Clustering

ARTIN

Mathieu Lagrange

CENTRALE NANTES

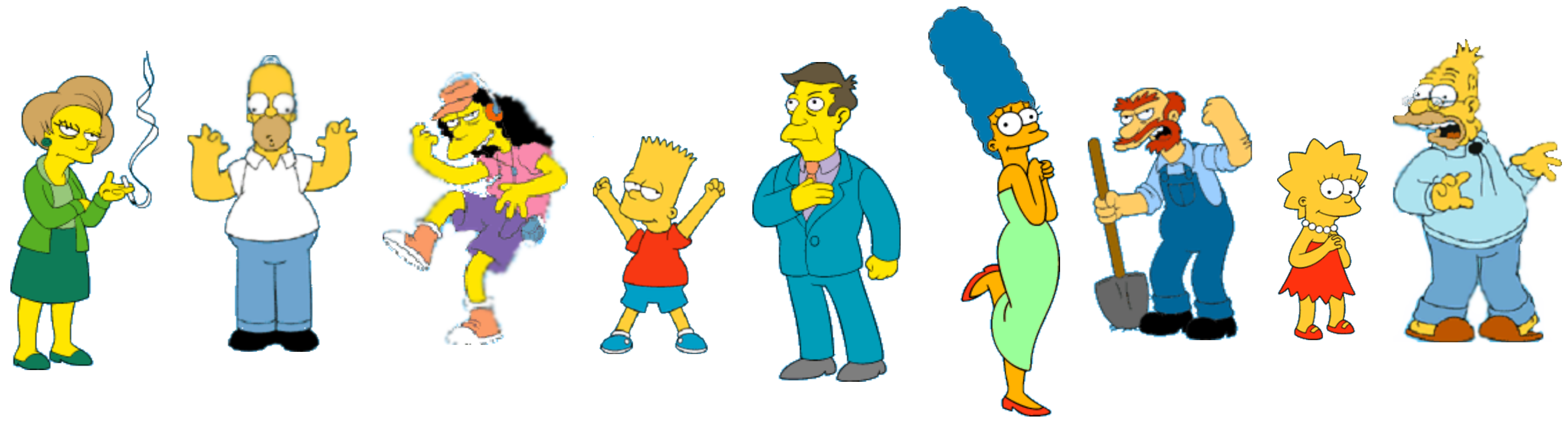
Slides mostly from Eamonn Keogh

What is Clustering?

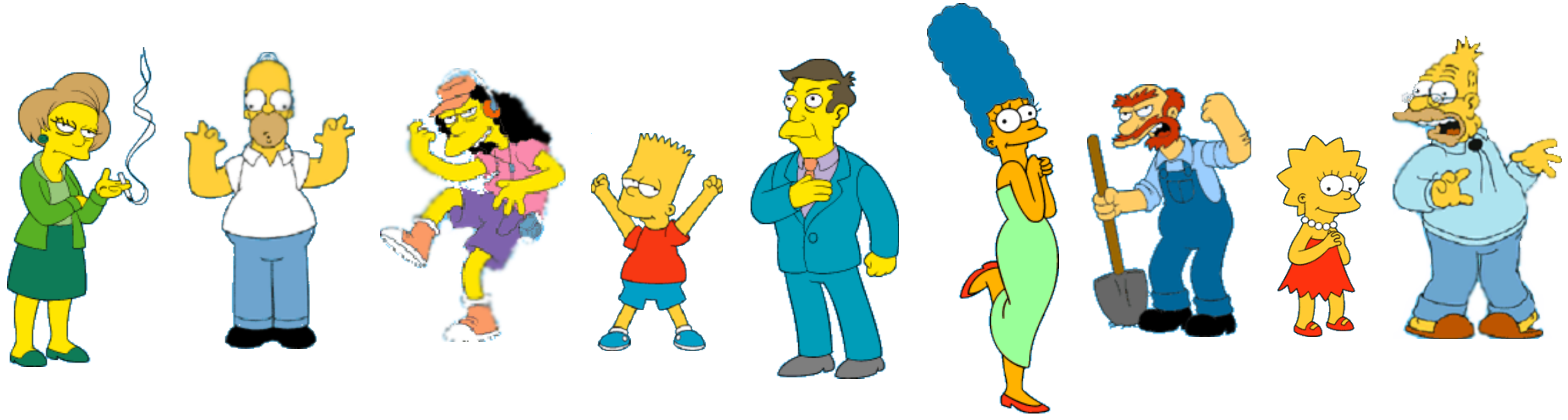
Also called *unsupervised learning*, sometimes called *classification* by statisticians and *sorting* by psychologists and *segmentation* by people in marketing

- Organizing data into classes such that there is
  - high intra-class similarity
  - low inter-class similarity
- Finding the class labels and the number of classes directly from the data (in contrast to classification).
- More informally, finding natural groupings among objects.

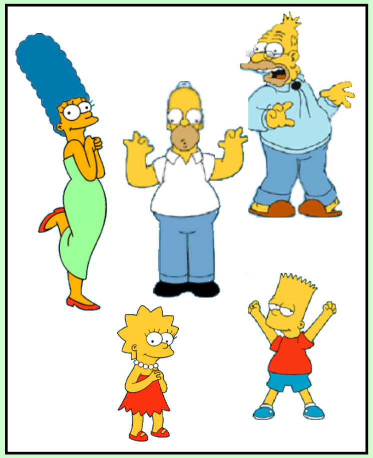
What is a natural grouping among these objects?



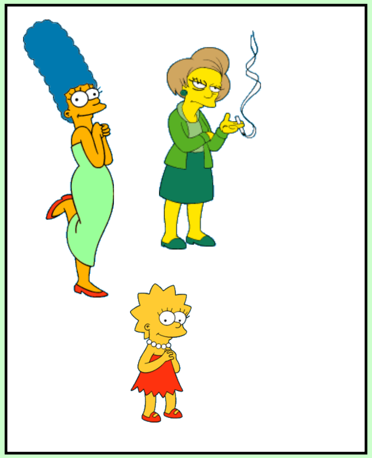
What is a natural grouping among these objects?



Clustering is subjective


Simpson's Family

School Employees


Females

Males

What is Similarity?

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

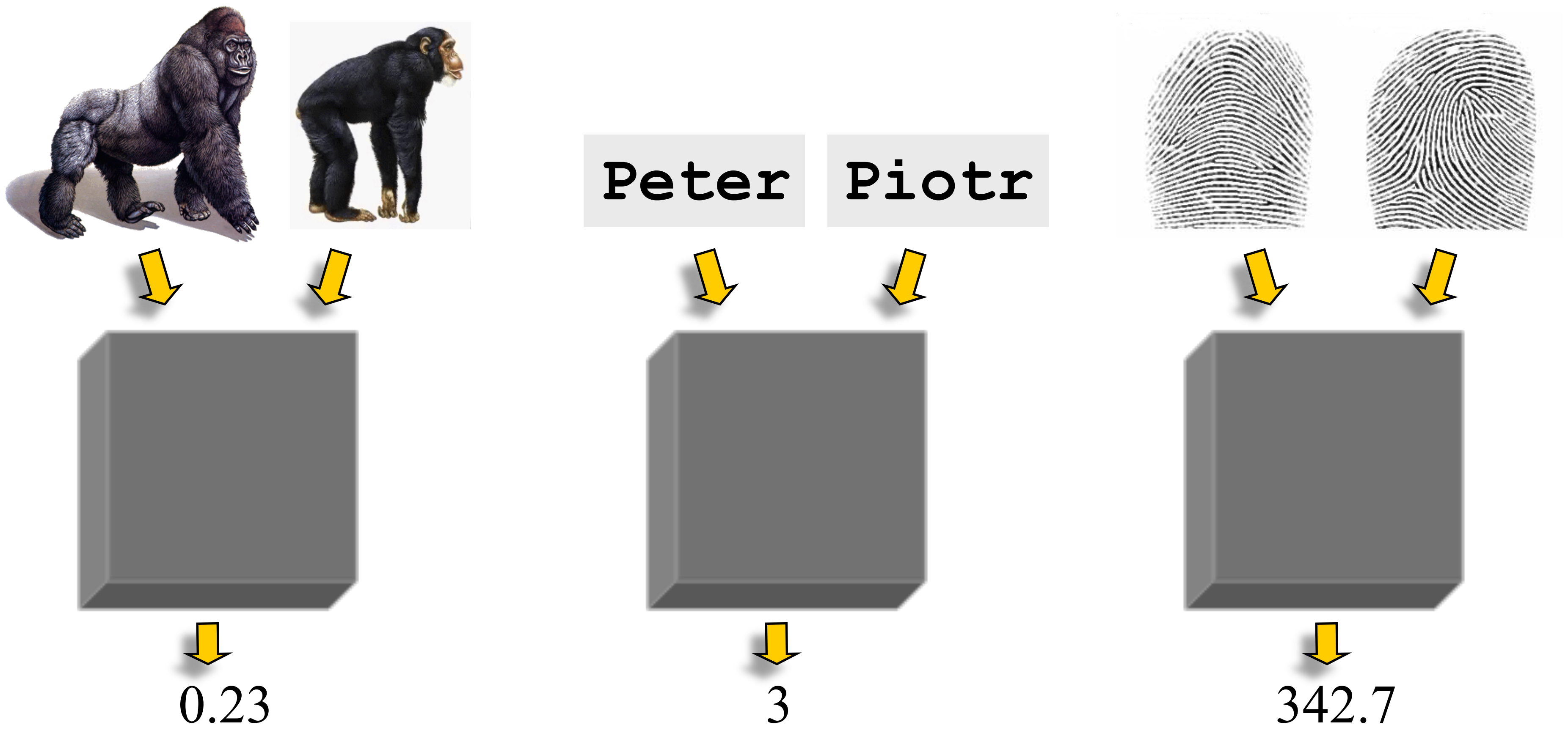
Webster's Dictionary

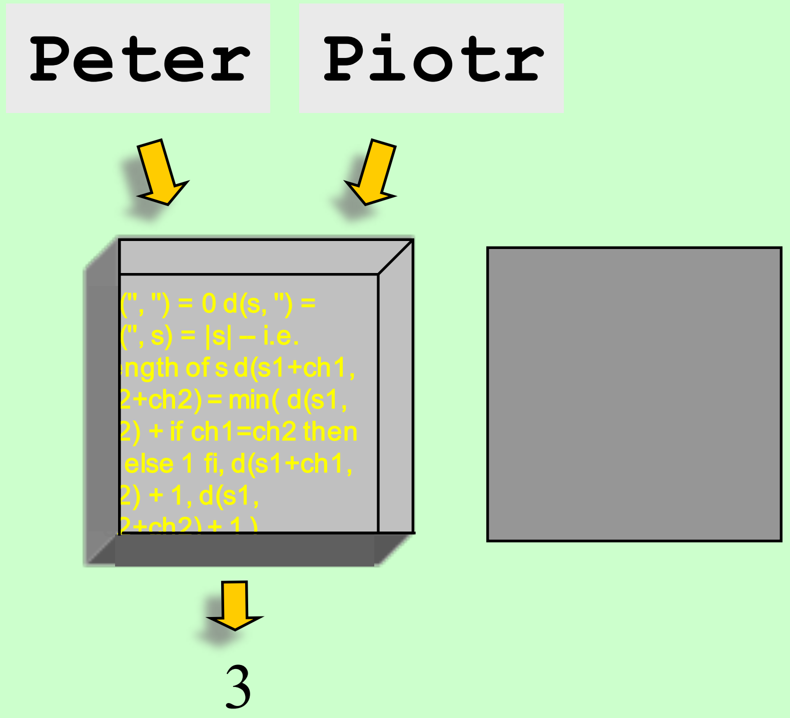
Similarity is hard to define, but… "*We know it when we see it*"

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 $D(O_1,O_2)$





When we peek inside one of these black boxes, we see some function on two variables. These functions might very simple or very complex.
In either case it is natural to ask, what properties should these functions have?

What properties should a distance measure have?

• $D(A,B) = D(B,A)$	*Symmetry*
• $D(A,A) = 0$	*Constancy of Self-Similarity*
• $D(A,B) = 0$ If $A= B$	*Positivity (Separation)*
• $D(A,B) \leq D(A,C) + D(B,C)$	*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)*

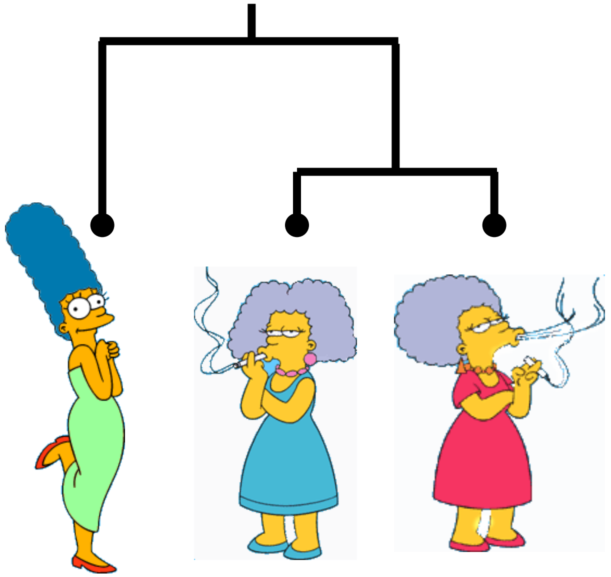
Otherwise there are objects in your world that are different, but you cannot tell apart.

$D(A,B) \leq D(A,C) + D(B,C)$ *Triangular Inequality*

Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl."

A generic technique for measuring similarity

To measure the similarity between two objects, transform one of the objects into the other, and measure how much effort it took. The measure of effort becomes the distance measure.



The distance between Patty and Selma.

```
Change dress color,    1 point
Change earring shape, 1 point
Change hair part,      1 point
```

D(Patty,Selma) = **3**

The distance between Marge and Selma.

```
Change dress color,    1 point
Add earrings,          1 point
Decrease height,       1 point
Take up smoking,       1 point
Lose weight,           1 point
```

D(Marge,Selma) = **5**

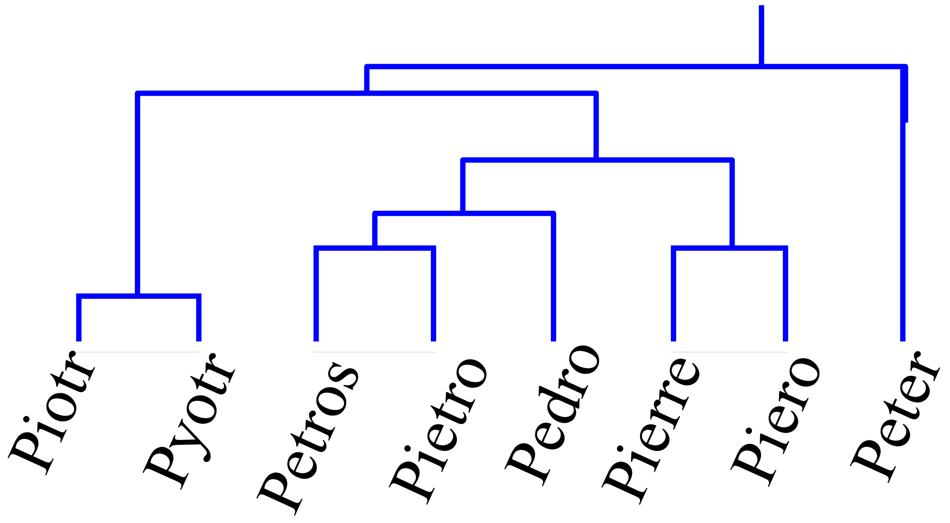
This is called the "edit distance" or the "transformation distance"

Edit Distance Example

It is possible to transform any string *Q* into string *C*, using only *Substitution*, *Insertion* and *Deletion*.
Assume that each of these operators has a cost associated with it.

The similarity between two strings can be defined as the cost of the cheapest transformation from *Q* to *C*.

Note that for now we have ignored the issue of how we can find this cheapest transformation



How similar are the names "Peter" and "Piotr"?

Assume the following cost function

Substitution	1 Unit
Insertion	1 Unit
Deletion	1 Unit

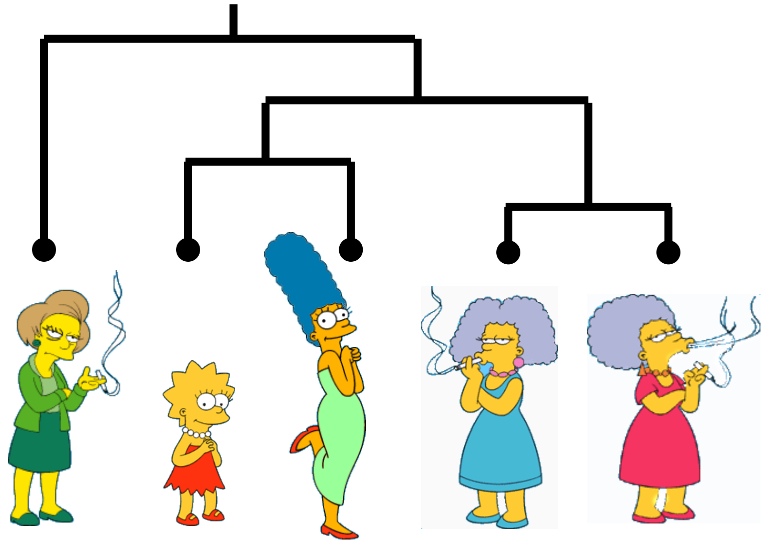
D(**`Peter,Piotr`**) is 3

$$\begin{tikzcd}
\texttt{Peter} \arrow[d, "\text{Substitution (i for e)}"] \\
\texttt{Piter} \arrow[d, "\text{Insertion (o)}"] \\
\texttt{Pioter} \arrow[d, "\text{Deletion (e)}"] \\
\texttt{Piotr}
\end{tikzcd}$$

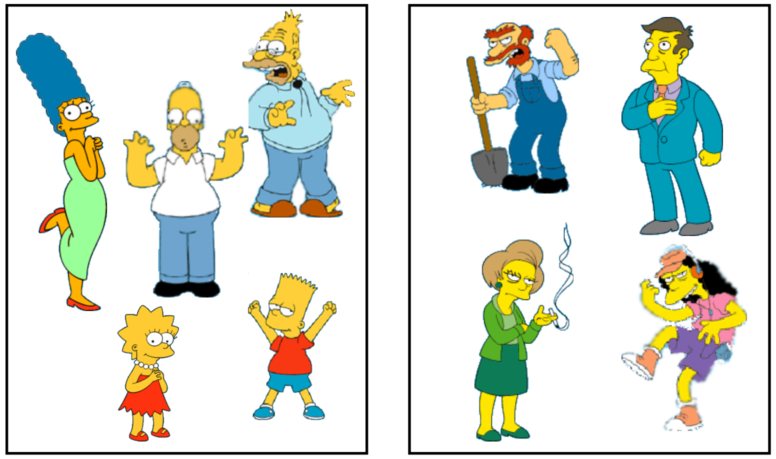
Two Types of Clustering

- **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion (we will see an example called BIRCH)
- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion

Hierarchical



Partitional

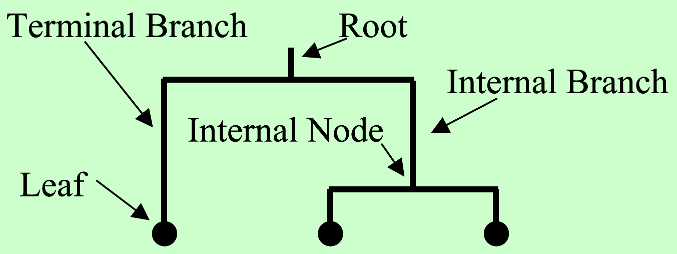


Desirable Properties of a Clustering Algorithm

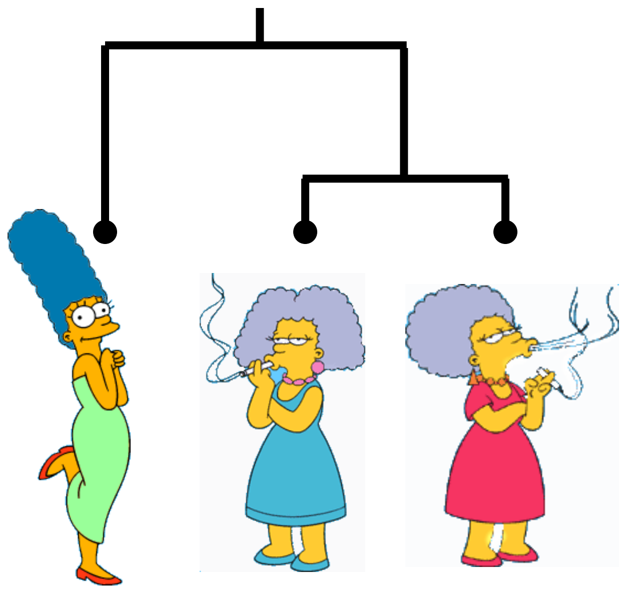
- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- Incorporation of user-specified constraints
- Interpretability and usability

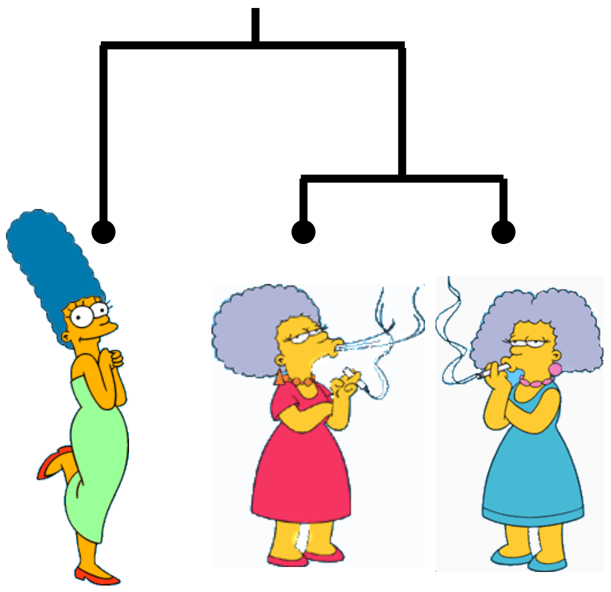
A Useful Tool for Summarizing Similarity Measurements

In order to better appreciate and evaluate the examples given in the early part of this talk, we will now introduce the *dendrogram*.



The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share.





A Demonstration of Hierarchical Clustering using String Edit Distance

Pedro (Portuguese)

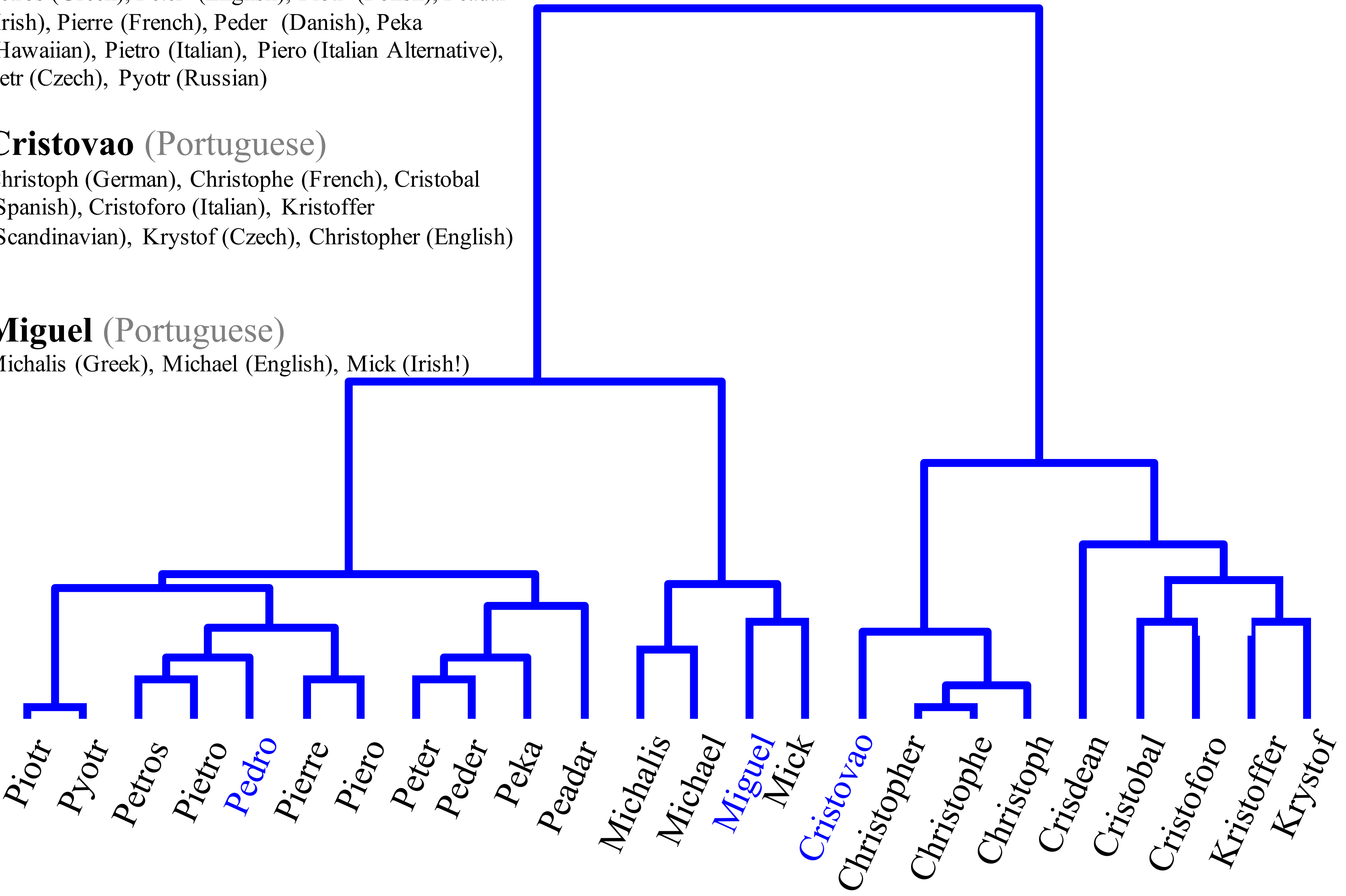
Petros (Greek), Peter (English), Piotr (Polish), Peadar (Irish), Pierre (French), Peder (Danish), Peka (Hawaiian), Pietro (Italian), Piero (Italian Alternative), Petr (Czech), Pyotr (Russian)

Cristovao (Portuguese)

Christoph (German), Christophe (French), Cristobal (Spanish), Cristoforo (Italian), Kristoffer (Scandinavian), Krystof (Czech), Christopher (English)

Miguel (Portuguese)

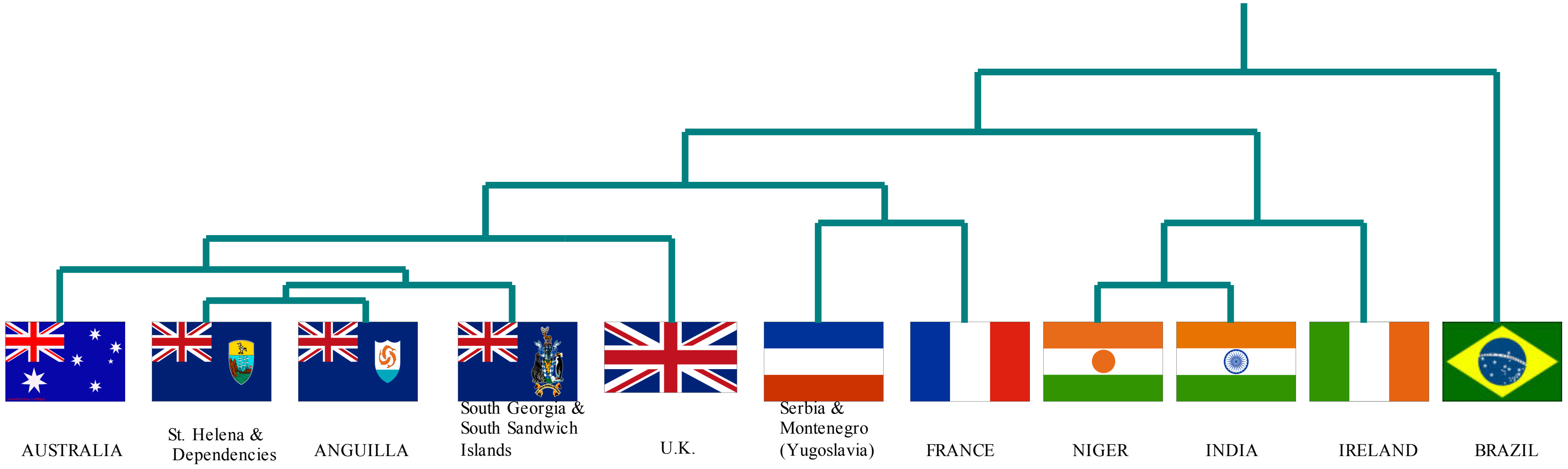
Michalis (Greek), Michael (English), Mick (Irish!)



Hierarchal clustering can sometimes show patterns that are meaningless or spurious

- For example, in this clustering, the tight grouping of Australia, Anguilla, St. Helena etc is meaningful, since all these countries are former UK colonies.

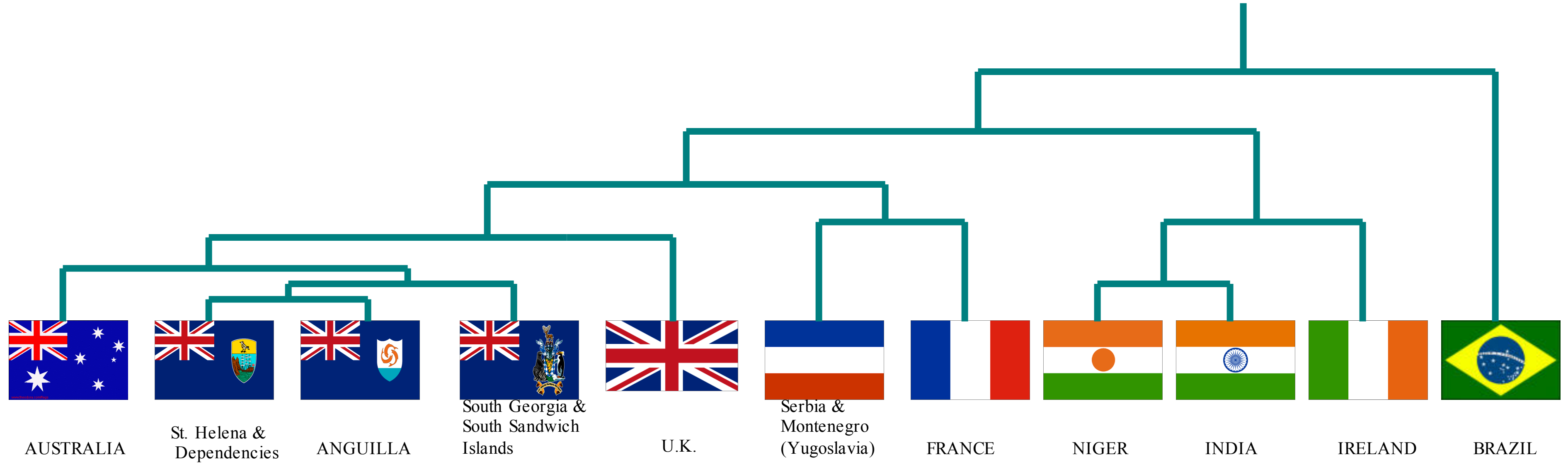
- However the tight grouping of Niger and India is completely spurious, there is no connection between the two.



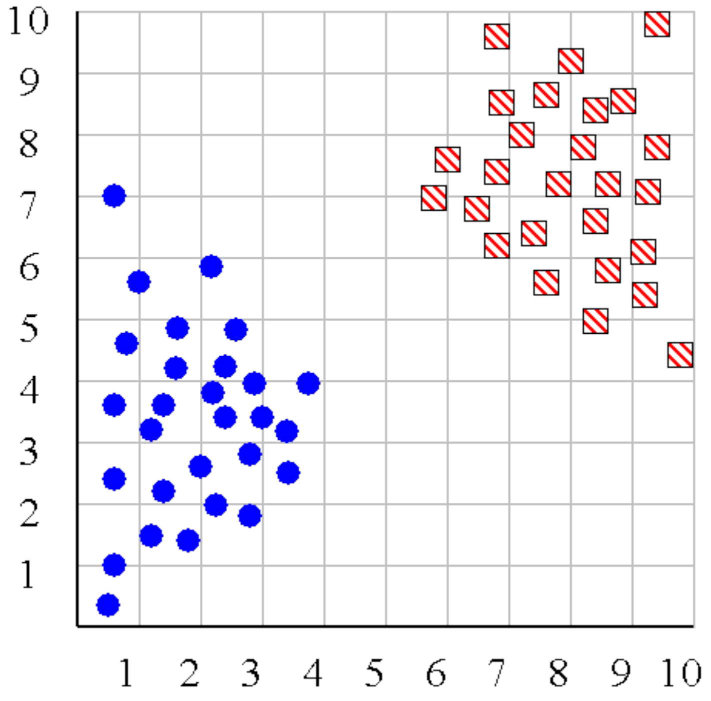
AUSTRALIA
St. Helena & Dependencies
ANGUILLA
South Georgia & South Sandwich Islands
U.K.
Serbia & Montenegro (Yugoslavia)
FRANCE
NIGER
INDIA
IRELAND
BRAZIL

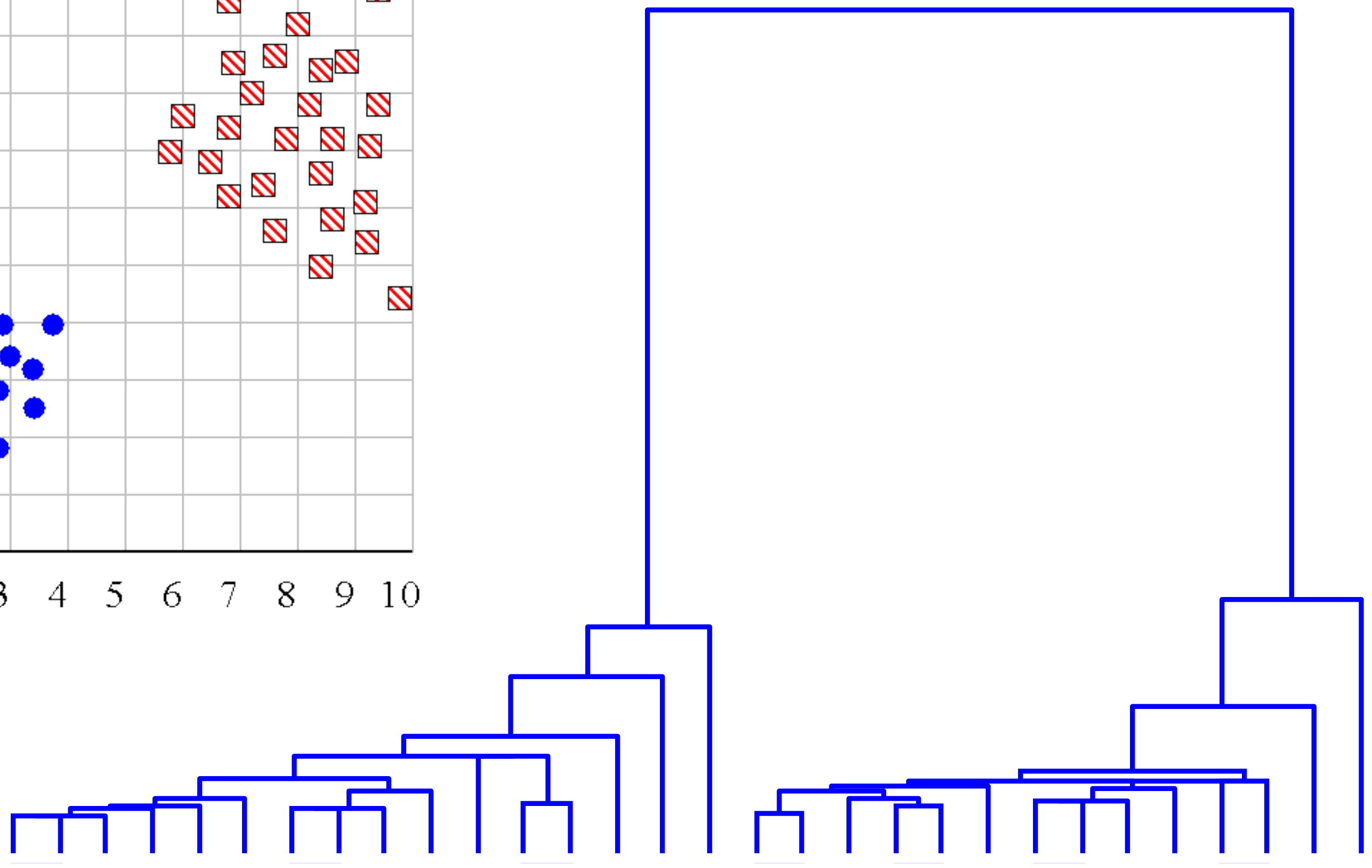
• The flag of Niger is orange over white over green, with an orange disc on the central white stripe, symbolizing the sun. The orange stands the Sahara desert, which borders Niger to the north. Green stands for the grassy plains of the south and west and for the River Niger which sustains them. It also stands for fraternity and hope. White generally symbolizes purity and hope.

• The Indian flag is a horizontal tricolor in equal proportion of deep saffron on the top, white in the middle and dark green at the bottom. In the center of the white band, there is a wheel in navy blue to indicate the Dharma Chakra, the wheel of law in the Sarnath Lion Capital. This center symbol or the 'CHAKRA' is a symbol dating back to 2nd century BC. The saffron stands for courage and sacrifice; the white, for purity and truth; the green for growth and auspiciousness.


AUSTRALIA
St. Helena & Dependencies
ANGUILLA
South Georgia & South Sandwich Islands
U.K.
Serbia & Montenegro (Yugoslavia)
FRANCE
NIGER
INDIA
IRELAND
BRAZIL

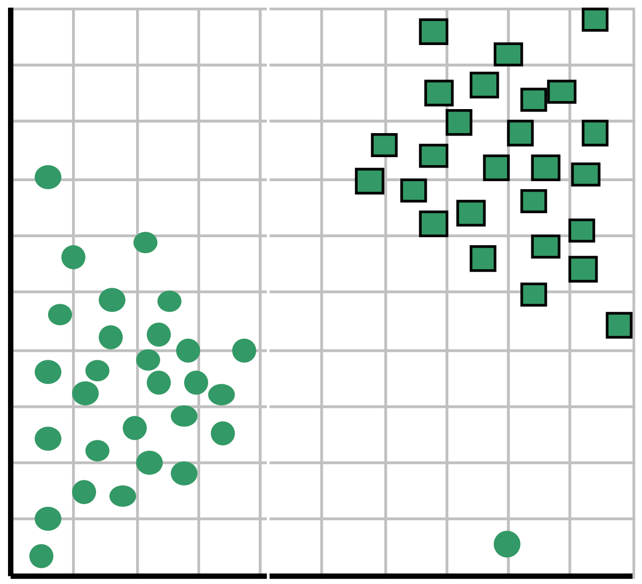
We can look at the dendrogram to determine the "correct" number of clusters. In this case, the two highly separated subtrees are highly suggestive of two clusters. (Things are rarely this clear cut, unfortunately)



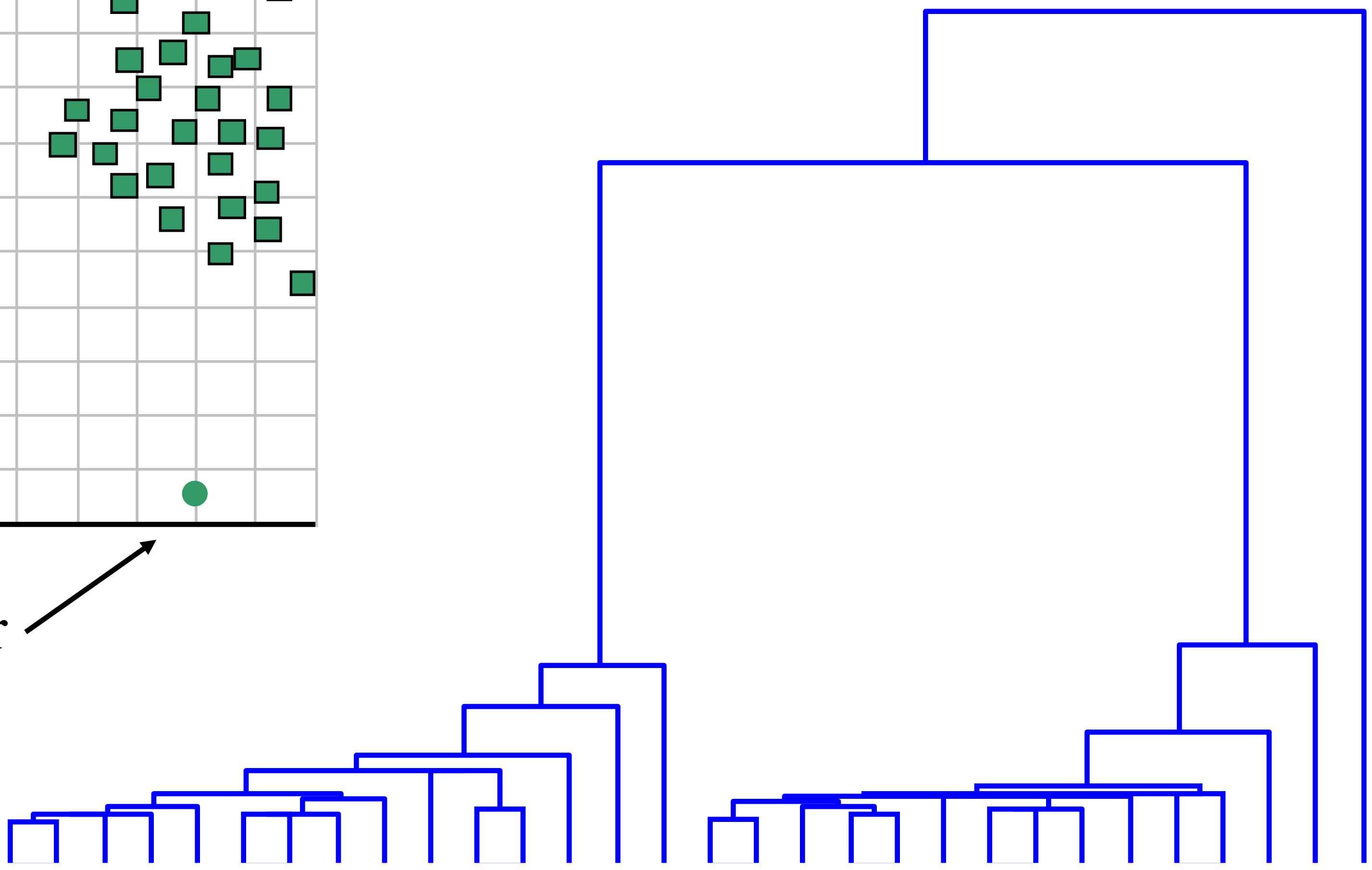


One potential use of a dendrogram is to detect outliers

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



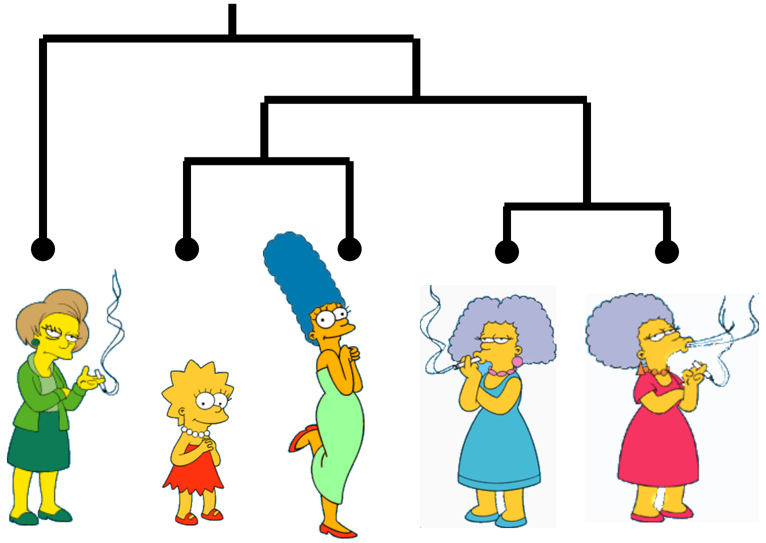
Outlier



(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



Since we cannot test all possible trees we will have to heuristic search of all possible trees. We could do this..

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.

Top-Down (divisive): Starting with all the data in a single cluster, consider every possible way to divide the cluster into two. Choose the best division and recursively operate on both sides.

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

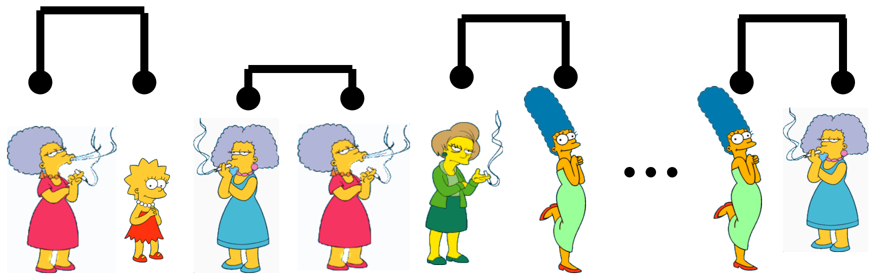
D(,) = 8

D(,) = 1

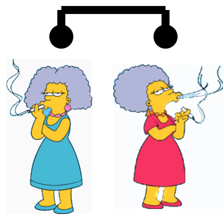
Bottom-Up (agglomerative):

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…



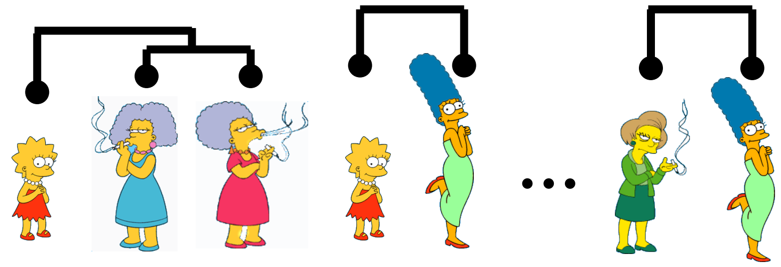
Choose the best



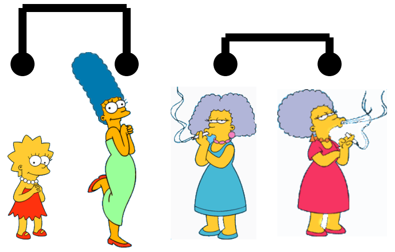
Bottom-Up (agglomerative):

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.

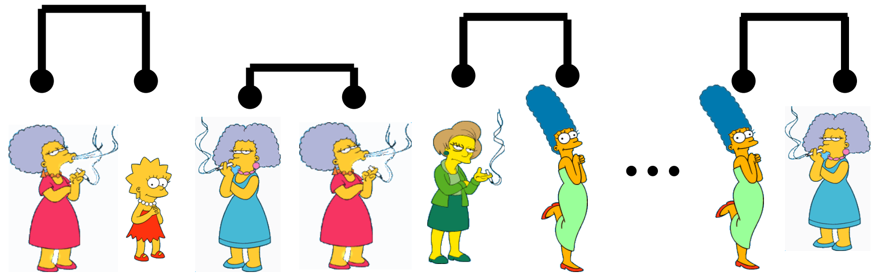
Consider all possible merges…



Choose the best



Consider all possible merges…

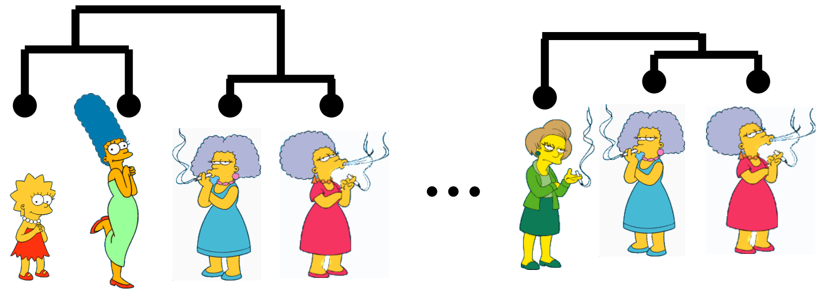


Choose the best

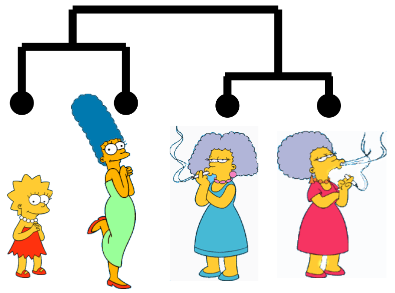
Bottom-Up (agglomerative):

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.

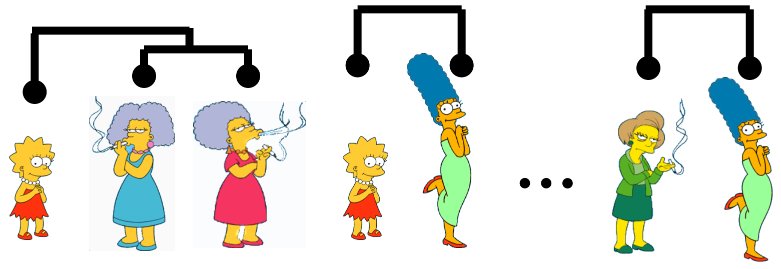
Consider all possible merges…



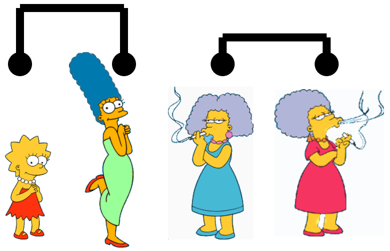
Choose the best



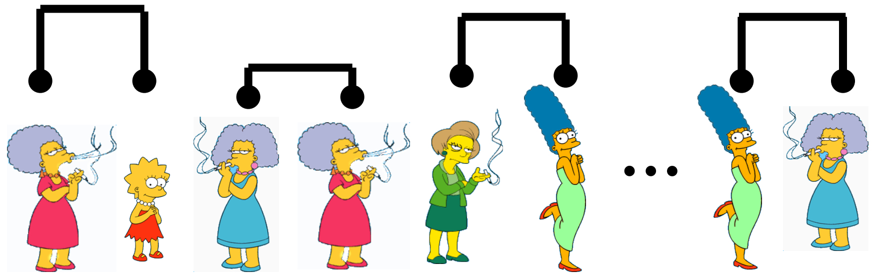
Consider all possible merges…



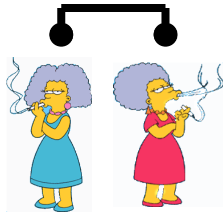
Choose the best



Consider all possible merges…

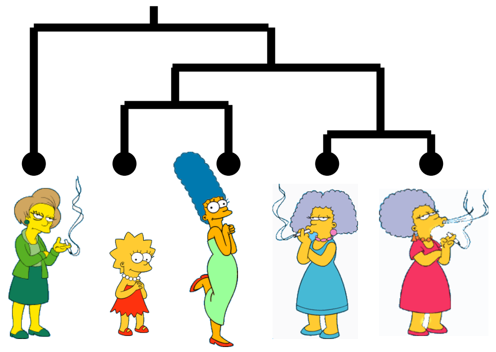


Choose the best

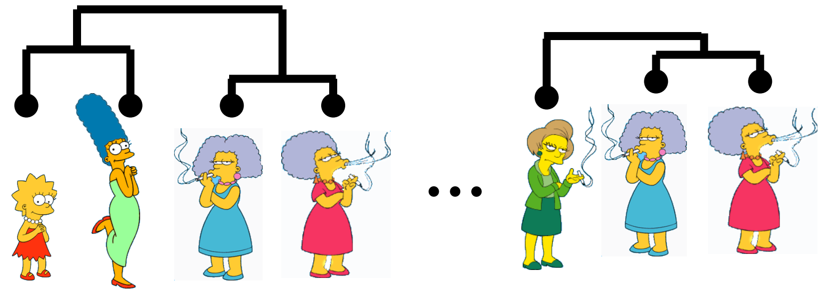


Bottom-Up (agglomerative):

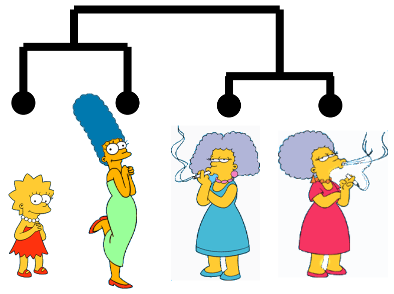
Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



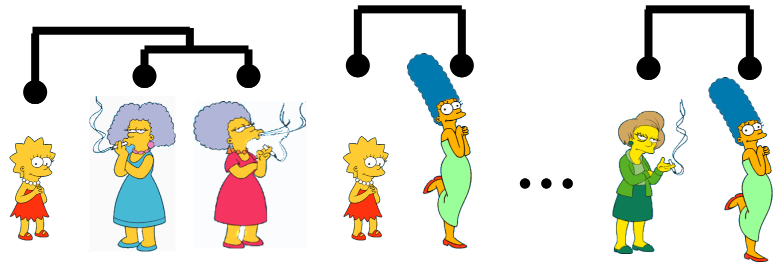
Consider all possible merges…



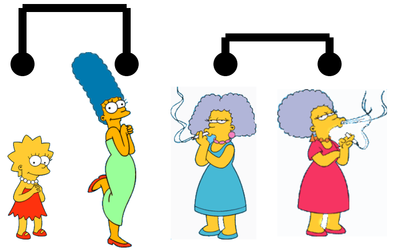
Choose the best



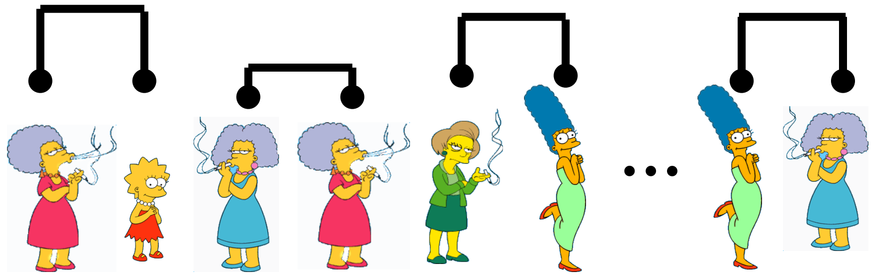
Consider all possible merges…



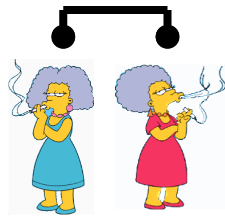
Choose the best



Consider all possible merges…

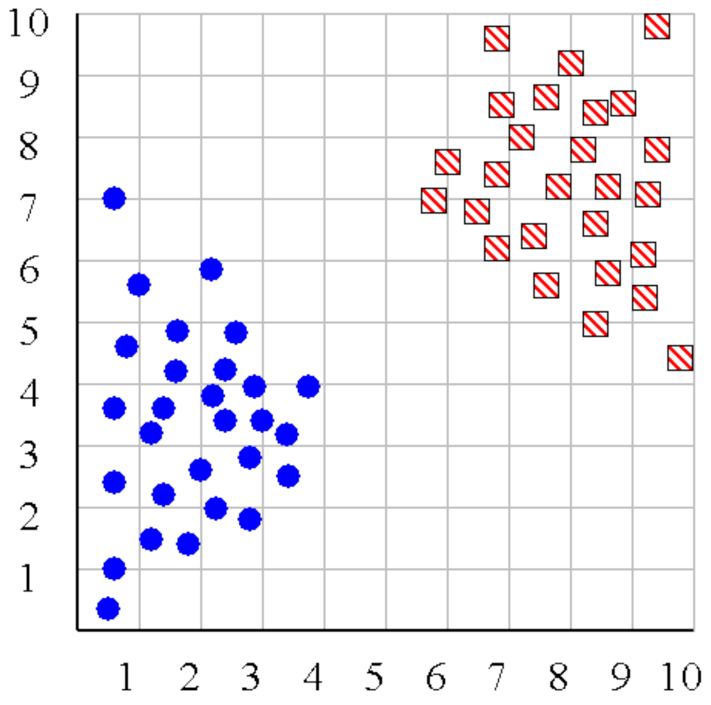


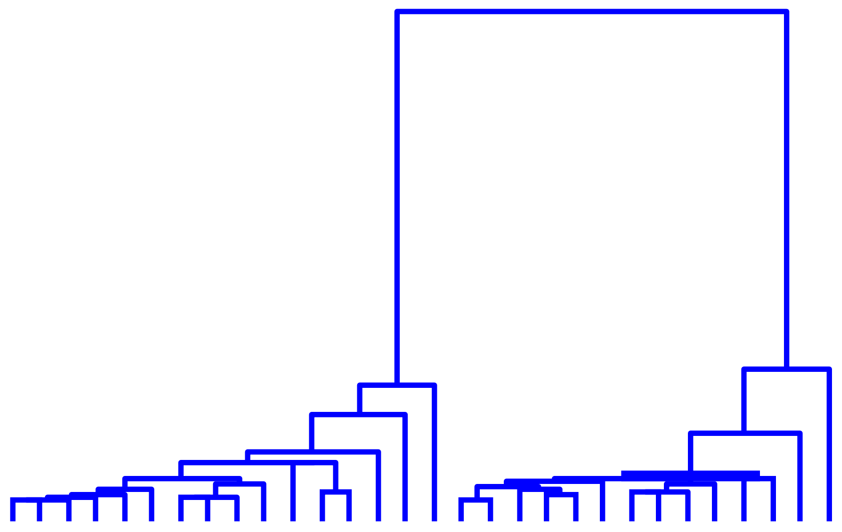
Choose the best



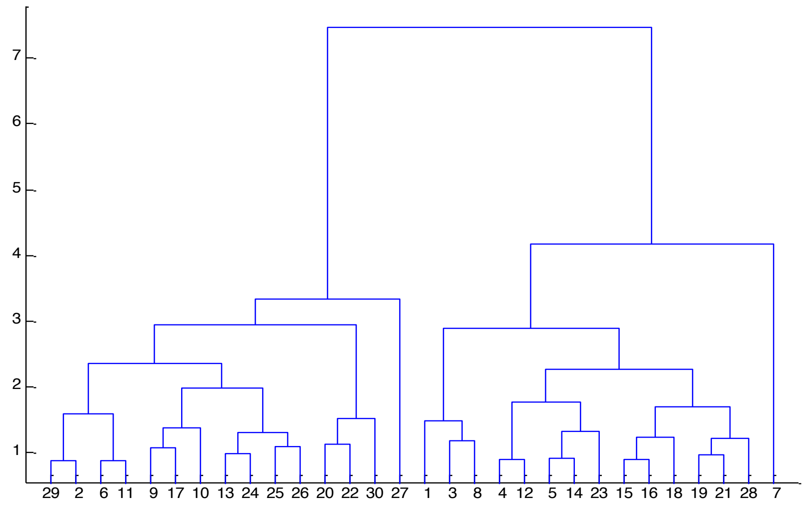
We know how to measure the distance between two objects, but defining the distance between an object and a cluster, or defining the distance between two clusters is non obvious.

- **Single linkage (nearest neighbor):** In this method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters.
- **Complete linkage (furthest neighbor):** In this method, the distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors").
- **Group average linkage:** In this method, the distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters.
- **Wards Linkage**: In this method, we try to minimize the variance of the merged clusters

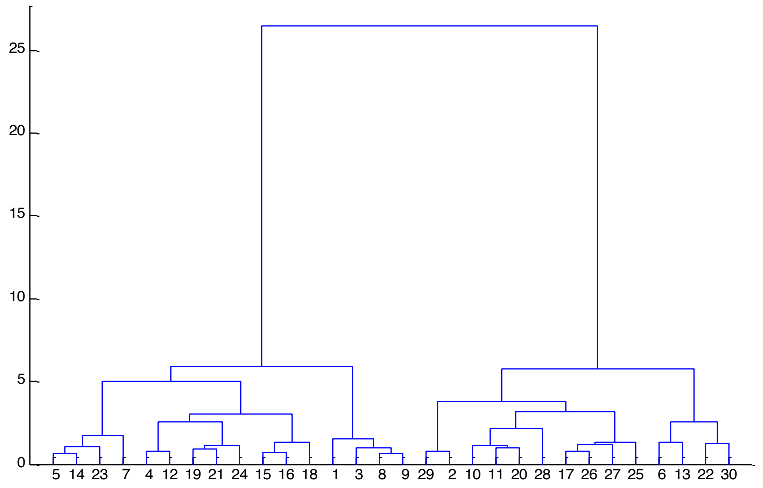




Single linkage



Average linkage



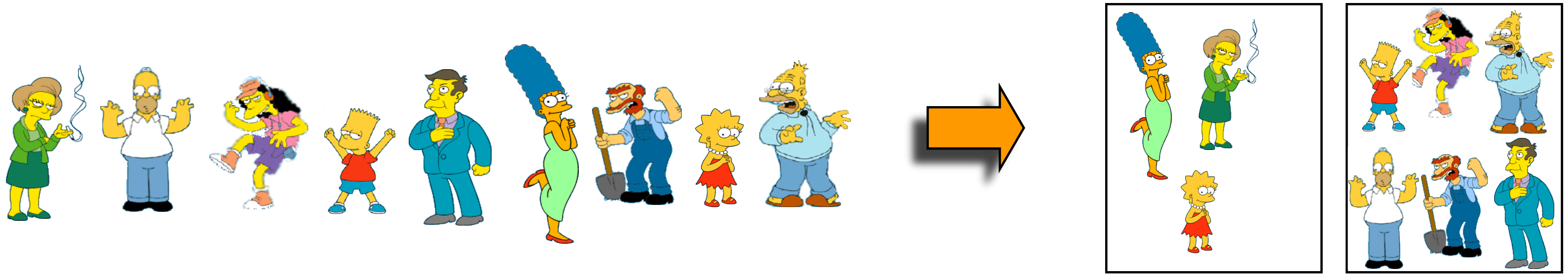
Wards linkage

Summary of Hierarchal Clustering Methods

- No need to specify the number of clusters in advance.
- Hierarchal nature 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.

Partitional Clustering

- Nonhierarchical, each instance is placed in exactly one of K nonoverlapping clusters.
- Since only one set of clusters is output, the user normally has to input the desired number of clusters K.

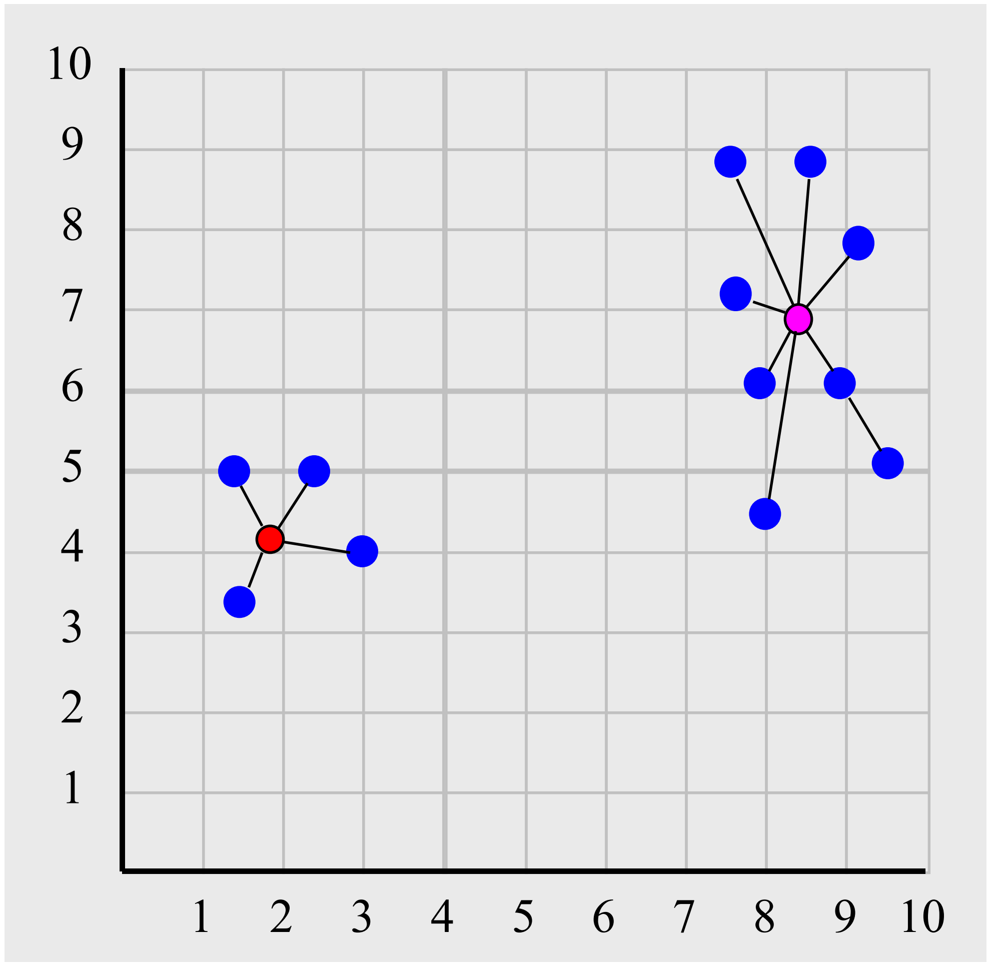


Squared Error

$$se_{K_i} = \sum_{j=1}^{m} \|t_{ij} - C_k\|^2$$

$$se_K = \sum_{j=1}^{k} se_{K_j}$$

Objective Function



Algorithm *k-means*

1. Decide on a value for *k*.

2. Initialize the *k* cluster centers (randomly, if necessary).

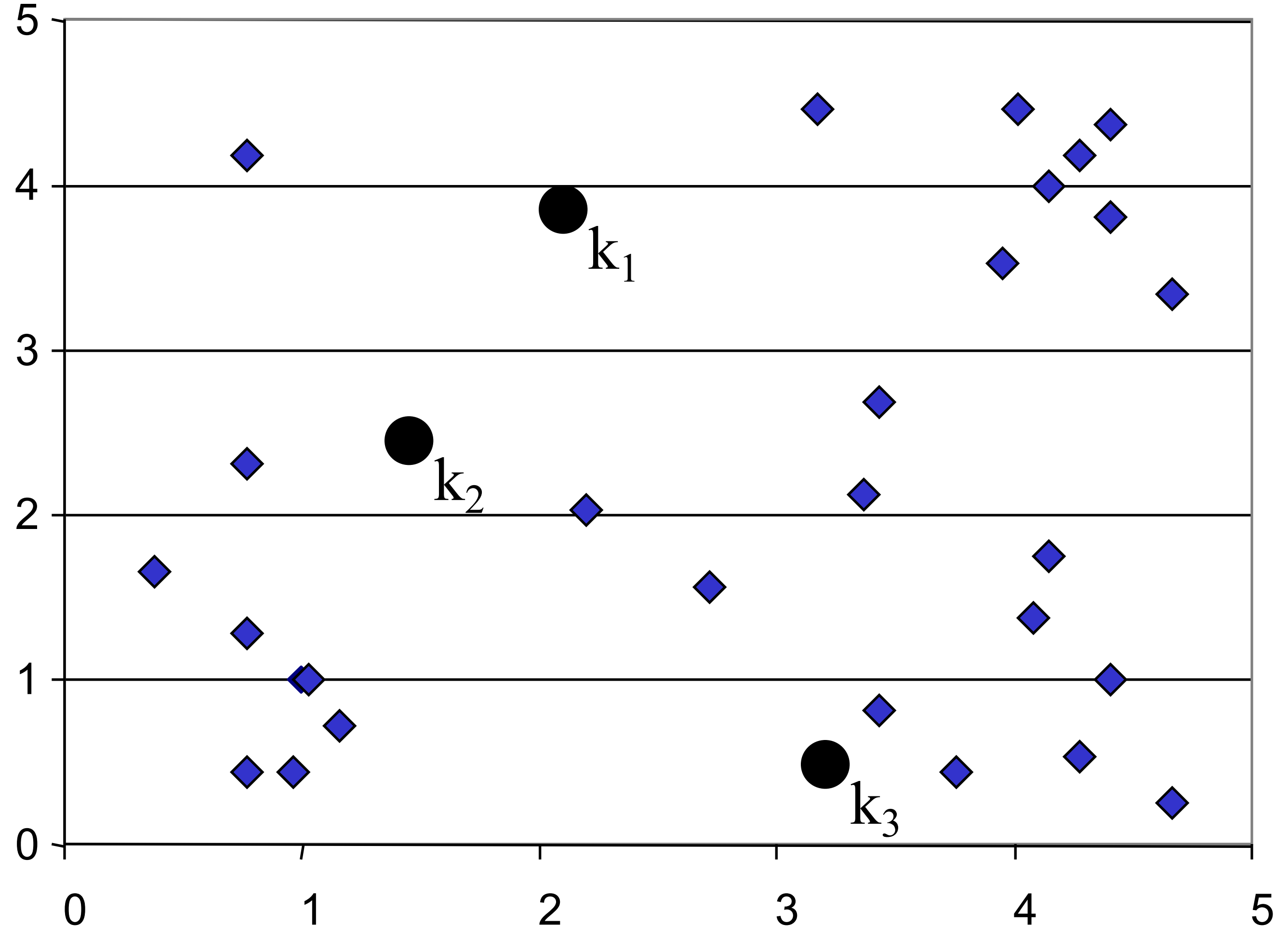
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

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

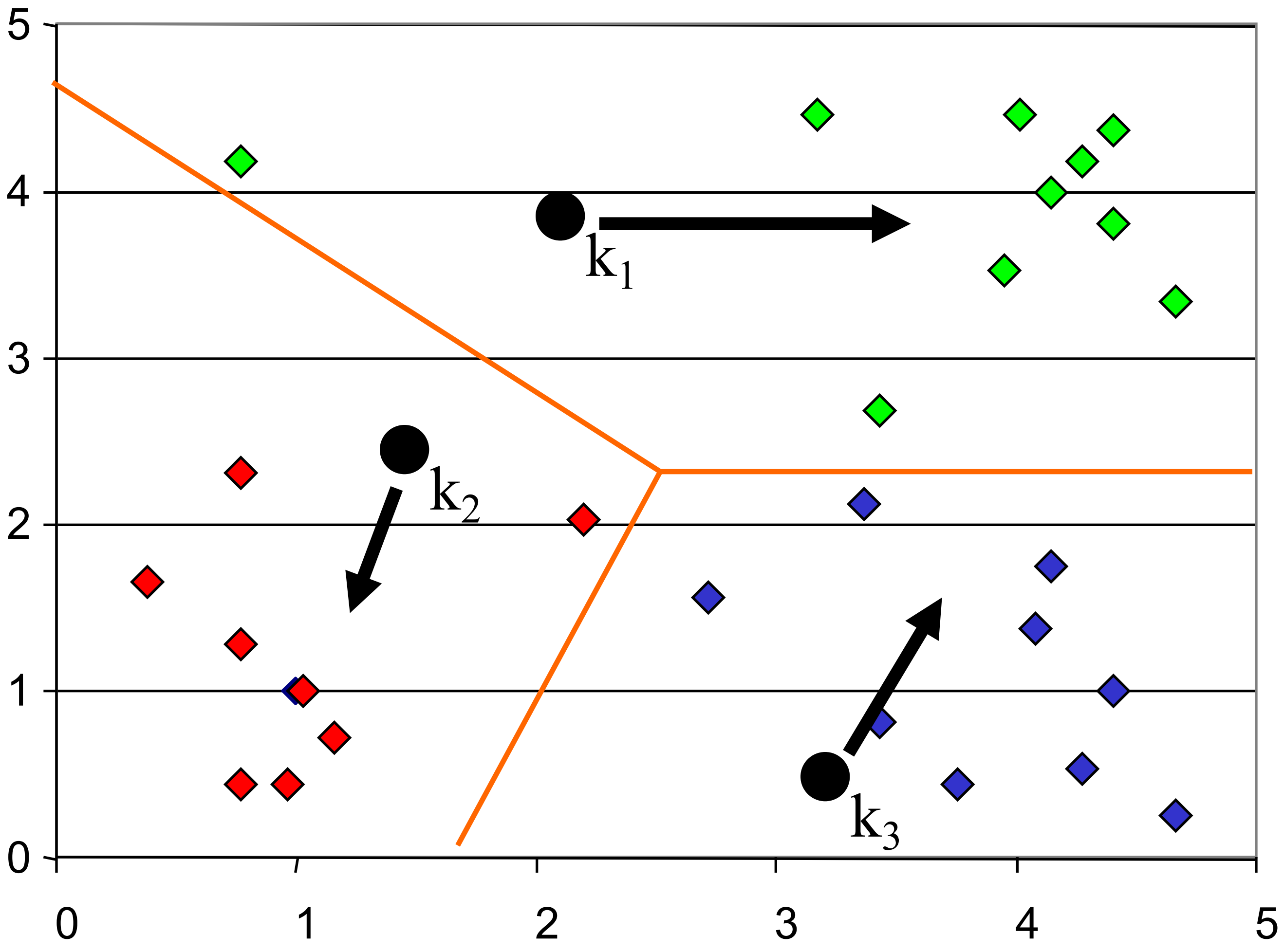
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



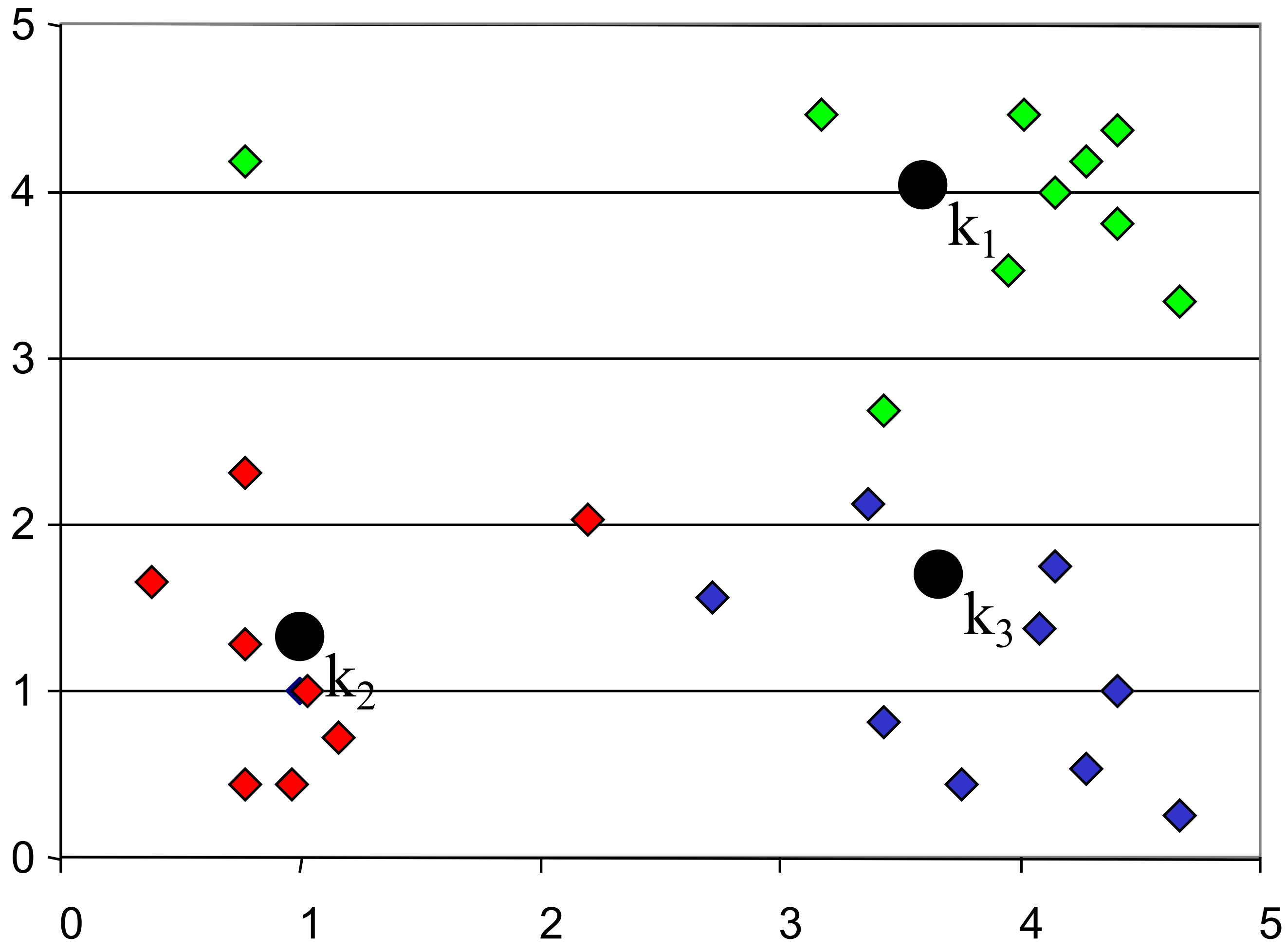
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



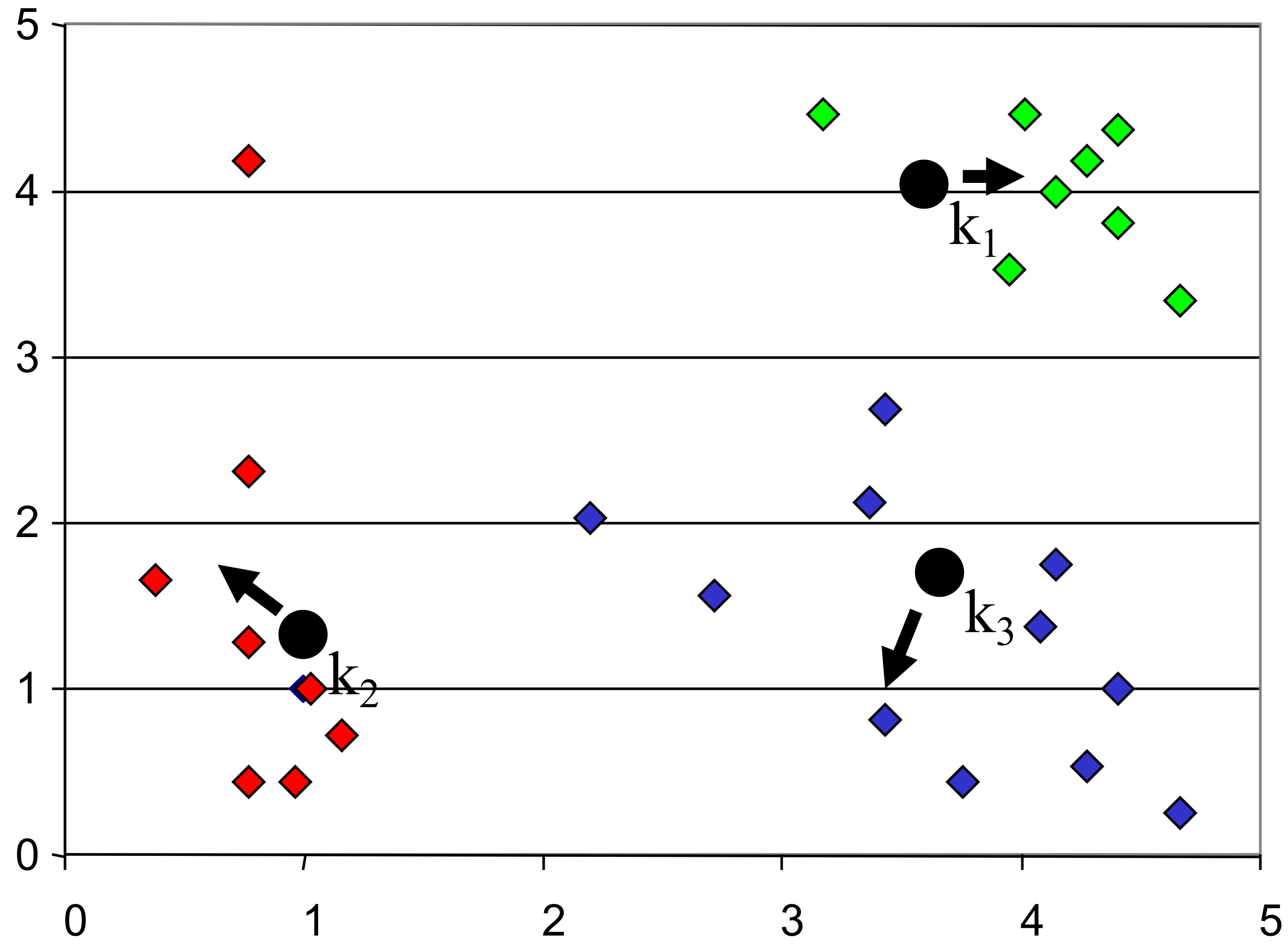
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



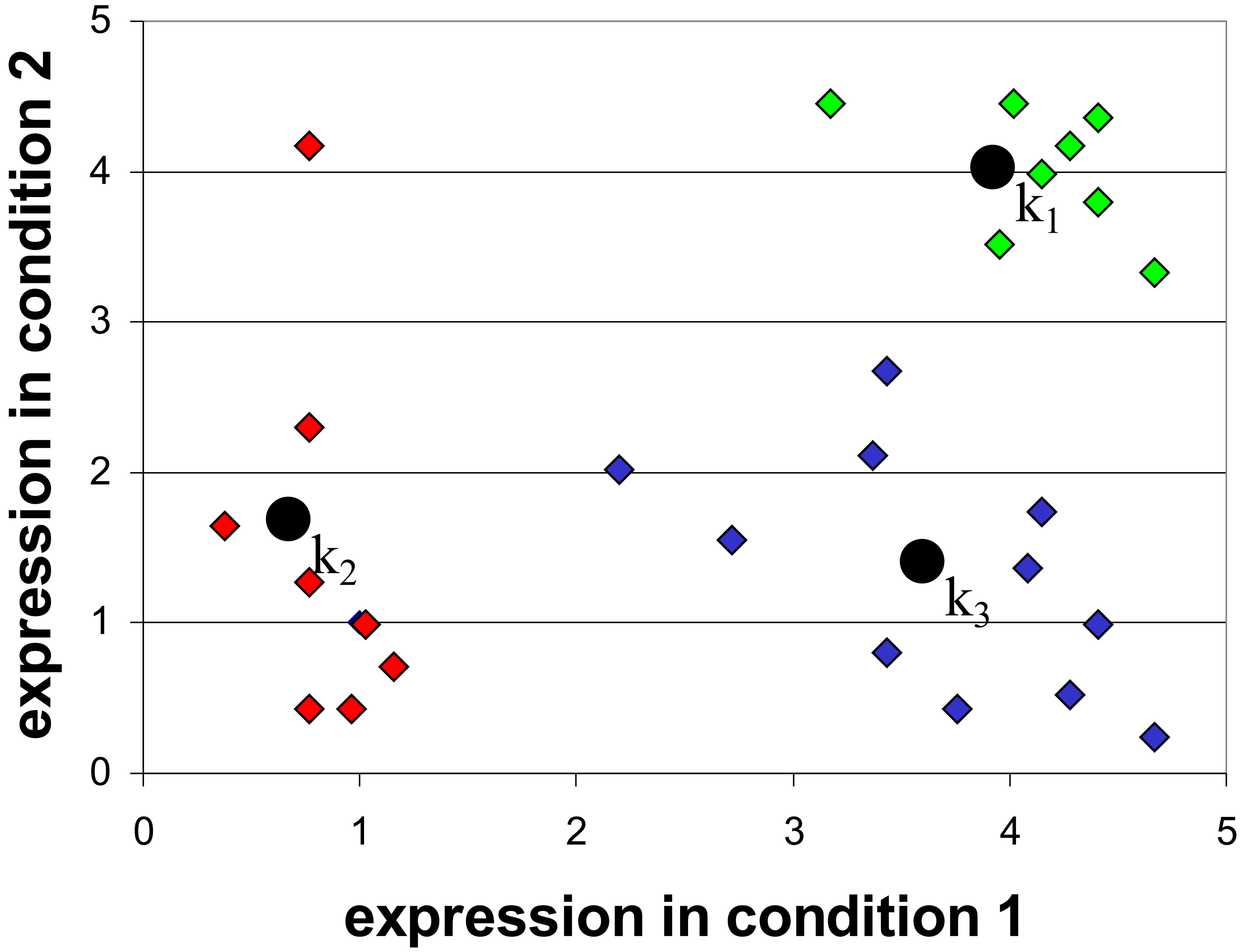
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



Comments on the *K-Means* Method

- <u>Strength</u>
  - *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, $t << n$.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- <u>Weakness</u>
  - Applicable only when *mean* is defined, then what about categorical data?
  - 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*

EM Algorithm

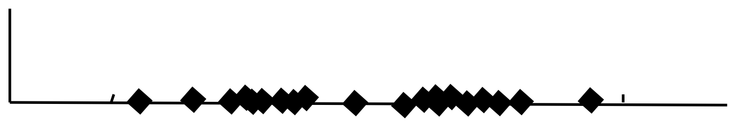
- Initialize K cluster centers
- Iterate between two steps
  - **E**xpectation step: assign points to clusters

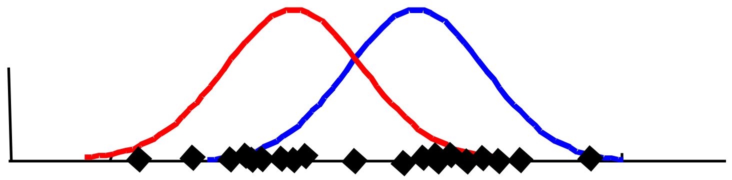
$$P(d_i \in c_k) = w_k \Pr(d_i \mid c_k) \Big/ \sum_j w_j \Pr(d_i \mid c_j)$$

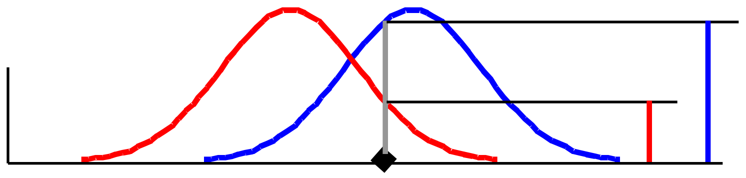
$$w_k = \frac{\sum_i \Pr(d_i \in c_k)}{N}$$

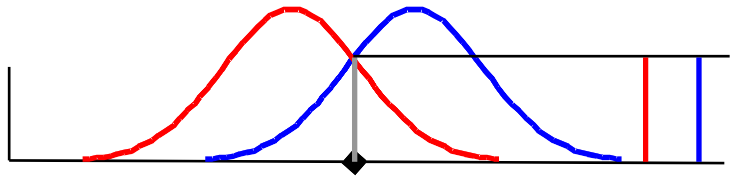
  - **M**aximation step: estimate model parameters

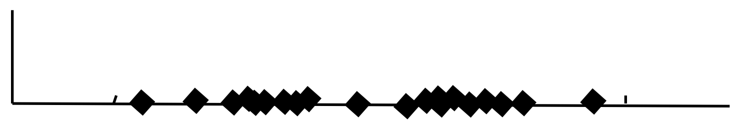
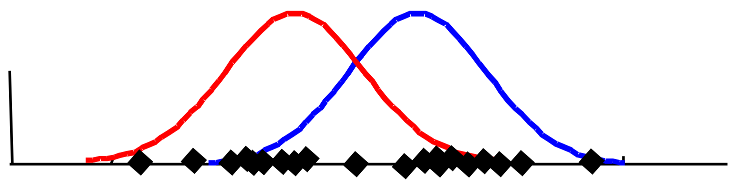
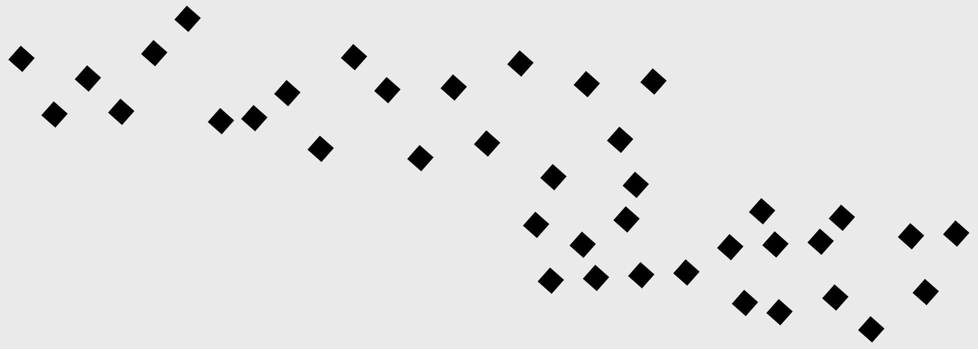
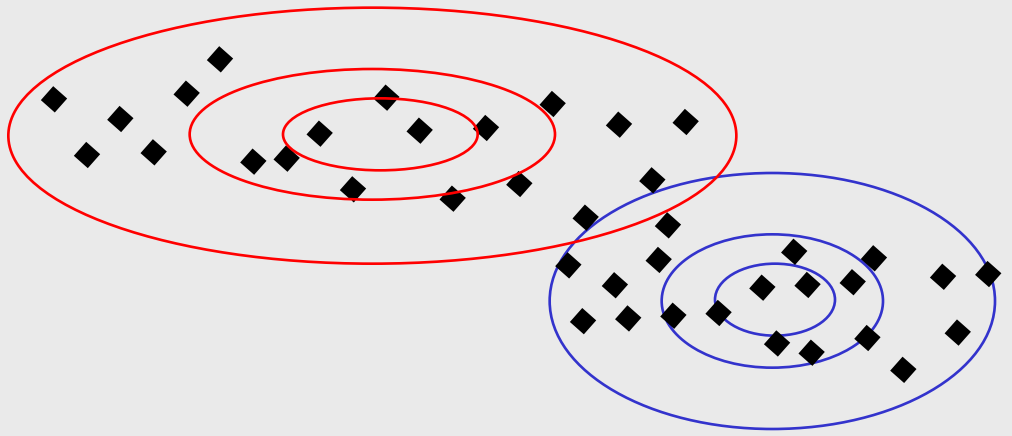
$$\mu_k = \frac{1}{m} \sum_{i=1}^{m} \frac{d_i P(d_i \in c_k)}{\sum_k P(d_i \in c_j)}$$

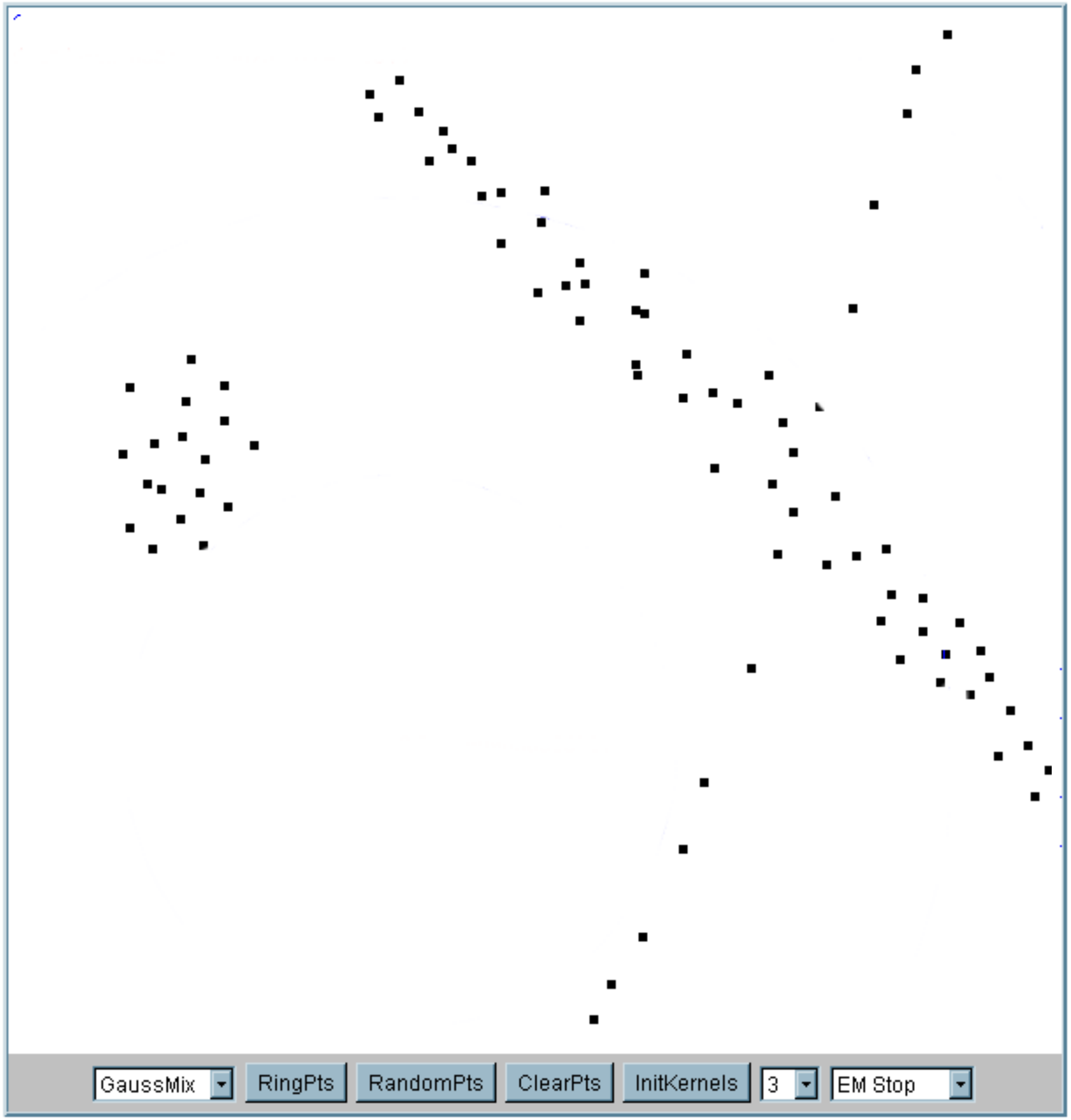






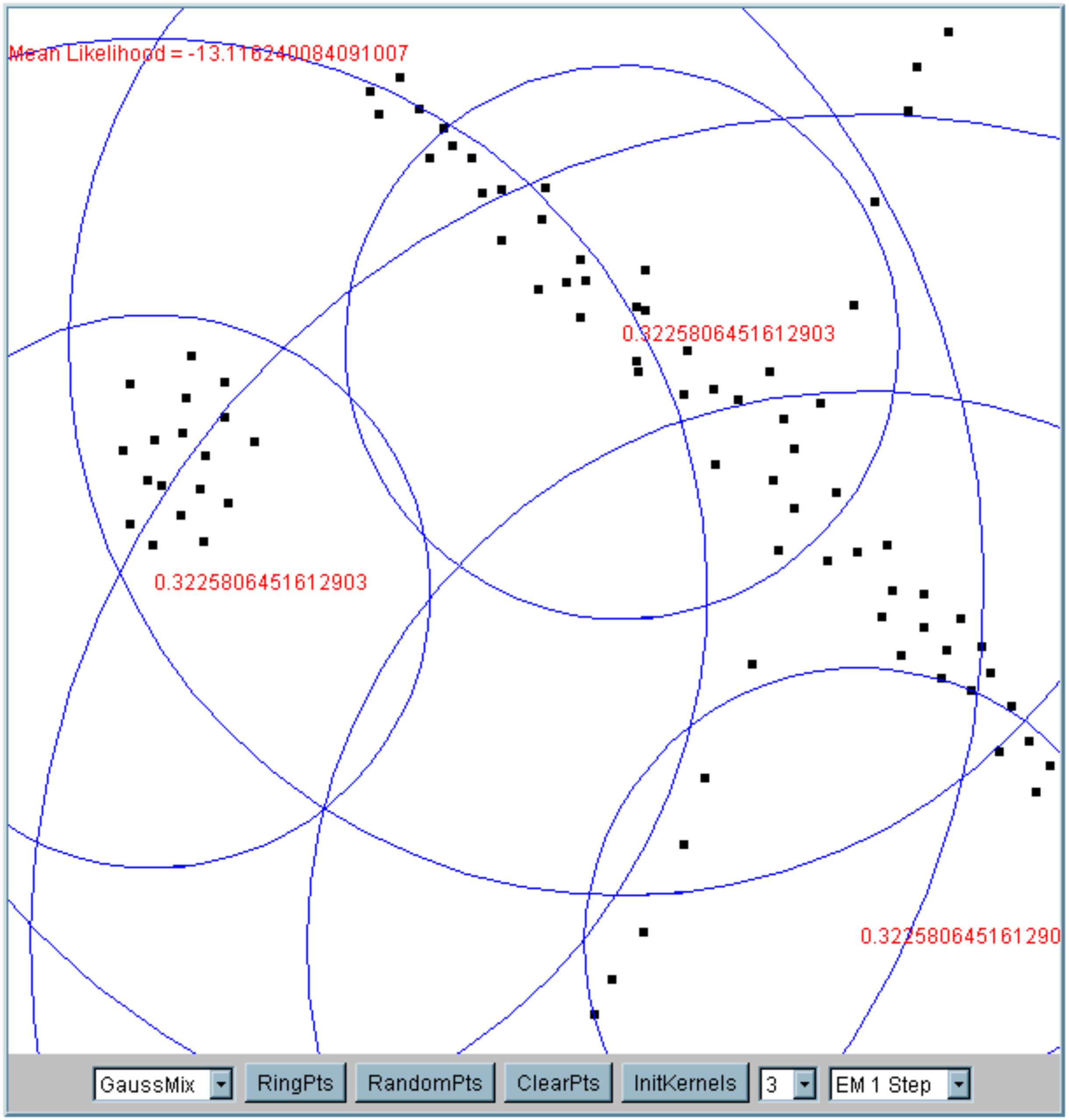


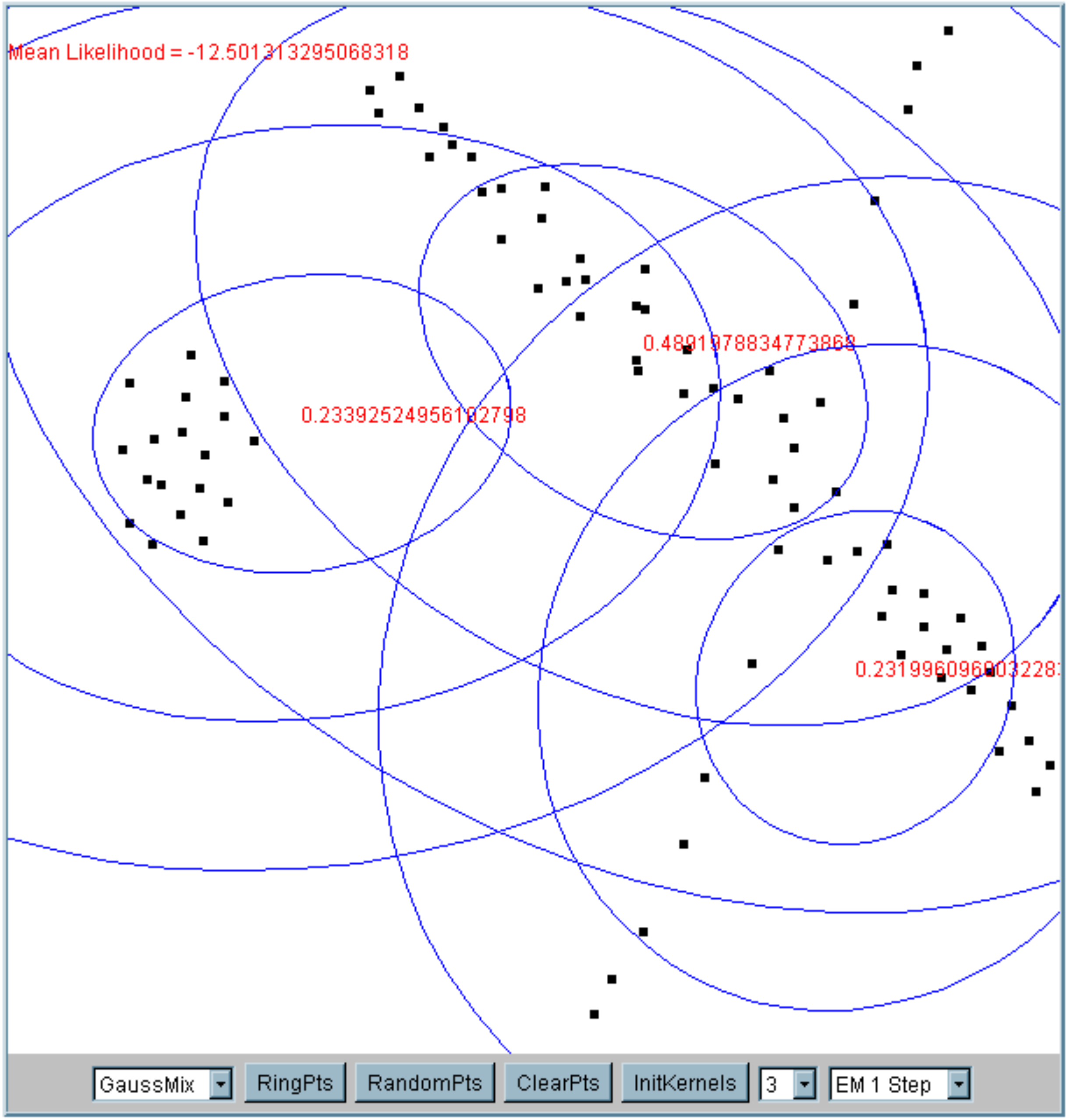

GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM Stop

Iteration 1

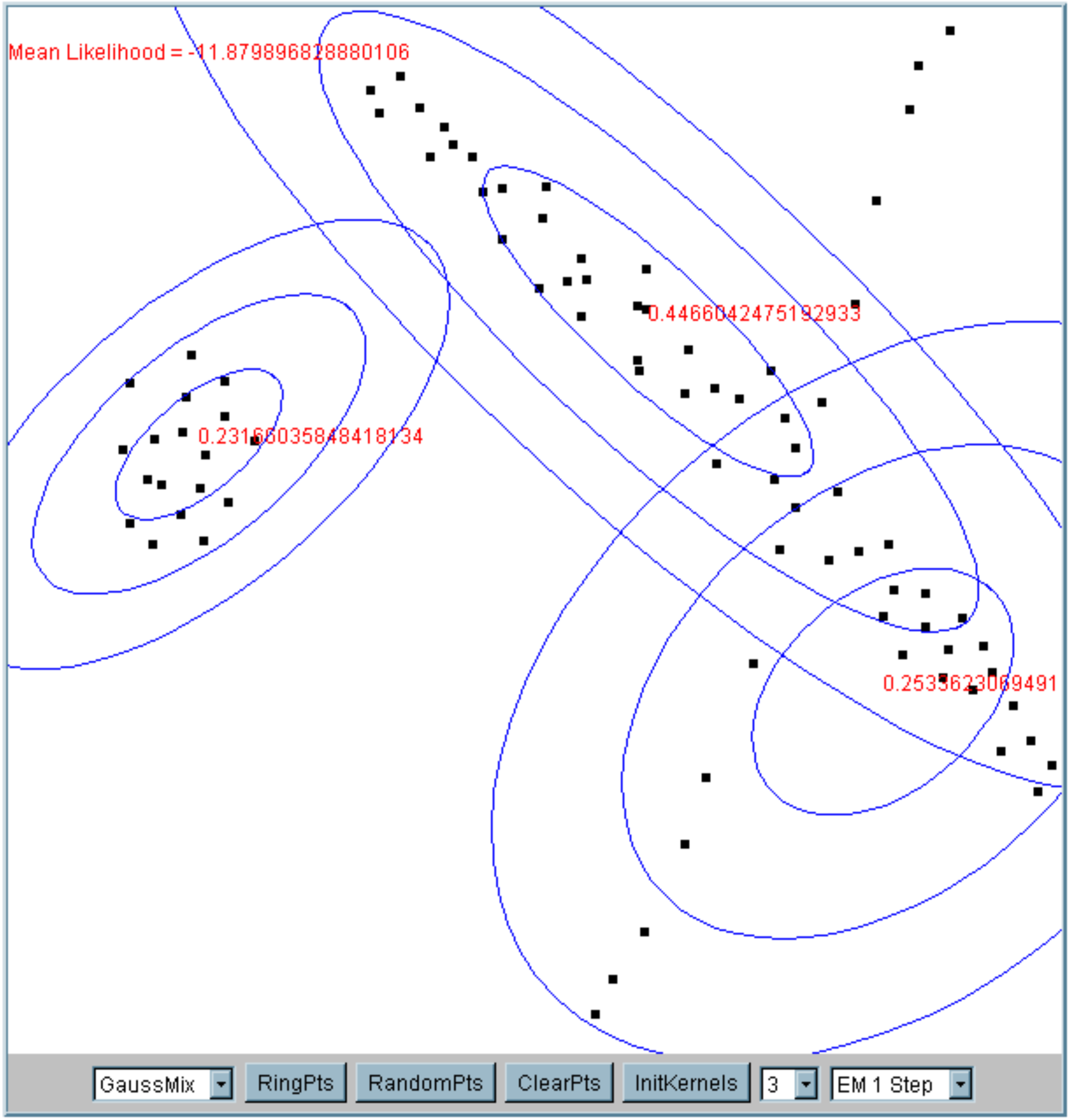
The cluster means are randomly assigned



Iteration 2


Mean Likelihood = -12.501313295068318
0.4891978834773868
0.23392524956102798
0.231996096003228
GaussMix
RingPts
RandomPts
ClearPts
InitKernels
3
EM 1 Step

Iteration 5

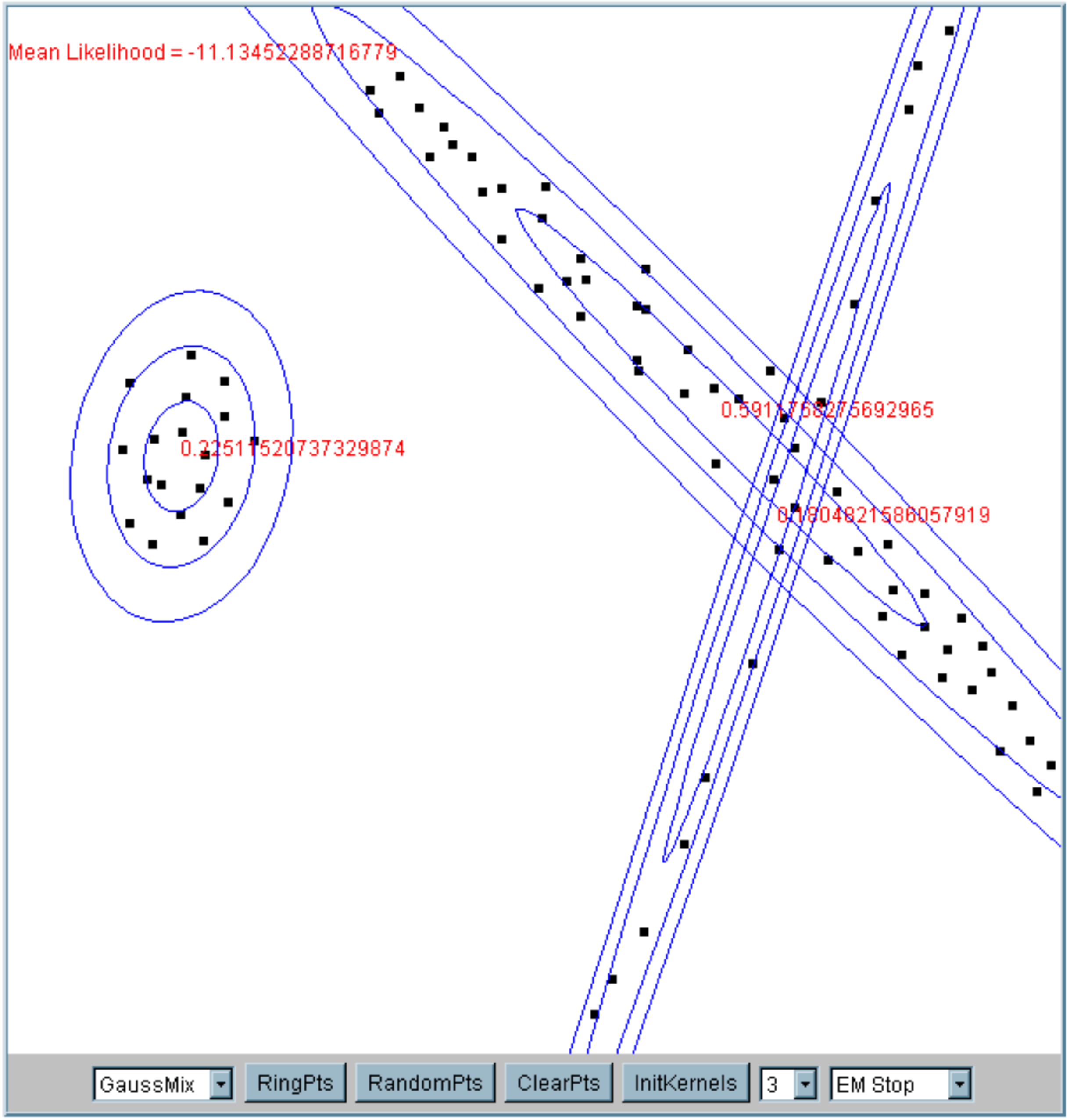

Mean Likelihood = -11.879896828880106

0.4466042475192933

0.23166035848418134

0.2536623069491

Iteration 25


Mean Likelihood = -11.13452288716779

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

Determining the number of clusters is a model selection problem that can be answered depending on the application:

- quantization: rate/distortion tradeoff
- density estimation: bias/variance tradeoff
- cluster analysis: data fitness

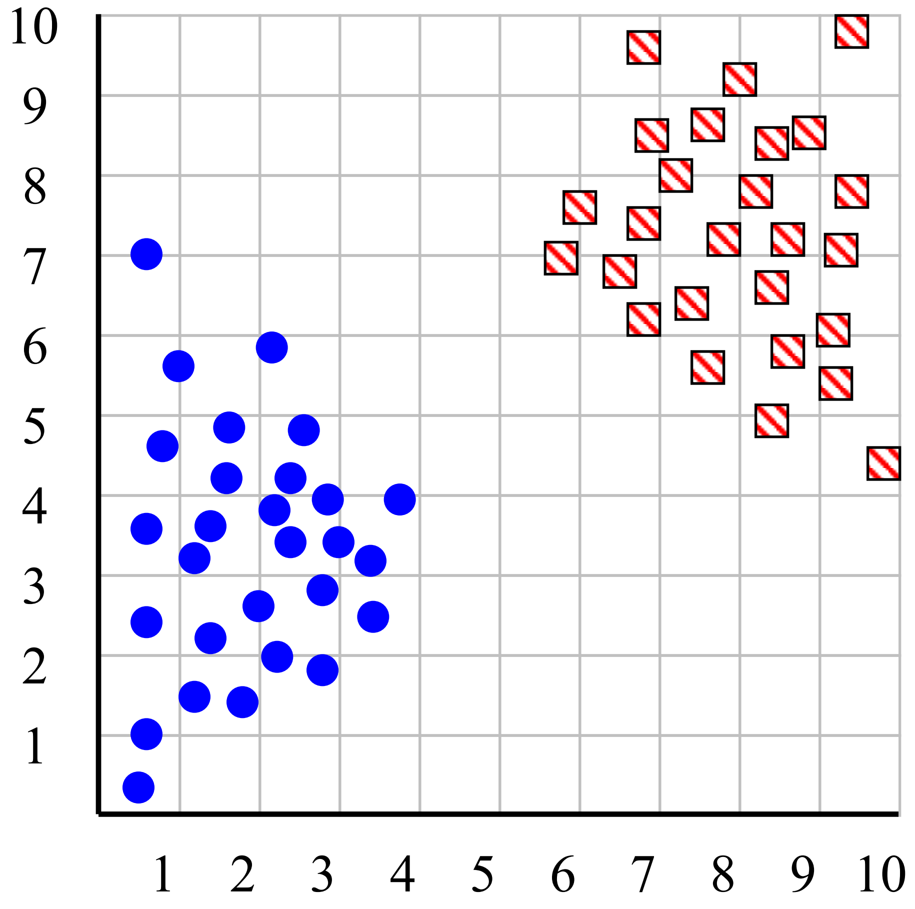
How can we tell the *right* number of clusters?

Many indicators can be considered to estimate the goodness of fit.

- But it is usually make sense to consider the one considered during the clustering.
- Usually some sort of average intra cluster distance
- The problem with this kind of criterion is that the average intra class distance DK (X) over the dataset X decreases monotonously with K.
- Need for some sort of normalization.

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

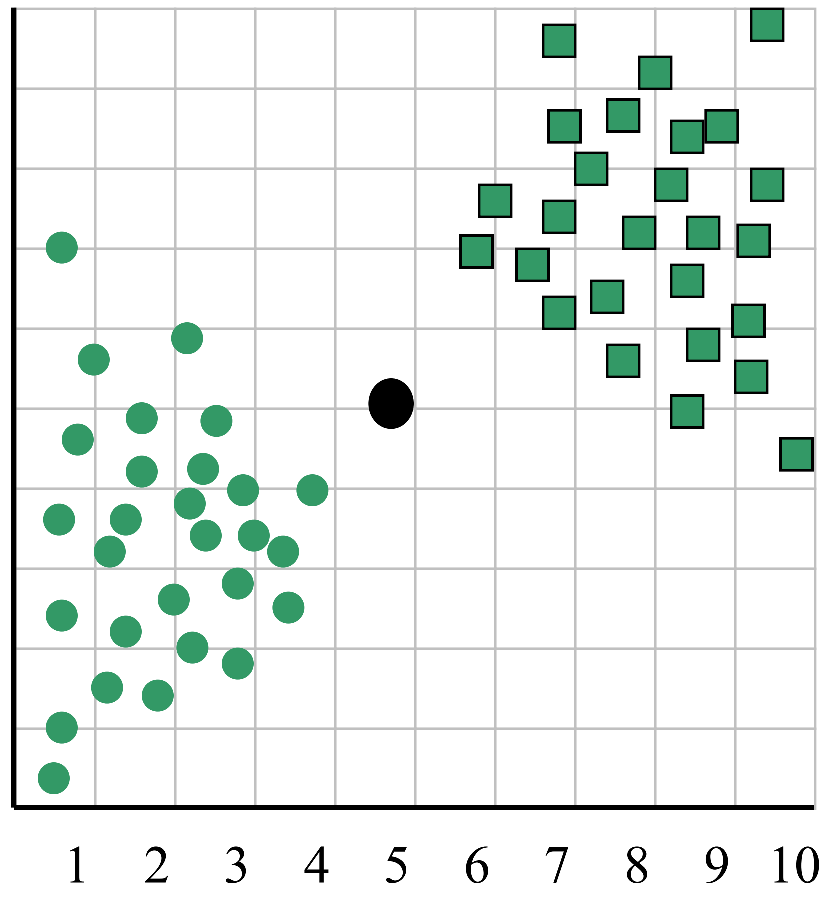
In general, this is a unsolved problem. However there are many approximate methods. In the next few slides we will see an example.



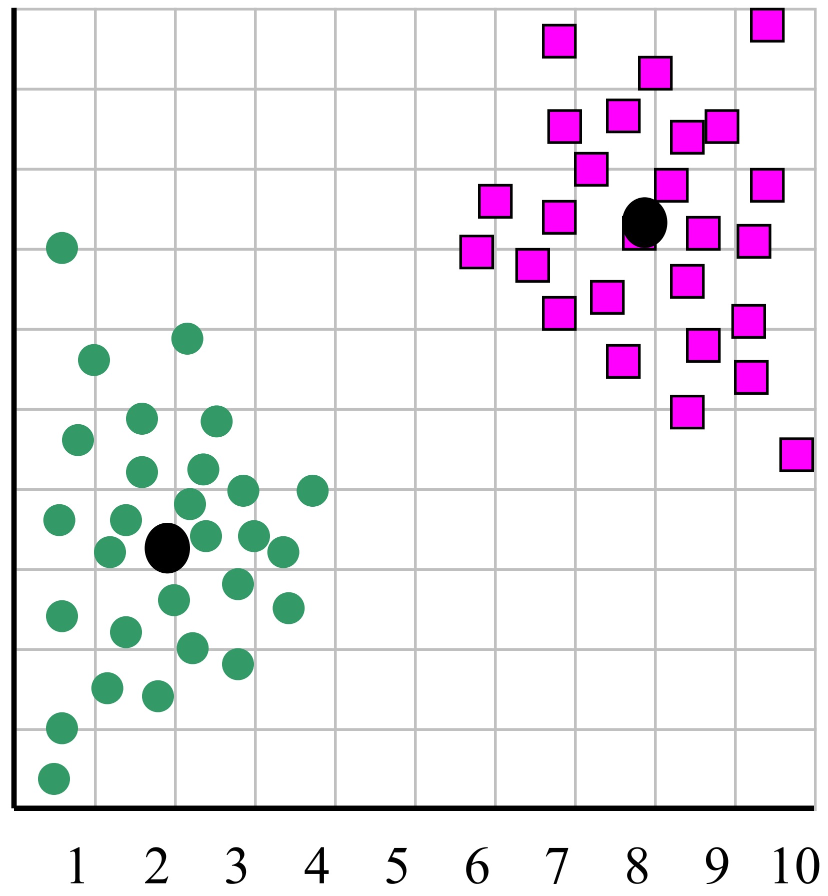
For our example, we will use the familiar katydid/grasshopper dataset.

However, in this case we are imagining that we do NOT know the class labels. We are only clustering on the X and Y axis values.

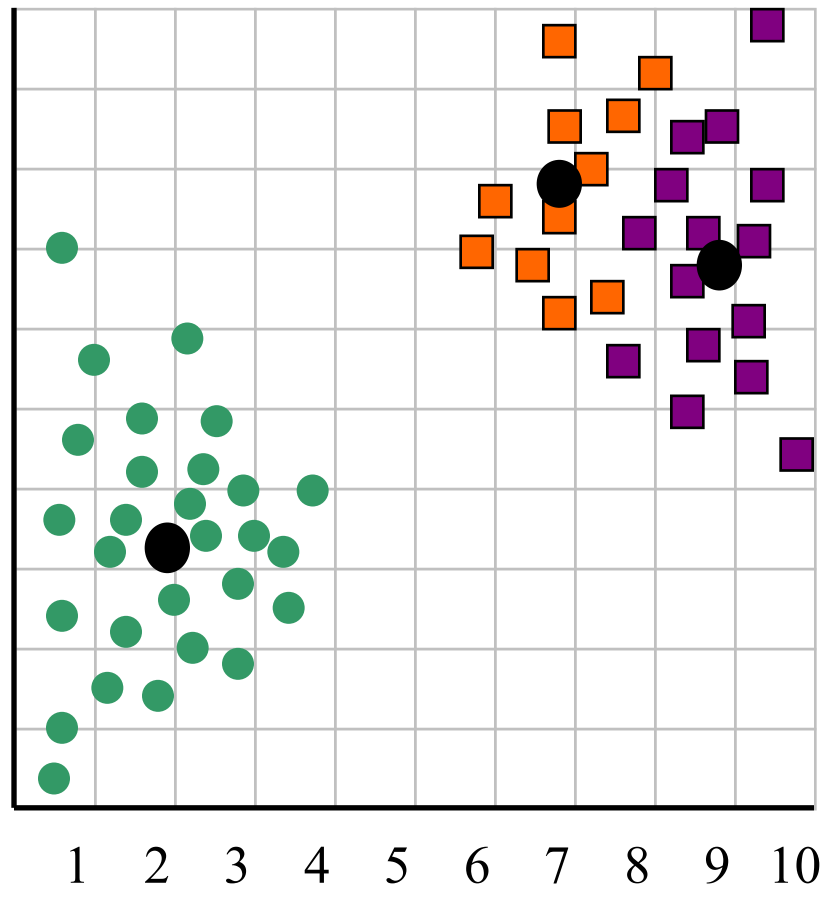
When k = 1, the objective function is 873.0



When k = 2, the objective function is 173.1

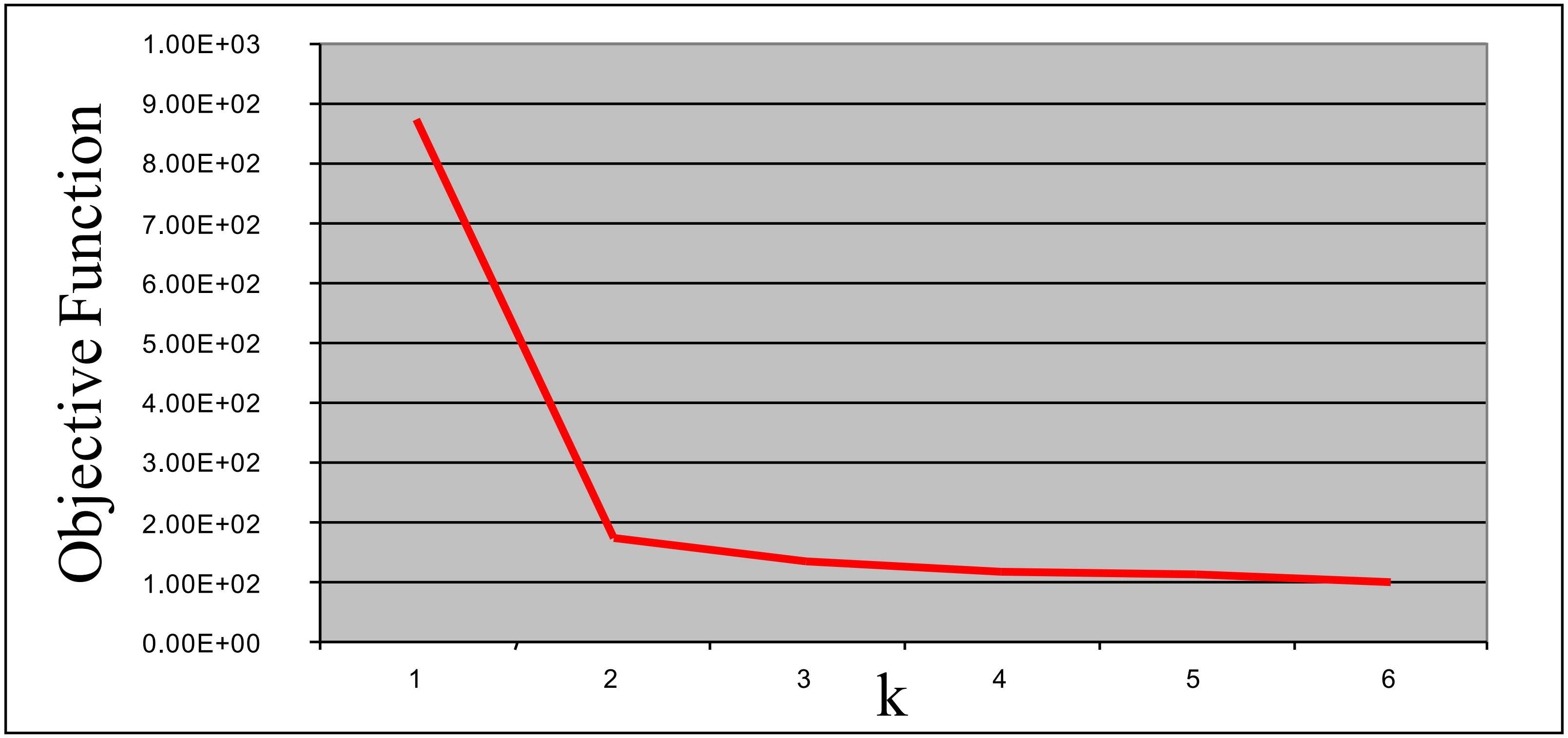


When k = 3, the objective function is 133.6



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

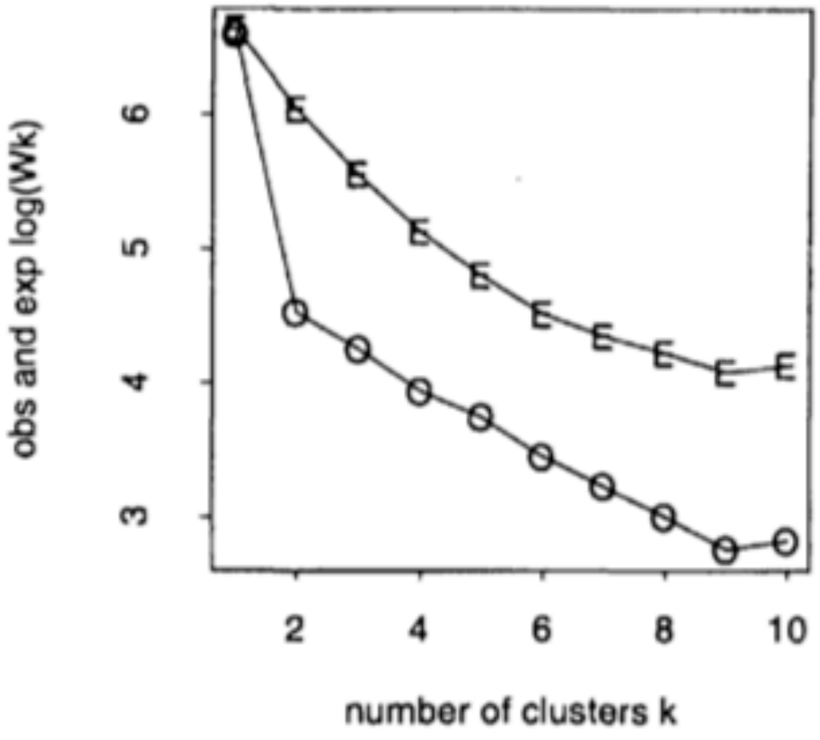
The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as "knee finding" or "elbow finding".



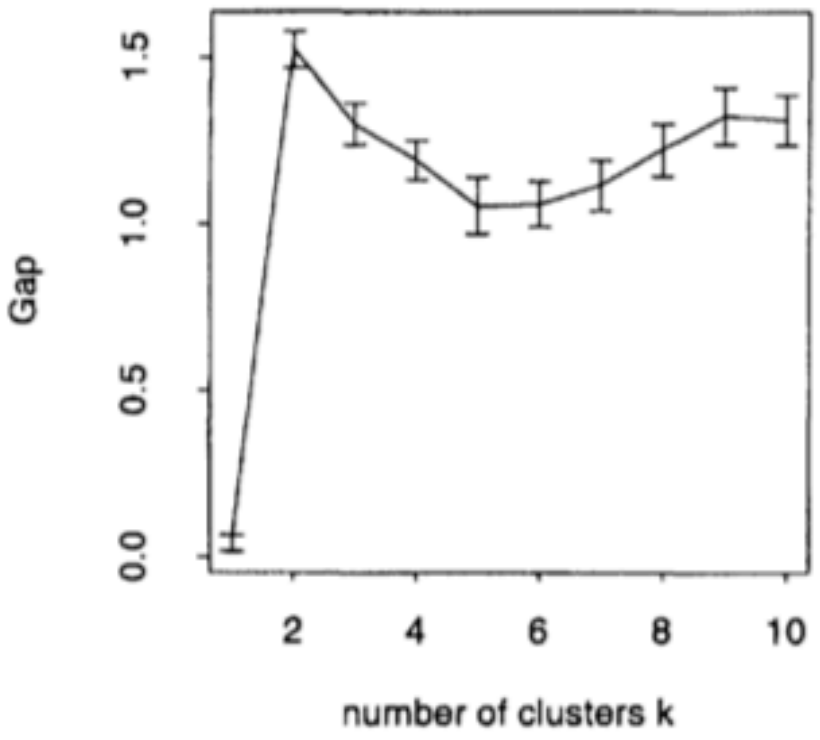
Note that the results are not always as clear cut as in this toy example

The Gap Statistic

- normalizes Dk with is expectation under an appropriate null reference of the distribution of the data.
- Gk (X) = Dk (X) / DK (null)



(c)



(d)

How do we compare a clustering to reference ?

[0, 0, 1, 1, 2, 1, 2, 2] = ? = [2, 2, 0, 0, 1, 0, 1, 1]

Need to find a workaround for label permutation

Easy when clustering are almost similar

Numerous methods (see scikit-learn page on Clustering performance evaluation):

1. Adjusted Rand Index (ARI)
2. Normalized Mutual Information (NMI)

Principle of the Rand Index (RI)

Given two partitions X and Y define the following:
a, the number of pairs of elements that are in the same subset in X and in the same subset in Y
b, the number of pairs of elements that are in different subsets in X and in different subsets in Y
c, the number of pairs of elements that are in the same subset in X and in different subsets in Y
d, the number of pairs of elements that are in different subsets in X and in the same subset in Y

$RI(X, Y) = (a+b)/(a+b+c+d)$

The Rand index does not ensure to obtain a value close to 0.0 for a random labelling. The Adjusted Rand Index **corrects for chance** and will give such a baseline.