

Fair Use Agreement

This agreement covers the use of all slides on this CD-Rom, please read carefully.

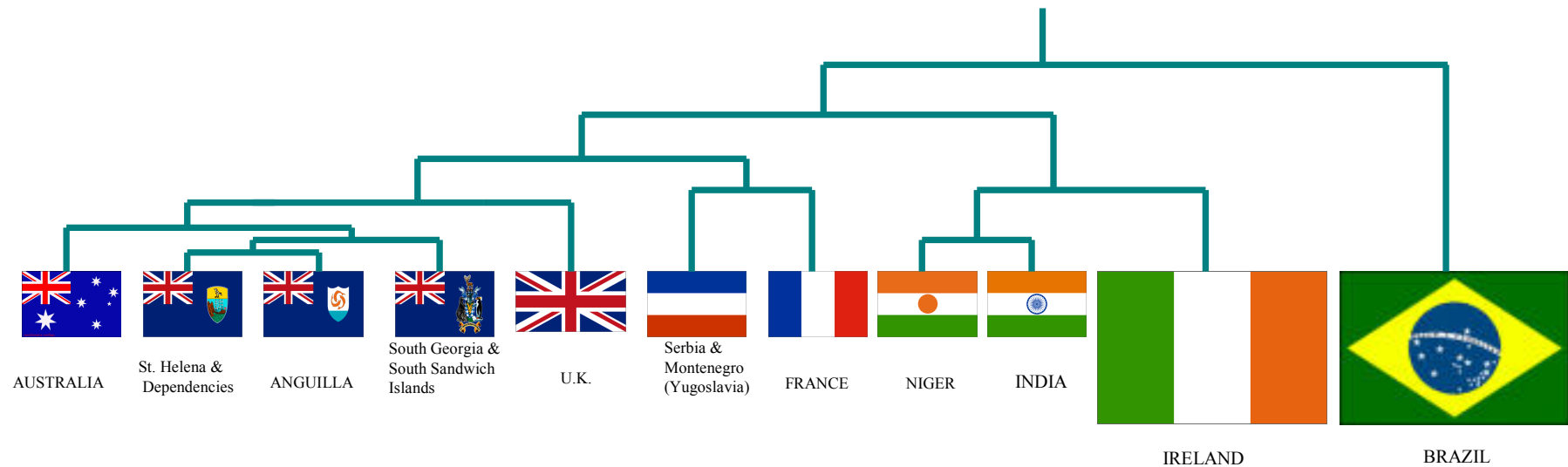
- You may freely use these slides for teaching, if
 - You send me an email telling me the class number/ university in advance.
 - My name and email address appears on the first slide (if you are using all or most of the slides), or on each slide (if you are just taking a few slides).
- You may freely use these slides for a conference presentation, if
 - You send me an email telling me the conference name in advance.
 - My name appears on each slide you use.
- You may not use these slides for tutorials, or in a published work (tech report/ conference paper/ thesis/ journal etc). If you wish to do this, email me first, it is highly likely I will grant you permission.

(c) Eamonn Keogh, eamonn@cs.ucr.edu

A Gentle Introduction to Machine Learning

Dr Eamonn Keogh

University of California - Riverside
eamonn@cs.ucr.edu

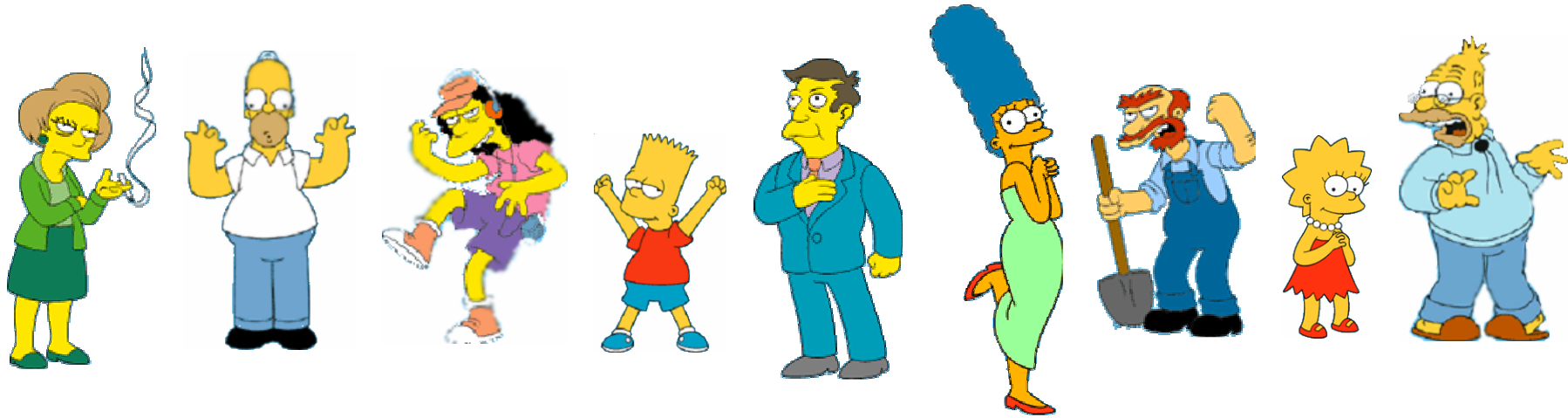


What is Clustering?

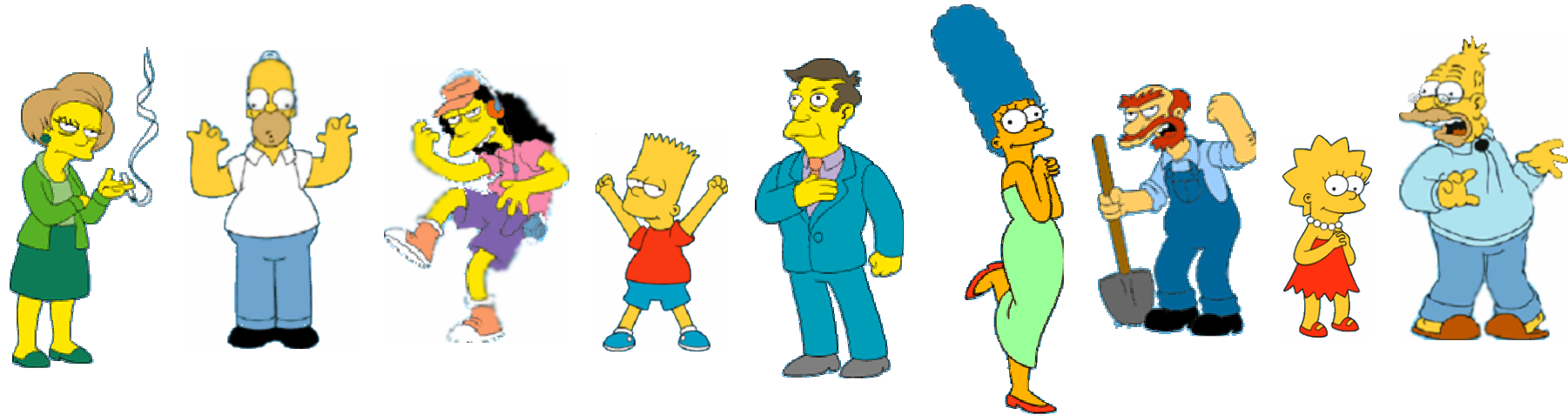
Also called *unsupervised learning*, sometimes called *classification* by statisticians and *sorting* by psychologists and *segmentation* by people in marketing

- 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.

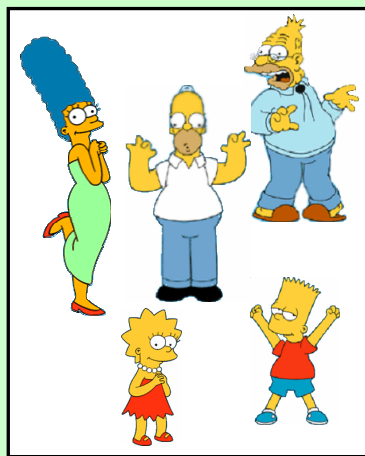
What is a natural grouping among these objects?



What is a natural grouping among these objects?



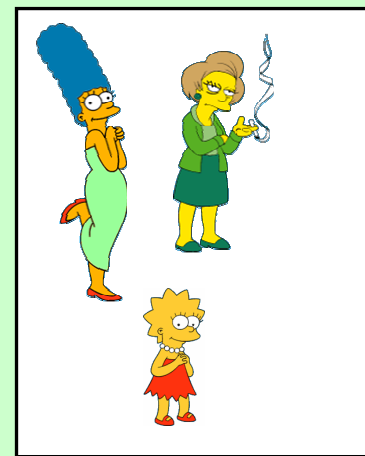
Clustering is subjective



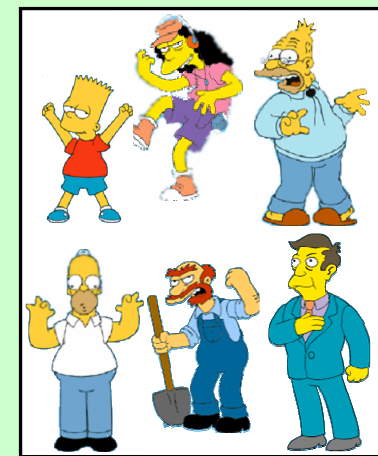
Simpson's Family



School Employees



Females



Males

What is Similarity?

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

Webster's Dictionary

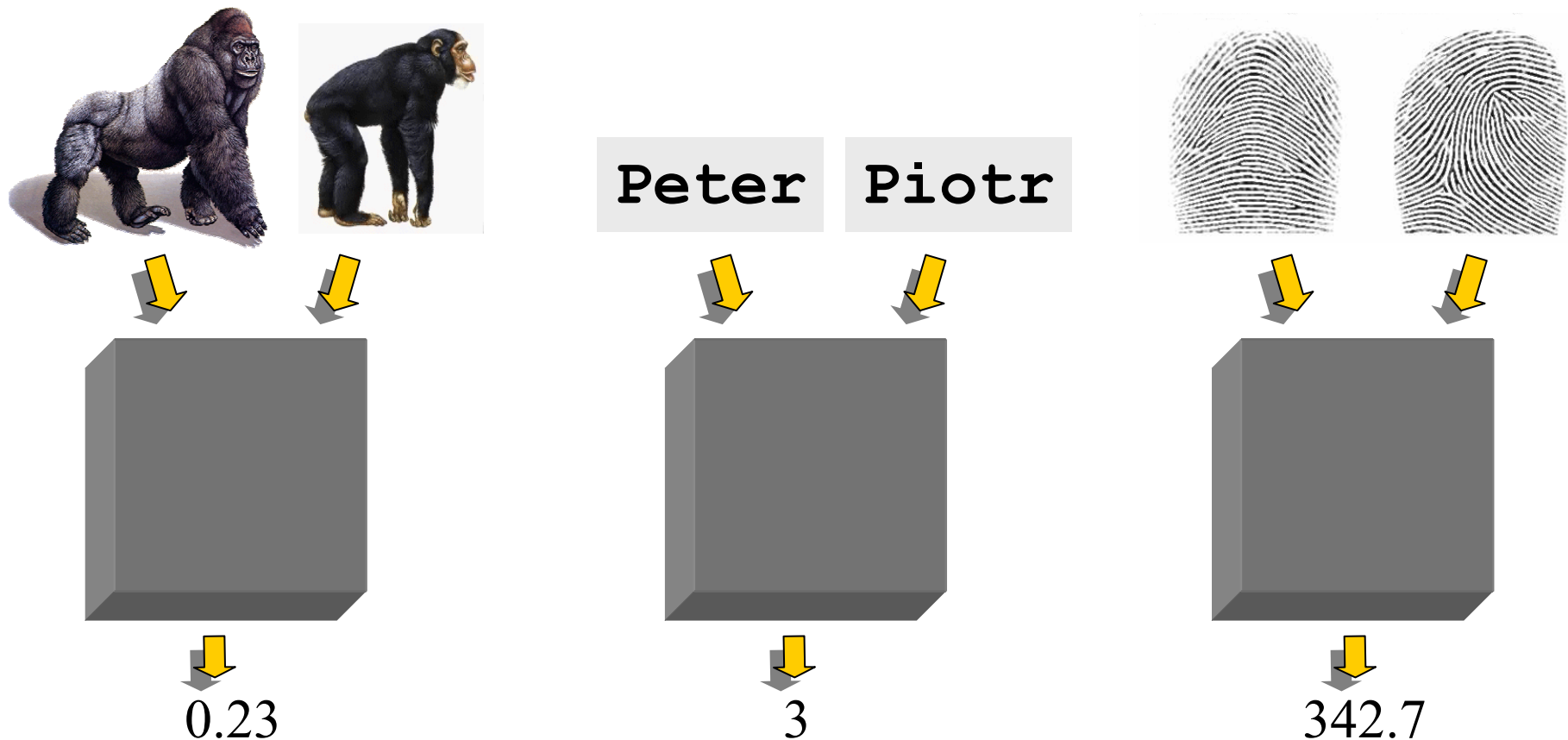


Similarity is hard to define, but...
"We know it when we see it"

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

Defining Distance Measures

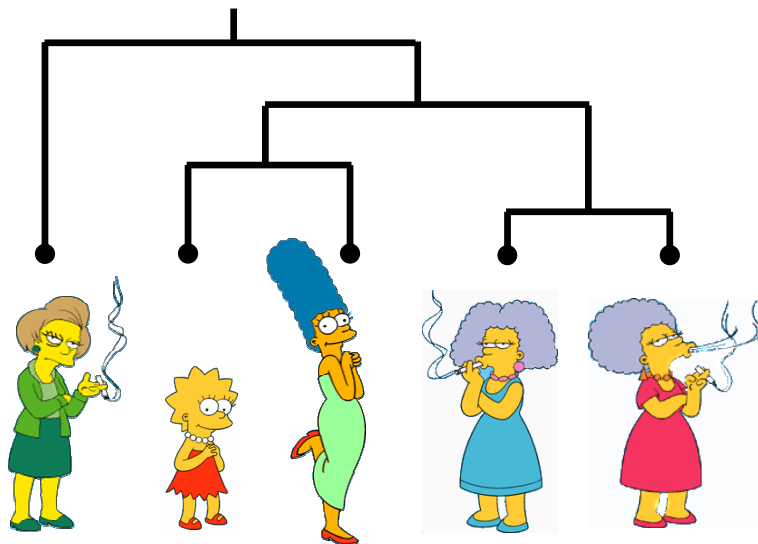
Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1, O_2)$



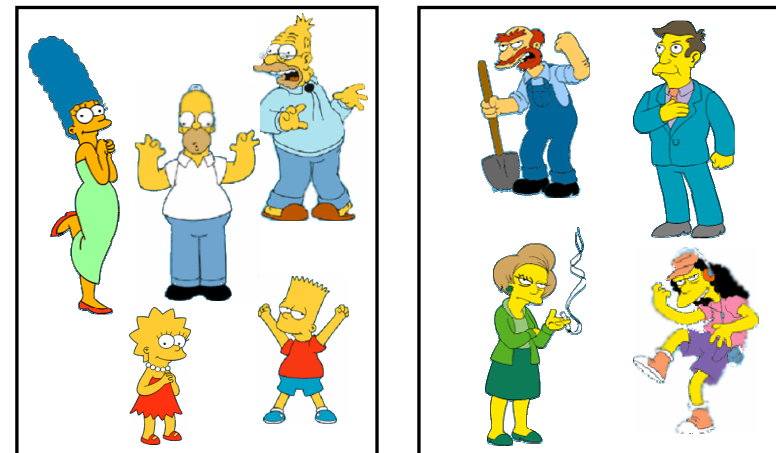
Two Types of Clustering

- **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

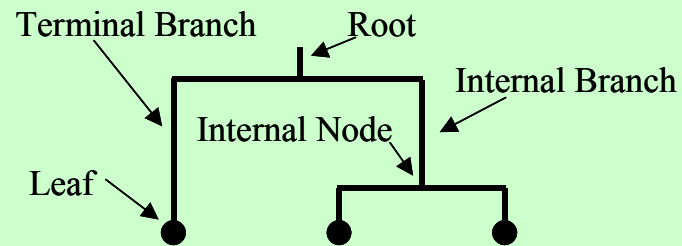


Desirable Properties of a Clustering Algorithm

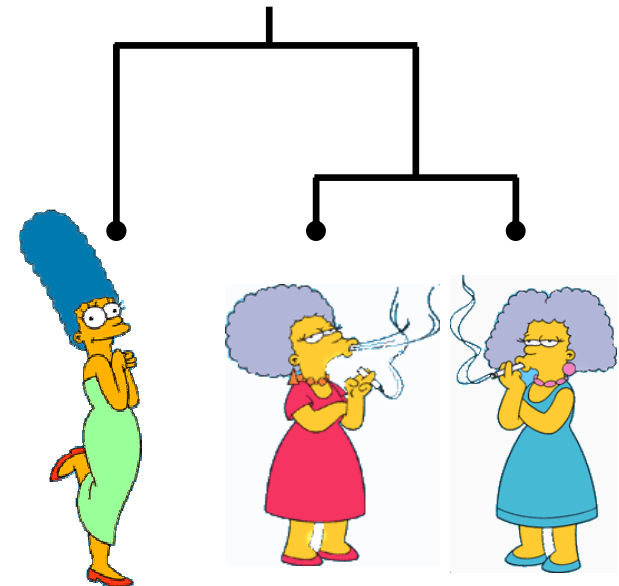
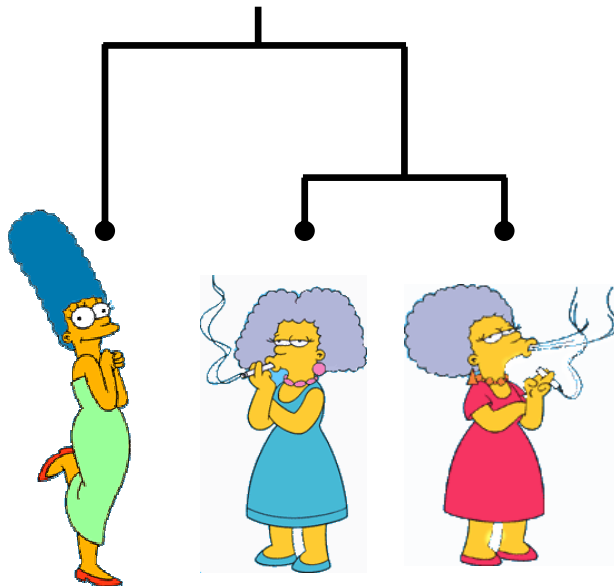
- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- Incorporation of user-specified constraints
- Interpretability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



Note that hierarchies are commonly used to organize information, for example in a web portal.

Yahoo's hierarchy is manually created, we will focus on automatic creation of hierarchies in data mining.

Web Site Directory - Sites organized by subject

[Suggest your site](#)

Business & Economy

[B2B](#), [Finance](#), [Shopping](#), [Jobs](#)...

Regional

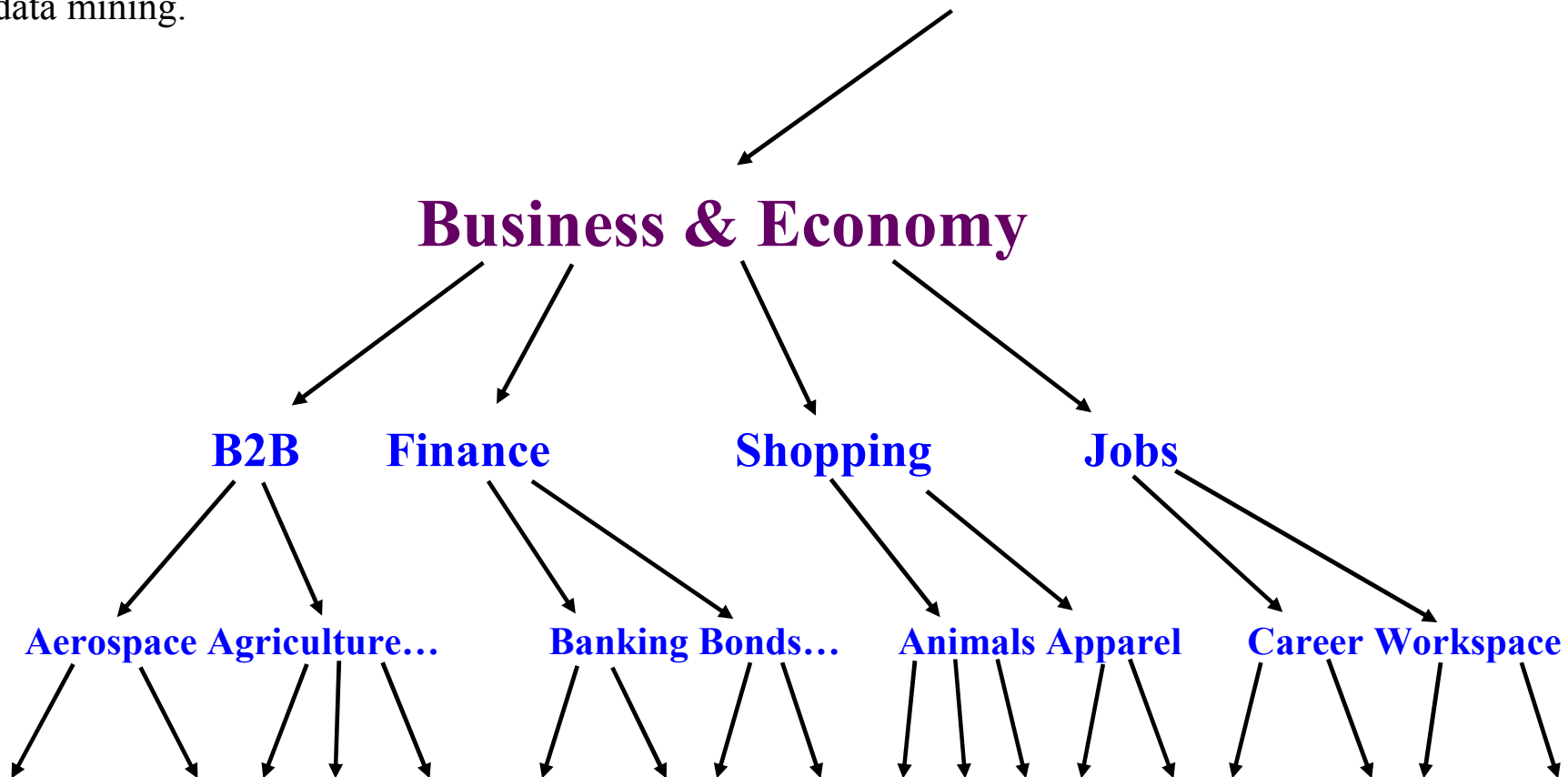
[Countries](#), [Regions](#), [US States](#)...

Computers & Internet

[Internet](#), [WWW](#), [Software](#), [Games](#)...

Society & Culture

[People](#), [Environment](#), [Religion](#)...



A Demonstration of Hierarchical Clustering using String Edit Distance

Pedro (Portuguese)

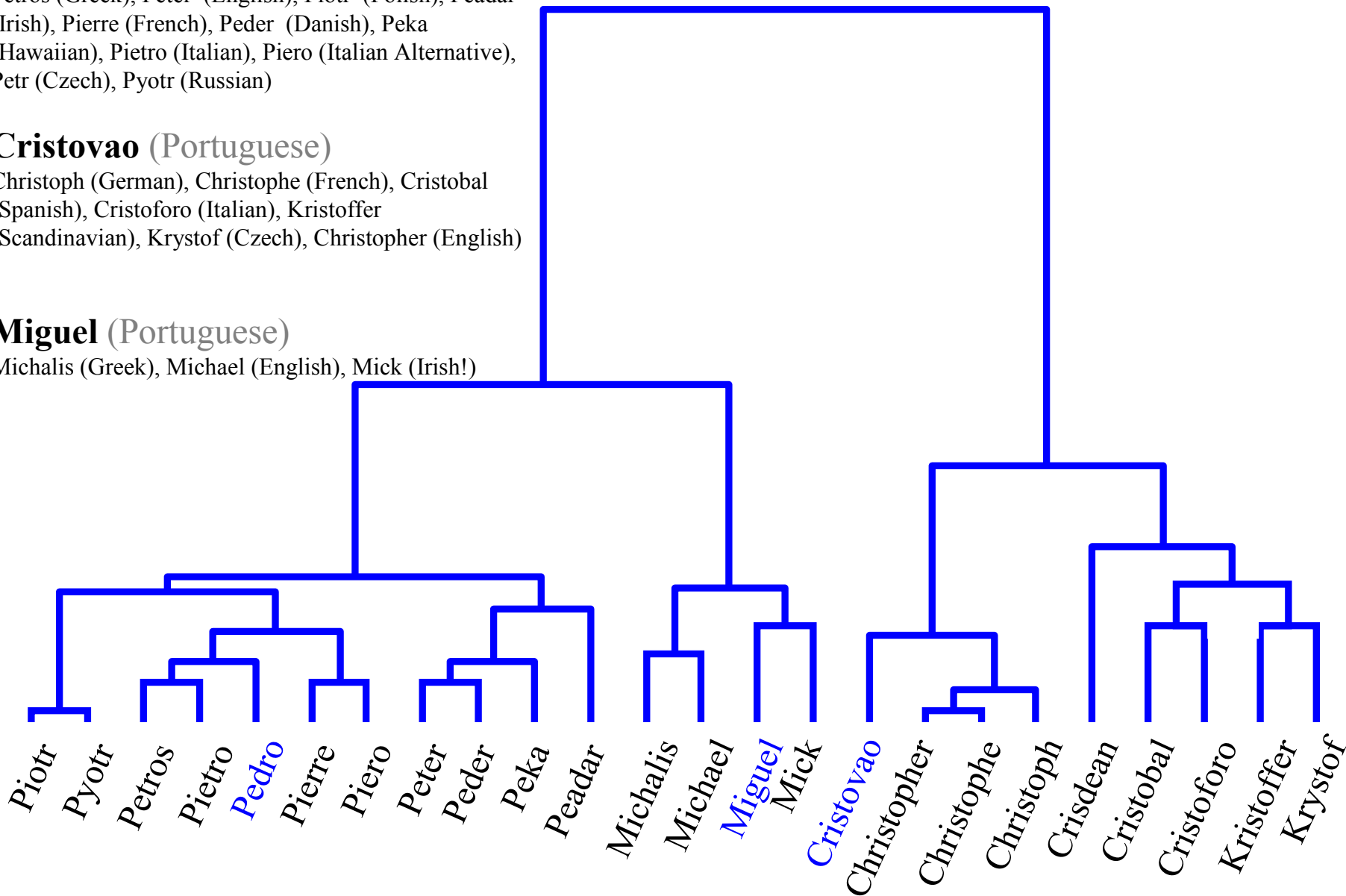
Petros (Greek), Peter (English), Piotr (Polish), Peadar (Irish), Pierre (French), Peder (Danish), Peka (Hawaiian), Pietro (Italian), Piero (Italian Alternative), Petr (Czech), Pyotr (Russian)

Cristovao (Portuguese)

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

Miguel (Portuguese)

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

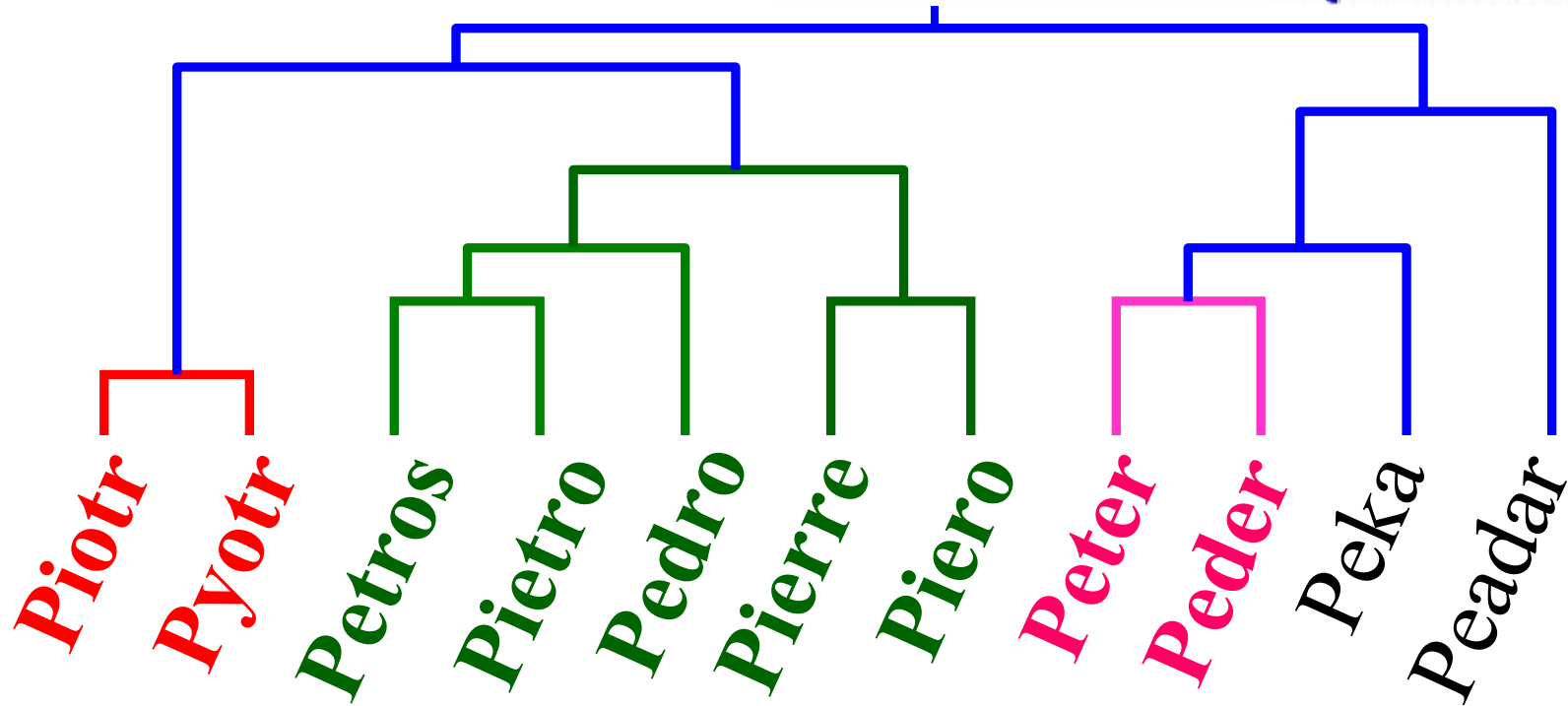
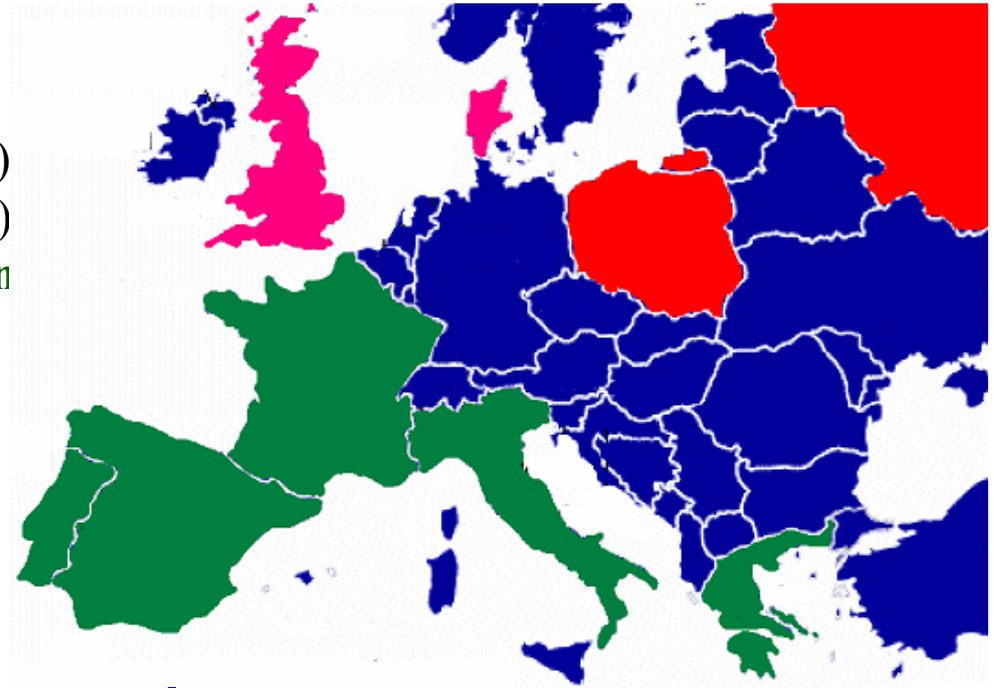


Pedro (Portuguese/Spanish)

Petros (Greek), Peter (English), Piotr (Polish)

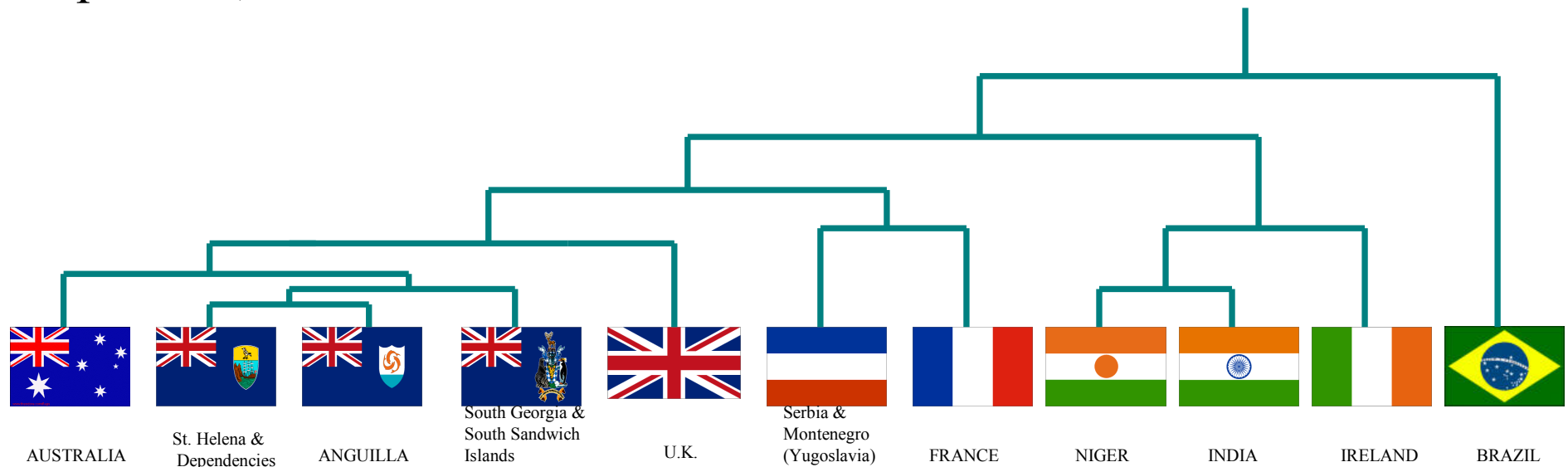
Peadar (Irish), Pierre (French), Peder (Danish)

Peka (Hawaiian), Pietro (Italian), Piero (Italian Alternative), Petr (Czech), Pyotr (Russian)

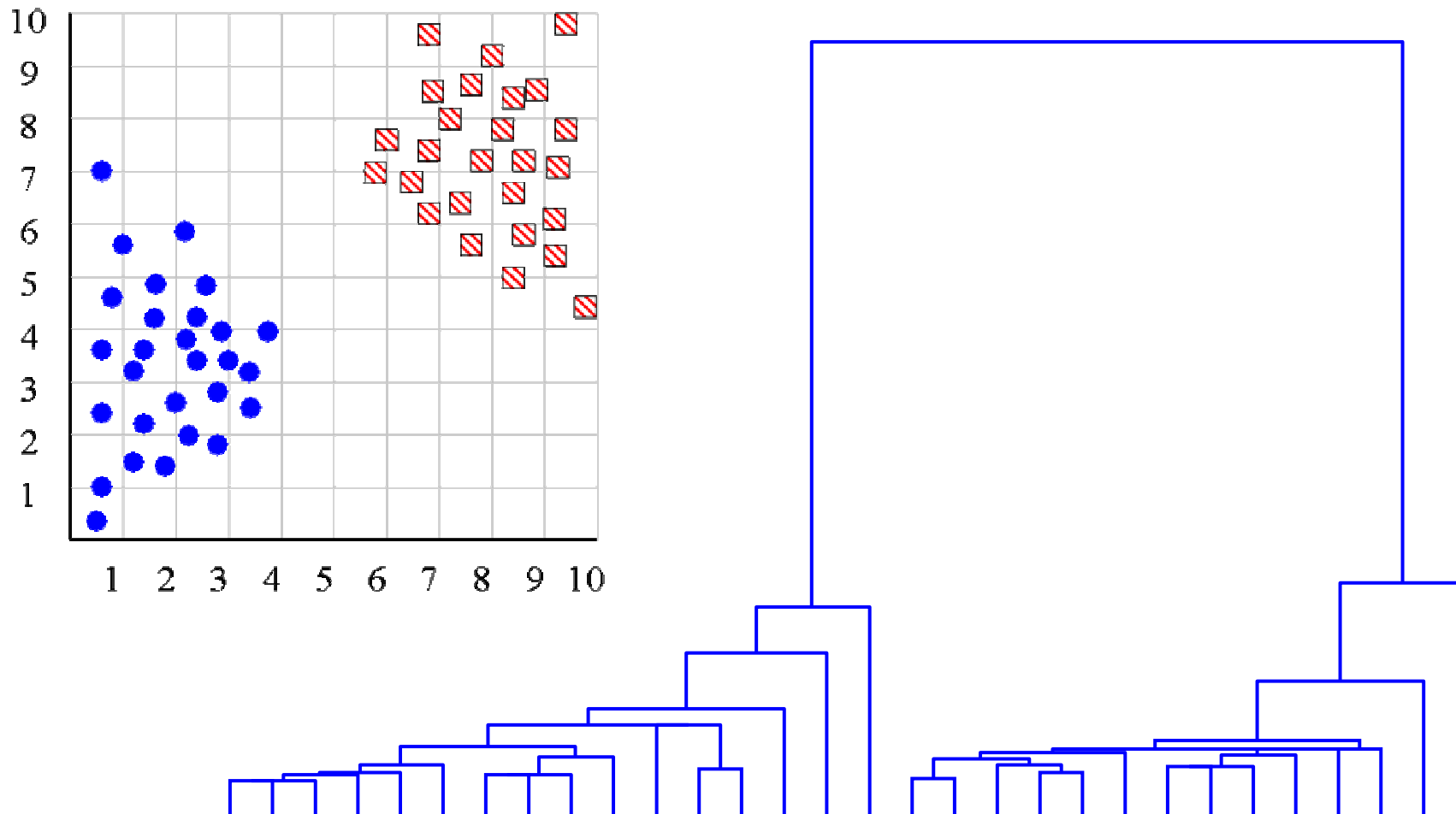


Hierarchical clustering can sometimes show patterns that are meaningless or spurious

- 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.

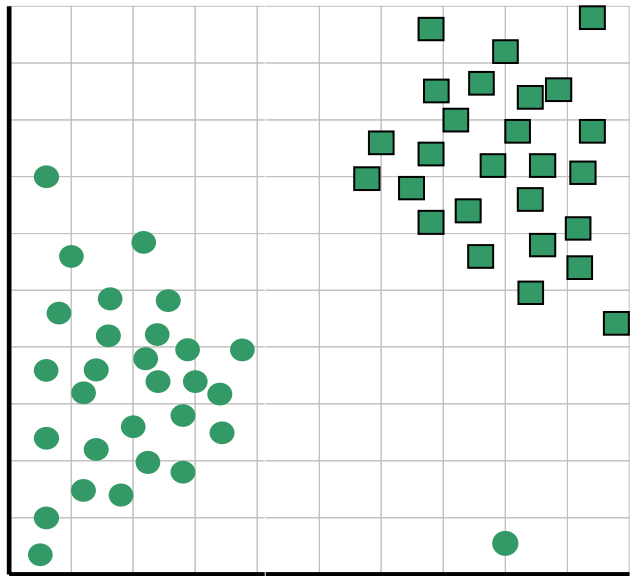


We can look at the dendrogram to determine the “correct” number of clusters. In this case, the two highly separated subtrees are highly suggestive of two clusters. (Things are rarely this clear cut, unfortunately)

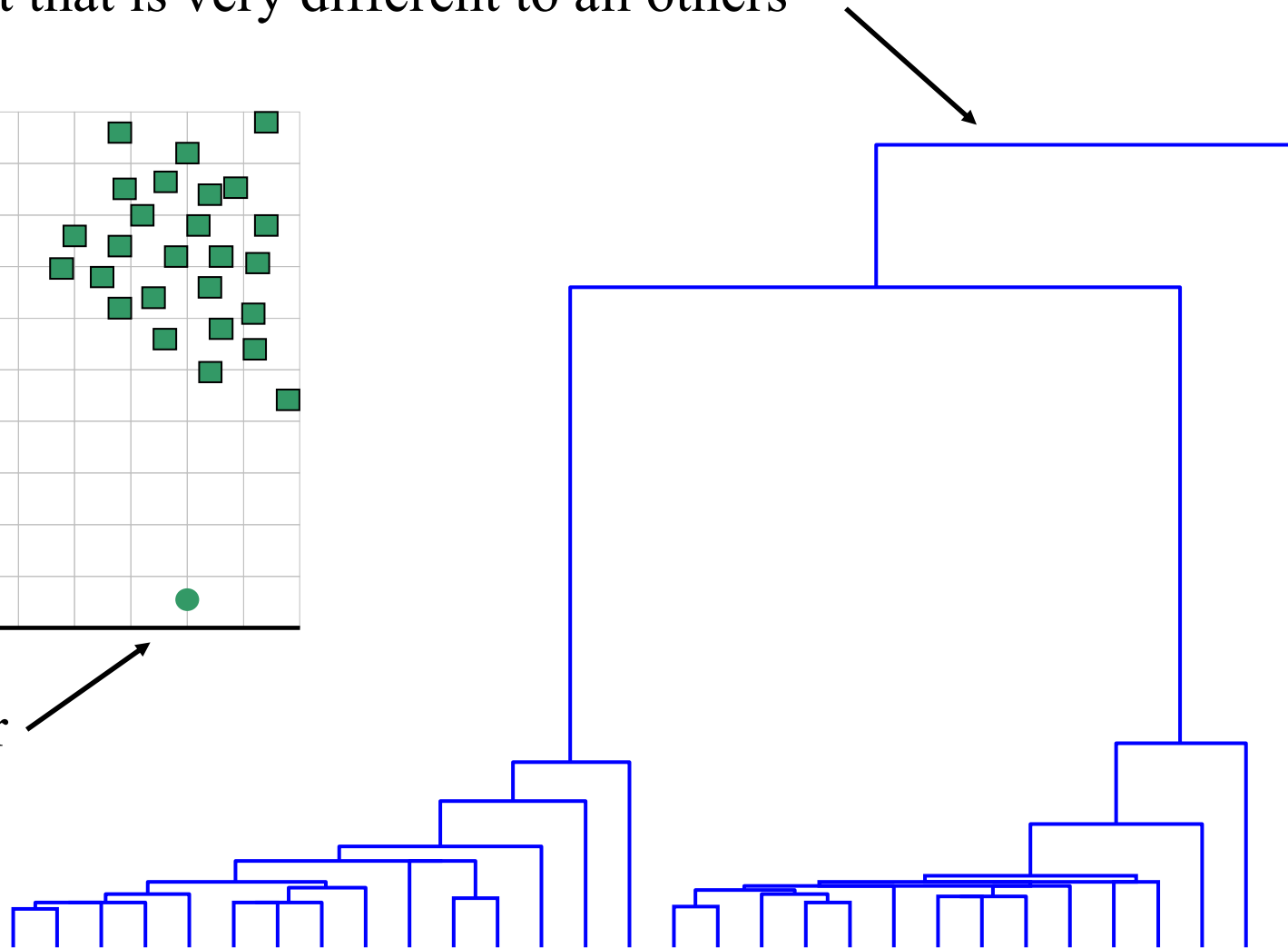


One potential use of a dendrogram is to detect outliers

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



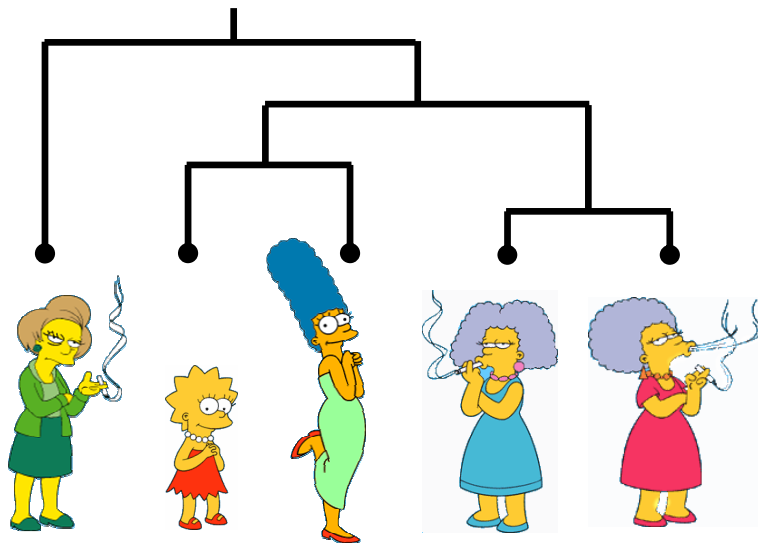
Outlier



(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..












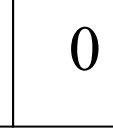
Bottom-Up (agglomerative): Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

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.

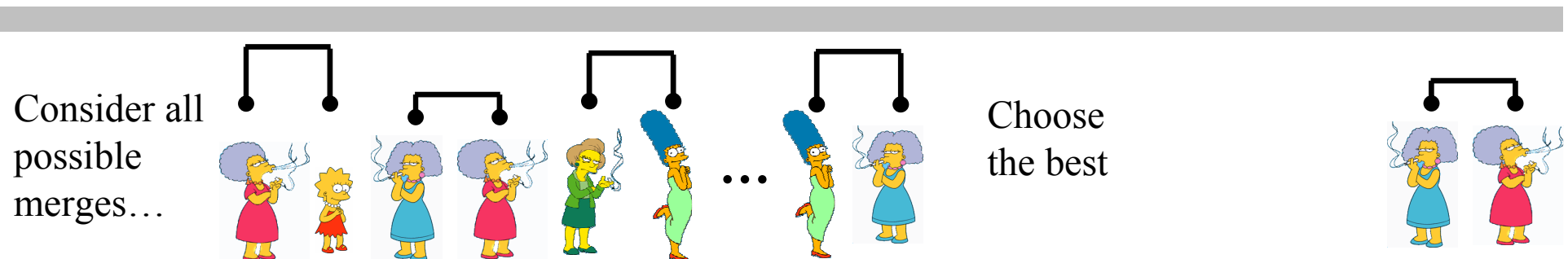
$$D(\text{Mrs. Krabappel, Lisa Simpson}) = 8$$

$$D(\text{Marge Simpson, Edna Krabappel}) = 1$$

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

Bottom-Up (agglomerative):

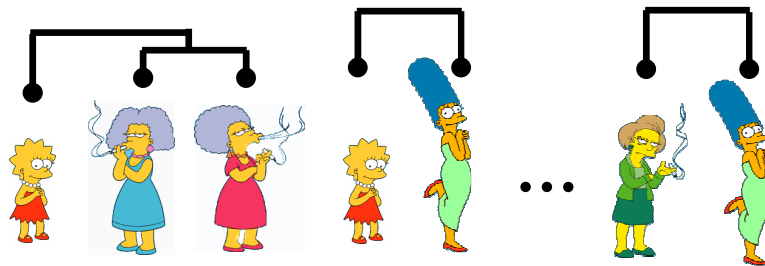
Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



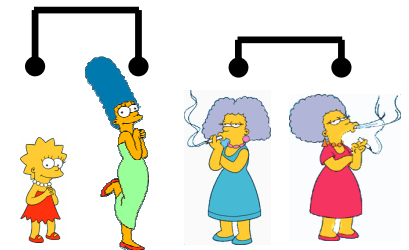
Bottom-Up (agglomerative):

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.

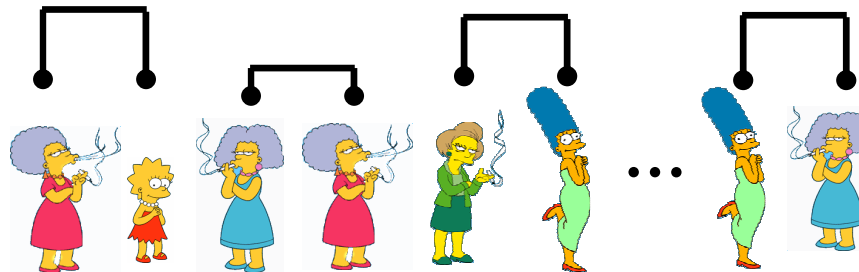
Consider all possible merges...



Choose the best



Consider all possible merges...



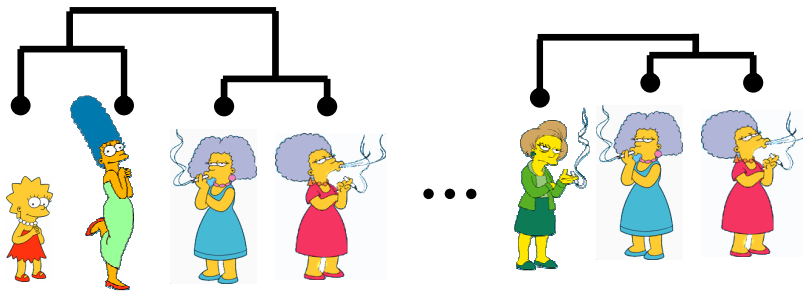
Choose the best



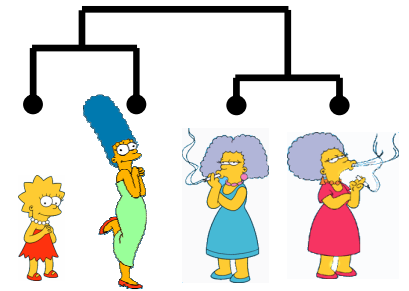
Bottom-Up (agglomerative):

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.

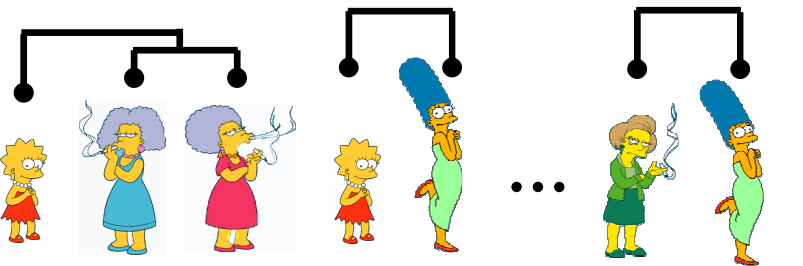
Consider all possible merges...



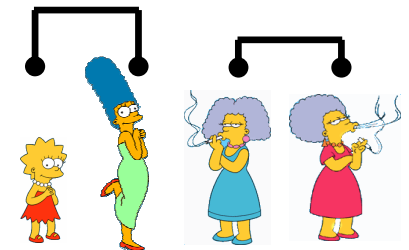
Choose the best



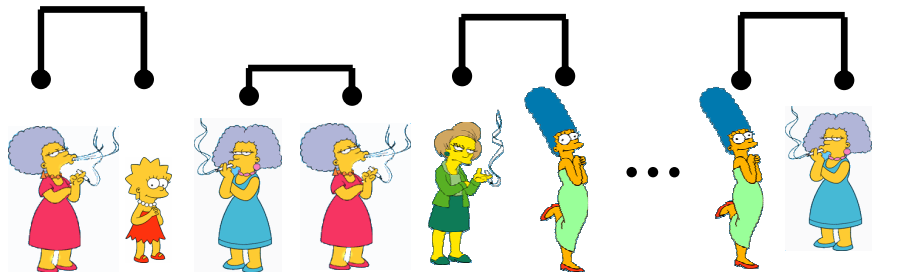
Consider all possible merges...



Choose the best



Consider all possible merges...

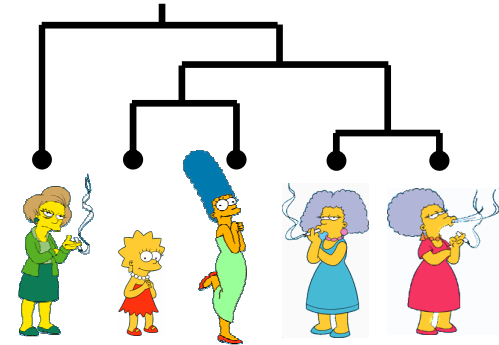


Choose the best

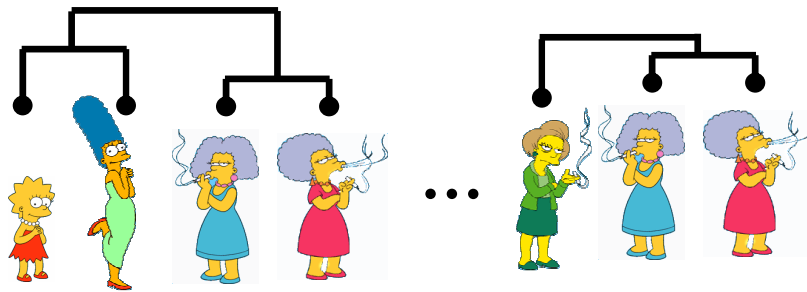


Bottom-Up (agglomerative):

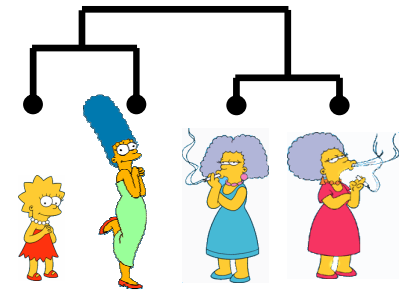
Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



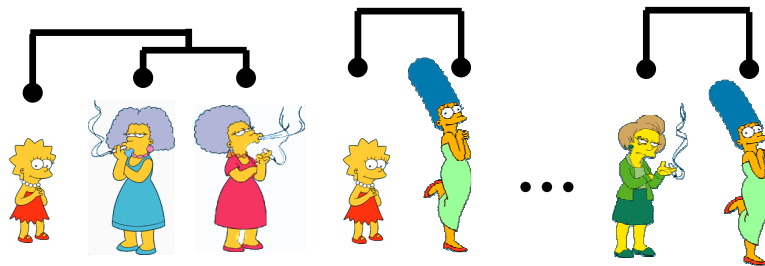
Consider all possible merges...



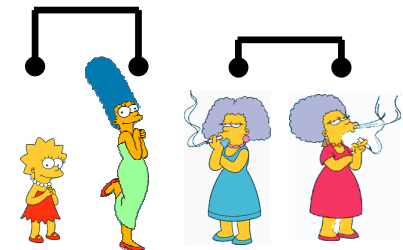
Choose the best



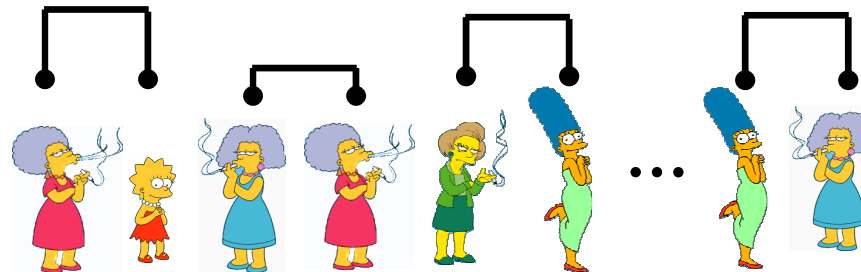
Consider all possible merges...



Choose the best



Consider all possible merges...

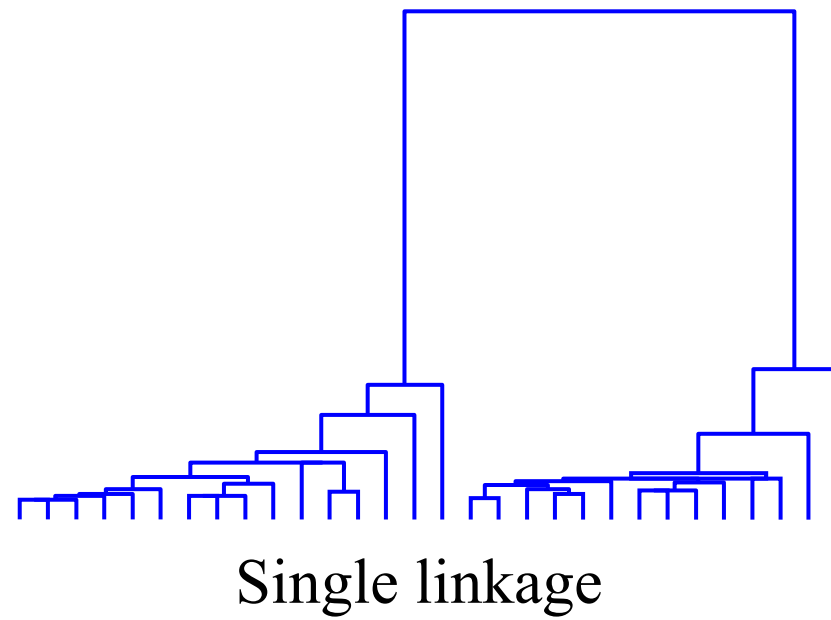
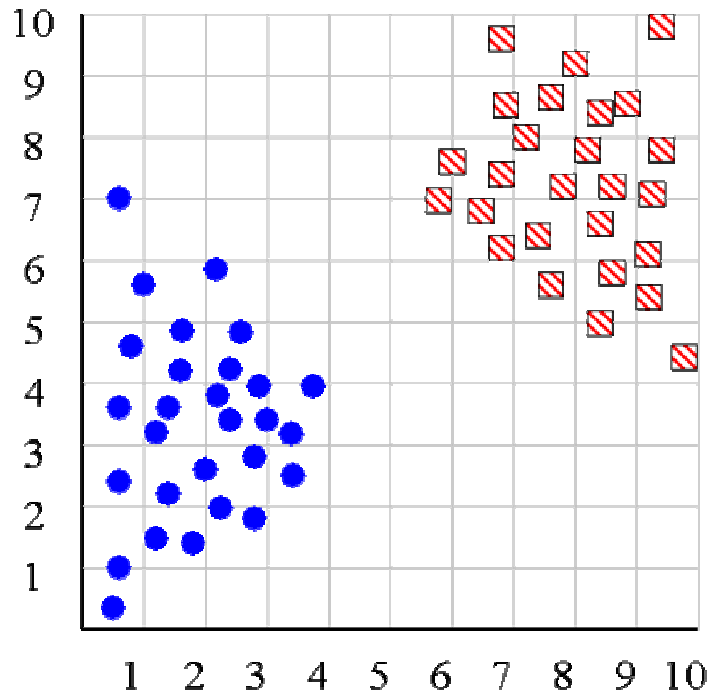


Choose the best

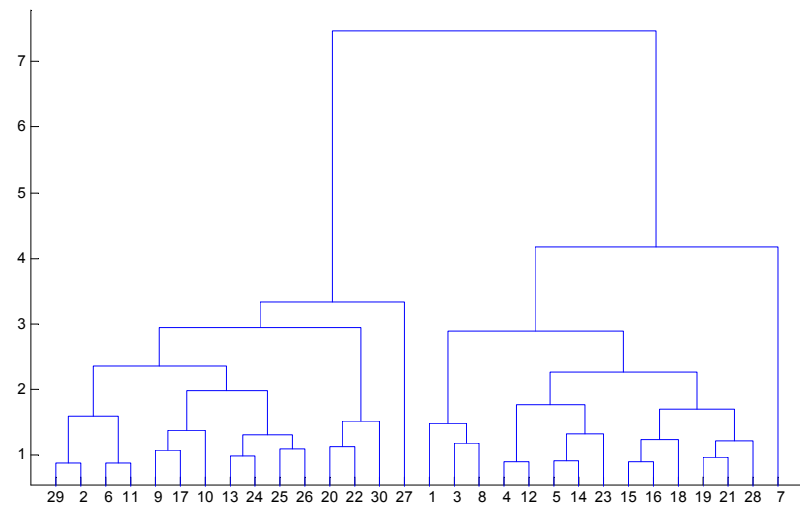


We know how to measure the distance between two objects, but defining the distance between an object and a cluster, or defining the distance between two clusters is non obvious.

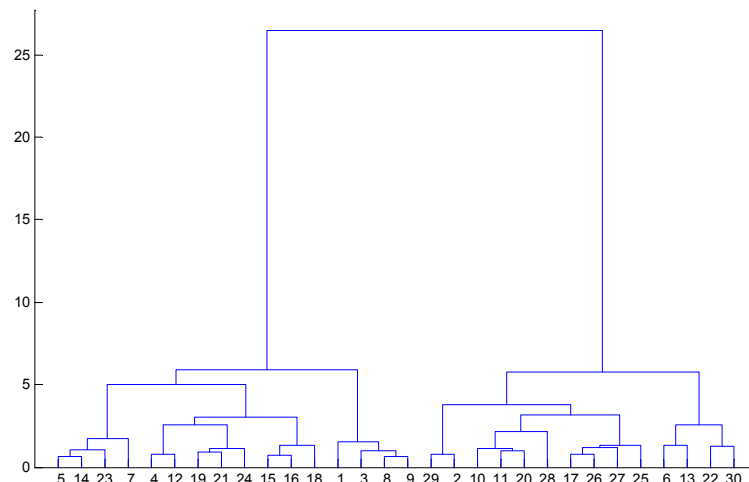
- **Single linkage (nearest neighbor):** In this method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters.
- **Complete linkage (furthest neighbor):** In this method, the distances between clusters are determined by 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



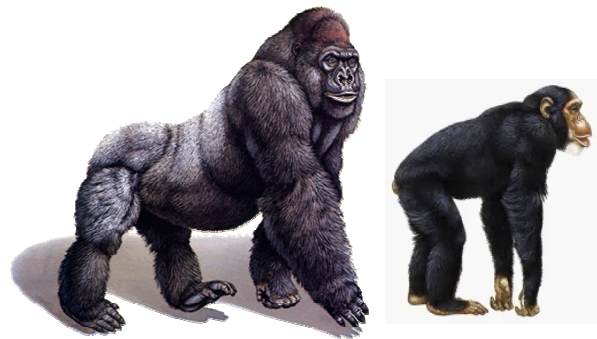
Average linkage



Wards linkage

Up to this point we have simply assumed that we can measure similarity, but

How do we measure similarity?



0.23

Peter Piotr



3



342.7

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

$D(\text{Patty}, \text{Selma}) = 3$

The distance between Marge and Selma.

Change dress color, 1 point

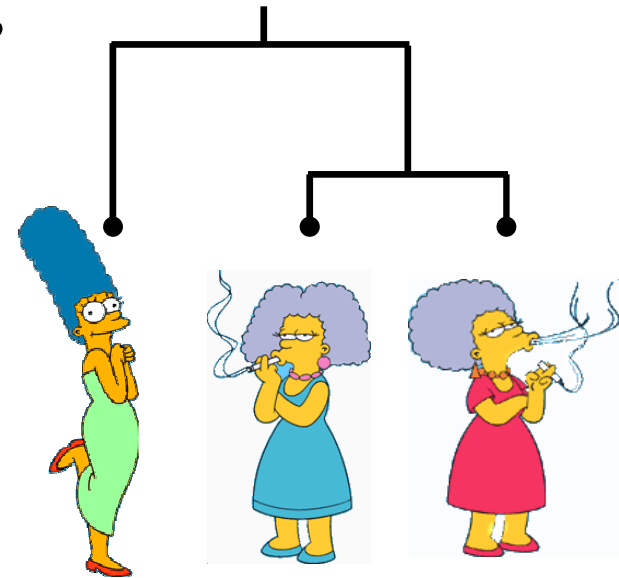
Add earrings, 1 point

Decrease height, 1 point

Take up smoking, 1 point

Lose weight, 1 point

$D(\text{Marge}, \text{Selma}) = 5$



This is called the “edit distance” or the “transformation distance”

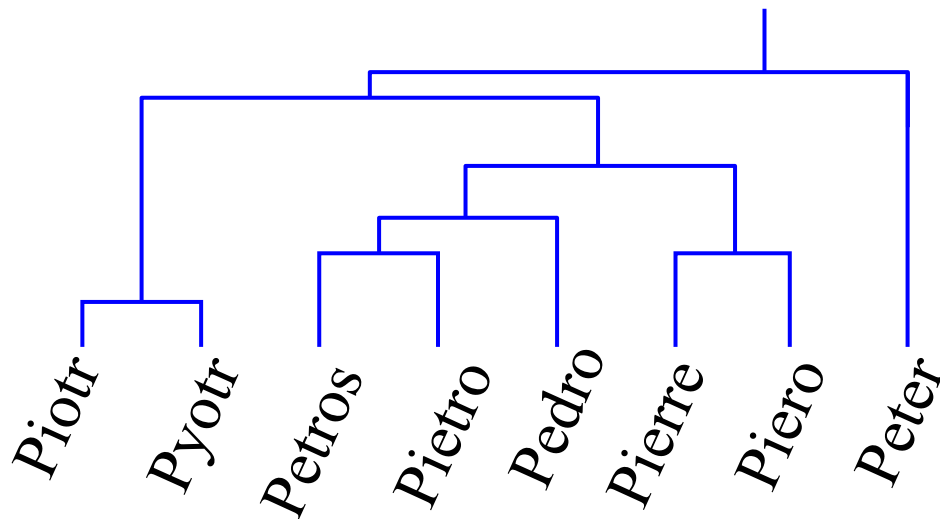
Edit Distance Example

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

<i>Substitution</i>	1 Unit
<i>Insertion</i>	1 Unit
<i>Deletion</i>	1 Unit

$D(\text{Peter}, \text{Piotr})$ is 3

Peter



Substitution (i for e)

Piter



Insertion (o)

Pioter

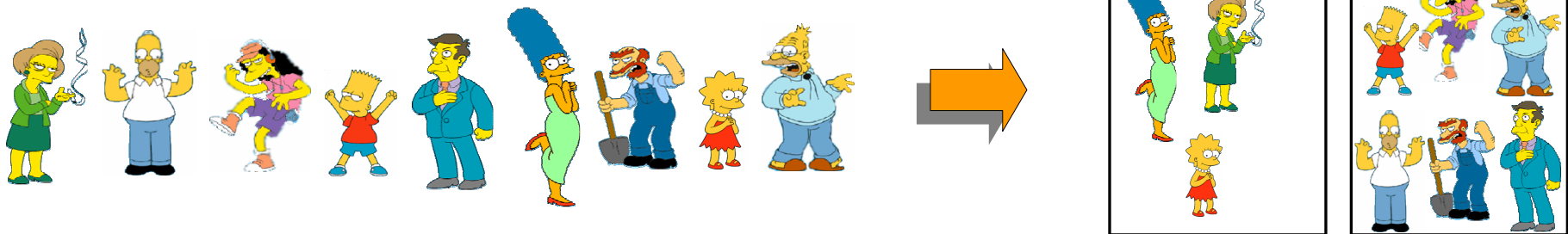


Deletion (e)

Piotr

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 .

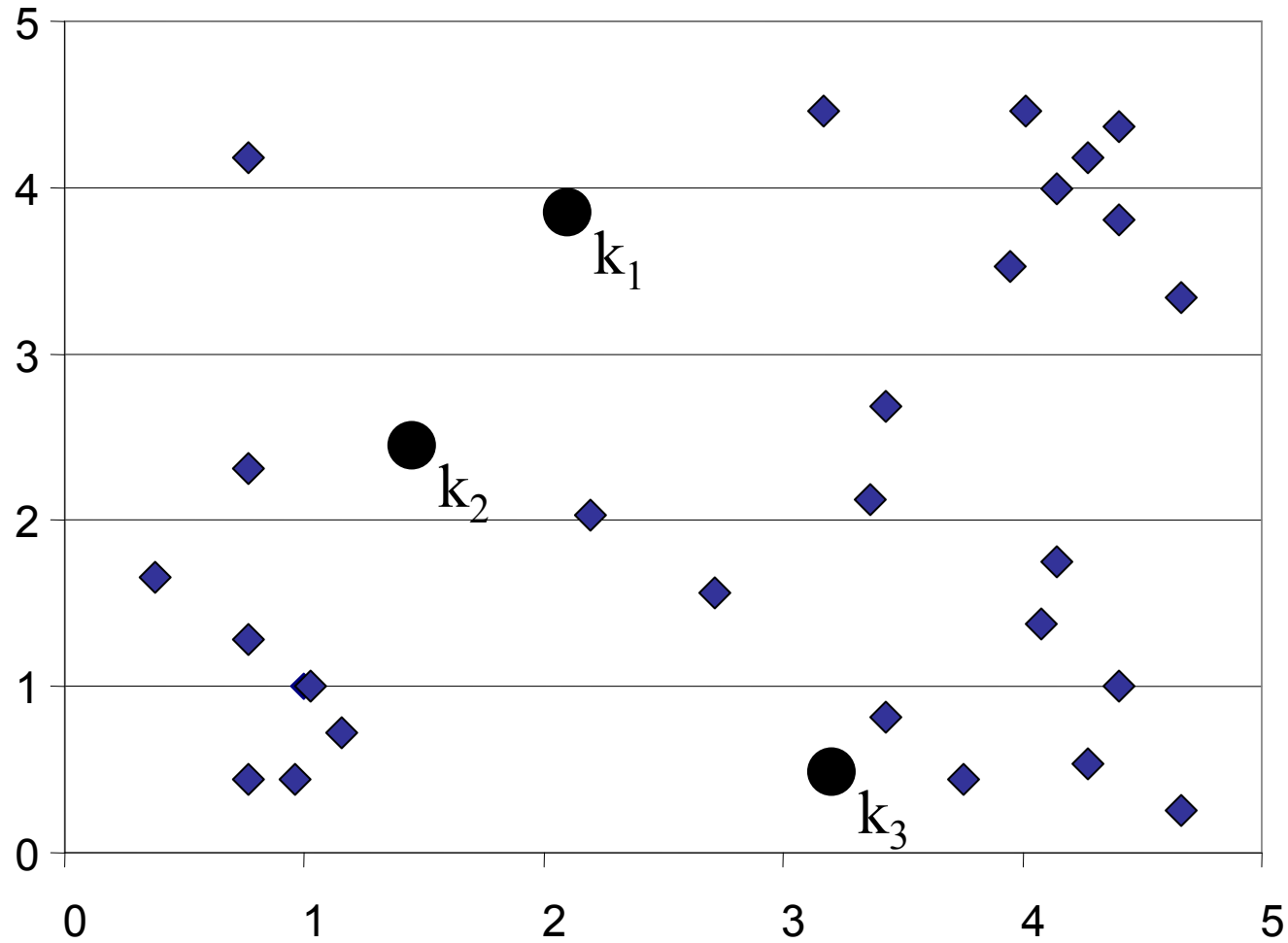


Algorithm *k-means*

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.

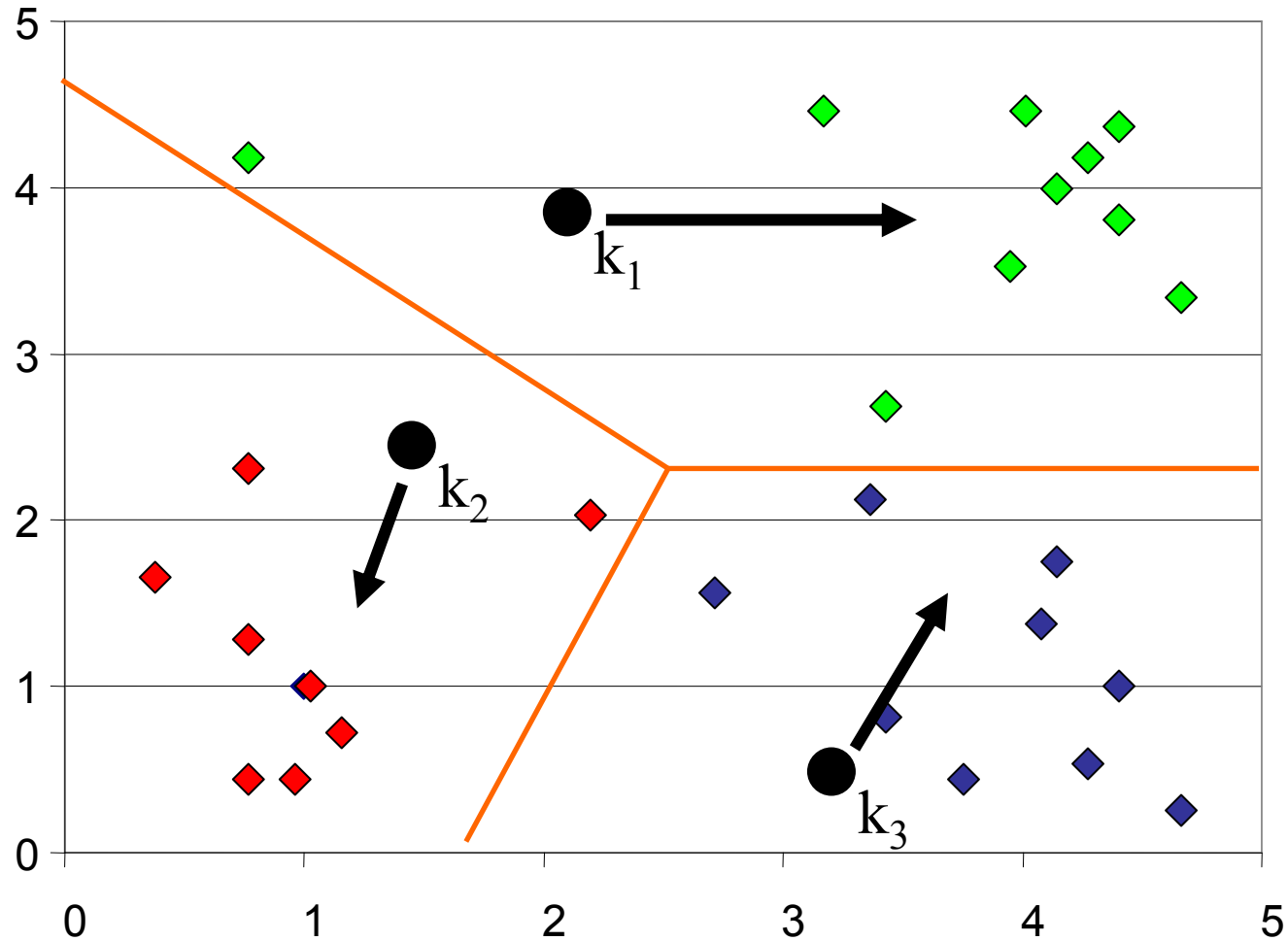
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



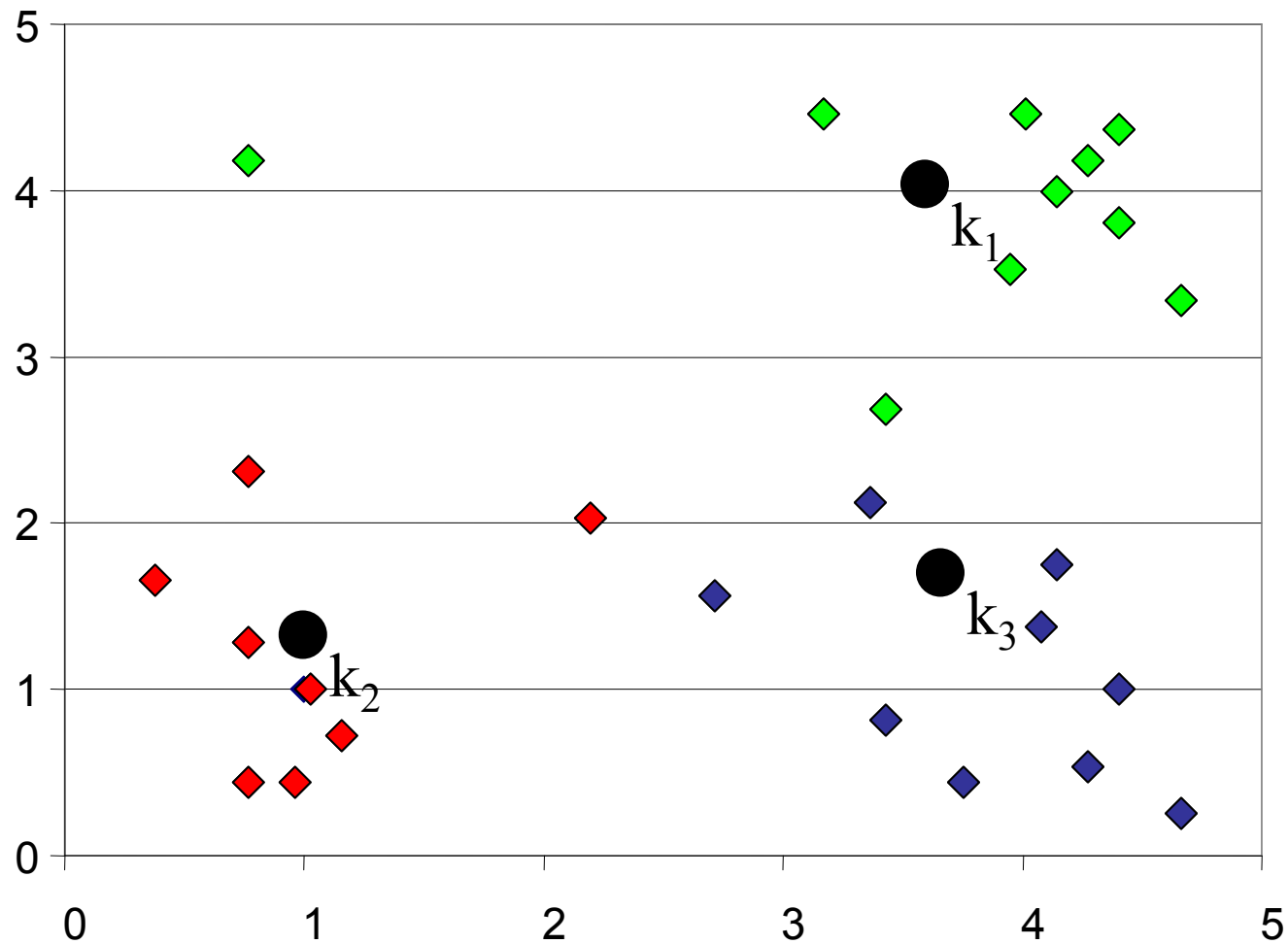
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



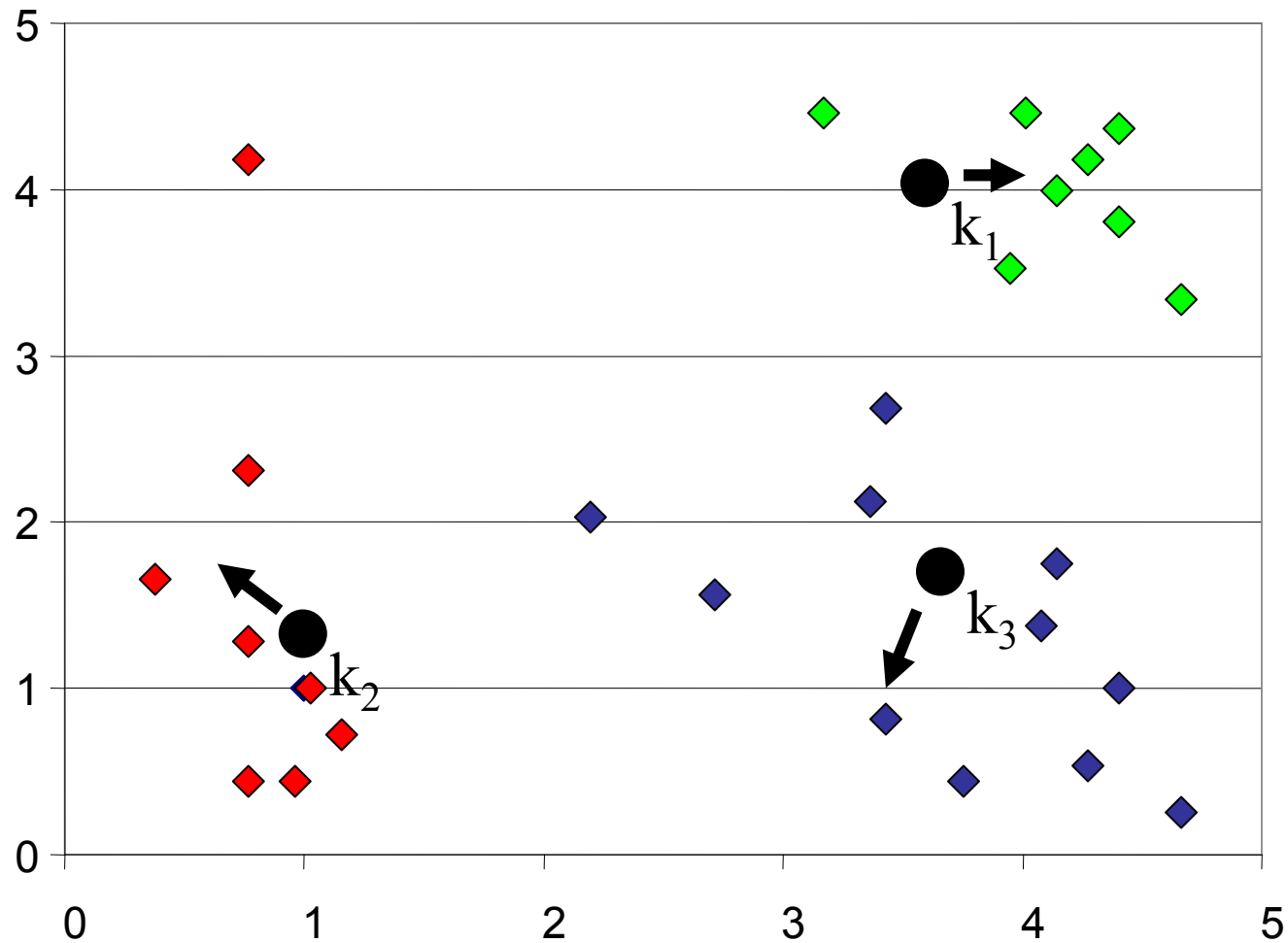
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



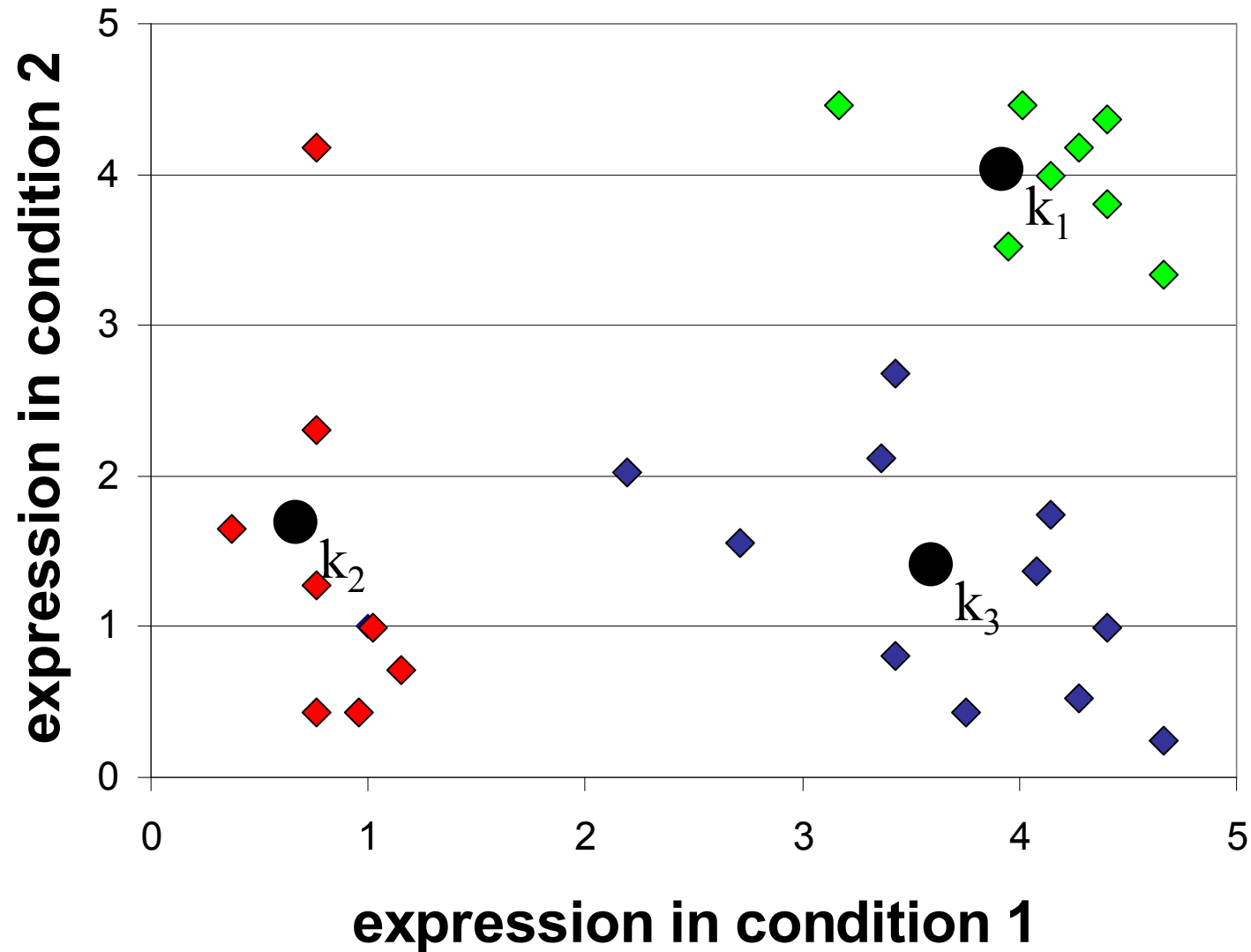
K-means Clustering: Step 4

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

- Strength

- *Relatively efficient: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.*
- *Often terminates at a local optimum. The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms*

- Weakness

- *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*

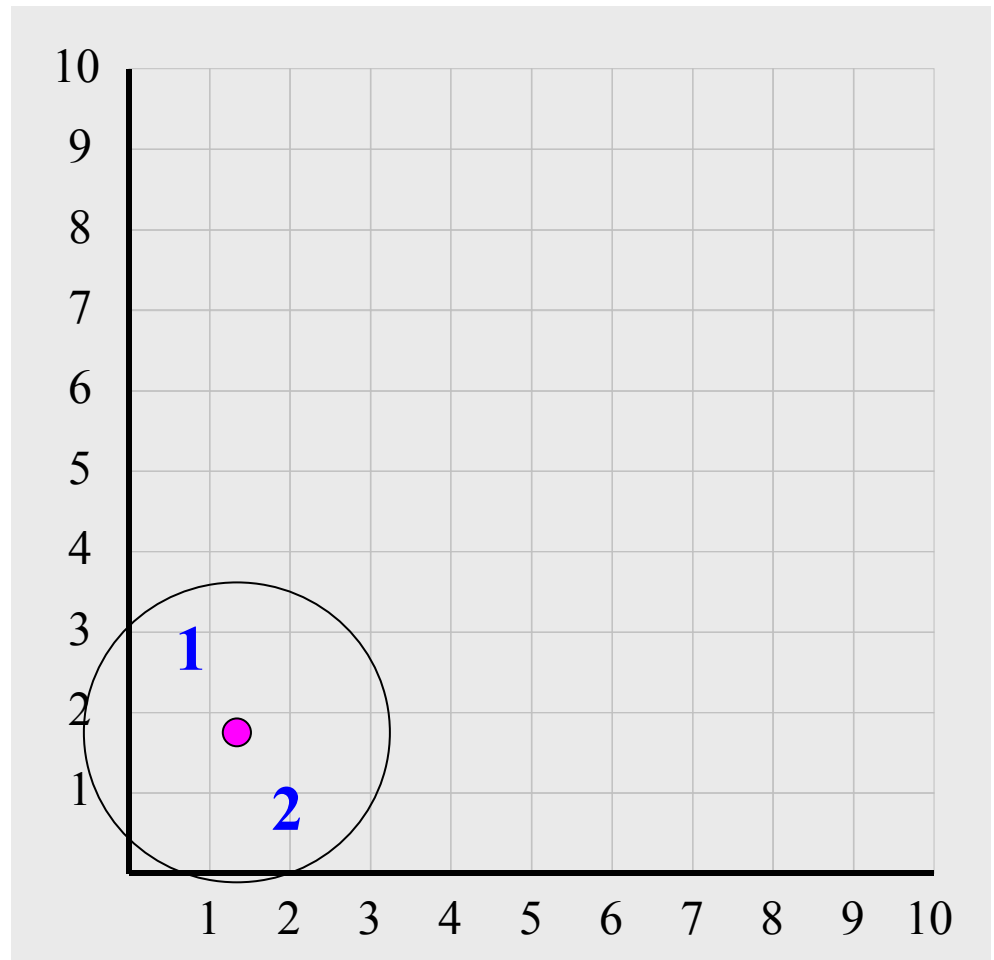
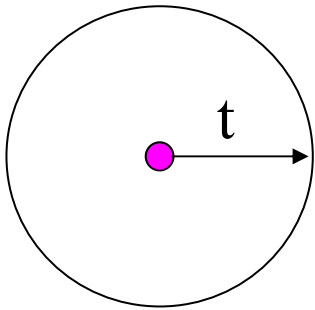
What happens if the data is streaming...

Nearest Neighbor Clustering

Not to be confused with Nearest Neighbor **Classification**

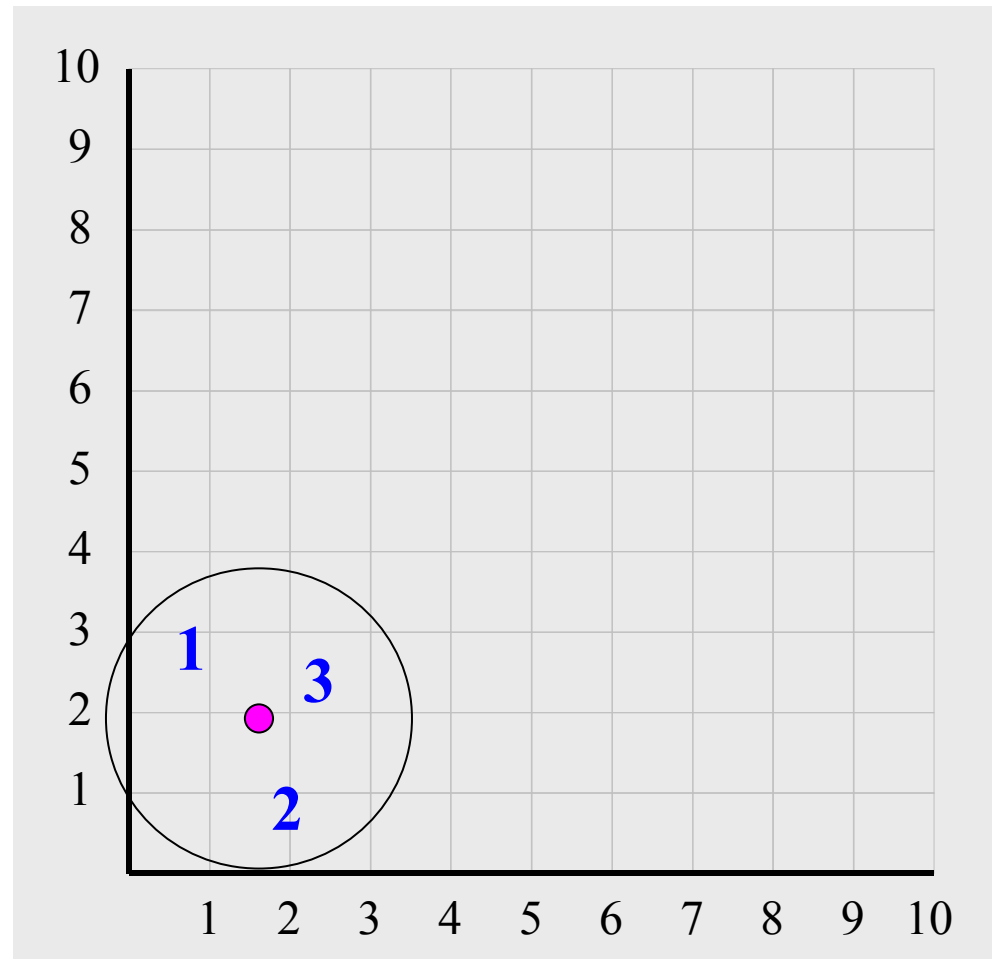
- Items are iteratively merged into the existing clusters that are closest.
- Incremental
- Threshold, t , used to determine if items are added to existing clusters or a new cluster is created.

Threshold t



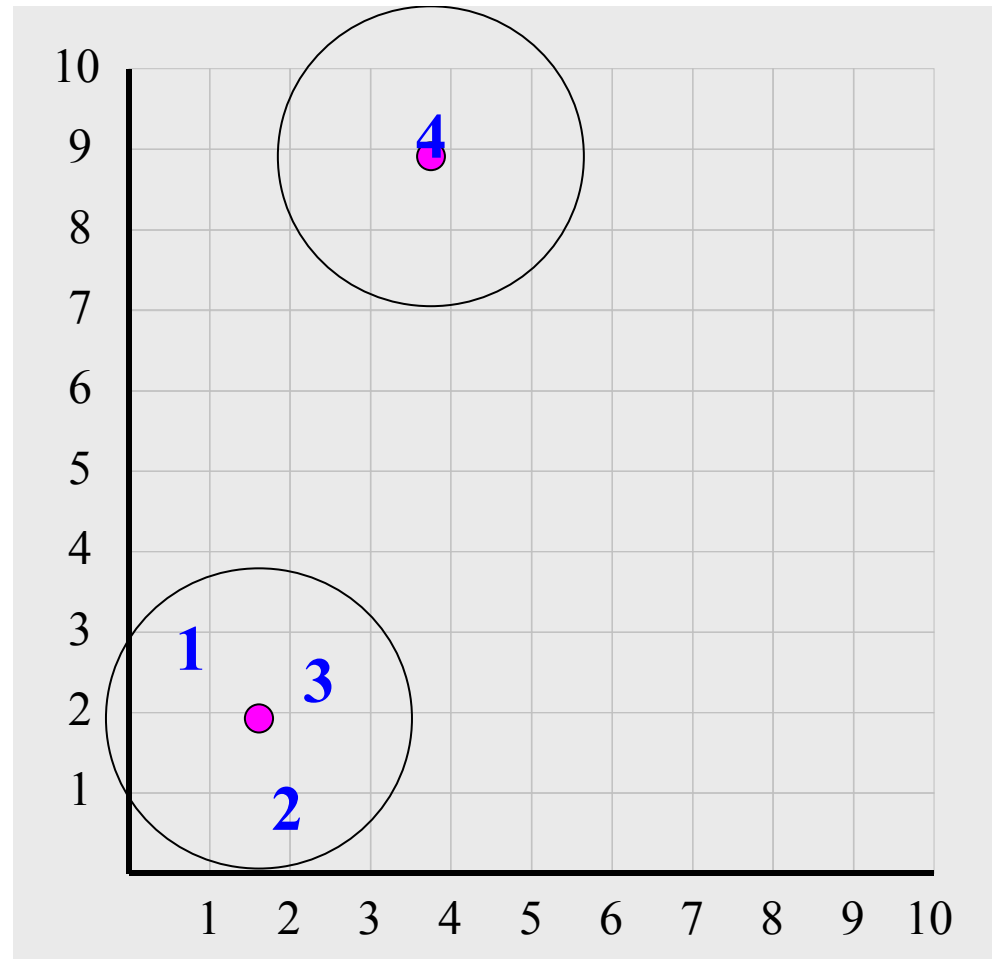
New data point arrives...

It is within the threshold for cluster 1, so add it to the cluster, and update cluster center.



New data point arrives...

It is **not** within the threshold for cluster 1, so create a new cluster, and so on..

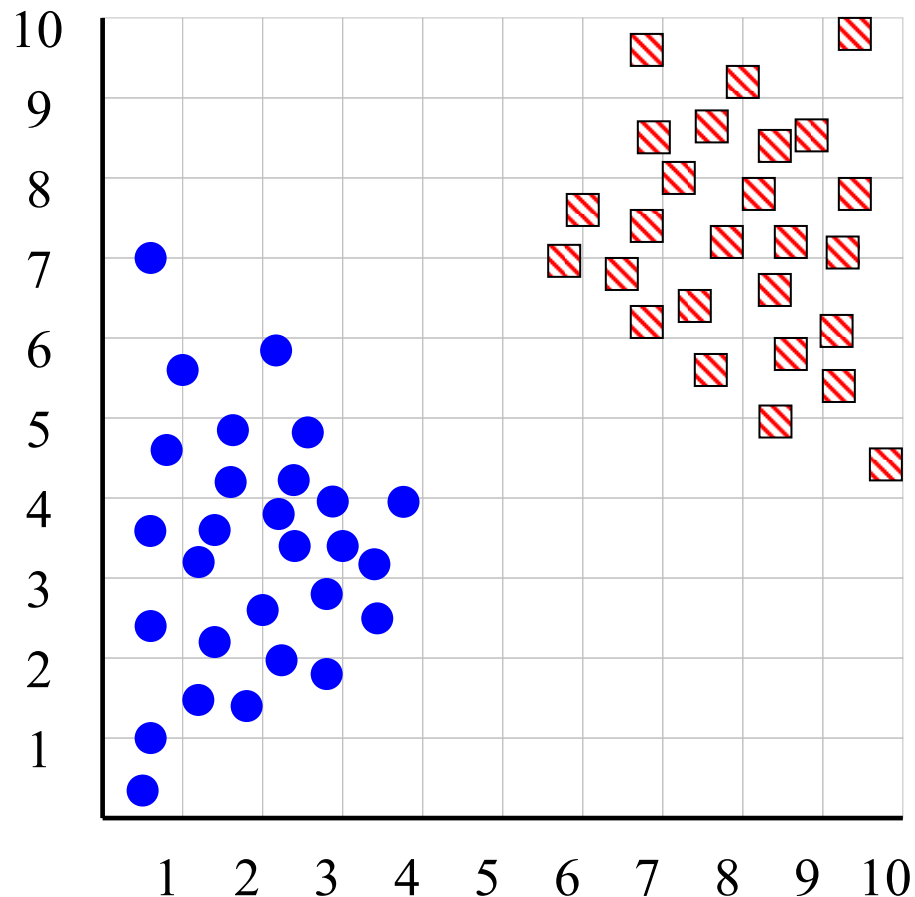


Algorithm is highly order dependent...

It is difficult to determine t in advance...

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

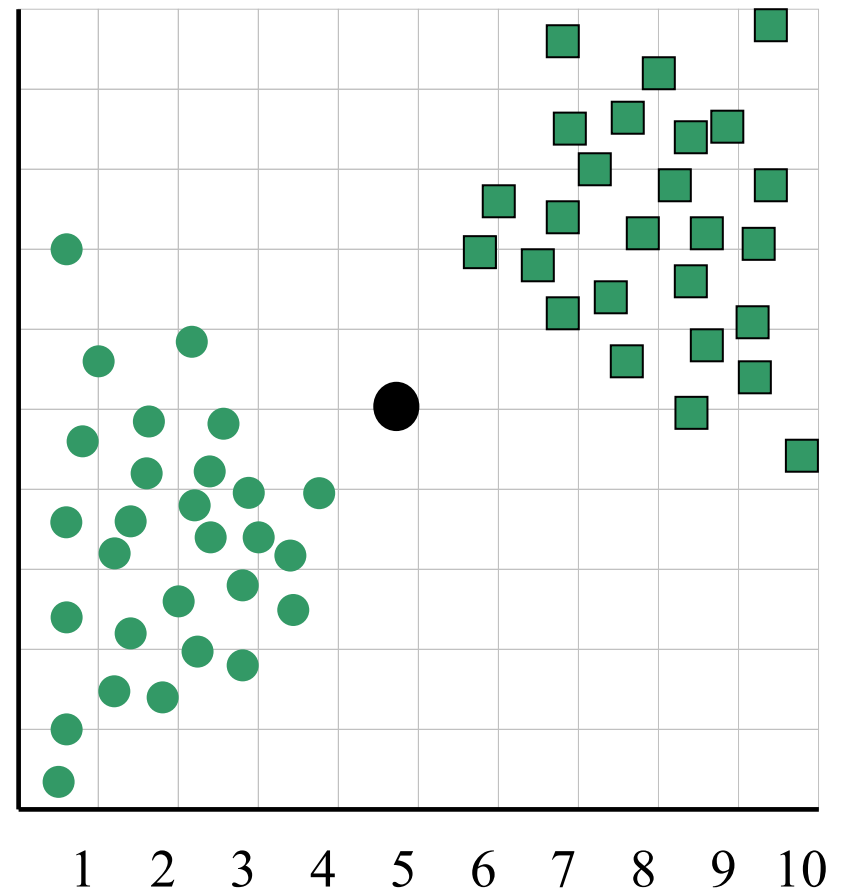
In general, this is a unsolved problem. However there are many approximate methods. In the next few slides we will see an example.



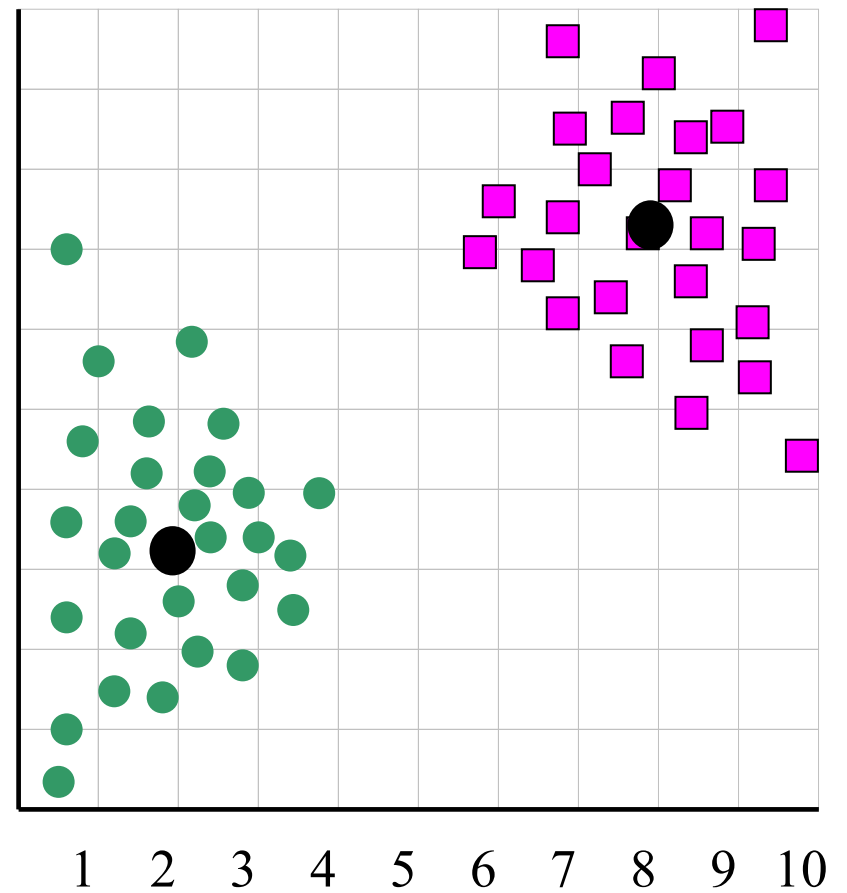
For our example, we will use the familiar **katydid**/**grasshopper** dataset.

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.

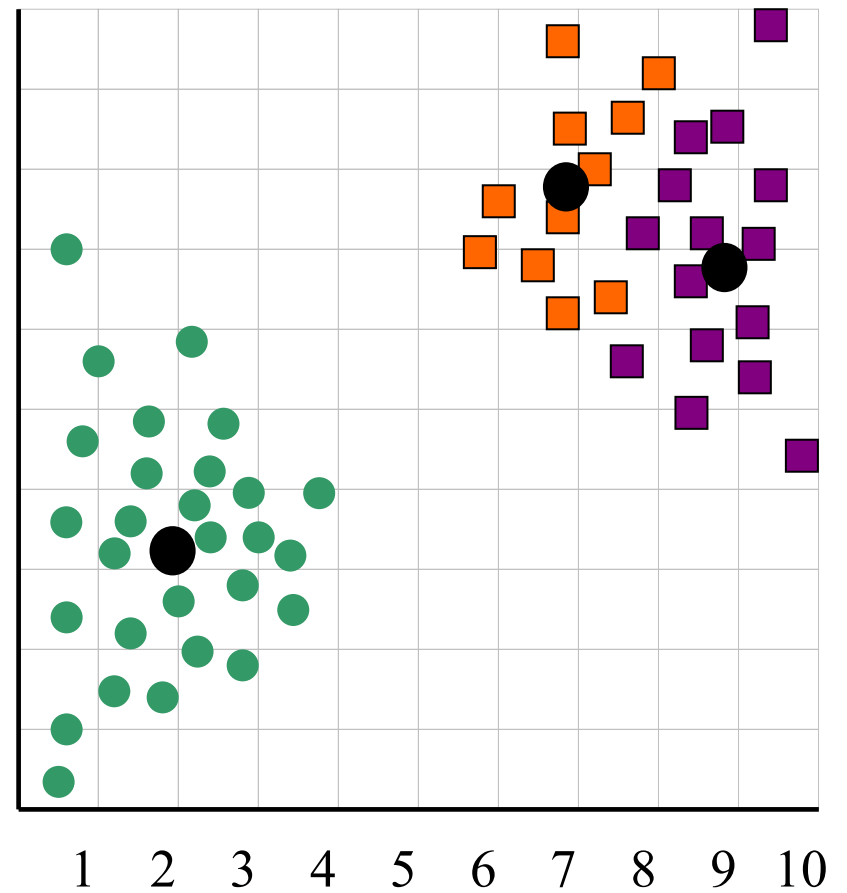
When $k = 1$, the objective function is 873.0



When $k = 2$, the objective function is 173.1

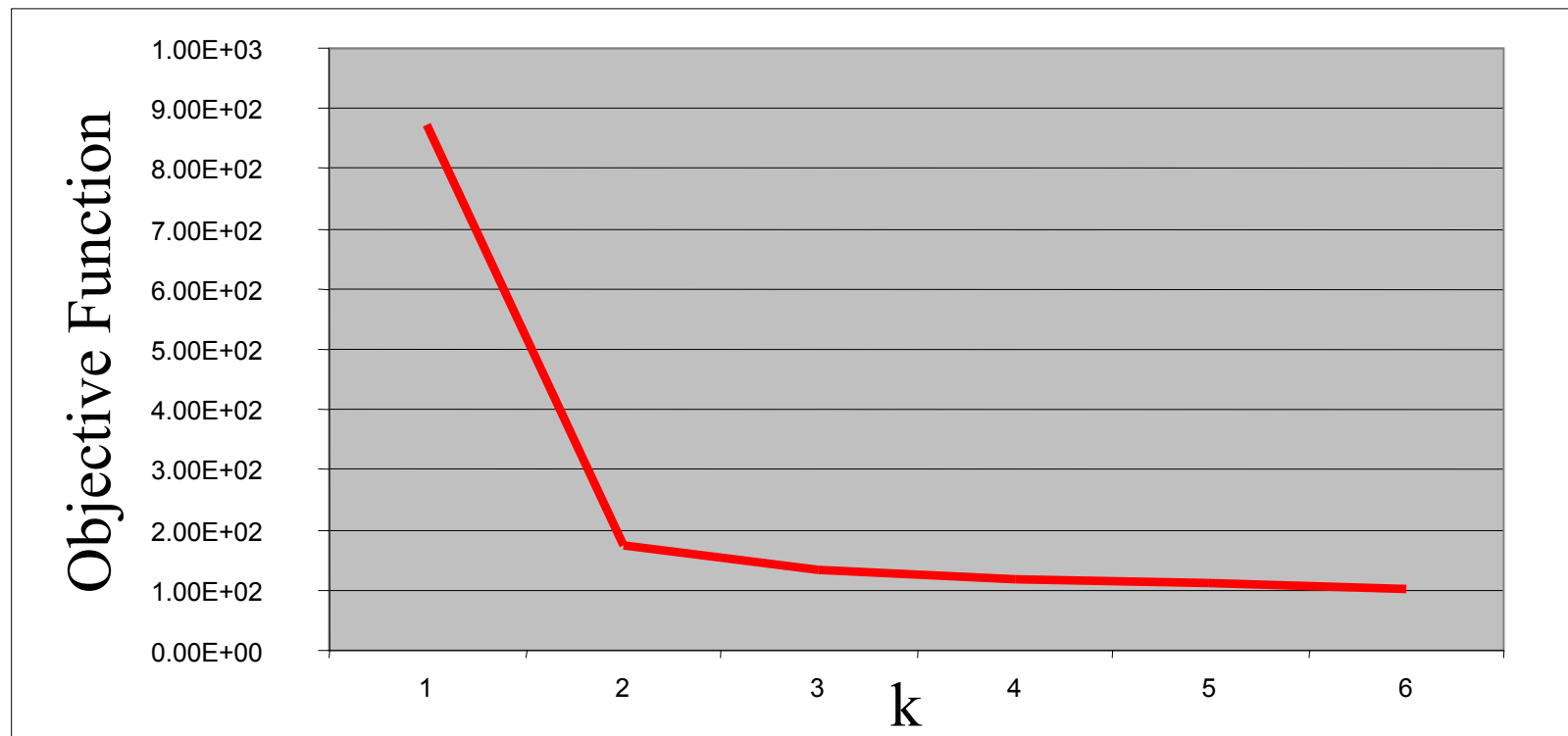


When $k = 3$, the objective function is 133.6



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