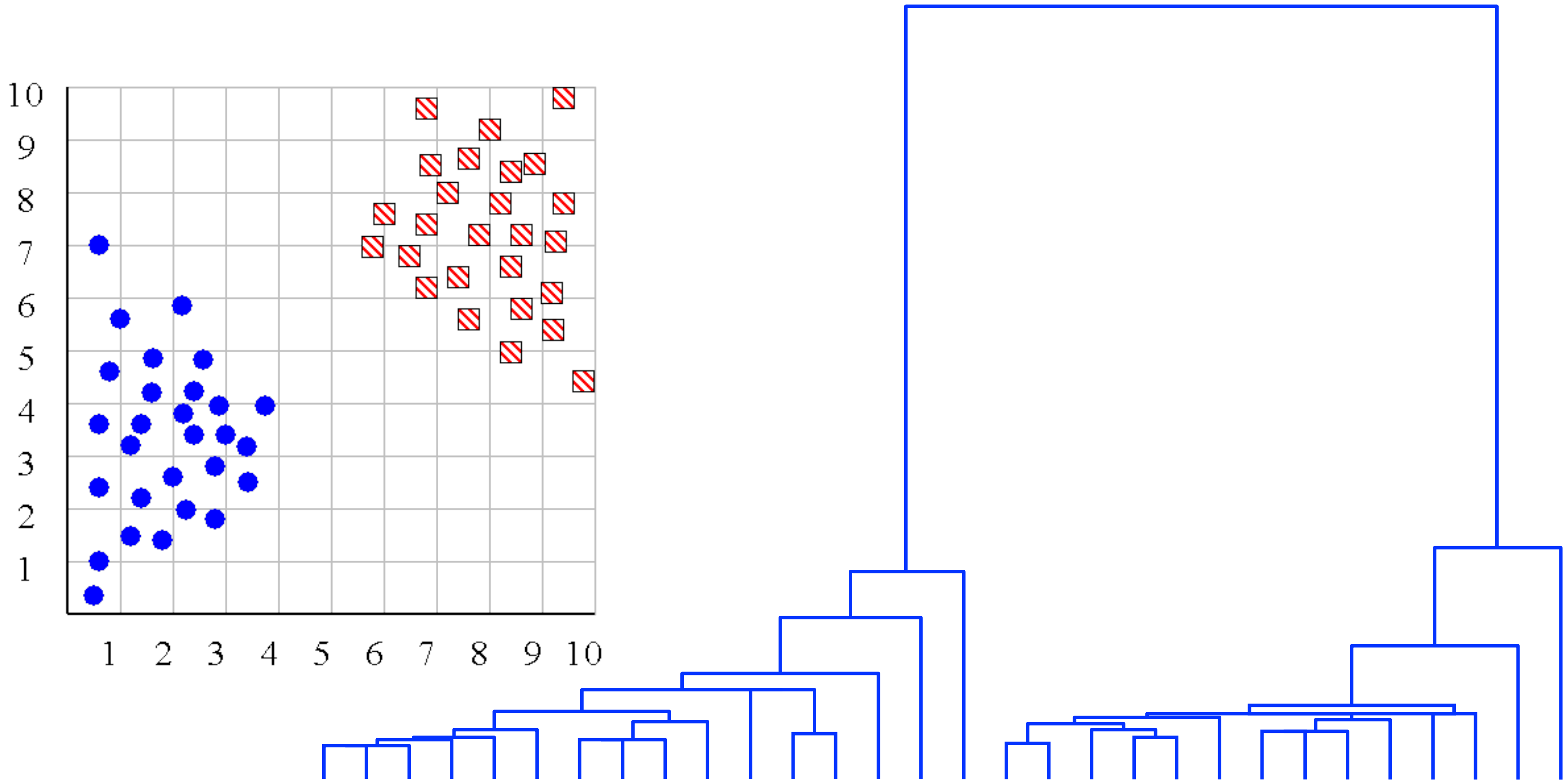
Height represents distance between objects / clusters

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.

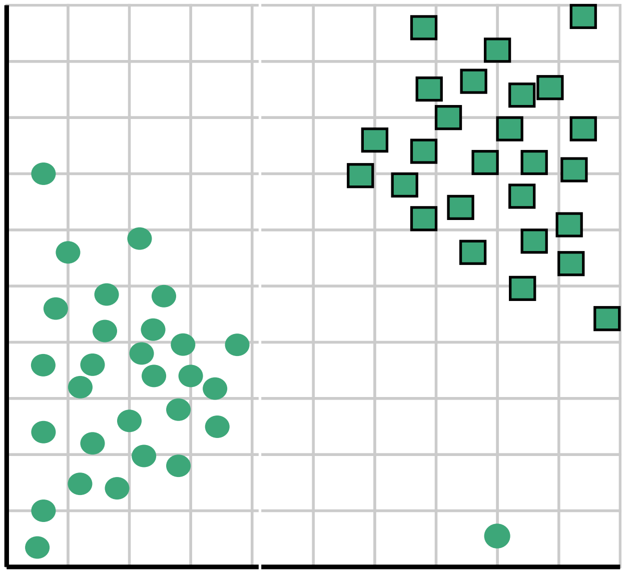
But what are the clusters?

In some cases we can determine the “correct” number of clusters. However, things are rarely this clear cut, unfortunately.

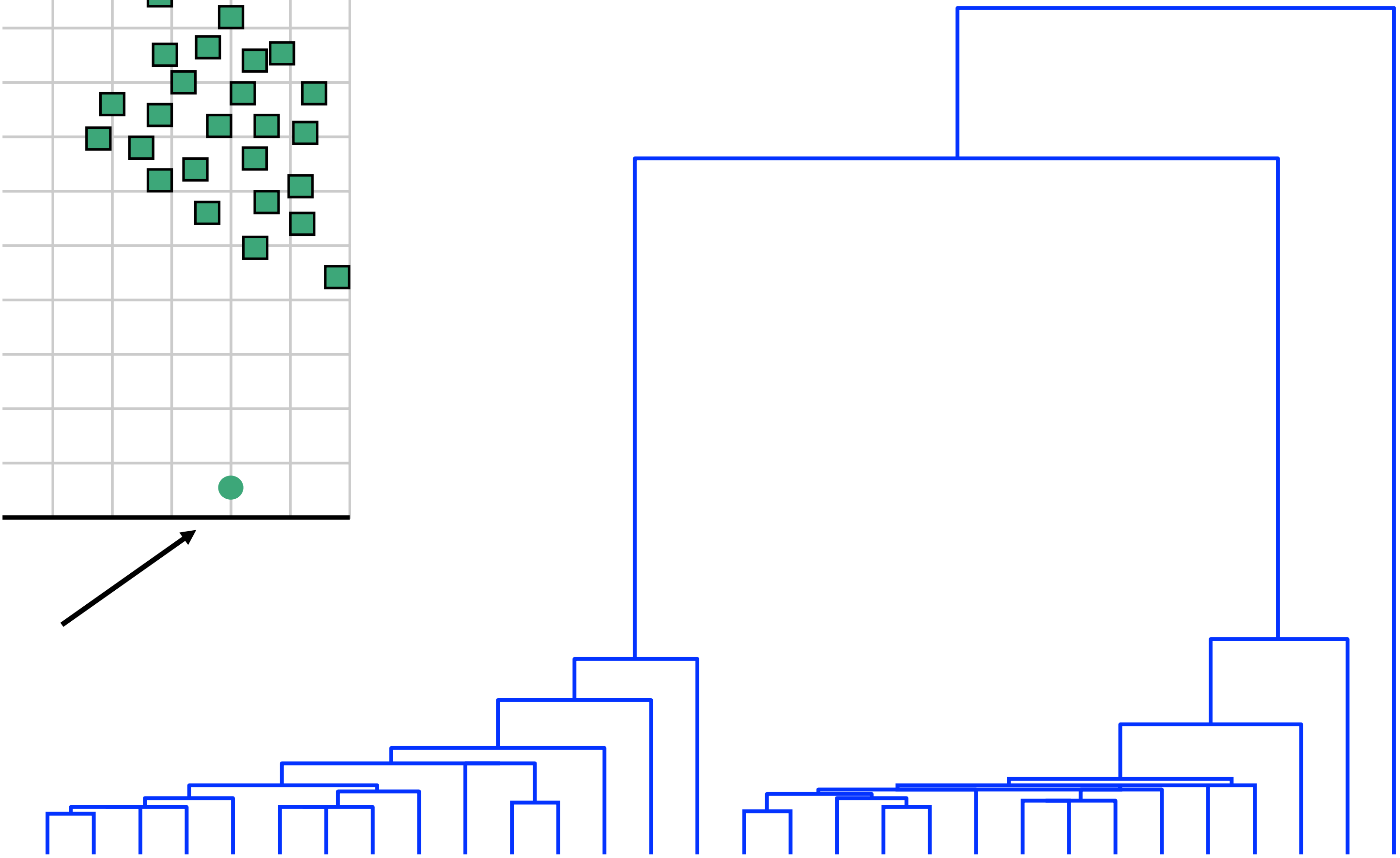


One potential use of a dendrogram is to detect outliers

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

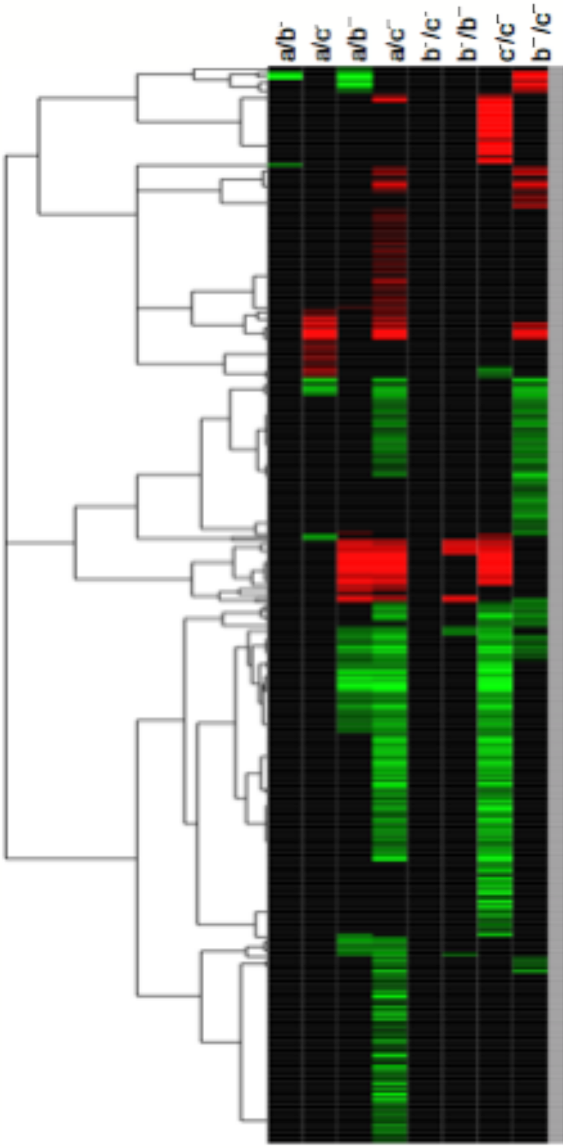


Outlier



Example: clustering genes

- Microarrays measure the activities of all genes in different conditions
- Group genes that perform the same function
- Clustering genes can help determine new functions for unknown genes
- An early “killer application” in this area
  - The most cited (>12,000) paper in PNAS!

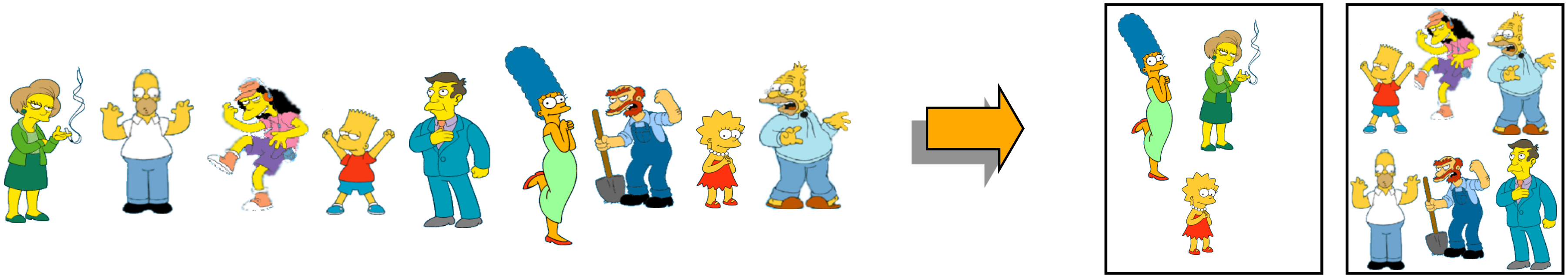


Partitioning Algorithms

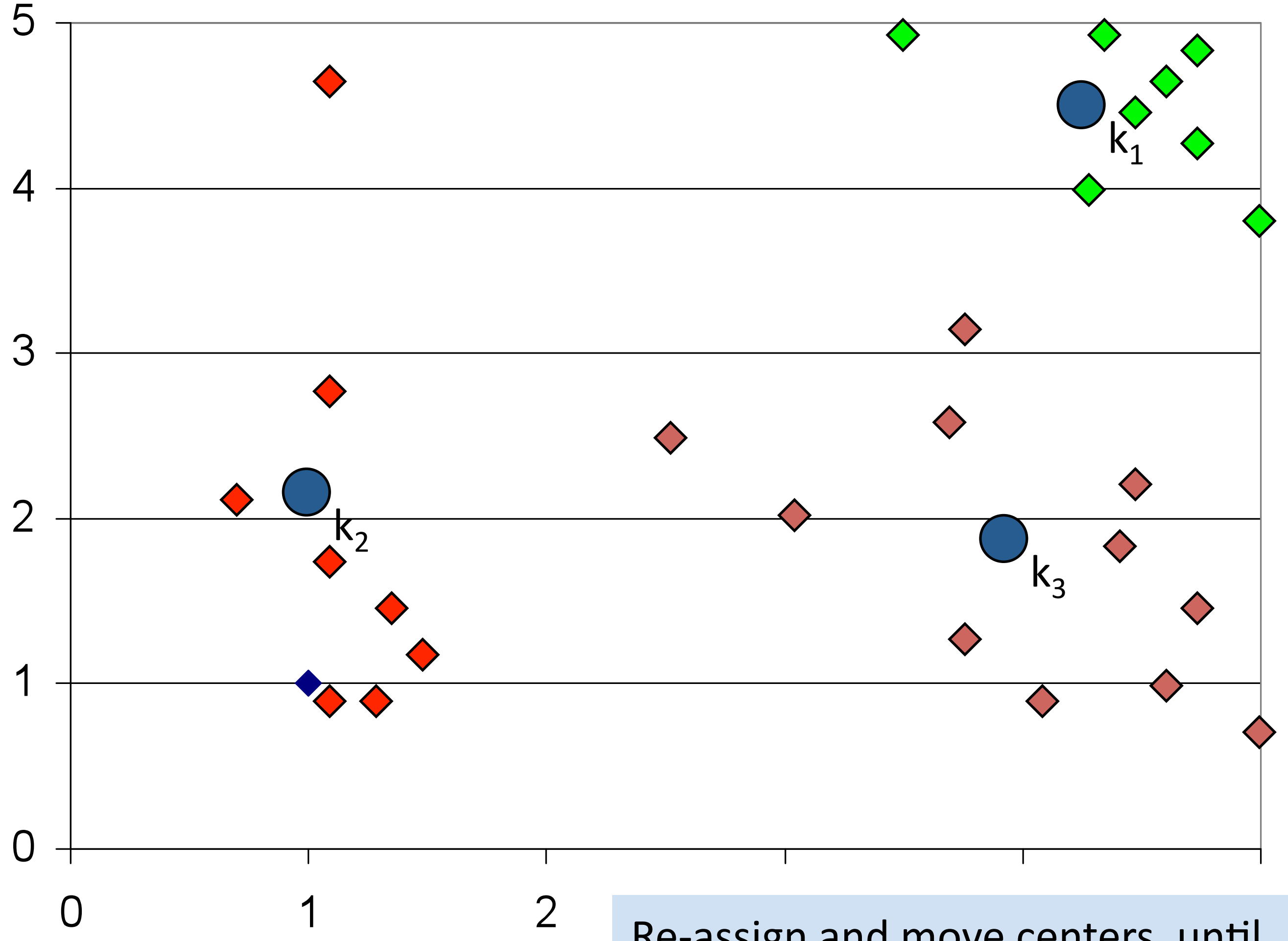
- Partitioning method: Construct a partition of *n* objects into a set of *K* clusters
  - Given: a set of objects and the number *K*
  - Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
    - Globally optimal: exhaustively enumerate all partitions
    - Effective heuristic methods: K-means algorithms

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.

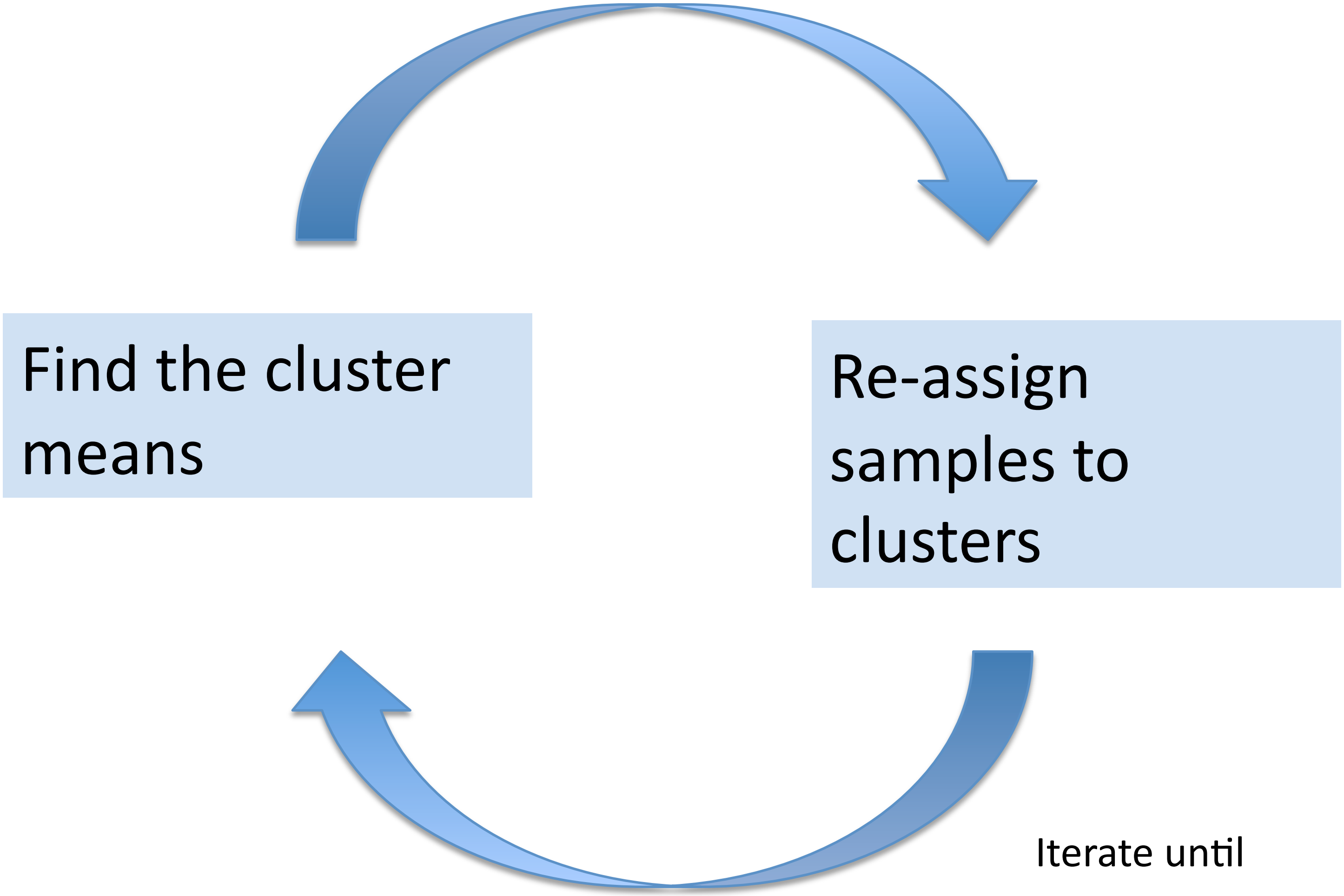


K-means Clustering



Re-assign and move centers, until
no objects change their cluster membership

K-Means Clustering Algorithm

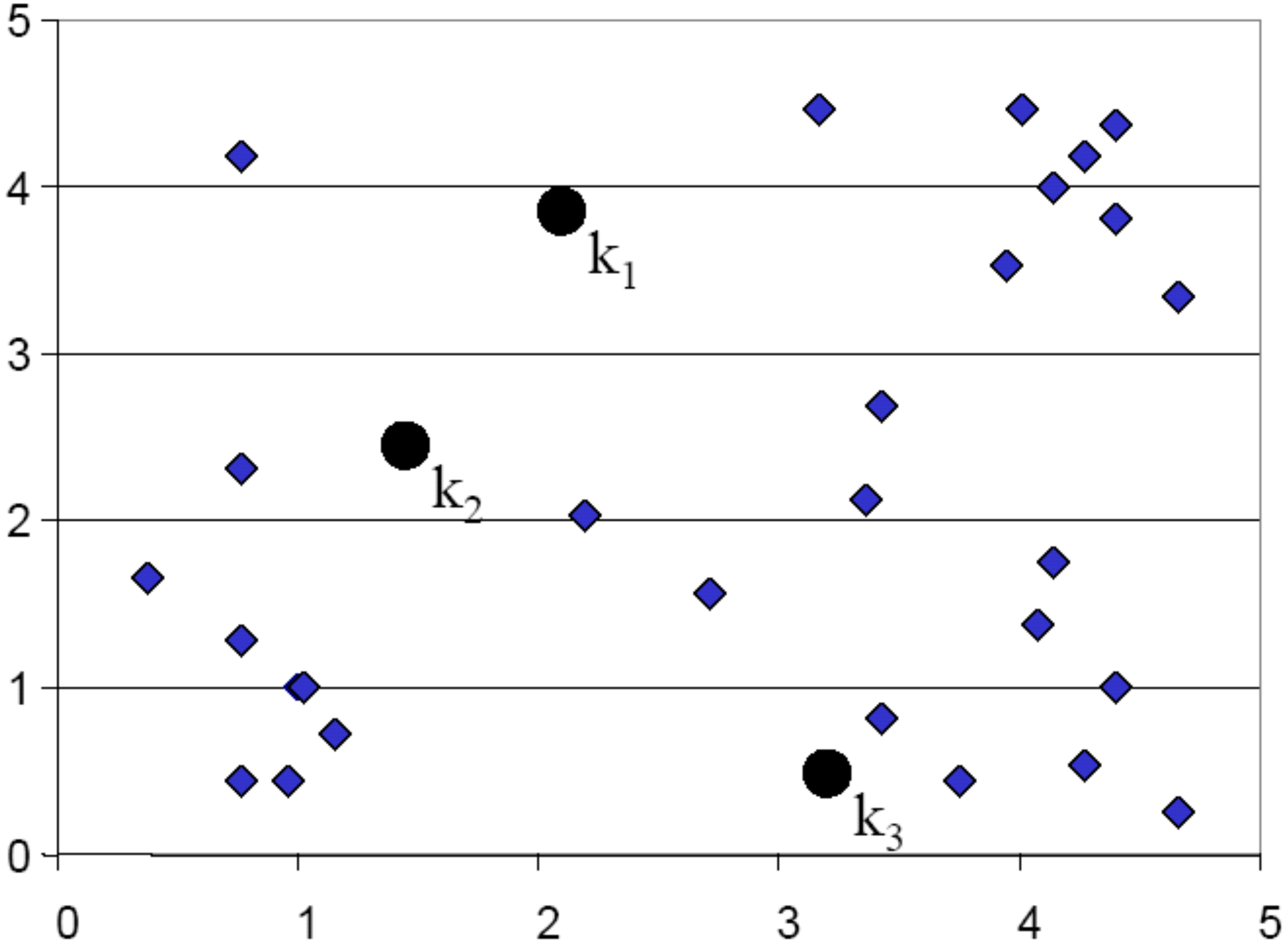


Find the cluster means

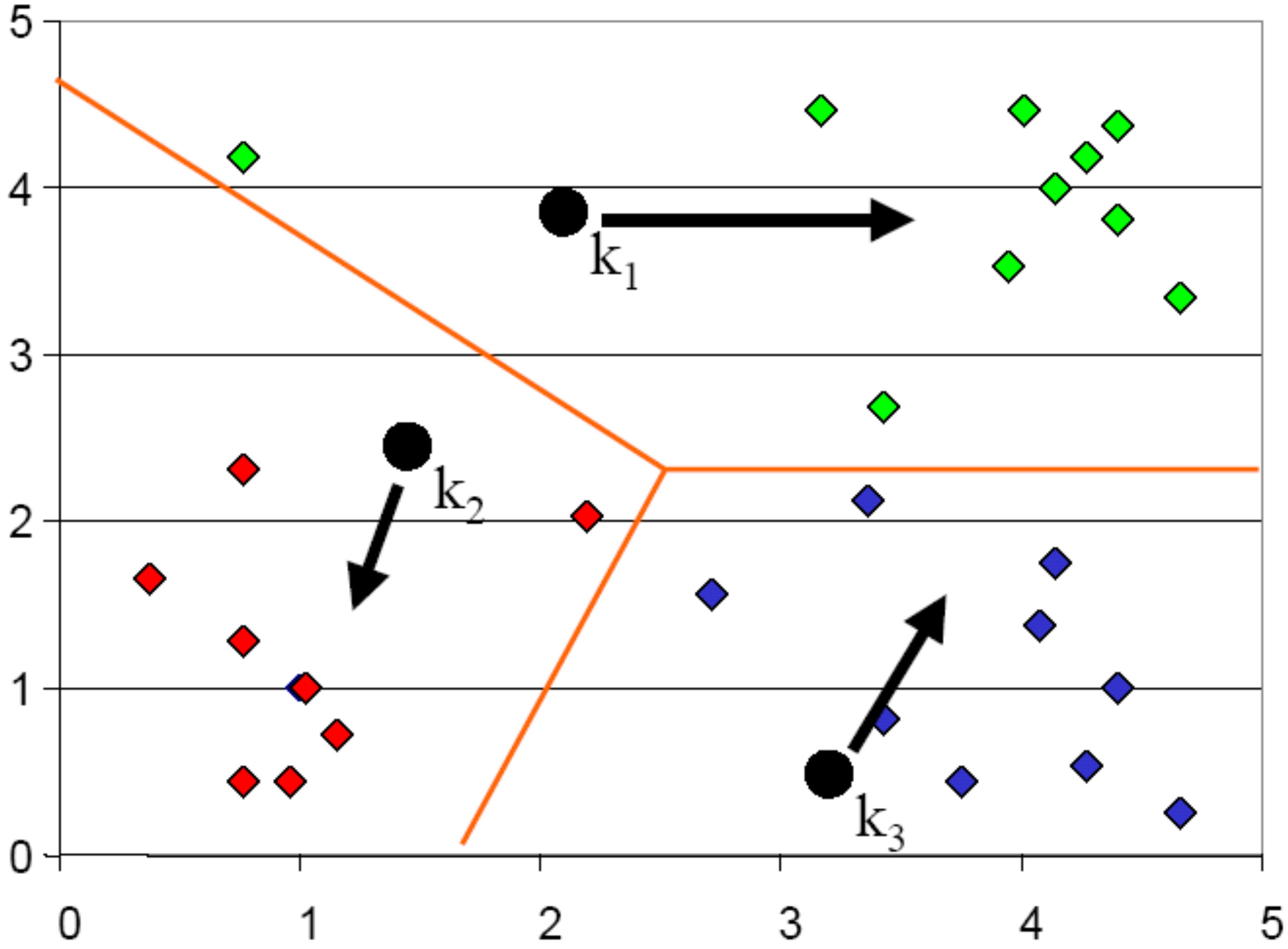
Re-assign samples to clusters

Iterate until convergence

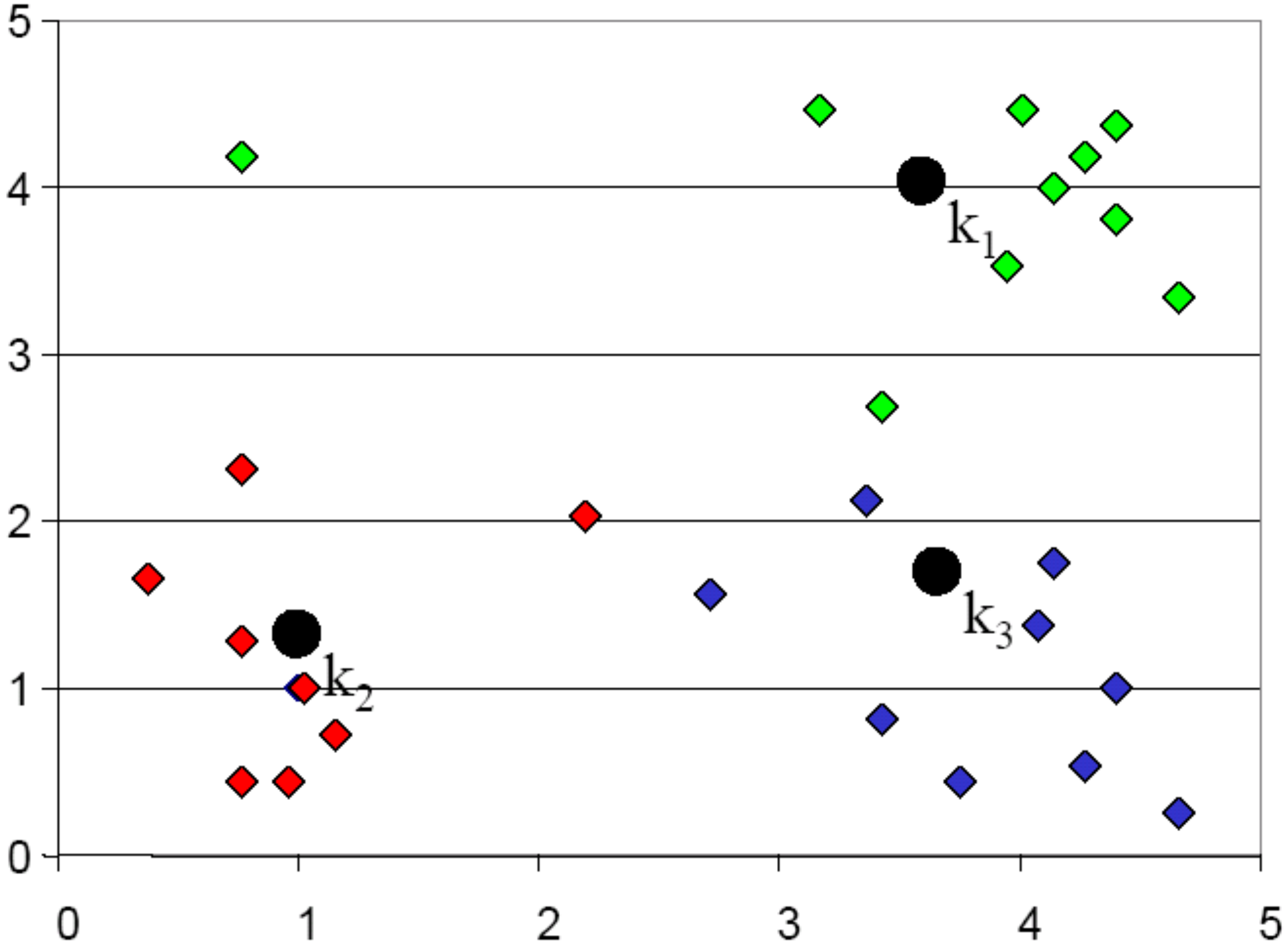
K-means Clustering: Step 1



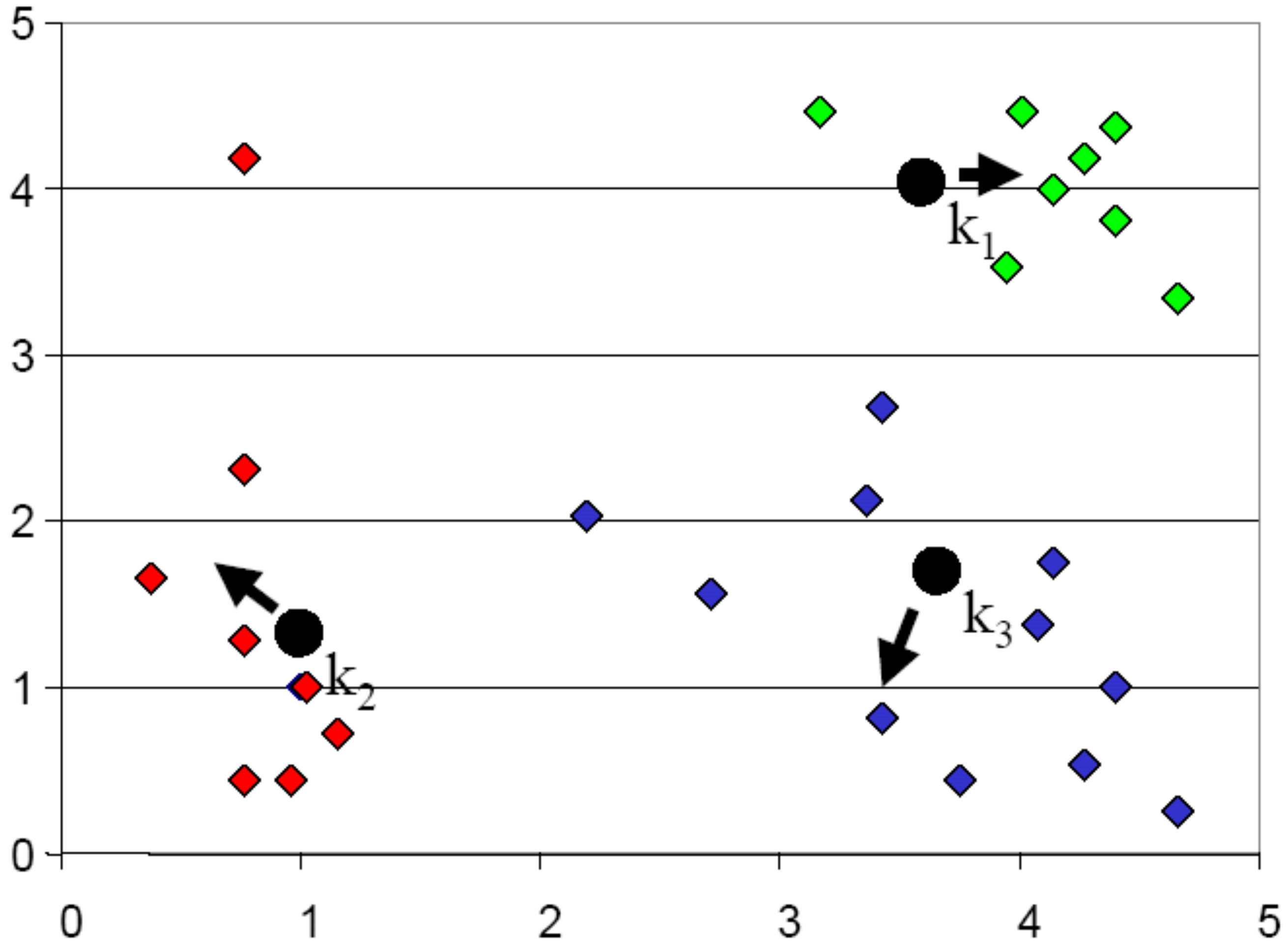
K-means Clustering: Step 2



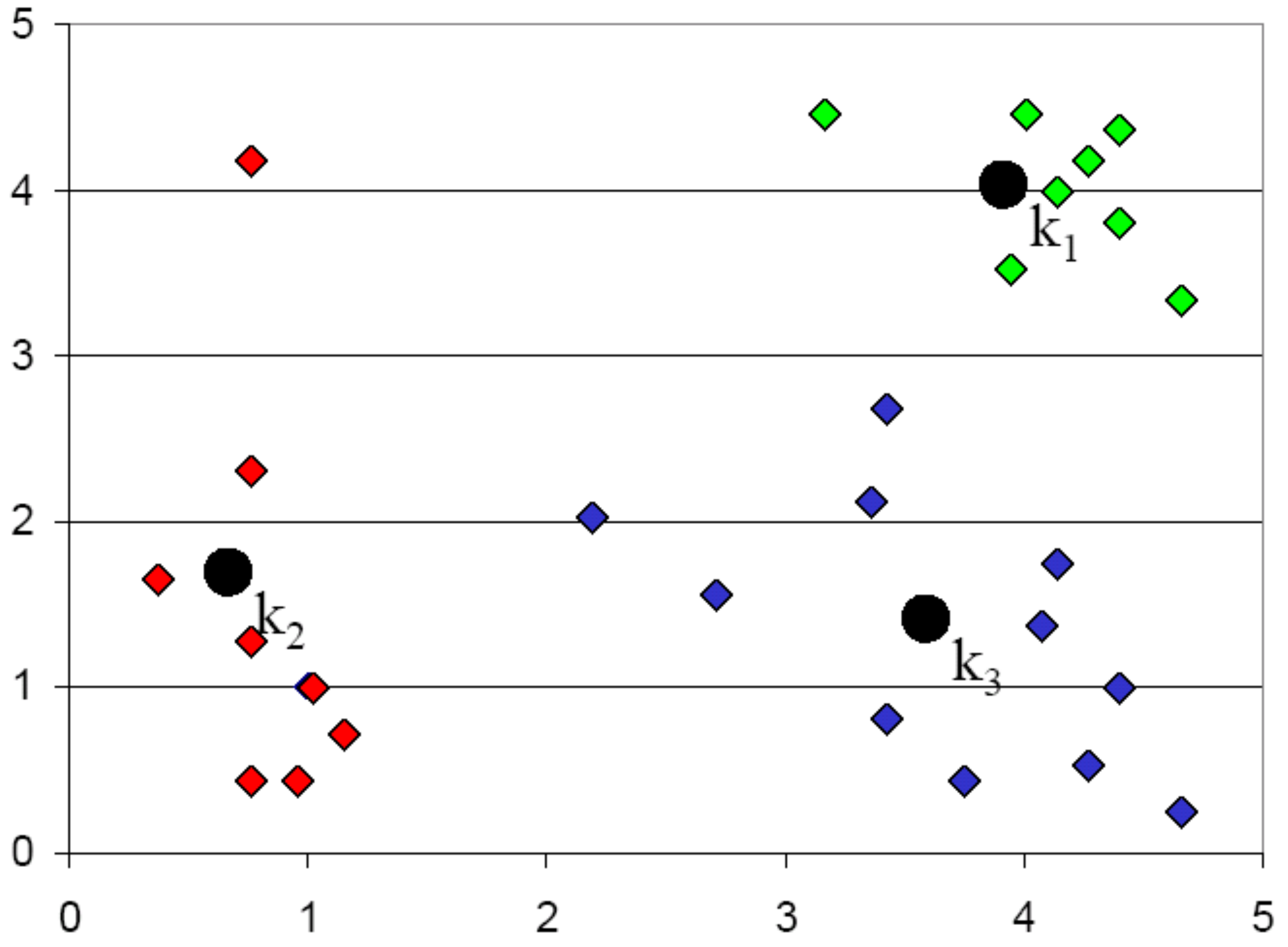
K-means Clustering: Step 3



K-means Clustering: Step 4



K-means Clustering: Step 5



K-Means

Algorithm

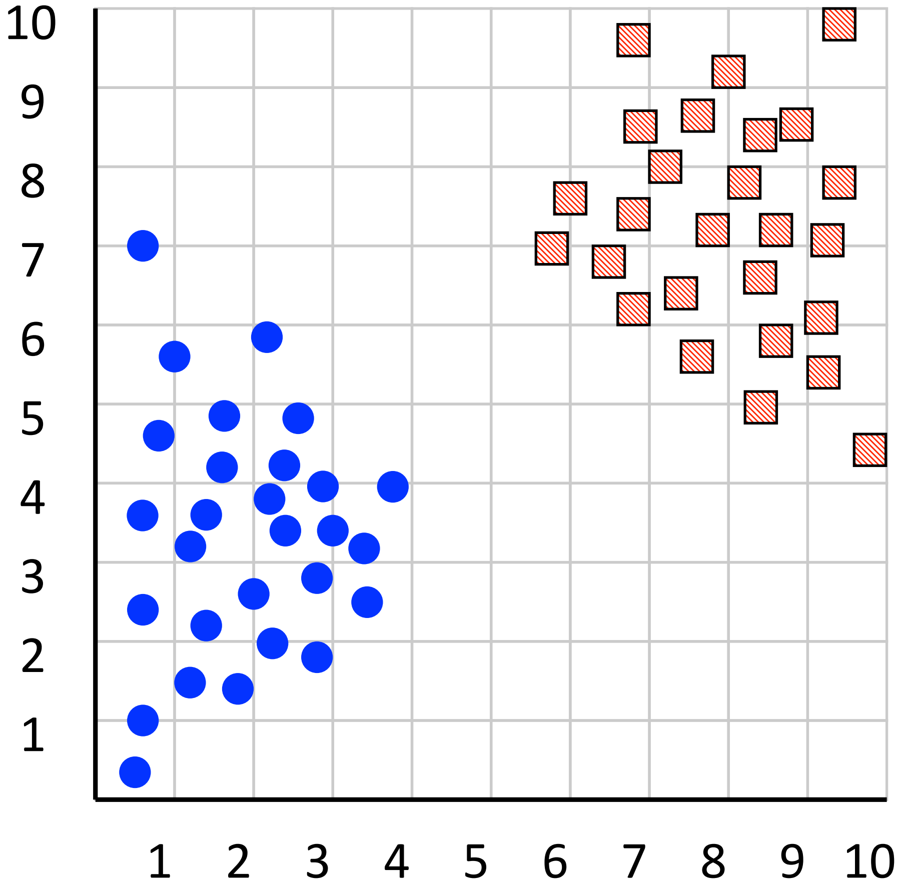
1. Decide on a value for *K*.
2. Initialize the *K* cluster centers randomly if necessary.
3. Decide the **class memberships** of the *N* objects by assigning them to the nearest cluster centroids (aka the center of gravity or mean)
4. Re-estimate the ***K* cluster centers**, by assuming the memberships found above are correct.

$$\vec{\mu}_k = \frac{1}{\mathcal{C}_k} \sum_{i \in \mathcal{C}_k} \vec{x}_i$$

5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise go to 3.

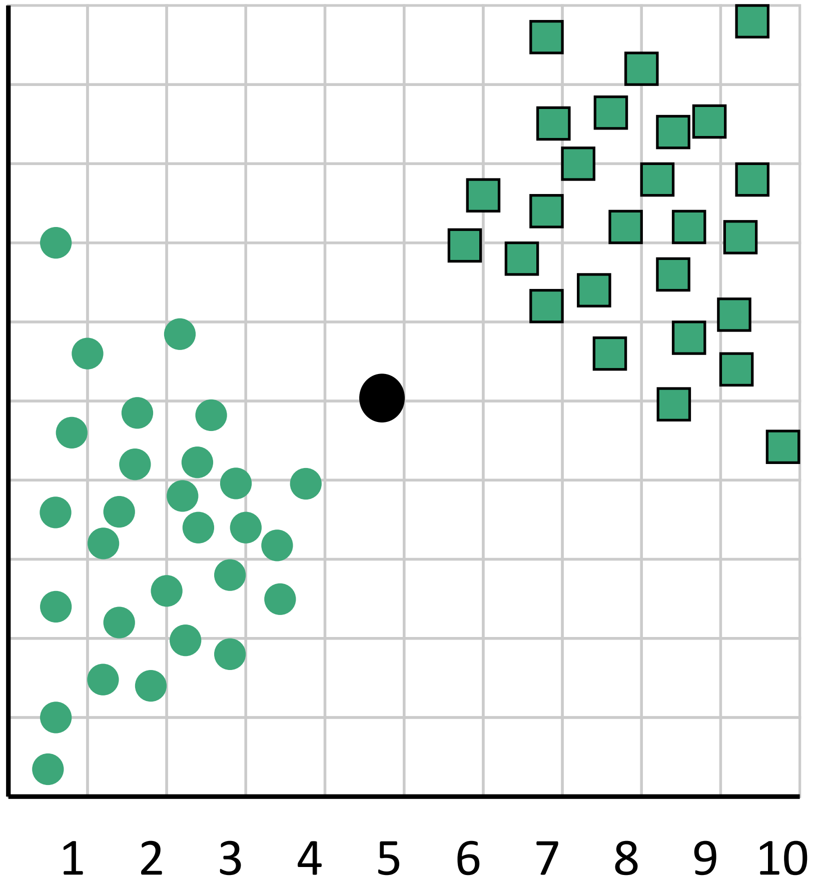
How can we tell the *right* number of clusters?

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



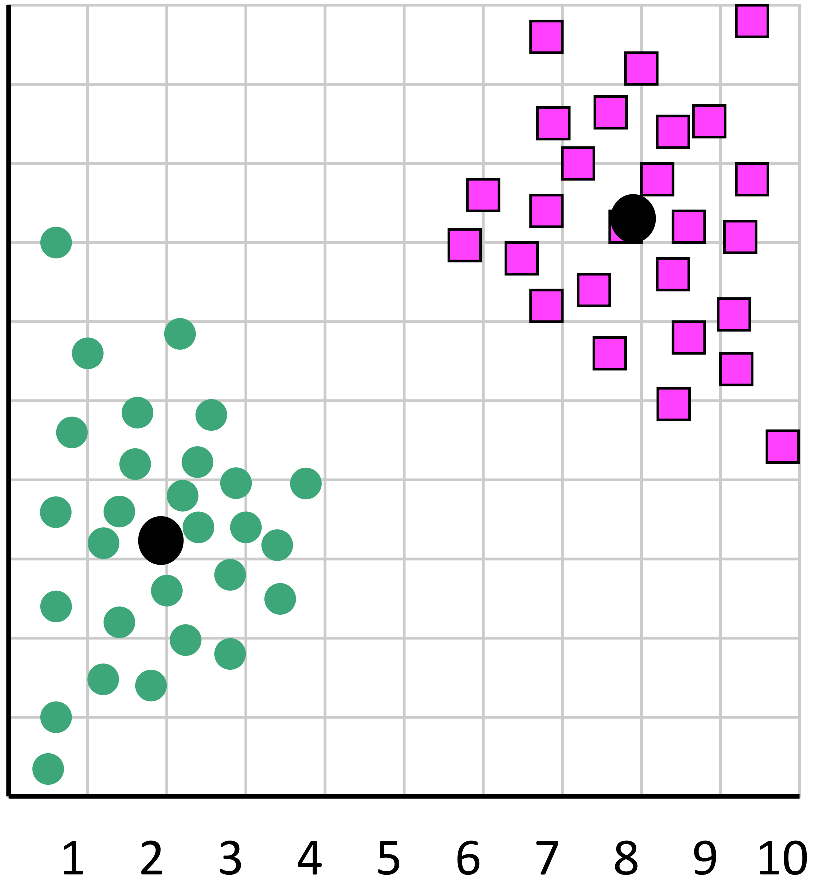
When k = 1, the objective function is 873.0

$$Obj = \sum_k \sum_i \|x_i - \mu_k\|_2^2$$



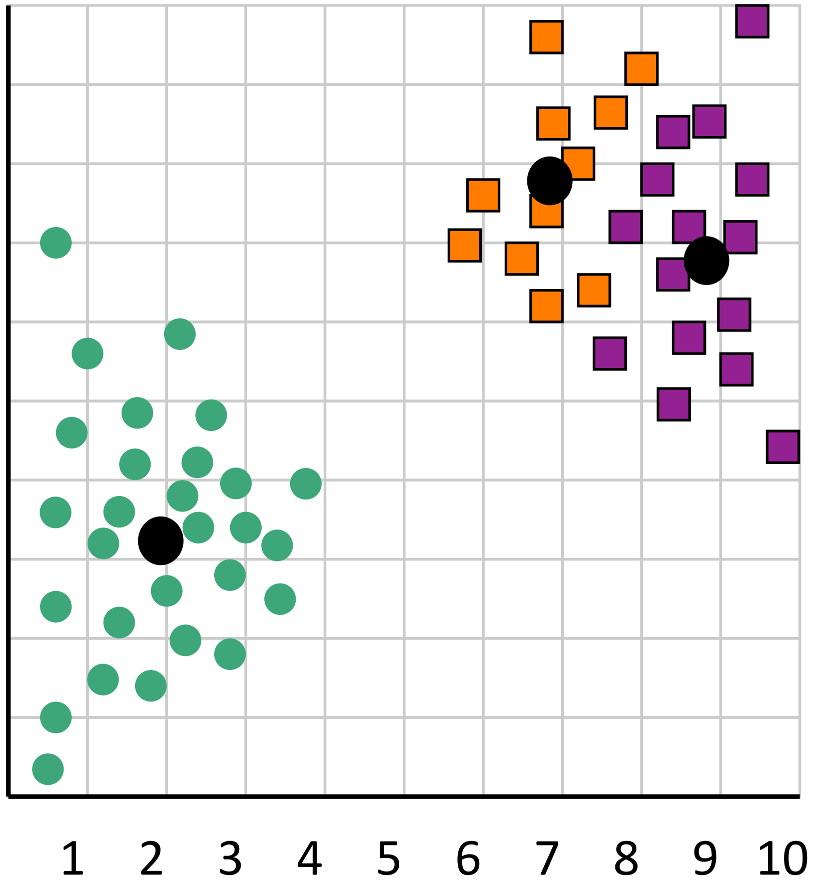
When k = 2, the objective function is 173.1

$$Obj = \sum_{k} \sum_{i} \|x_i - \mu_k\|_2^2$$



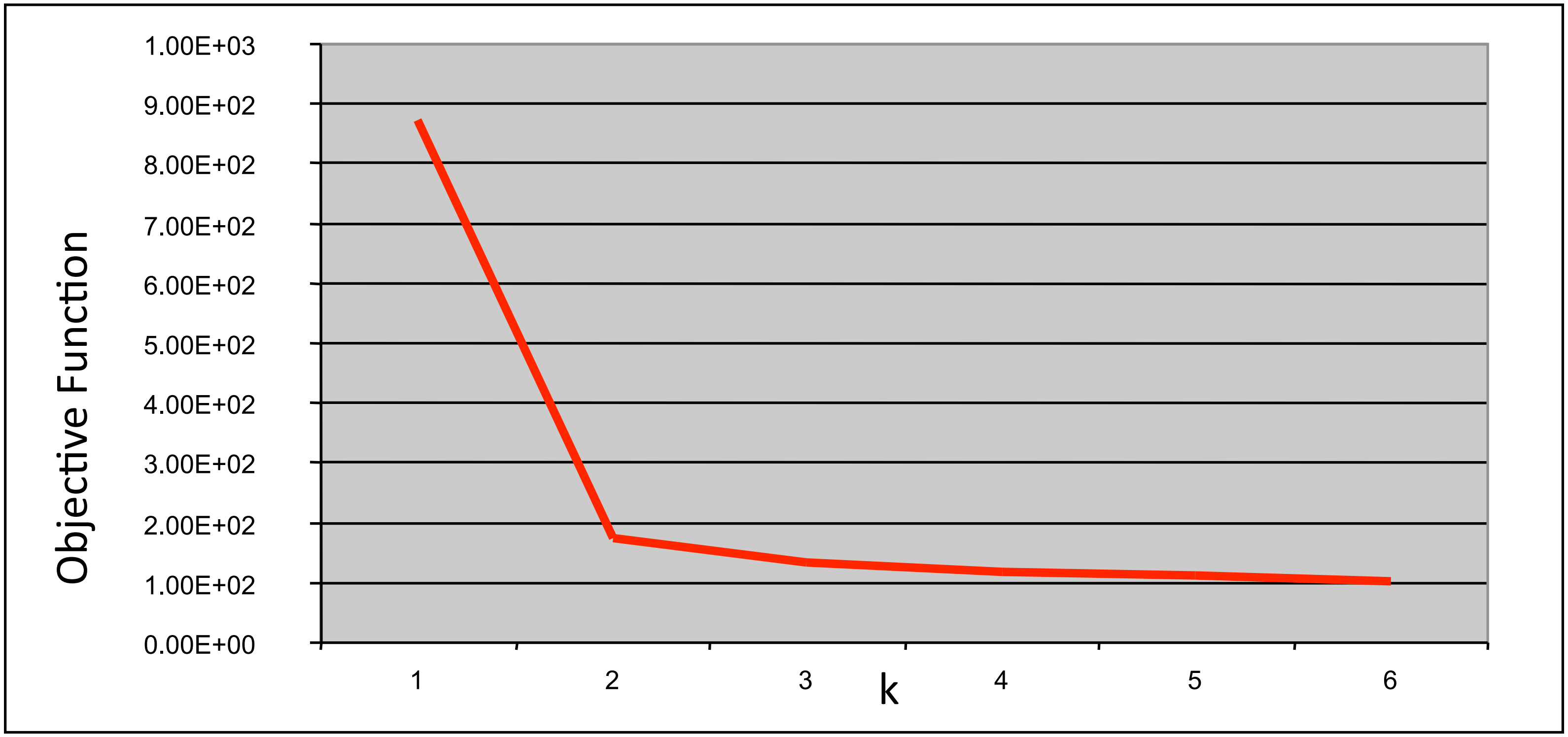
When k = 3, the objective function is 133.6

$$Obj = \sum_{k} \sum_{i} \| x_i - \mu_k \|_2^2$$



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

What you should know

- Clustering as an unsupervised learning method
- 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.