

Clustering

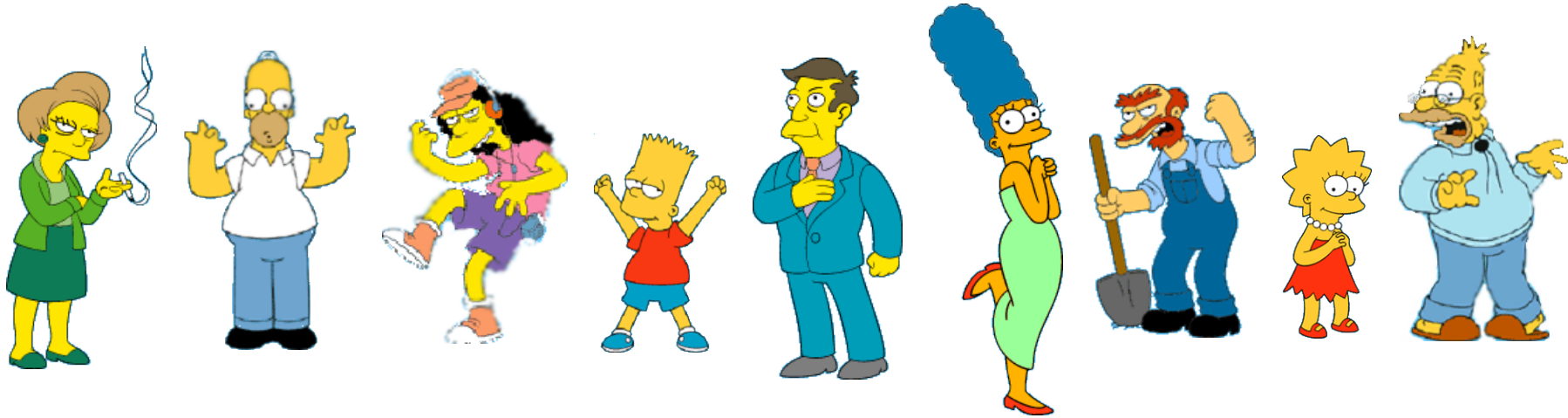
Slides by Eamonn Keogh

What is Clustering?

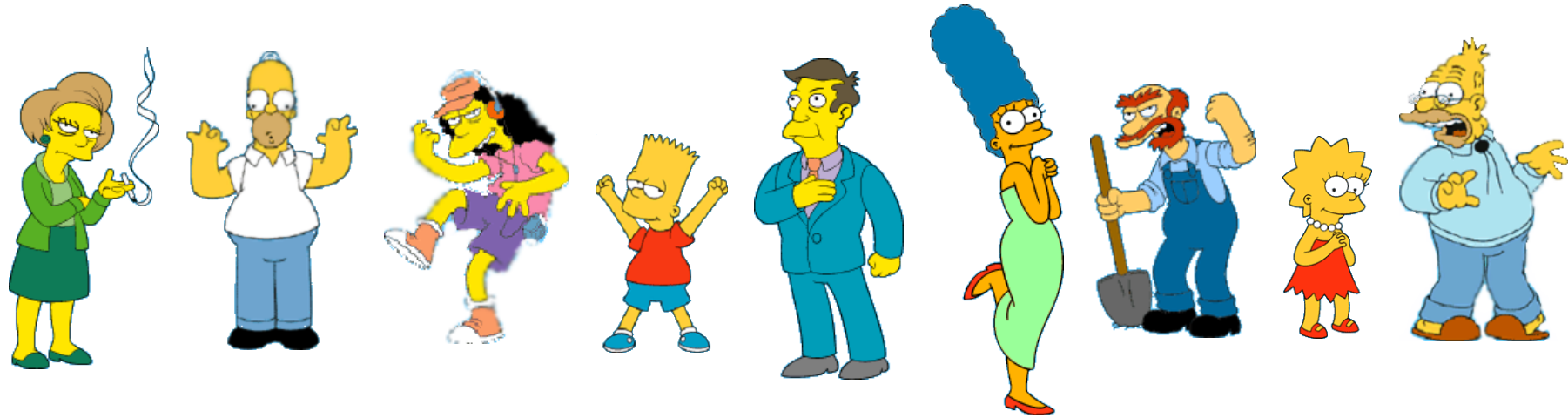
Also called *unsupervised learning*, sometimes called *classification* by statisticians and *sorting* by psychologists and *segmentation* by people in marketing

- 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.

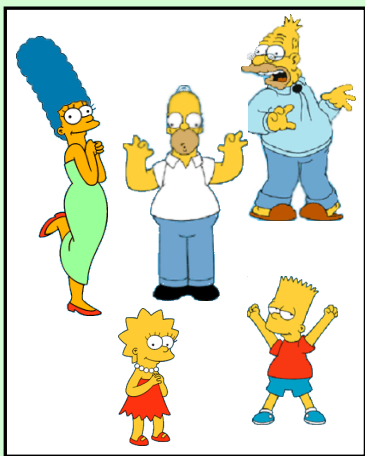
What is a natural grouping among these objects?



What is a natural grouping among these objects?



Clustering is subjective



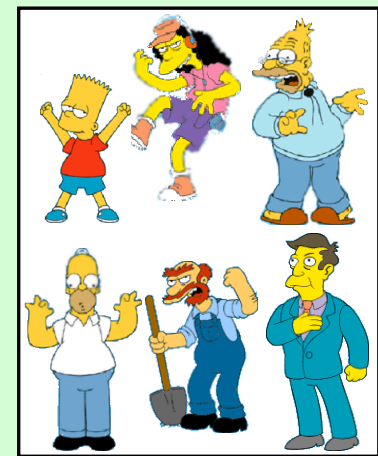
Simpson's Family



School Employees



Females



Males

What is Similarity?

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

Webster's Dictionary

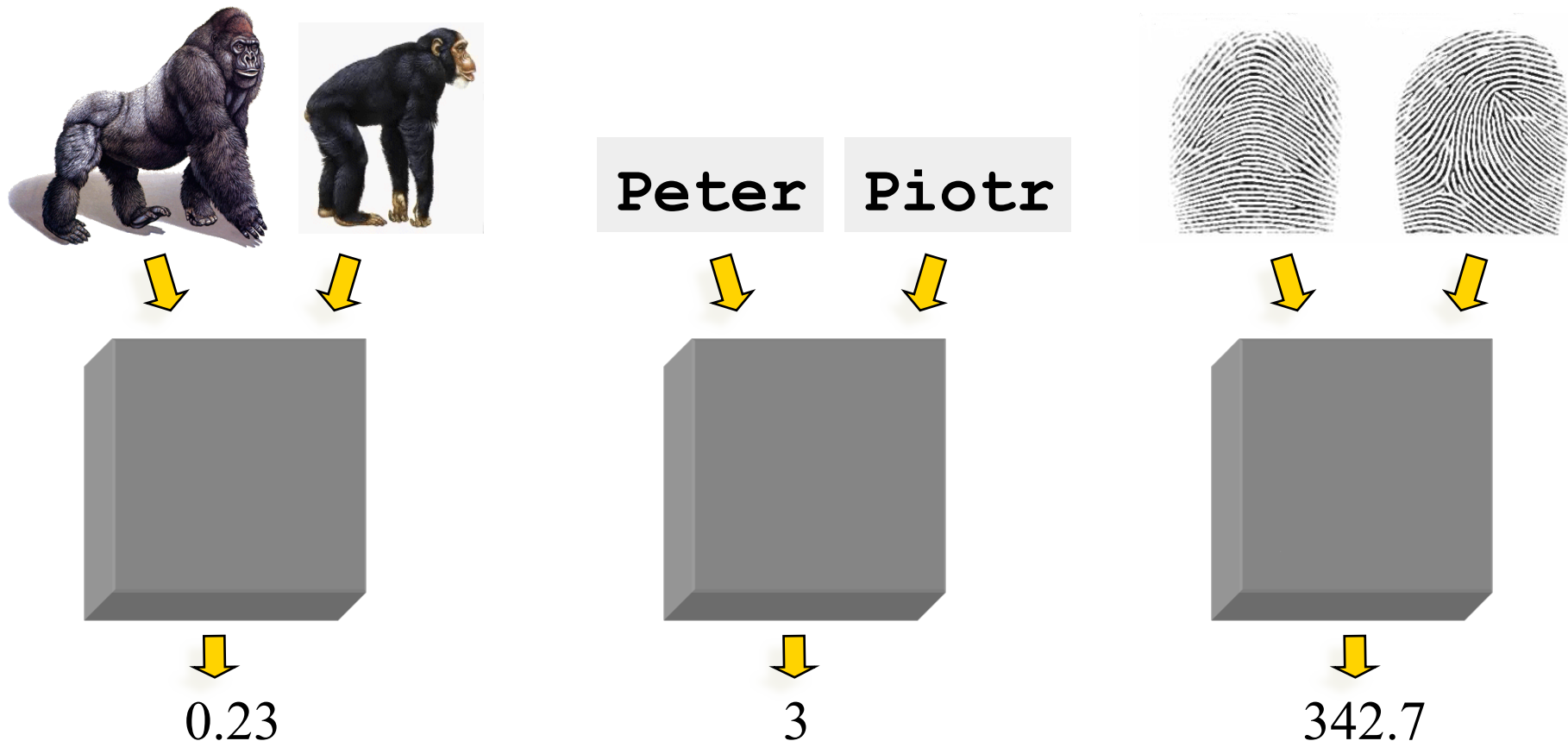


Similarity is hard to define, but...
"We know it when we see it"

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

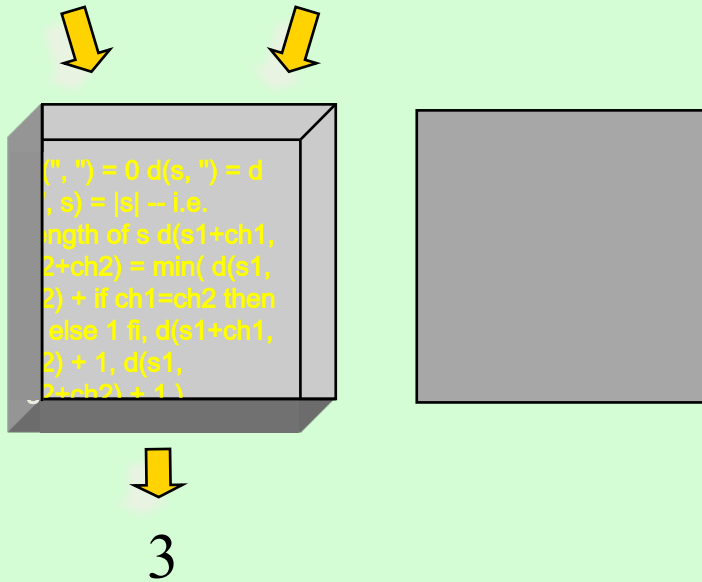
Defining Distance Measures

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 $D(O_1, O_2)$



Peter

Piotr



When we peek inside one of these black boxes, we see some function on two variables. These functions might be 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?

- $D(A,B) = D(B,A)$
- $D(A,A) = 0$
- $D(A,B) = 0$ If $A = B$
- $D(A,B) \leq D(A,C) + D(B,C)$

Symmetry

Constancy of Self-Similarity

Positivity (Separation)

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.”

$$D(A,B) = 0 \text{ Iif } A=B$$

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 Carl, and Bob is very like Carl, but Alex is very unlike Bob.”

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

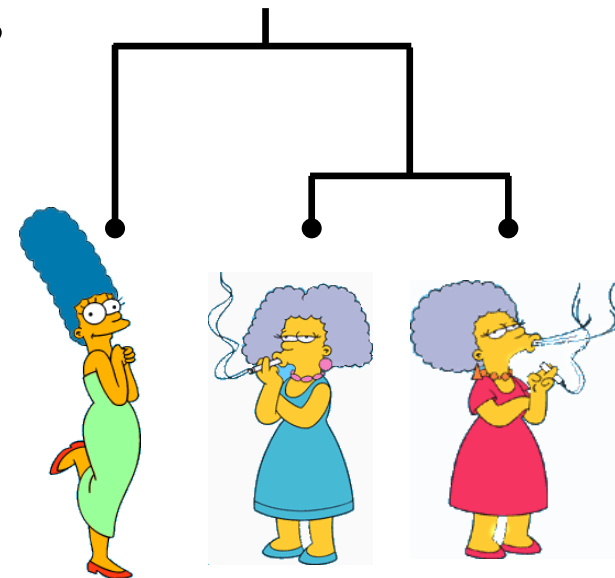
Add earrings, 1 point

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”

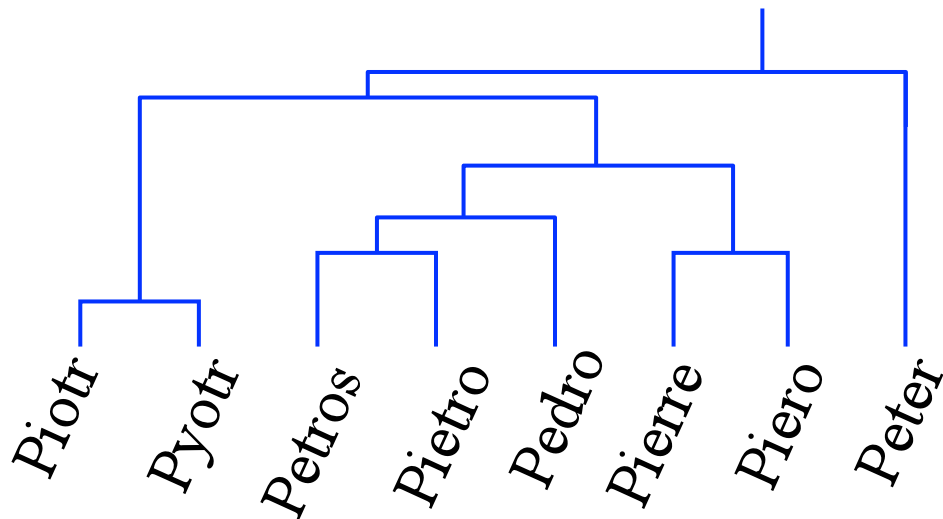
Edit Distance Example

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



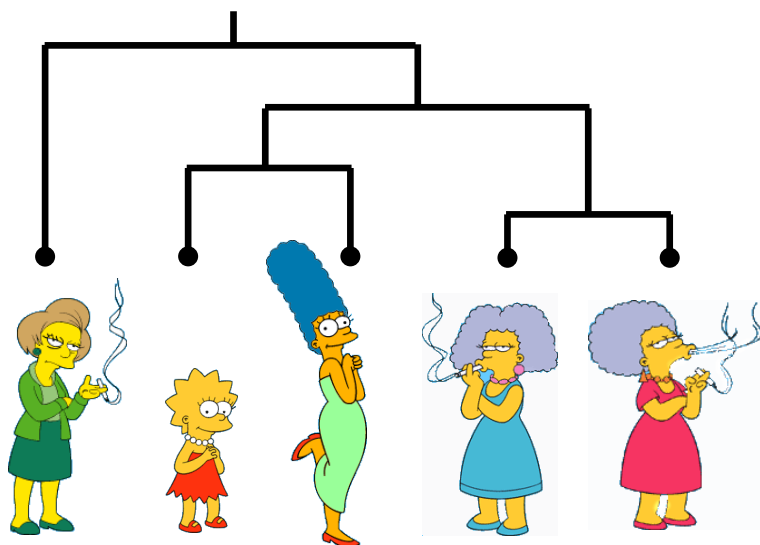
Deletion (e)

Piotr

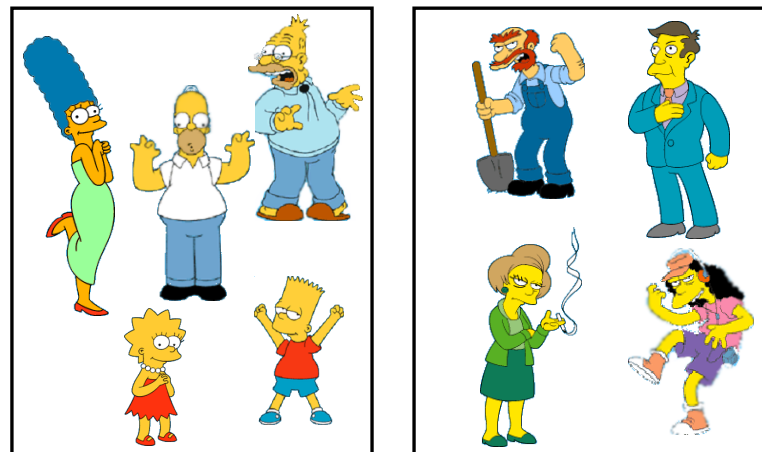
Two Types of Clustering

- **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

Hierarchical



Partitional

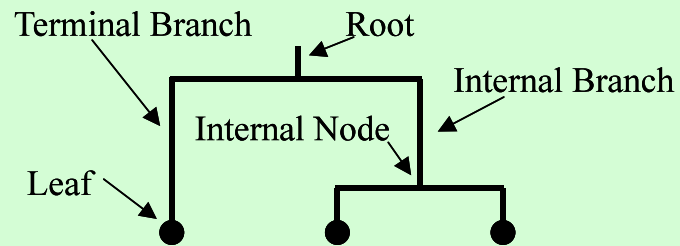


Desirable Properties of a Clustering Algorithm

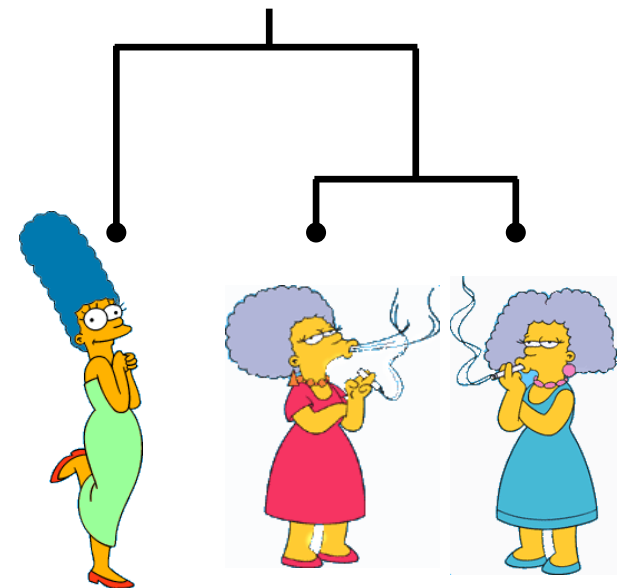
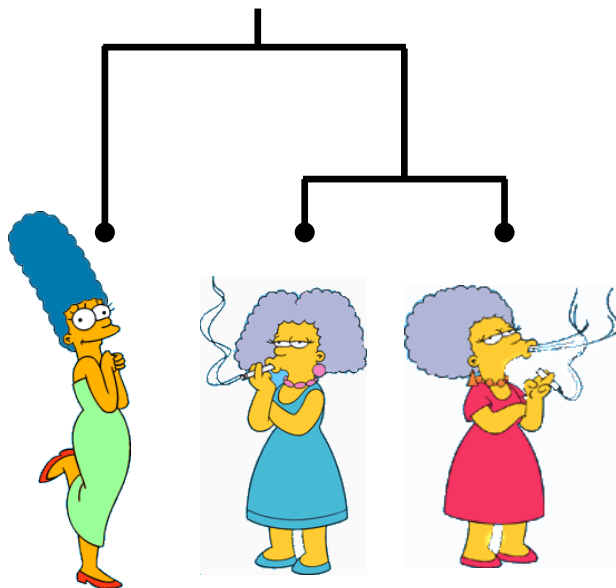
- Scalability (in terms of both time and space)
- Ability to deal with different data types
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- Incorporation of user-specified constraints
- Interpretability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

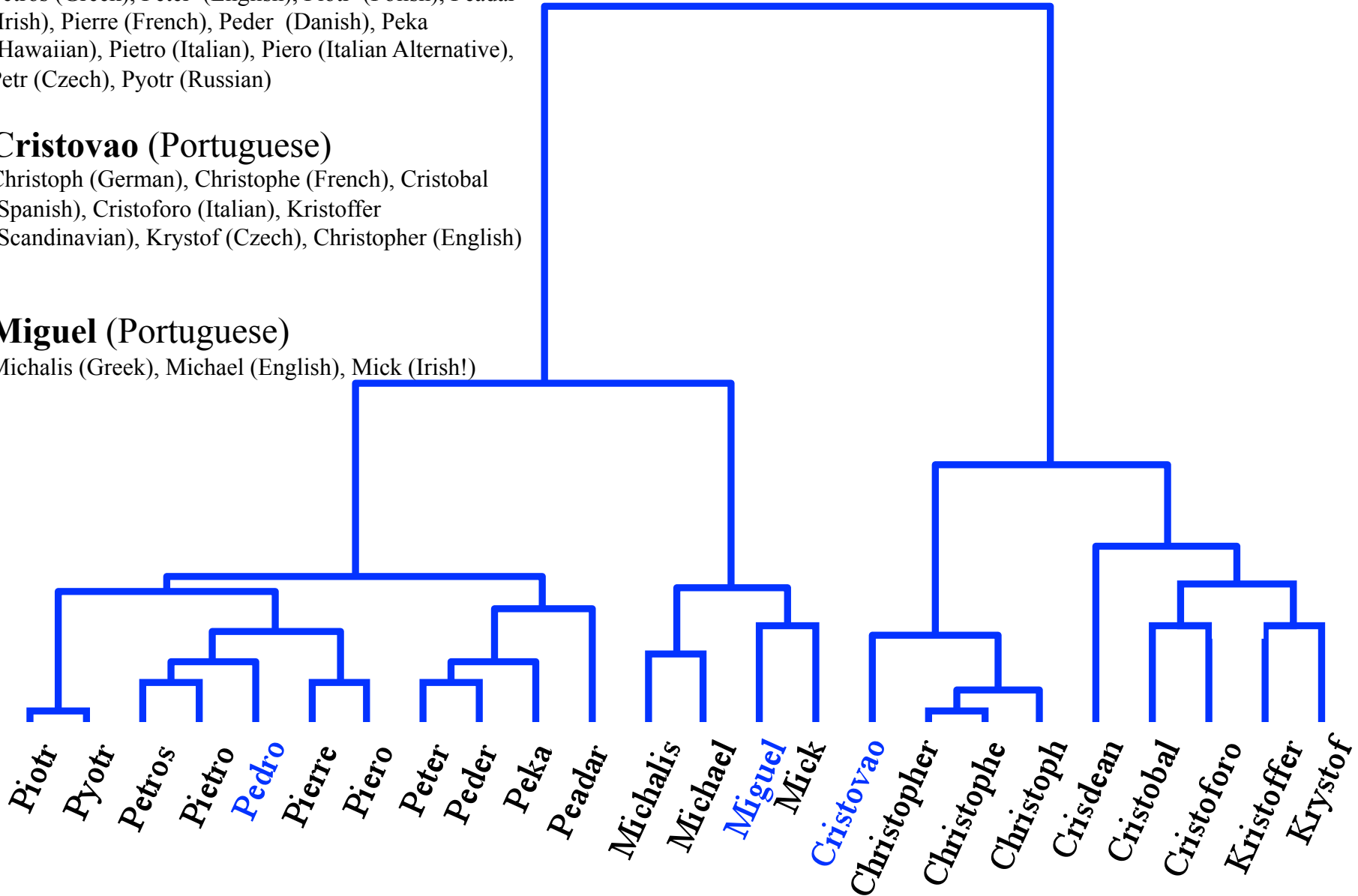
Petros (Greek), Peter (English), Piotr (Polish), Peadar (Irish), Pierre (French), Peder (Danish), Peka (Hawaiian), Pietro (Italian), Piero (Italian Alternative), Petr (Czech), Pyotr (Russian)

Cristovao (Portuguese)

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

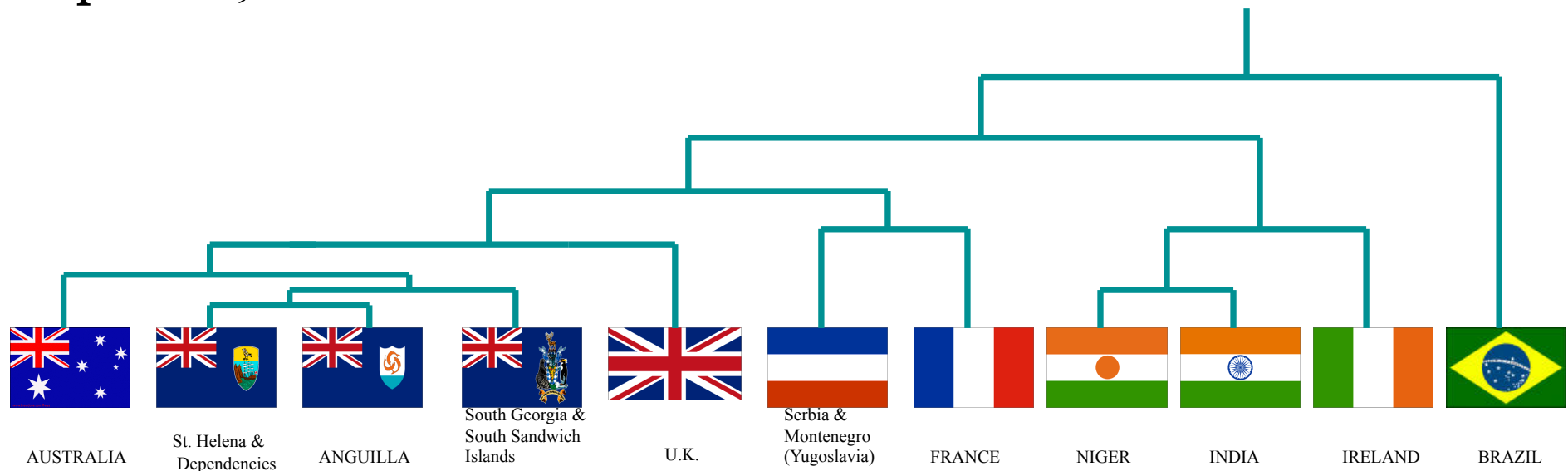
Miguel (Portuguese)

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

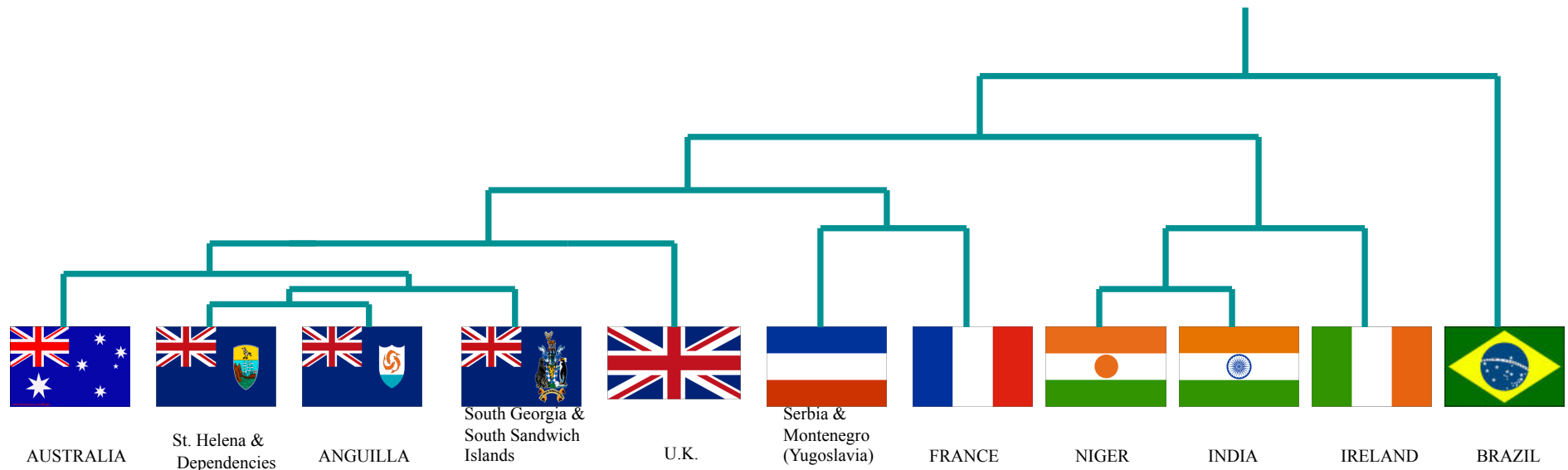


Hierarchical clustering can sometimes show patterns that are meaningless or spurious

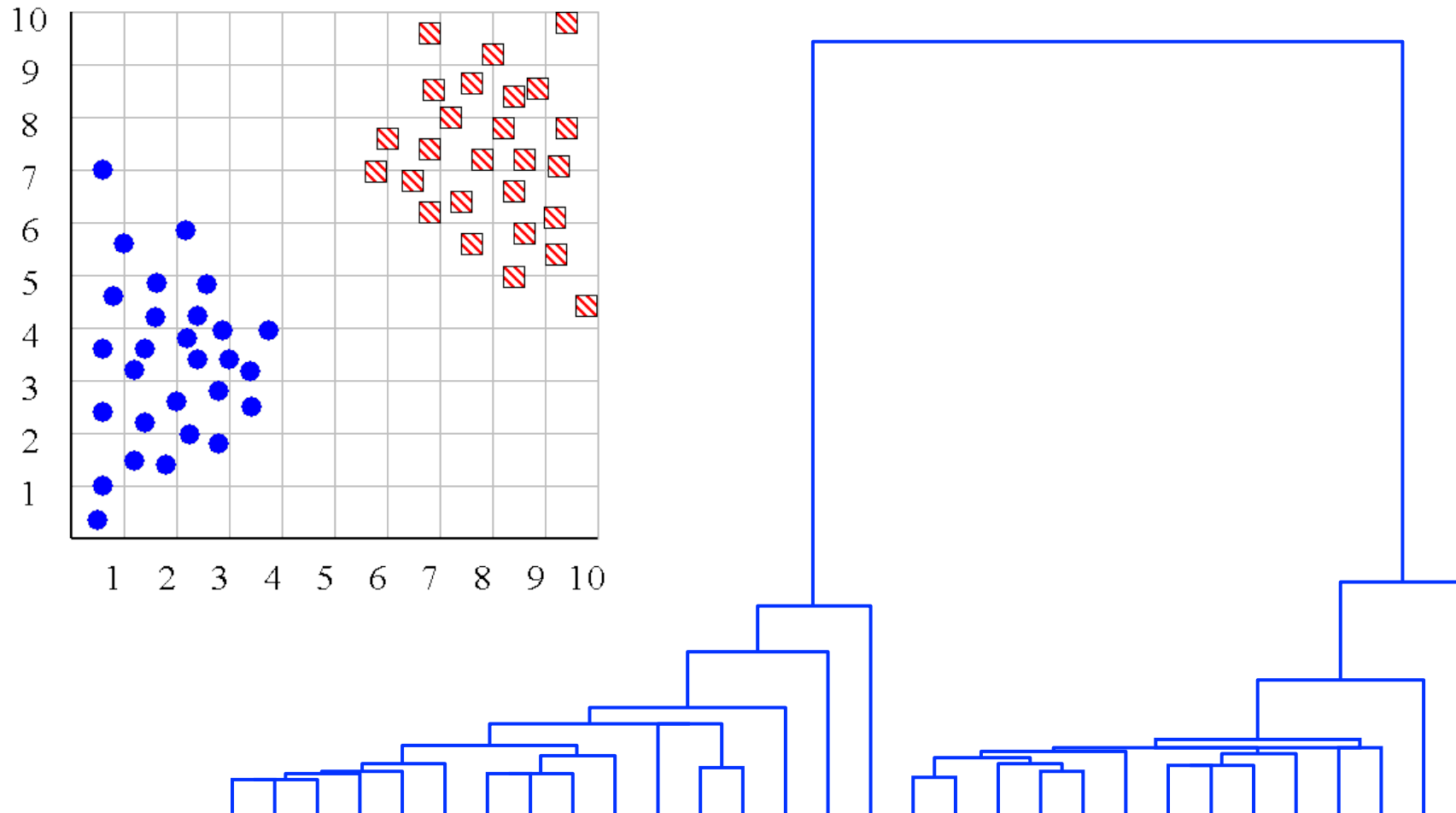
- For example, in this clustering, the tight grouping of Australia, Anguilla, St. Helena etc is meaningful, since all these countries are former UK colonies.
- However the tight grouping of Niger and India is completely spurious, there is no connection between the two.



- 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.

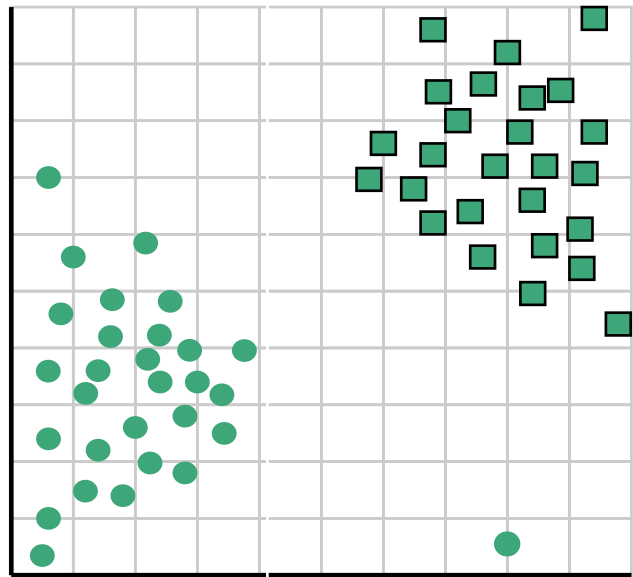


We can look at the dendrogram to determine the “correct” number of clusters. In this case, the two highly separated subtrees are highly suggestive of two clusters. (Things are rarely this clear cut, unfortunately)

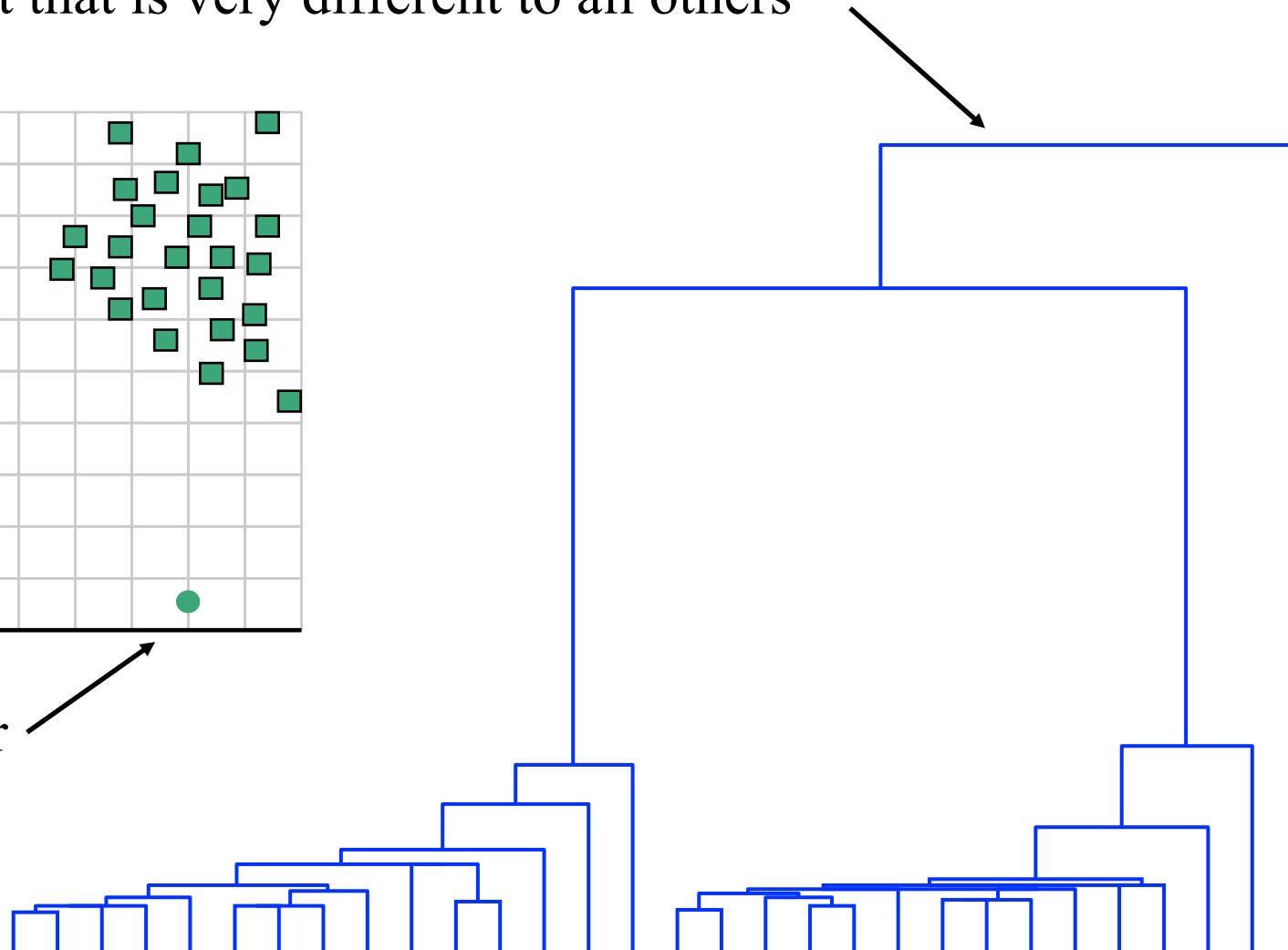


One potential use of a dendrogram is to detect outliers

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



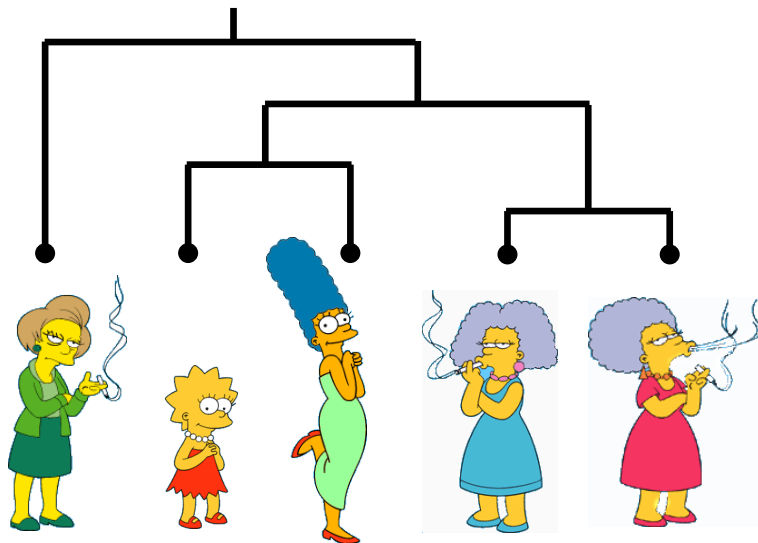
Outlier



(How-to) Hierarchical Clustering

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

Number of Leafs	Number of Possible Dendrograms
2	1
3	3
4	15
5	105
...	...
18	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.

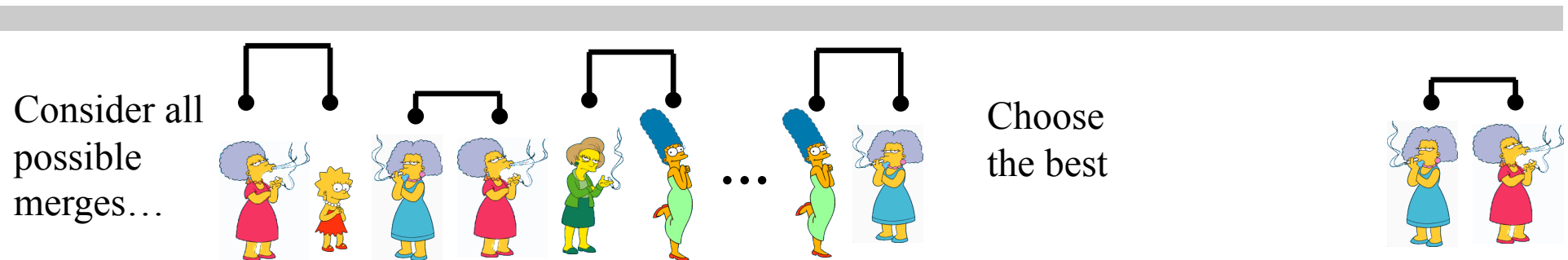
$$D(\text{Mrs. Krabappel, Lisa Simpson}) = 8$$

$$D(\text{Marge Simpson, Mrs. Simpson}) = 1$$

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

Bottom-Up (agglomerative):

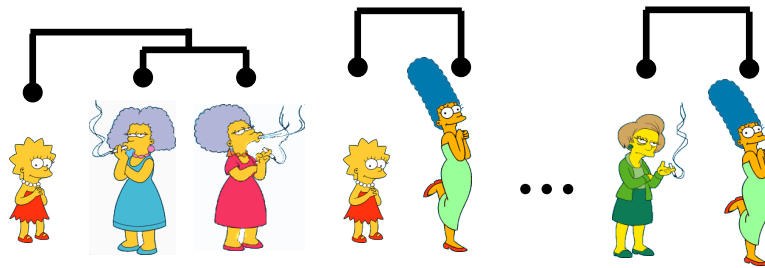
Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



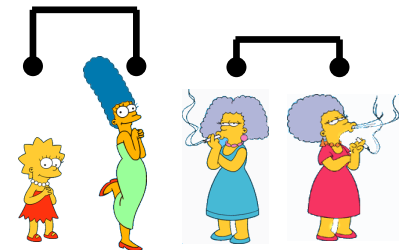
Bottom-Up (agglomerative):

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.

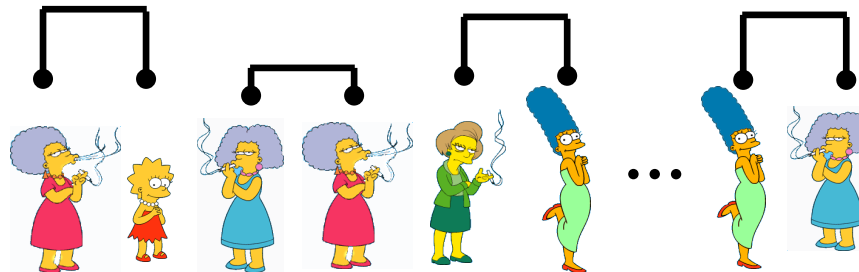
Consider all possible merges...



Choose the best



Consider all possible merges...



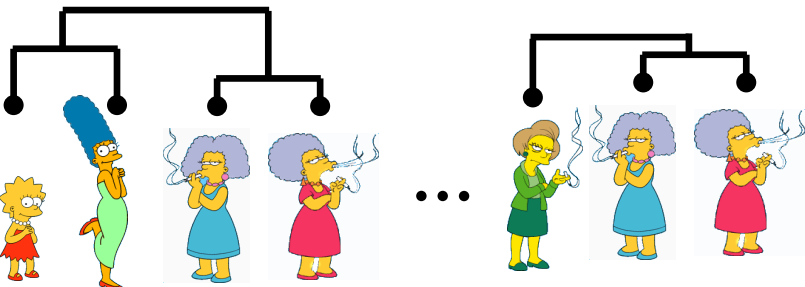
Choose the best



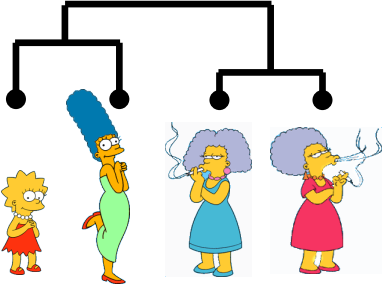
Bottom-Up (agglomerative):

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.

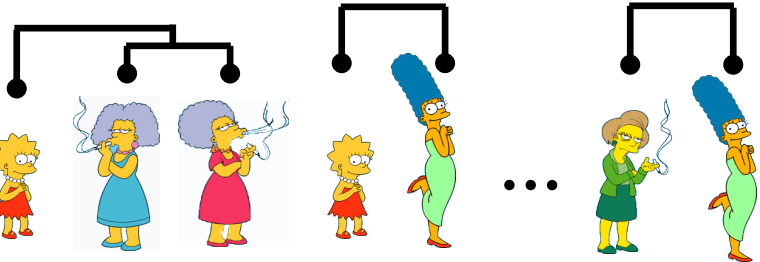
Consider all possible merges...



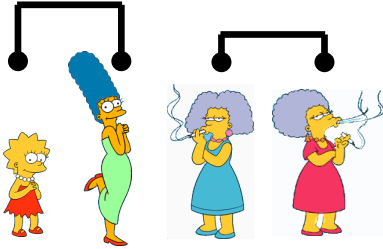
Choose the best



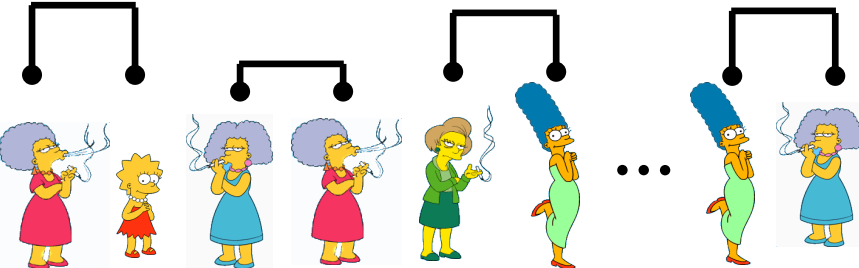
Consider all possible merges...



Choose the best



Consider all possible merges...

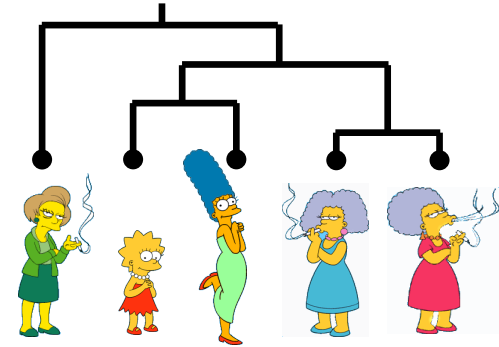


Choose the best

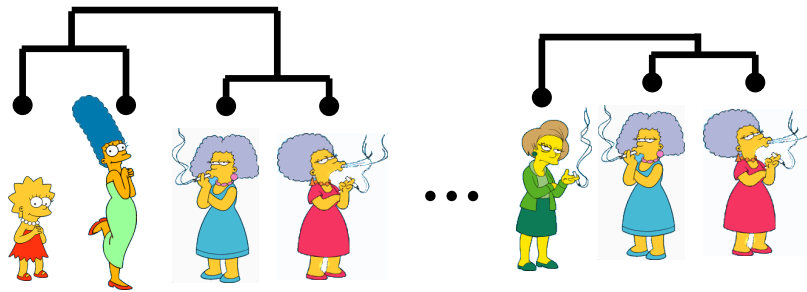


Bottom-Up (agglomerative):

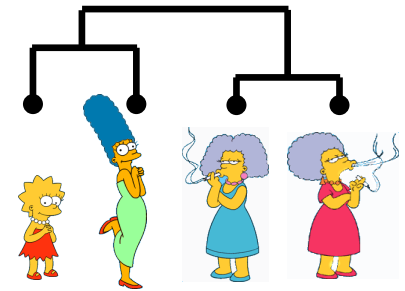
Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.



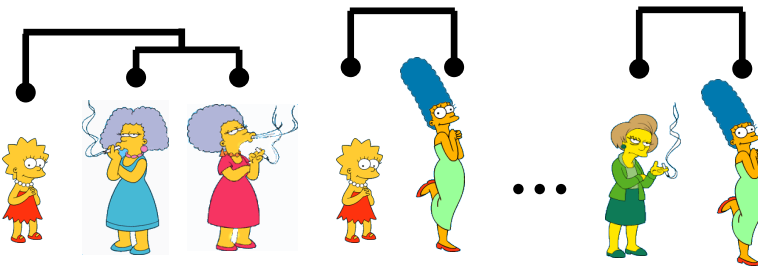
Consider all possible merges...



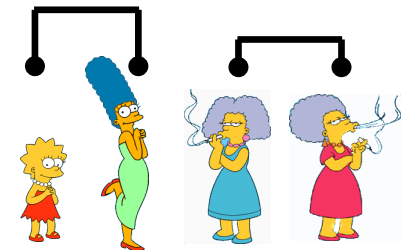
Choose the best



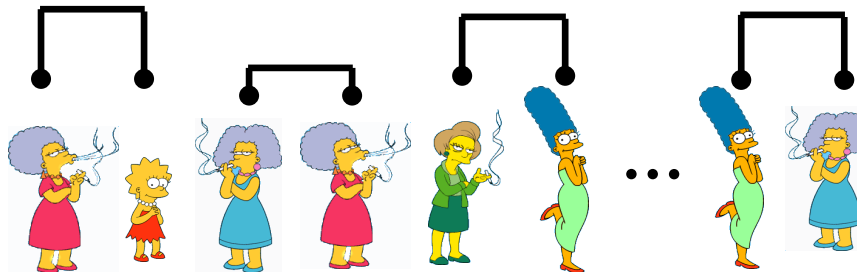
Consider all possible merges...



Choose the best



Consider all possible merges...

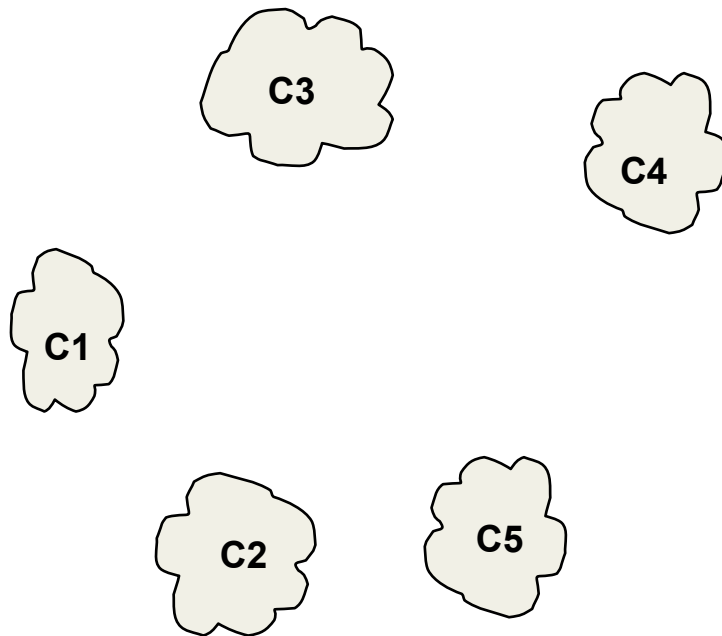


Choose the best



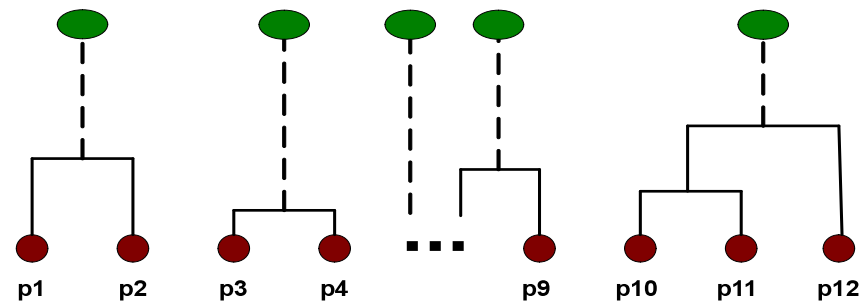
Intermediate State

- After some merging steps, we have some clusters



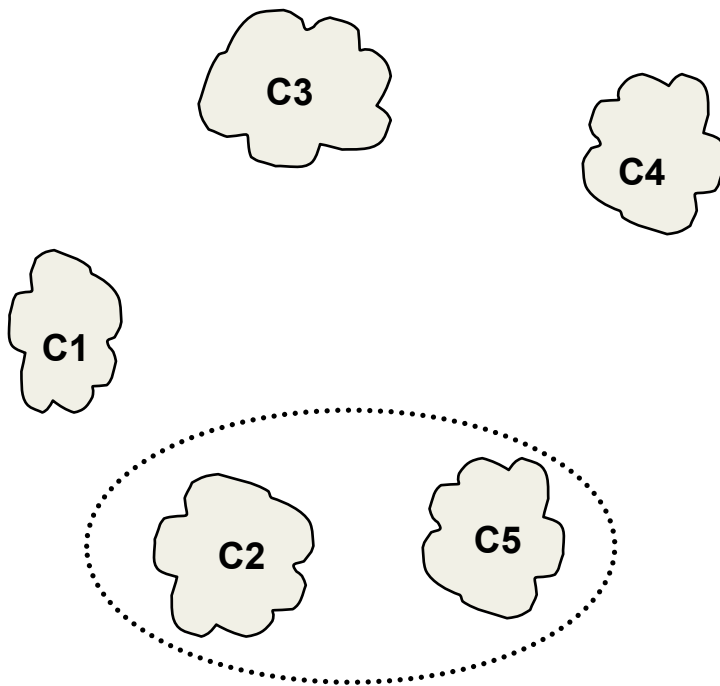
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix



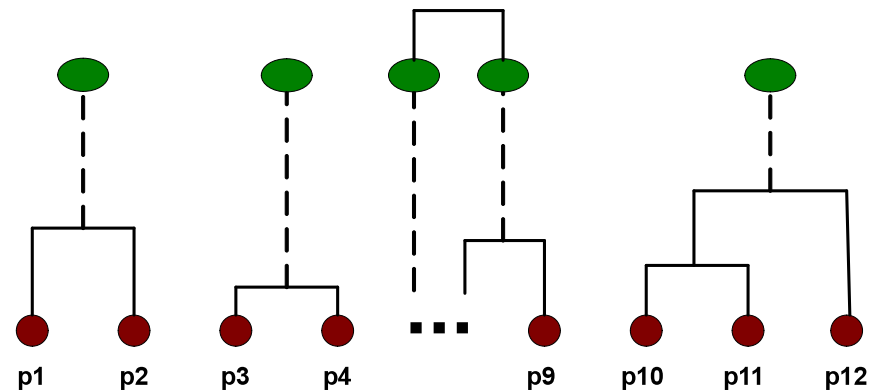
Intermediate State

- Merge the two closest clusters (C2 and C5) and update the distance matrix.



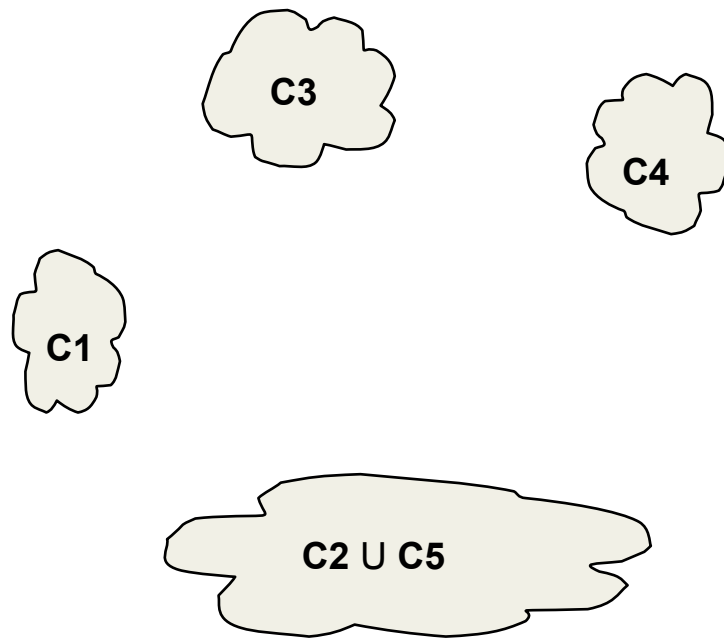
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix

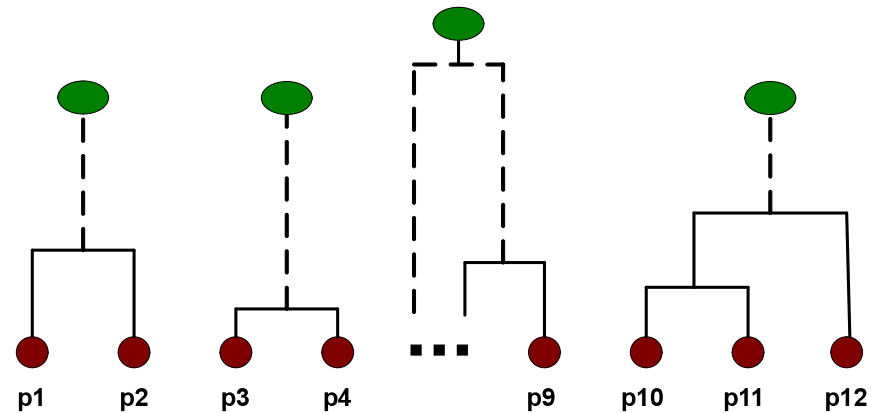


After Merging

- “How do we update the distance matrix?”



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		



Distance between two clusters

- **Single-link distance** between clusters C_i and C_j is the *minimum distance* between any object in C_i and any object in C_j
- The distance is **defined by the two most similar objects**

$$D_{sl}(C_i, C_j) = \min_{x,y} \{ d(x,y) \mid x \in C_i, y \in C_j \}$$

Single-link clustering: example

- Determined by one pair of points, i.e., by one link in the similarity graph.

	l1	l2	l3	l4	l5
l1	1.00	0.90	0.10	0.65	0.20
l2	0.90	1.00	0.70	0.60	0.50
l3	0.10	0.70	1.00	0.40	0.30
l4	0.65	0.60	0.40	1.00	0.80
l5	0.20	0.50	0.30	0.80	1.00

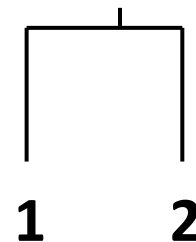
	l12	l3	l4	l5
l12	1,00	0,70	0,65	0,50
l3	0,70	1,00	0,40	0,30
l4	0,65	0,40	1,00	0,80
l5	0,50	0,30	0,80	1,00

Single-link clustering: example

- Determined by one pair of points, i.e., by one link in the proximity graph.

	112	13	14	15
112	1,00	0,70	0,65	0,50
13	0,70	1,00	0,40	0,30
14	0,65	0,40	1,00	0,80
15	0,50	0,30	0,80	1,00

	112	13	145
112	1,00	0,70	0,65
13	0,70	1,00	0,40
145	0,65	0,40	1,00

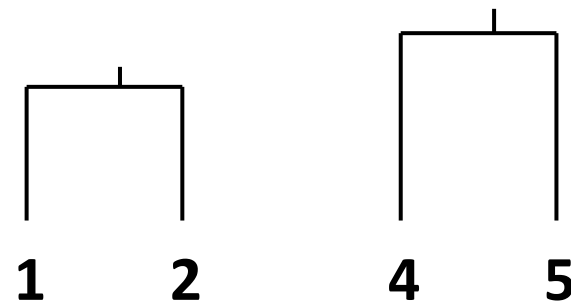


Single-link clustering: example

- Determined by one pair of points, i.e., by one link in the proximity graph.

	112	13	145
112	1,00	0,70	0,65
13	0,70	1,00	0,40
145	0,65	0,40	1,00

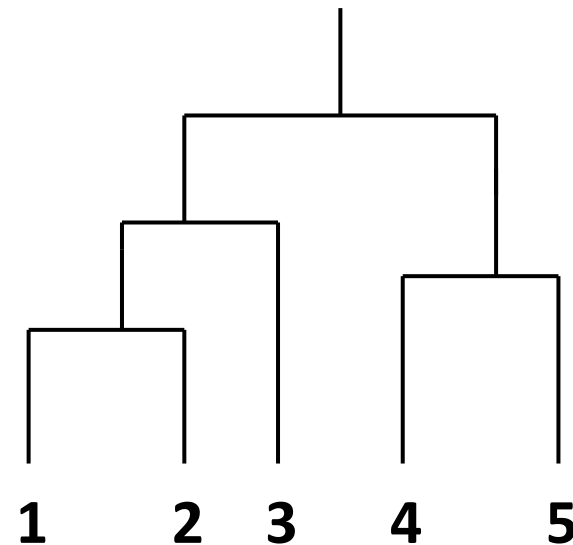
	1123	145
1123	1,00	0,65
145	0,65	1,00



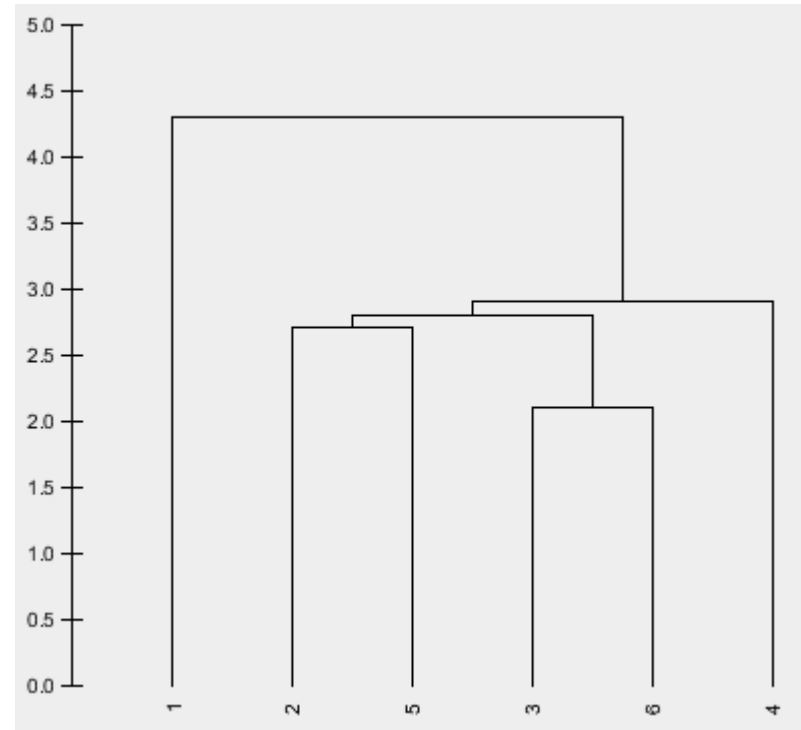
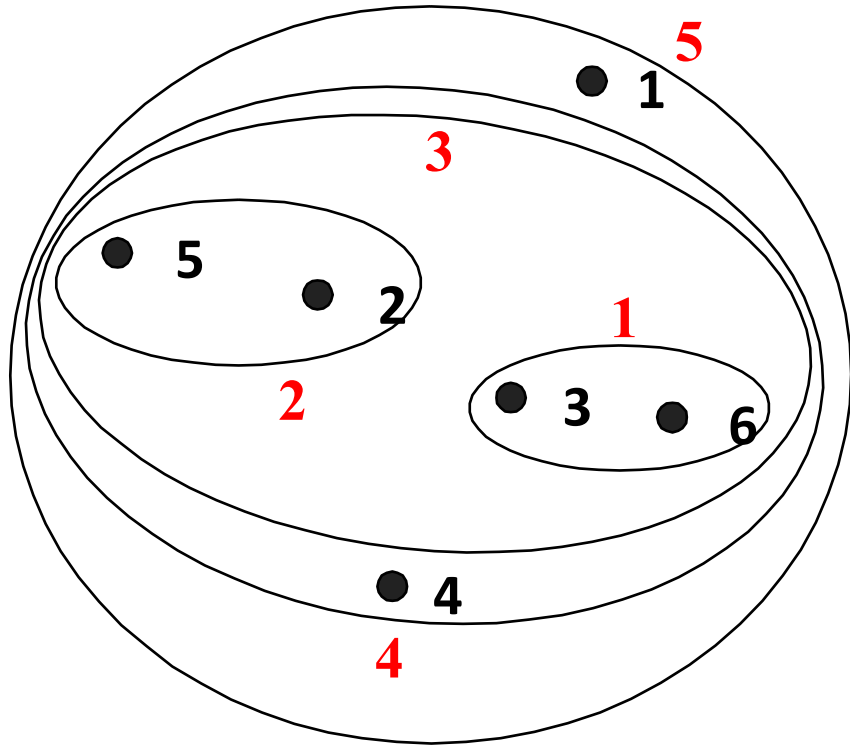
Single-link clustering: example

- Determined by one pair of points, i.e., by one link in the proximity graph.

	123	45
123	1,00	0,65
45	0,65	1,00



Single-link clustering: example

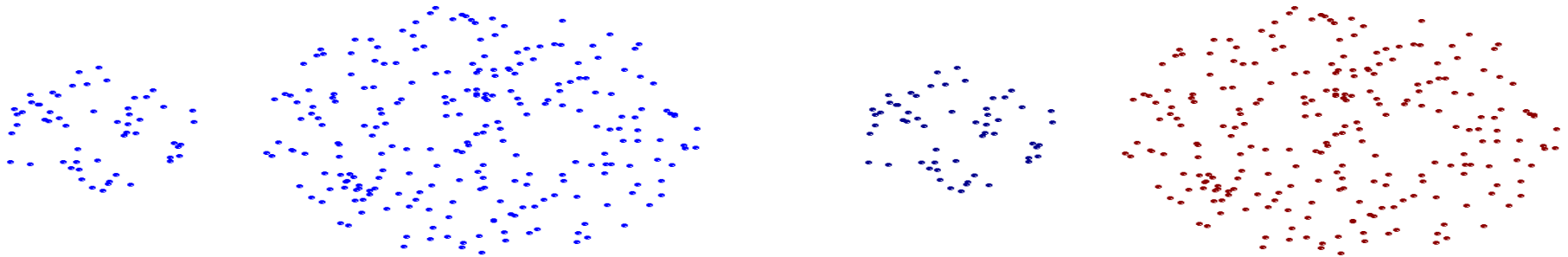


Nested Clusters

	1	2	3	4	5	6
1	0,0	4,5	4,3	7,3	6,7	4,6
2	4,5	0,0	2,8	3,9	2,7	5,0
3	4,3	2,8	0,0	2,9	5,5	2,1
4	7,3	3,9	2,9	0,0	5,7	4,4
5	6,7	2,7	5,5	5,7	0,0	7,7
6	4,6	5,0	2,1	4,4	7,7	0,0

Dendrogram

Strengths of single-link clustering

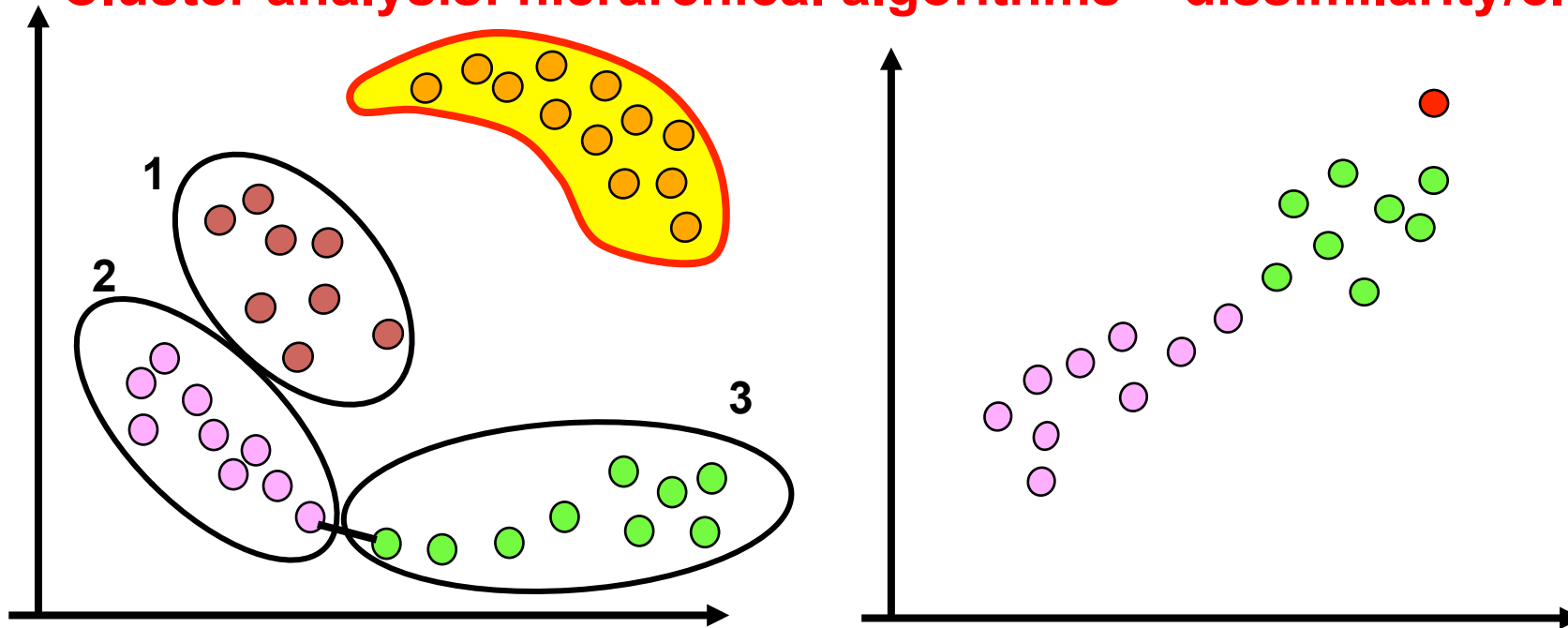


Original Points

Two Clusters

- **Can handle elliptical shapes**

Cluster analysis: hierarchical algorithms – dissimilarity/clusters



Single linkage: It is a flexible method and it can individuate also clusters with particular shapes (elongated, elliptical)

When clusters are not well separated this method may lead to unsatisfactory solutions due to the so called **chaining effect**.

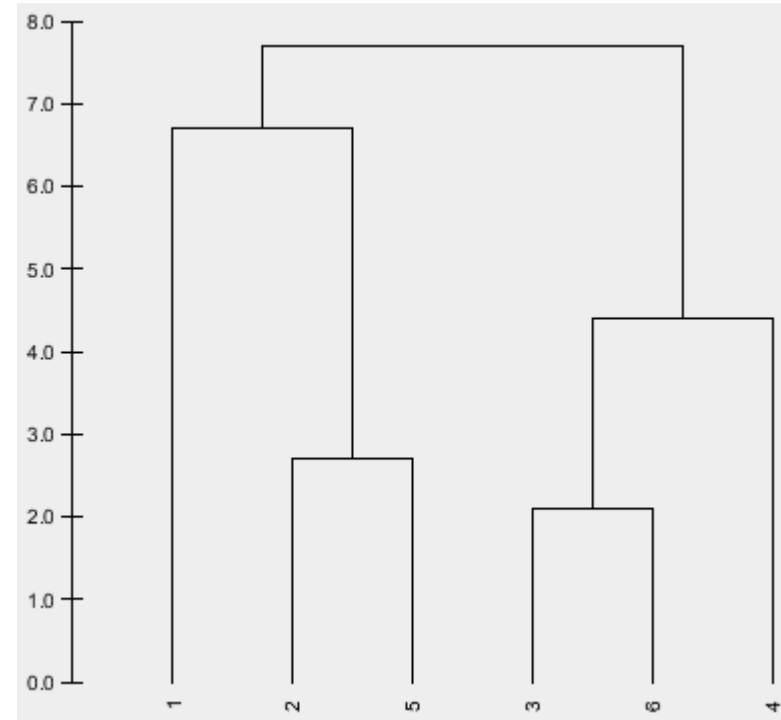
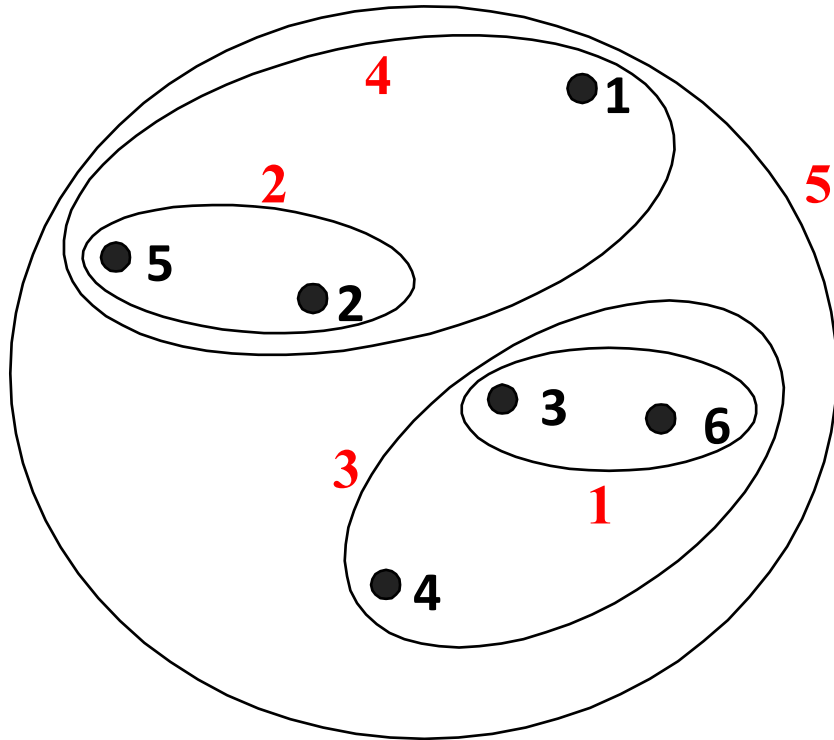
- in the left panel. Clusters 1 and 2 are (“globally”) closer.
- due to the presence of two very close cases in clusters 2 and 3, they will be joined instead.
- The example in the right panel evidences that this method may be useful in outliers detection.

Distance between two clusters

- **Complete-link distance** between clusters C_i and C_j is the *maximum distance* between any object in C_i and any object in C_j
- The distance is **defined by the two most dissimilar objects**

$$D_{cl}(C_i, C_j) = \max_{x,y} \{d(x,y) \mid x \in C_i, y \in C_j\}$$

Complete-link clustering: example

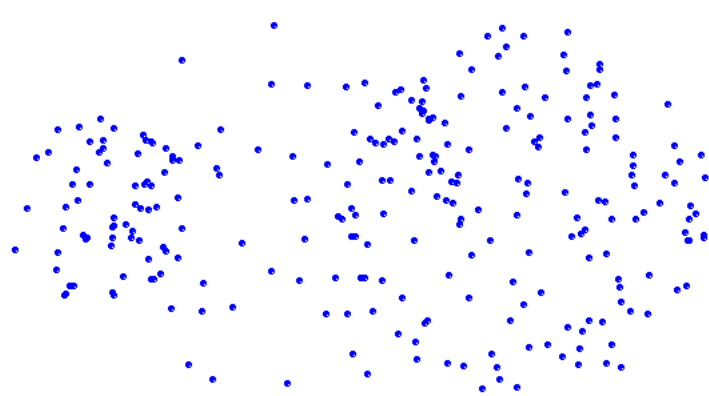


Nested Clusters

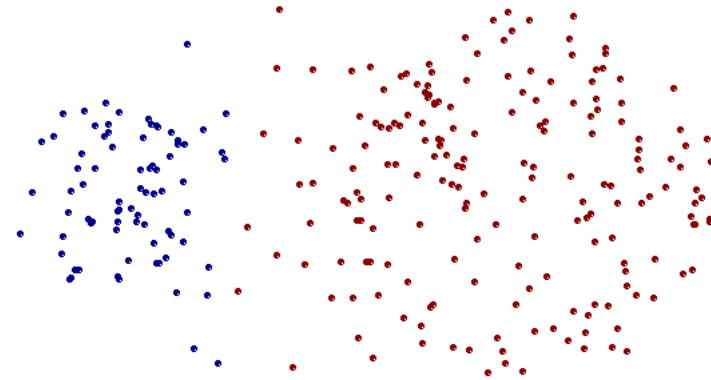
	1	2	3	4	5	6
1	0,0	4,5	4,3	7,3	6,7	4,6
2	4,5	0,0	2,8	3,9	2,7	5,0
3	4,3	2,8	0,0	2,9	5,5	2,1
4	7,3	3,9	2,9	0,0	5,7	4,4
5	6,7	2,7	5,5	5,7	0,0	7,7
6	4,6	5,0	2,1	4,4	7,7	0,0

Dendrogram

Strengths of complete-link clustering



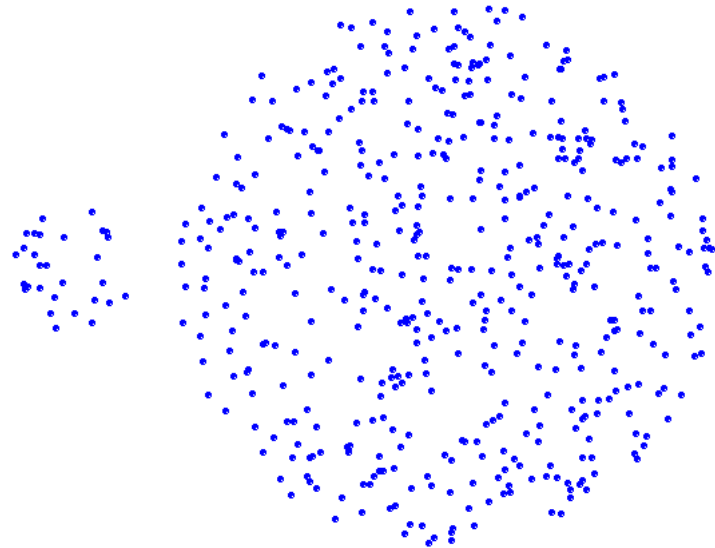
Original Points



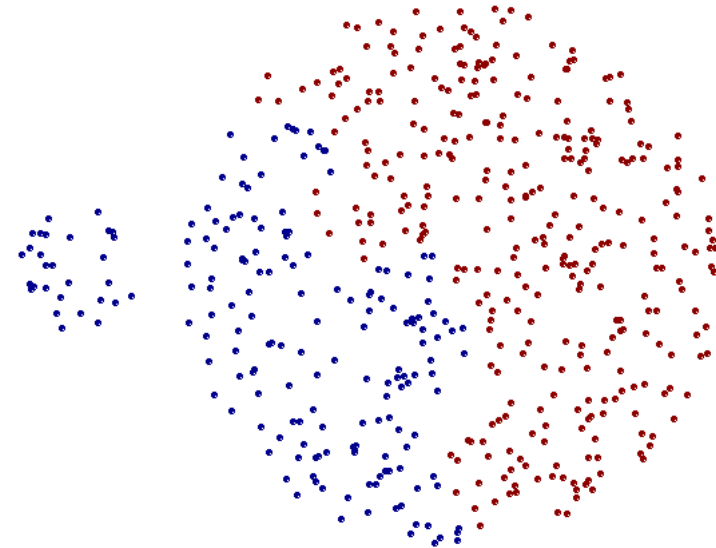
Two Clusters

- **More balanced clusters (with equal diameter)**
- **Less susceptible to noise**

Limitations of complete-link clustering



Original Points



Two Clusters

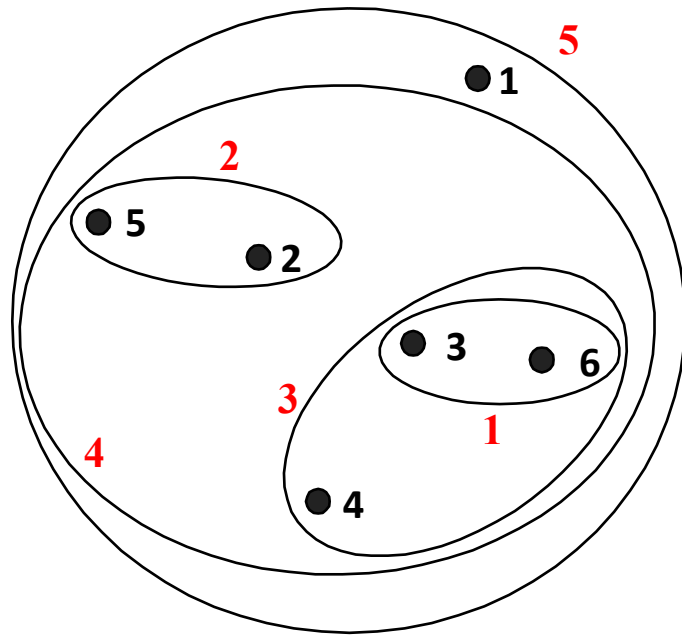
- **Tends to break large clusters**
- **All clusters tend to have the same diameter – small clusters are merged with larger ones**

Distance between two clusters

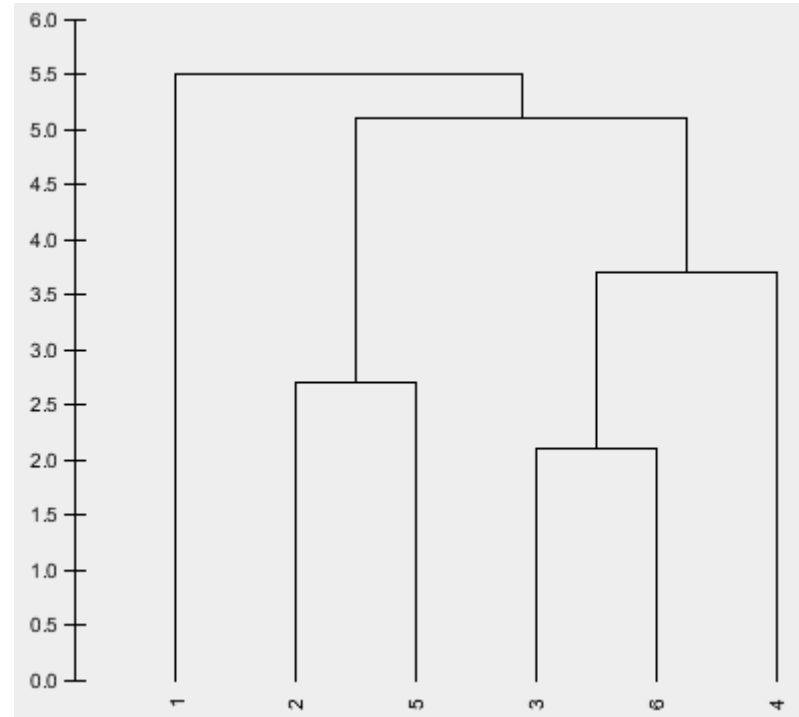
- **Group average distance** between clusters C_i and C_j is the *average distance* between any object in C_i and any object in C_j

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

Average-link clustering: example



Nested Clusters



Dendrogram

Average-link clustering: discussion

- Compromise between Single and Complete Link
- Strengths
 - Less susceptible to noise and outliers
- Limitations
 - Biased towards globular clusters

Distance between two clusters

- **Centroid distance** between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}(C_i, C_j) = d(r_i, r_j)$$

Distance between two clusters

- **Ward's distance** between clusters C_i and C_j is the *difference* between the *total within cluster sum of squares for the two clusters separately*, and the *within cluster sum of squares resulting from merging the two clusters* in cluster C_{ij}

$$D_w(C_i, C_j) = \sum_{x \in C_{ij}} (x - r_{ij})^2 - \left(\sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 \right)$$

- r_i : centroid of C_i
- r_j : centroid of C_j
- r_{ij} : centroid of C_{ij}

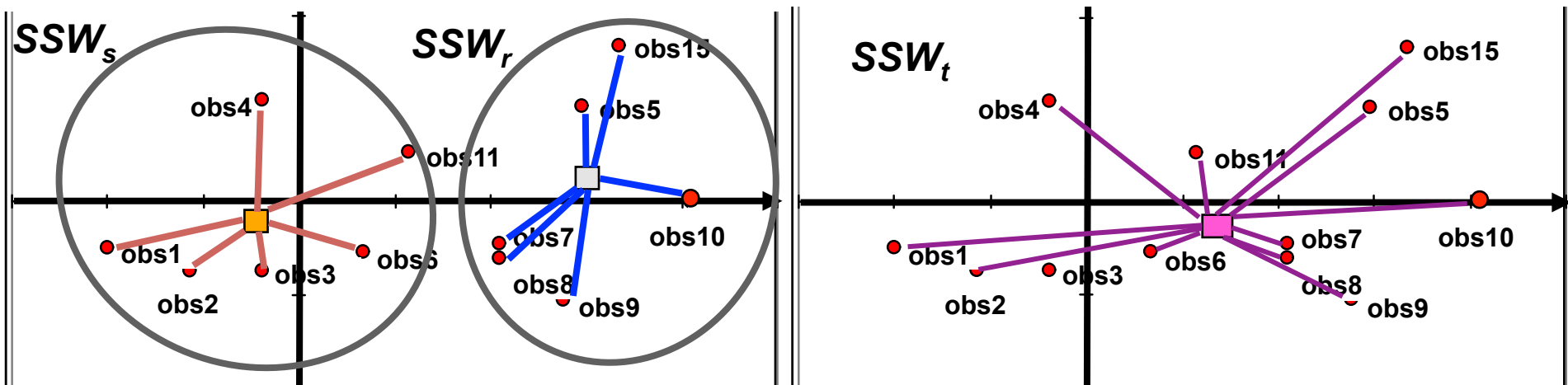
Cluster analysis: hierarchical algorithms – dissimilarity/clusters

Ward's method

It will be $SSW_t > SSW_r + SSW_s$

The quantity $SSW_t - (SSW_r + SSW_s)$ is called **between** sum of squares (SS).

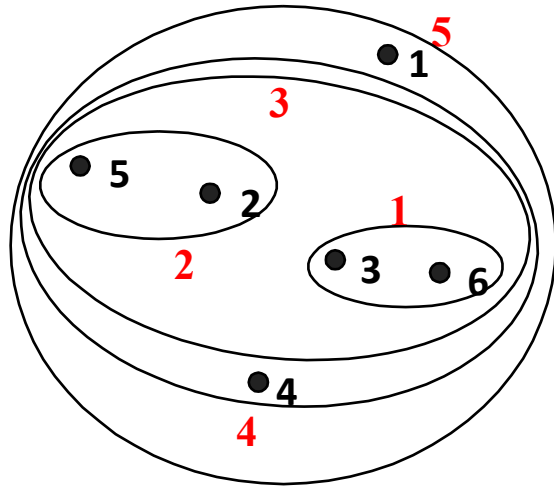
Ward's method: the two clusters with the smallest **Between SS** are joined.



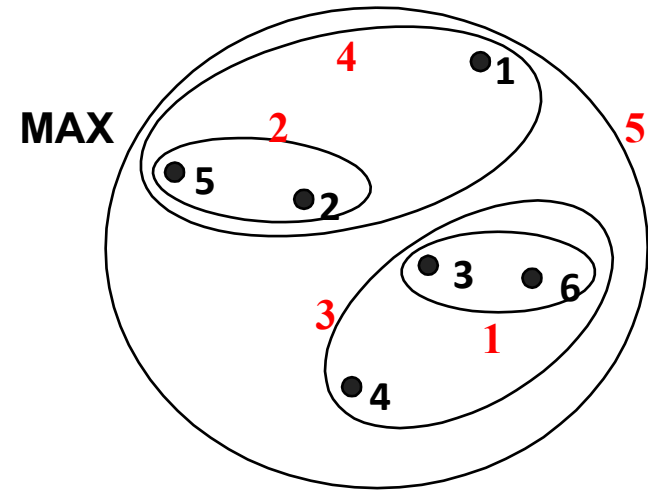
Ward's distance for clusters

- Similar to group average and centroid distance
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of k-means
 - Can be used to initialize k-means

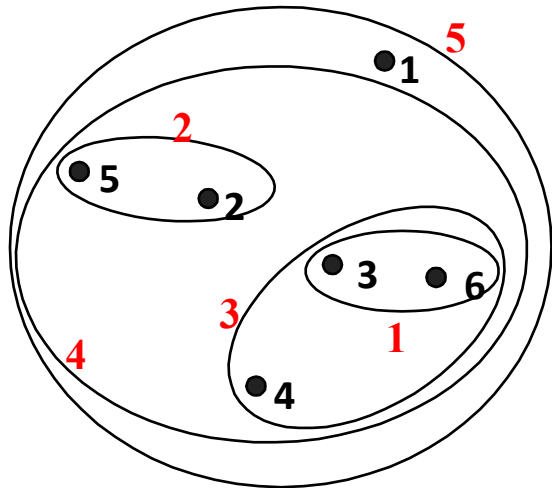
Hierarchical Clustering: Comparison



MIN

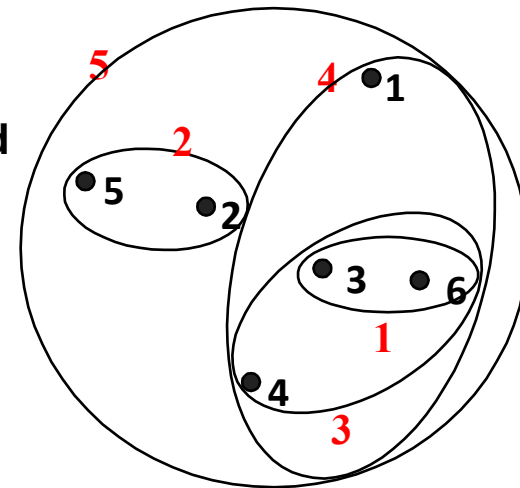


MAX



Group Average

Ward's Method



Divisive Clustering

■ Outline

- Define
 - N_C : Number of clusters
 - N_{EX} : Number of examples

1. Start with one large cluster
2. Find “worst” cluster
3. Split it
4. If $N_C < N_{EX}$ go to 2

■ How to choose the “worst” cluster

- Largest number of examples
- Largest variance
- Largest sum-squared-error
- ...

■ How to split clusters

- Mean-median in one feature direction
- Perpendicular to the direction of largest variance
- ...

■ The computations required by divisive clustering are more intensive than for agglomerative clustering methods

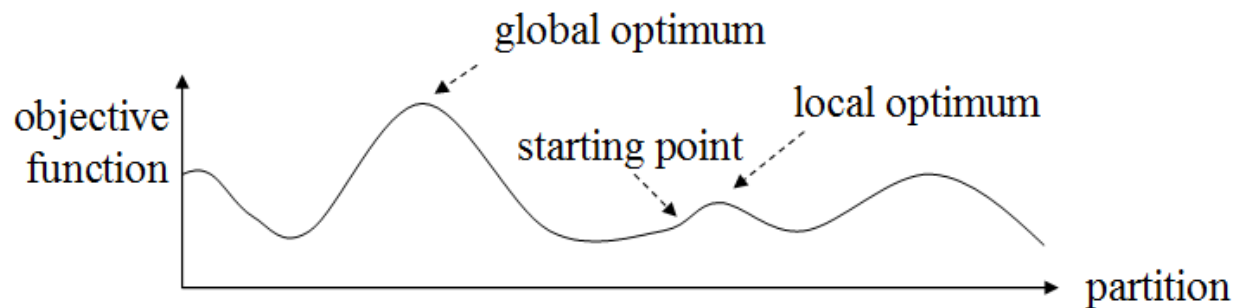
- For this reason, agglomerative approaches are more popular

Hierarchical Clustering: Time and Space requirements

- For a dataset X consisting of n points
- $O(n^2)$ **space**; it requires storing the distance matrix
- $O(n^3)$ **time** in most of the cases
 - There are n steps and at each step the size n^2 distance matrix must be updated and searched
 - Complexity can be reduced to $O(n^2 \log(n))$ time for some approaches by using appropriate data structures

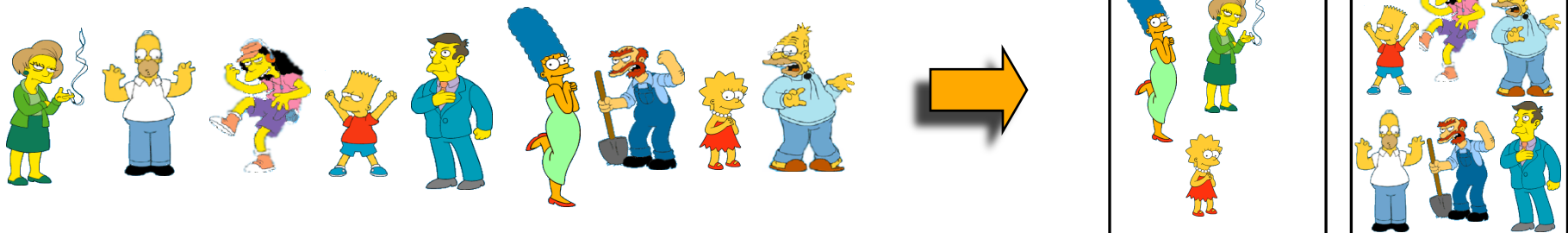
Summary of Hierarchical Clustering Methods

- No need to specify the number of clusters in advance.
- Hierarchical nature maps nicely onto human intuition for some domains
- They do not scale well: time complexity of at least $O(n^2 \log(n))$
- 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 .



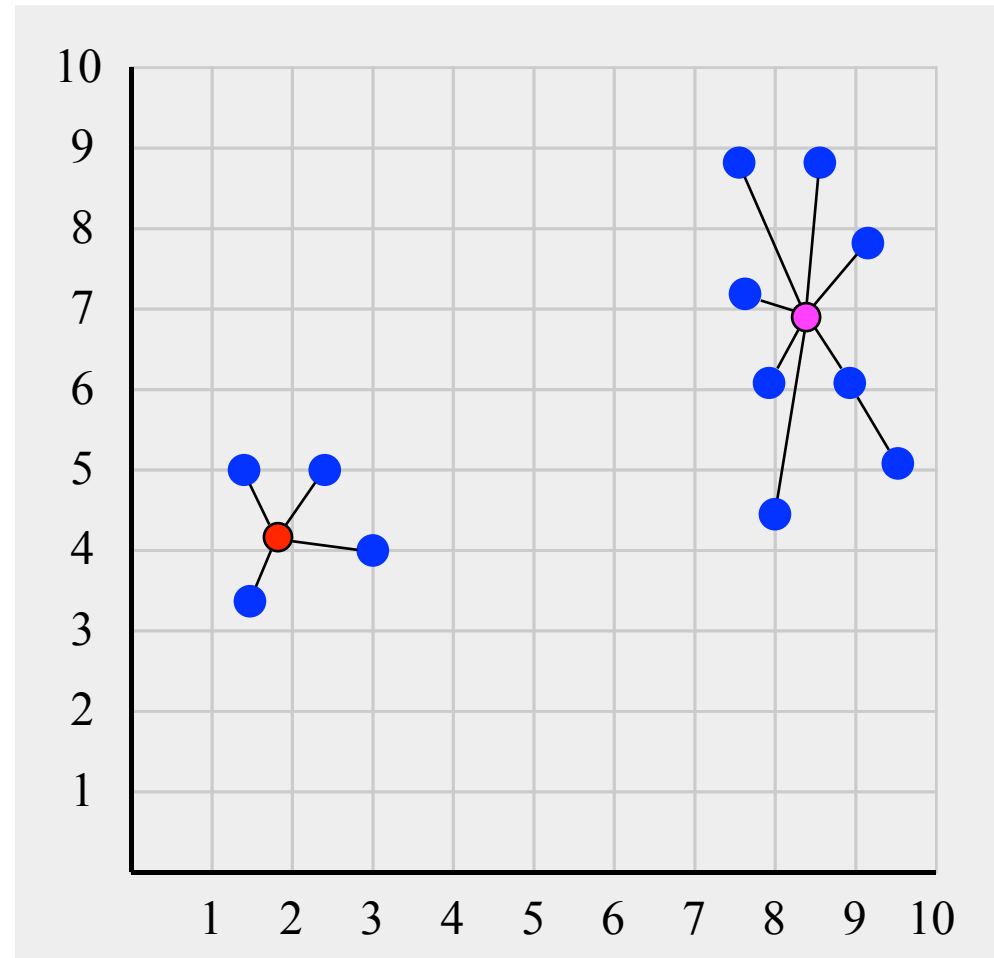
Squared Error

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

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



Objective Function

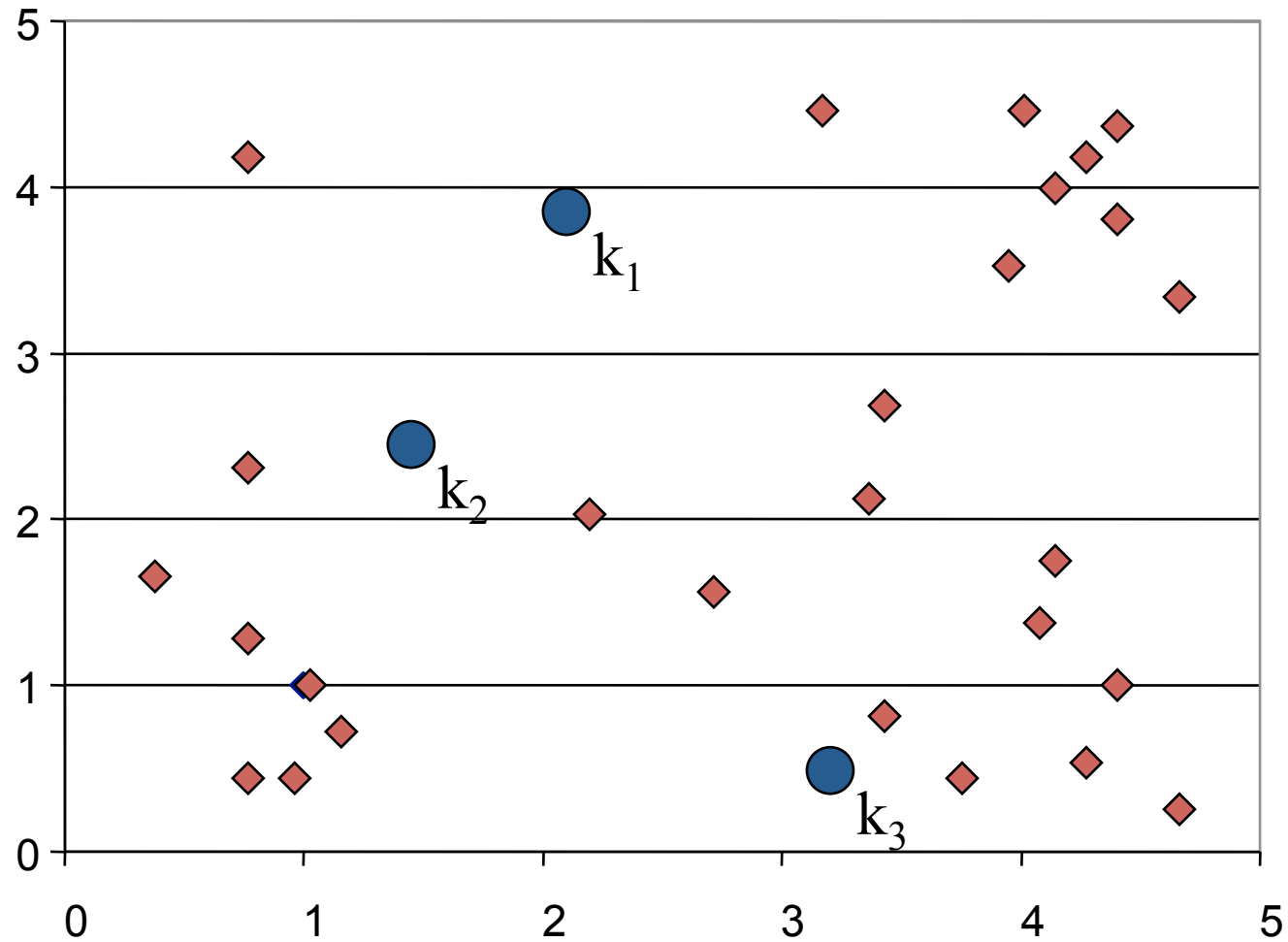


Algorithm *k-means*

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.

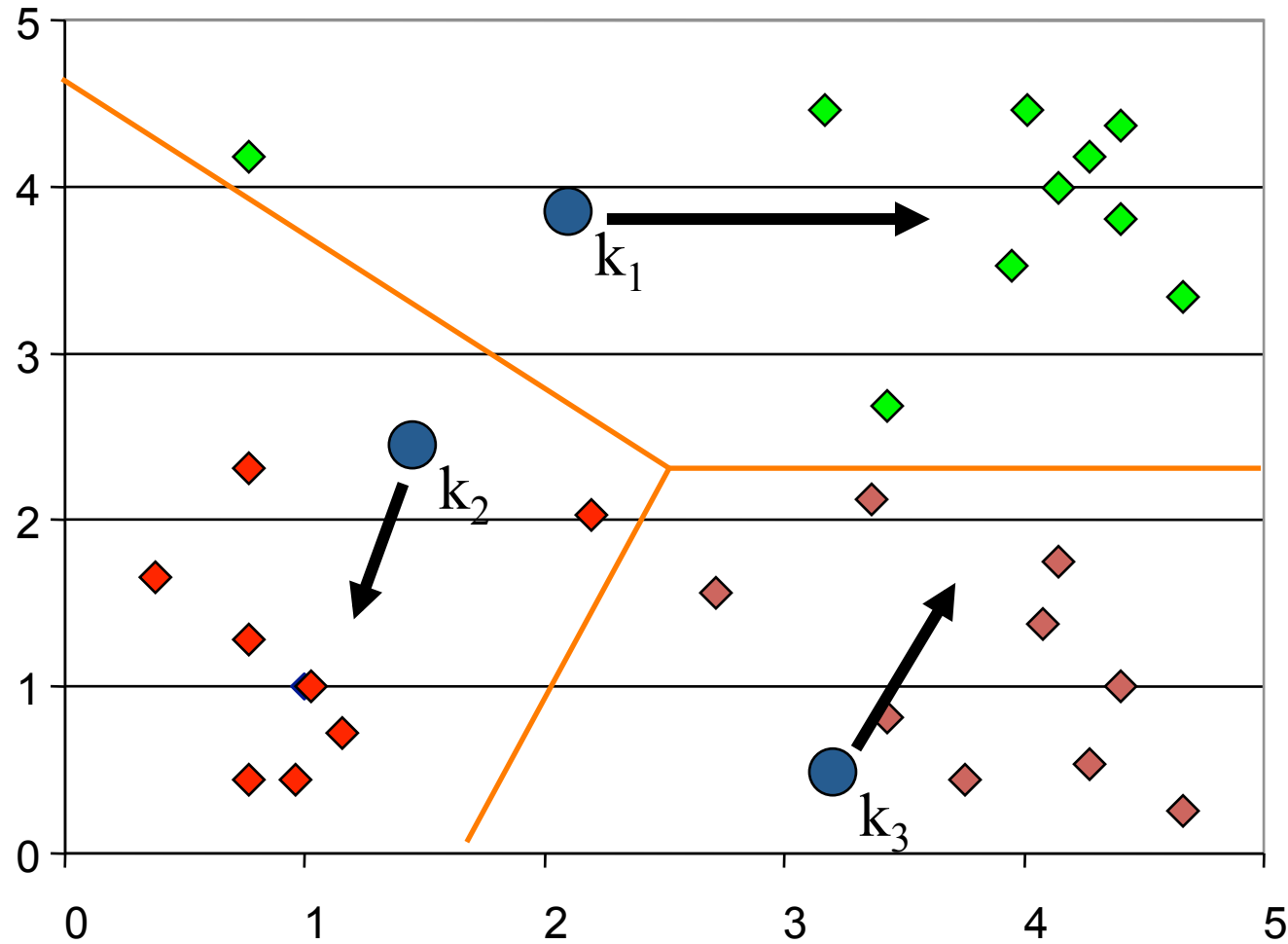
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 2

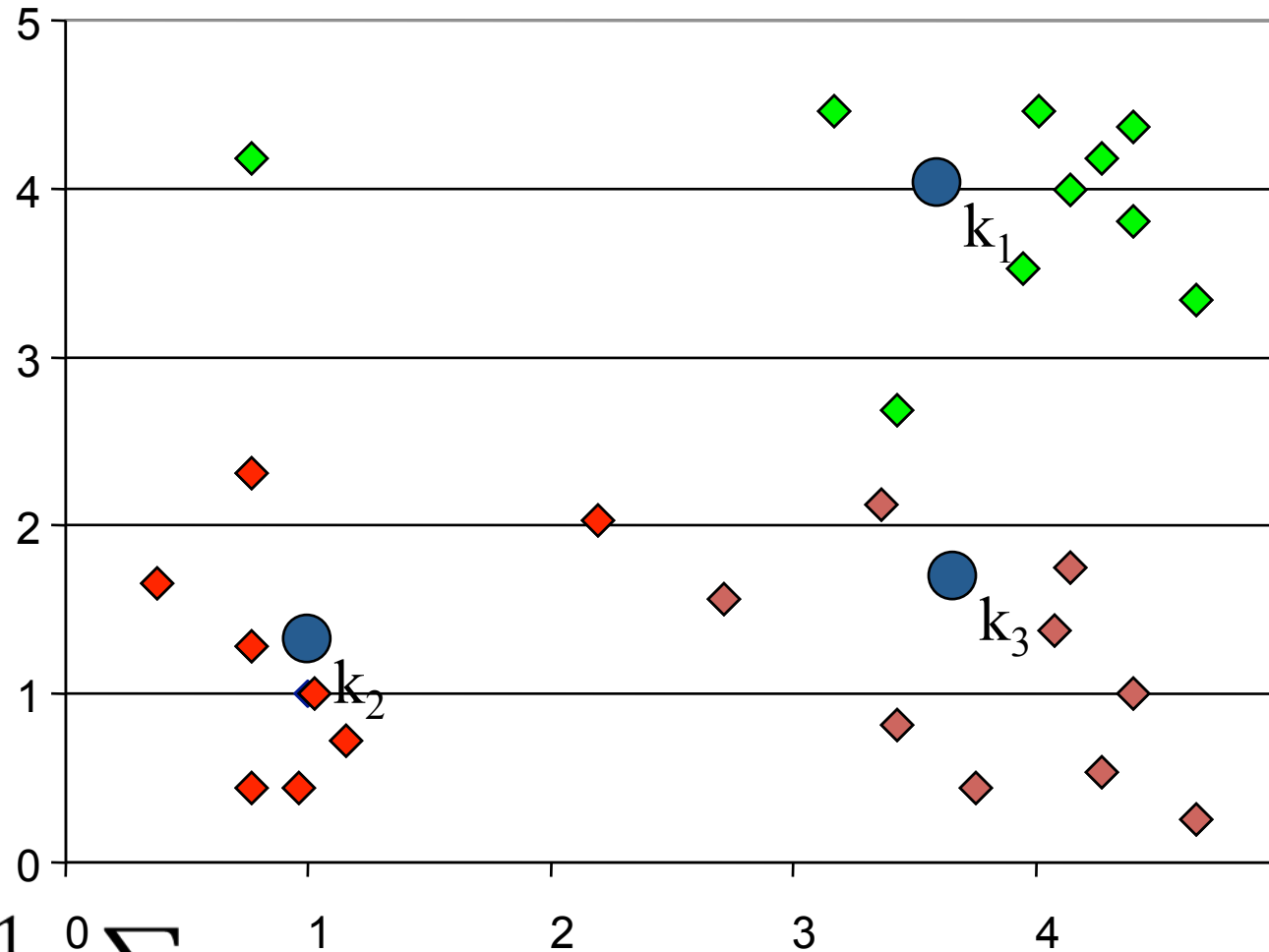
Algorithm: k-means, Distance Metric: Euclidean Distance



$$S_i^t = \{x_j : \left| |x_j - k_i^t| \right| \leq \left| |x_j - k_r^t| \right| \text{ for all } r = 1..k, r \neq i \}$$

K-means Clustering: Step 3

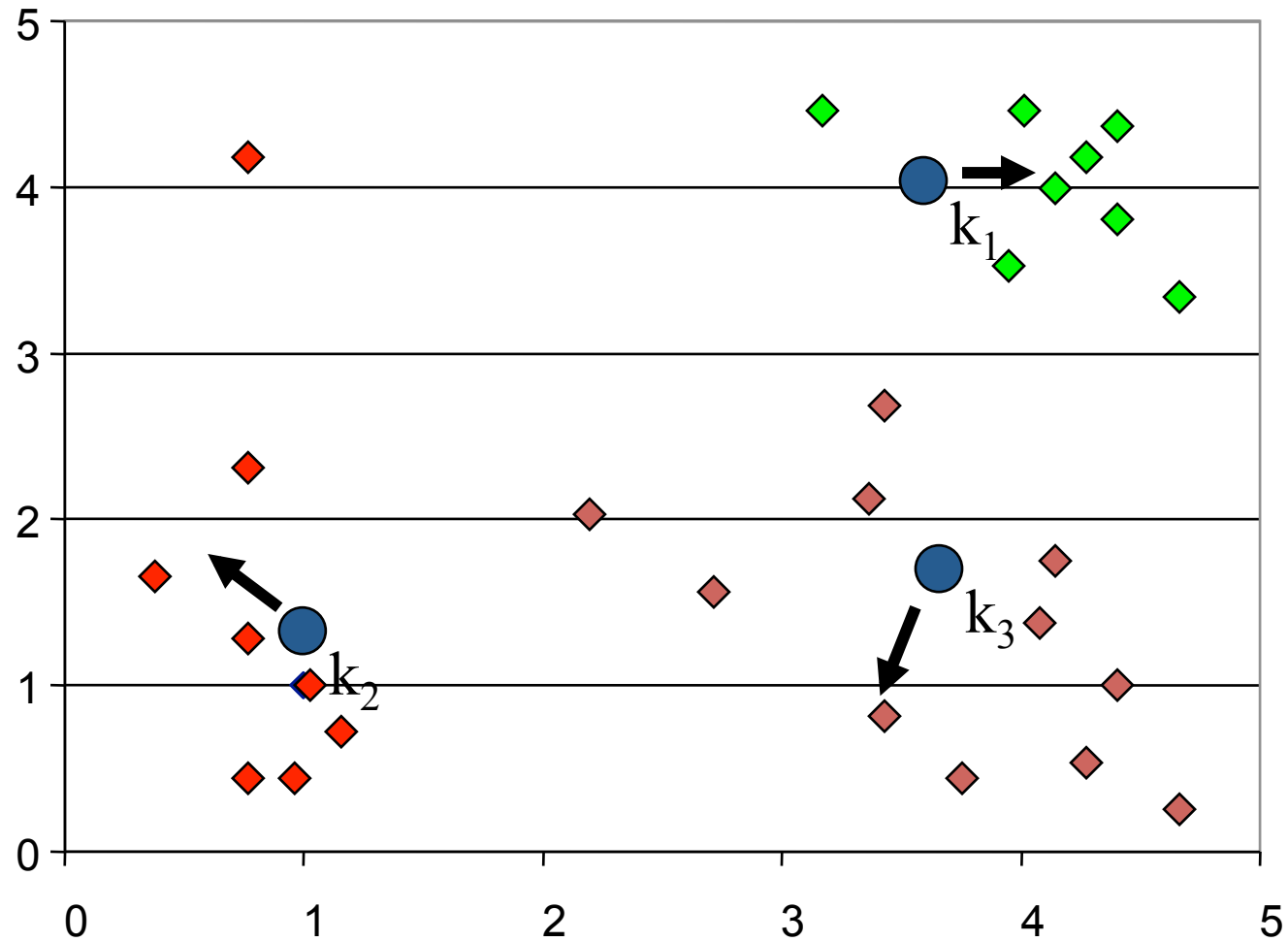
Algorithm: k-means, Distance Metric: Euclidean Distance



$$k_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} x_j$$

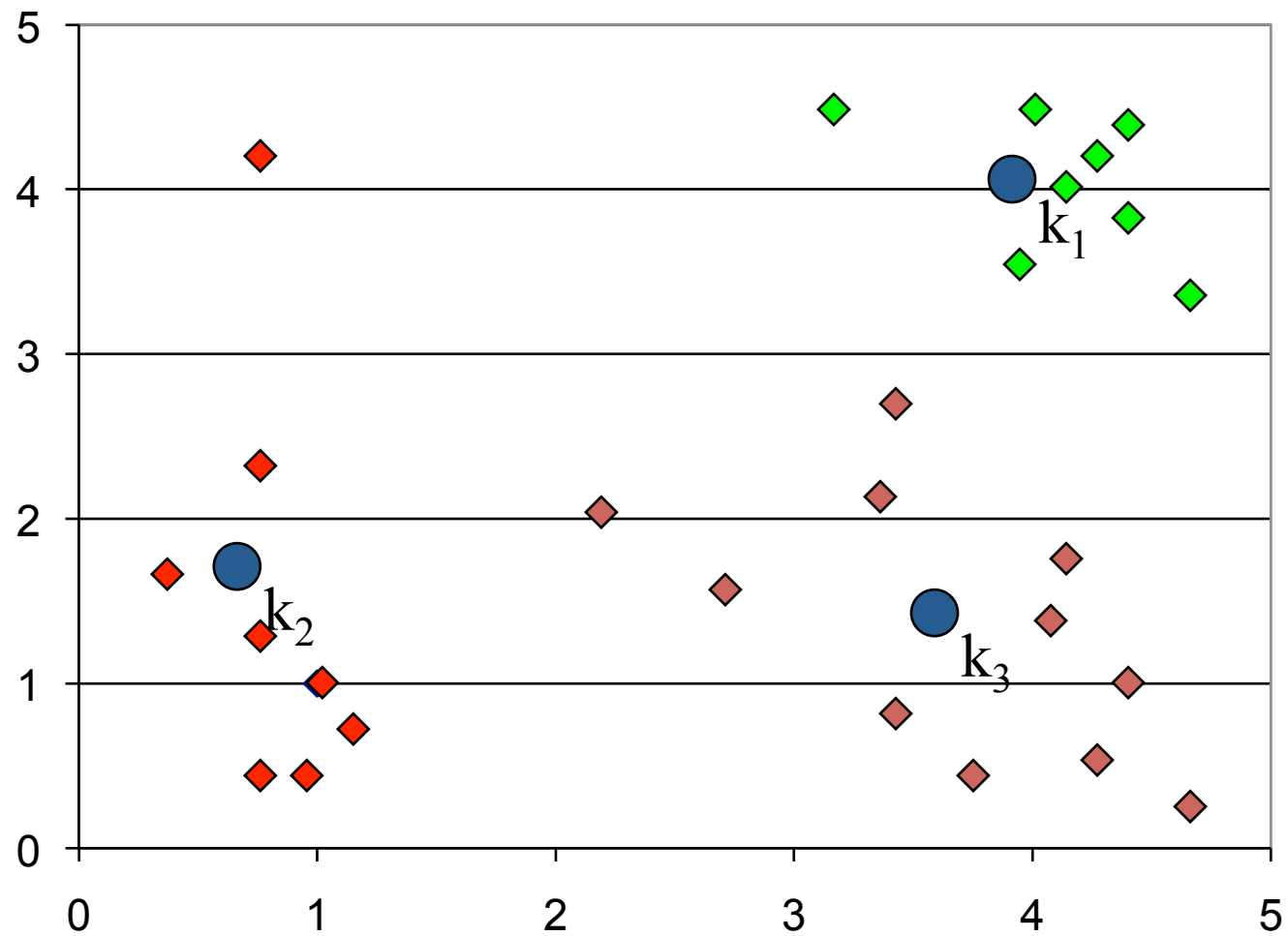
K-means Clustering: Step 4

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 extend the distance measurement.
 - Ahmad, Dey: **A k-mean clustering algorithm for mixed numeric and categorical data, Data & Knowledge Engineering, Nov. 2007**
- 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*
- Tends to build clusters of equal size

EM Algorithm

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

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

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

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$

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

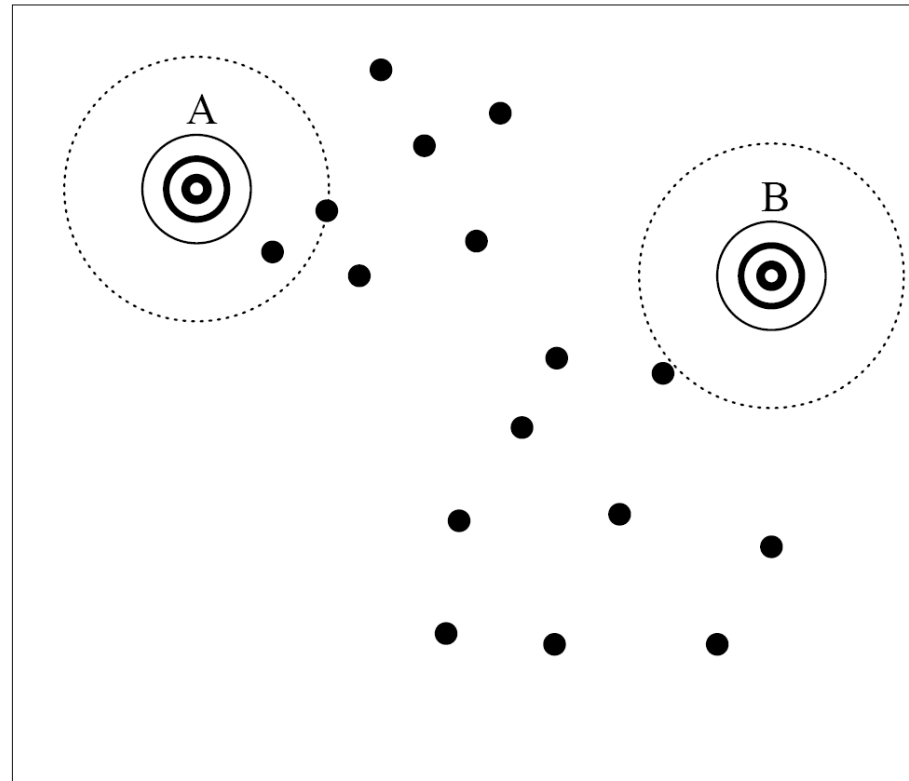
$$\mu_i \leftarrow \sum_j p_{ij} \mathbf{x}_j / p_i$$

$$\Sigma_i \leftarrow \sum_j p_{ij} \mathbf{x}_j \mathbf{x}_j^\top / p_i$$

$$p(x) = \sum_{i=1}^N w_i \mathcal{N}(x, \mu_i, \Sigma_i)$$

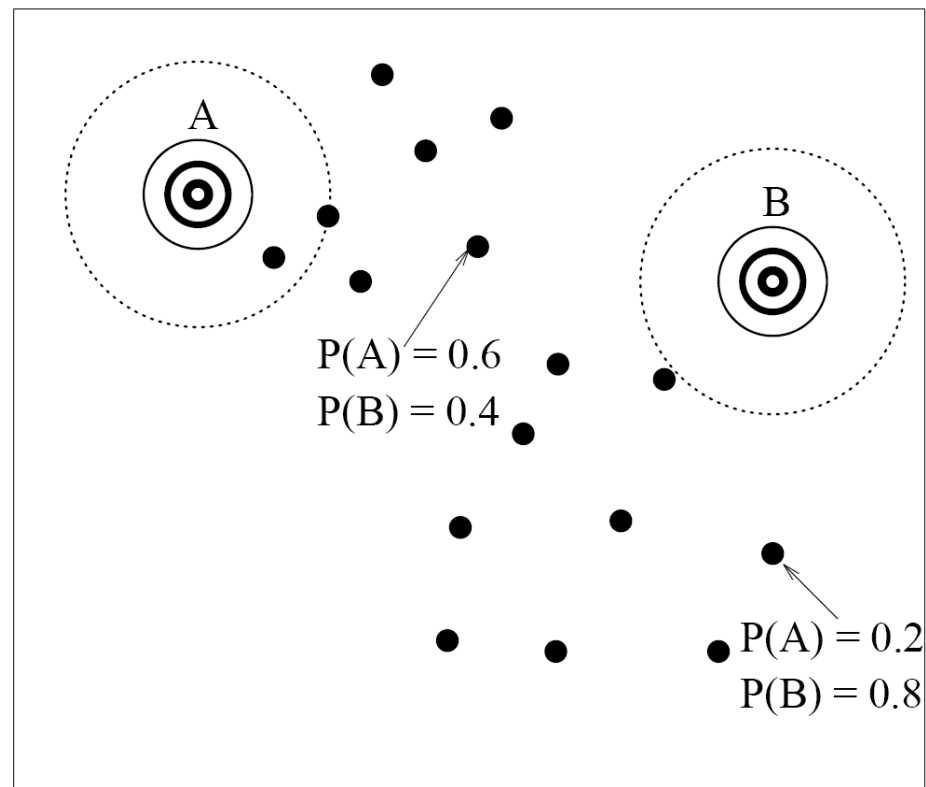
Processing : EM Initialization

- Initialization :
 - Assign random value to parameters



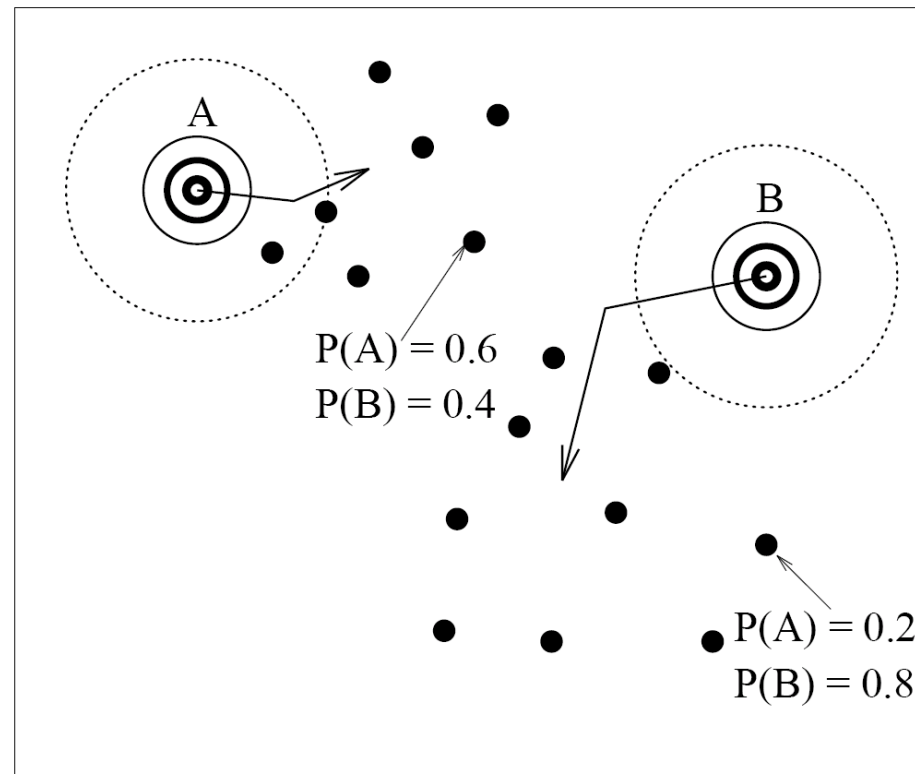
Processing : the E-Step

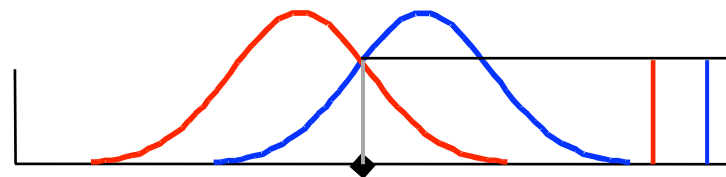
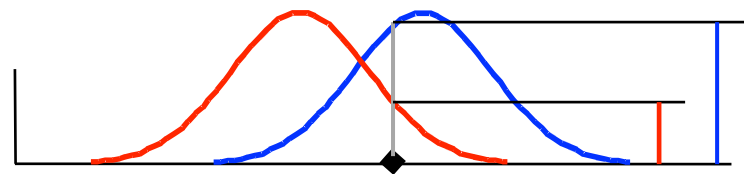
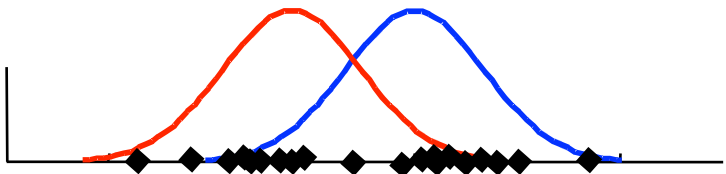
- Expectation :
 - Pretend to know the parameter
 - Assign data point to a component

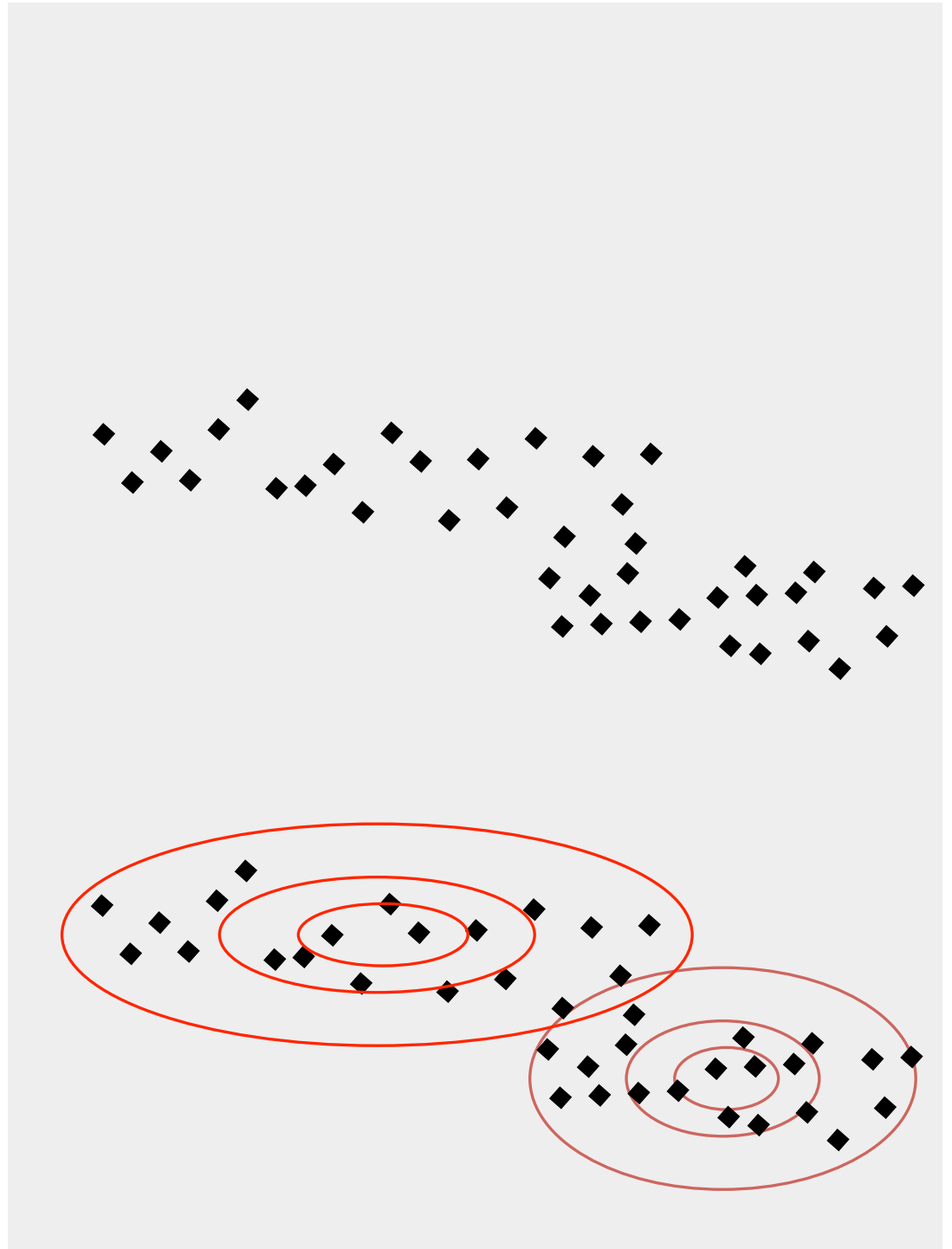
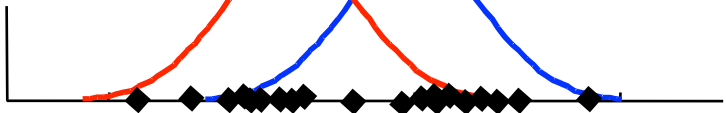


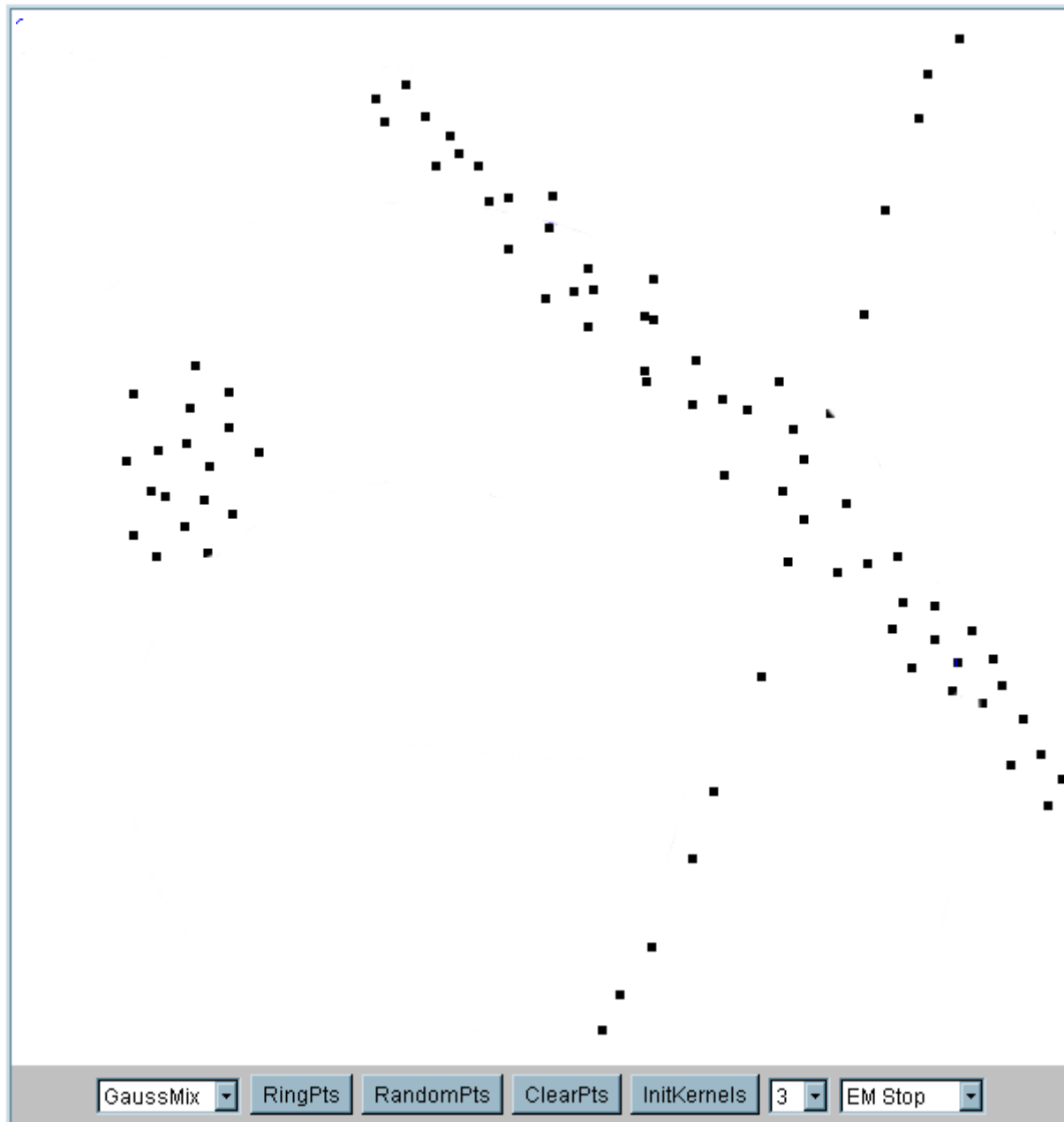
Processing : the M-Step (1/2)

- Maximization :
 - Fit the parameter to its set of points



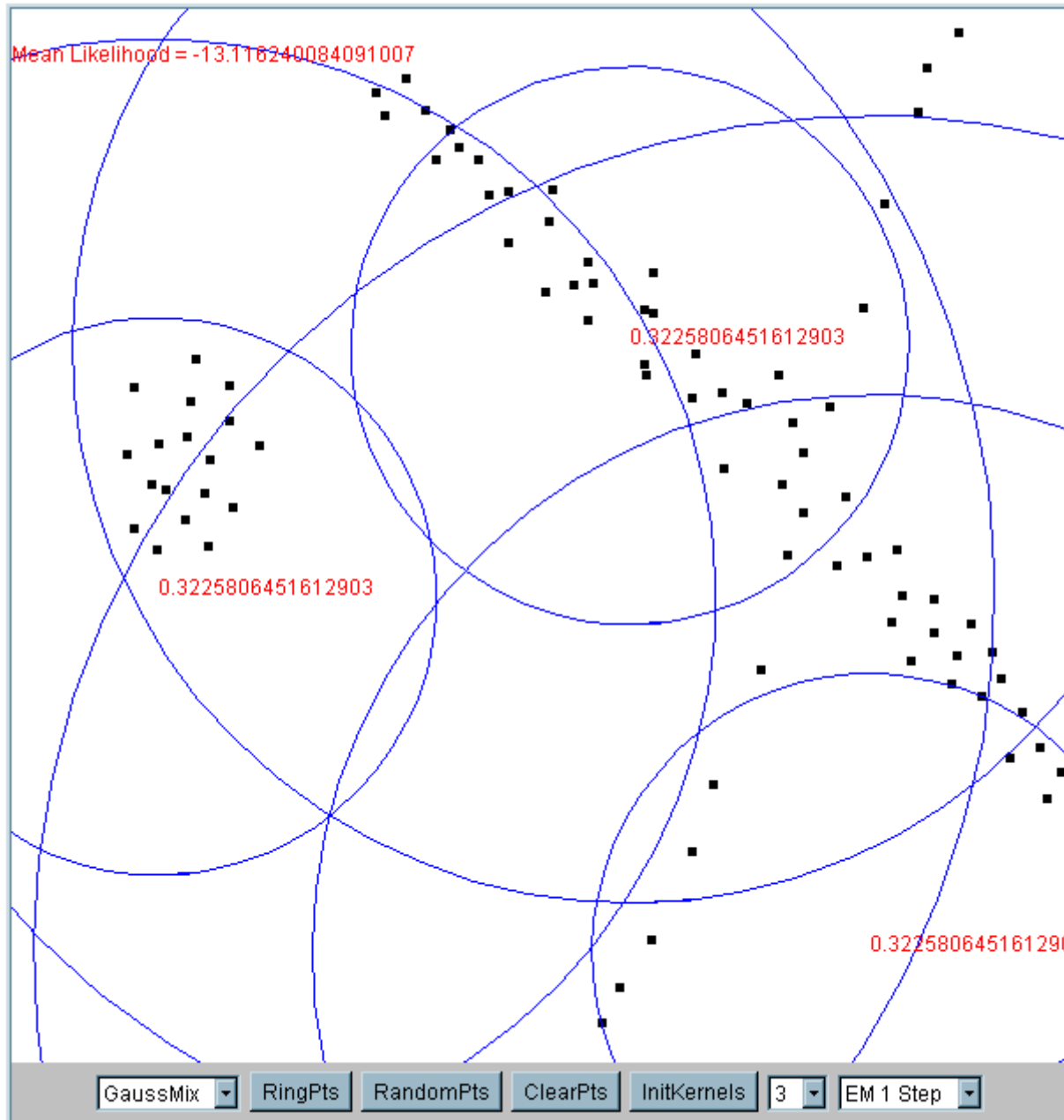




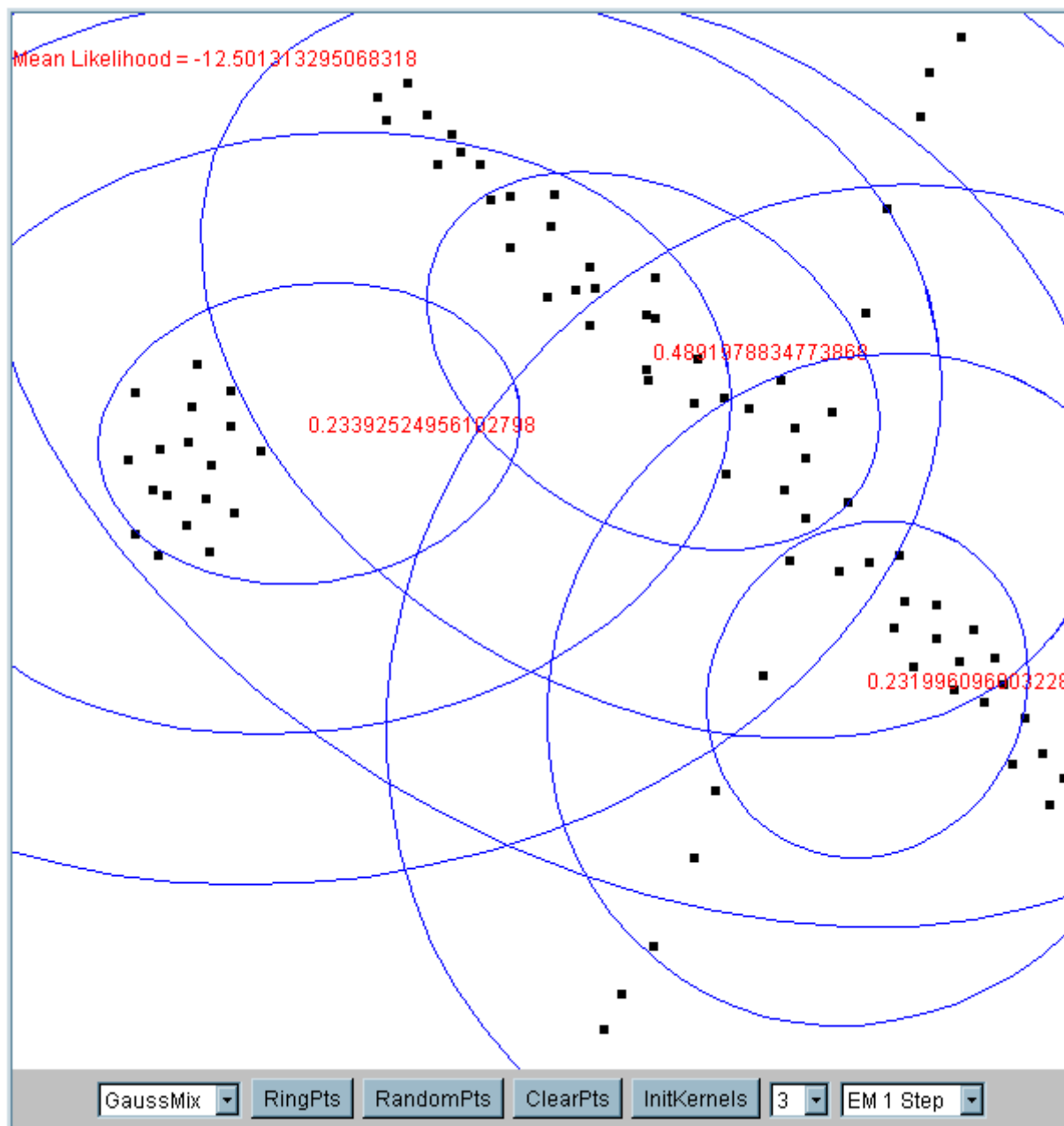


Iteration 1

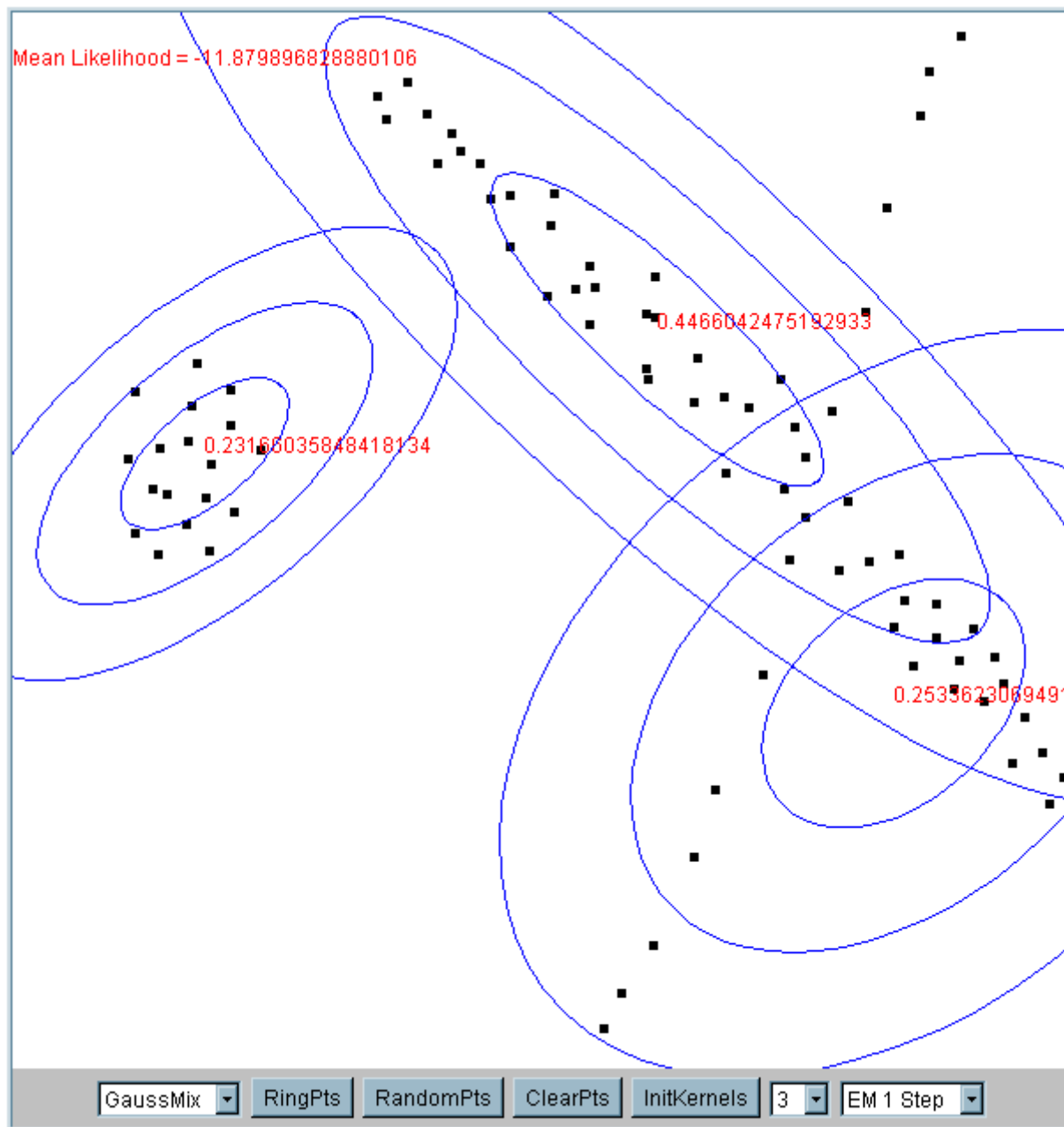
The cluster means are randomly assigned



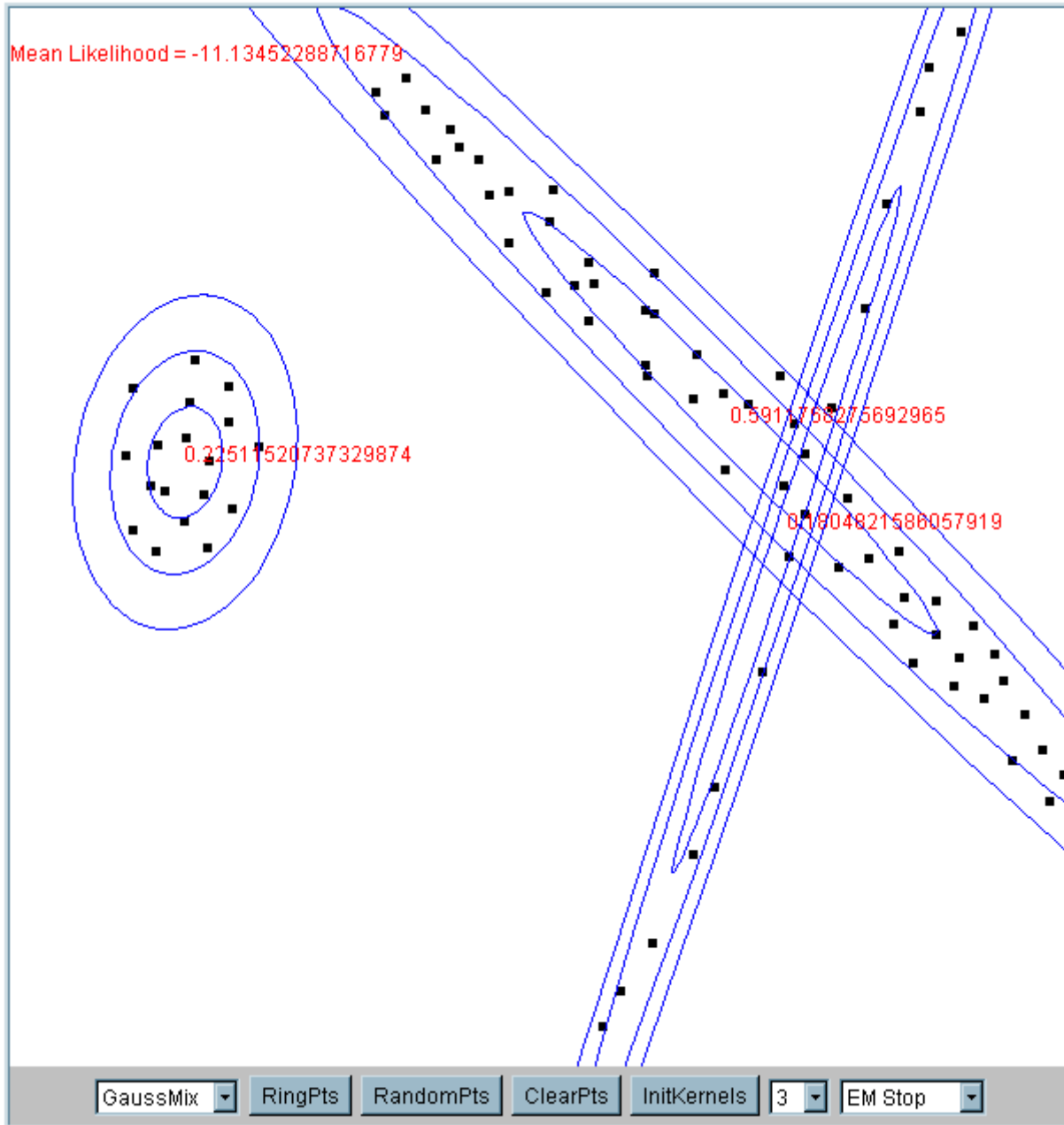
Iteration 2



Iteration 5

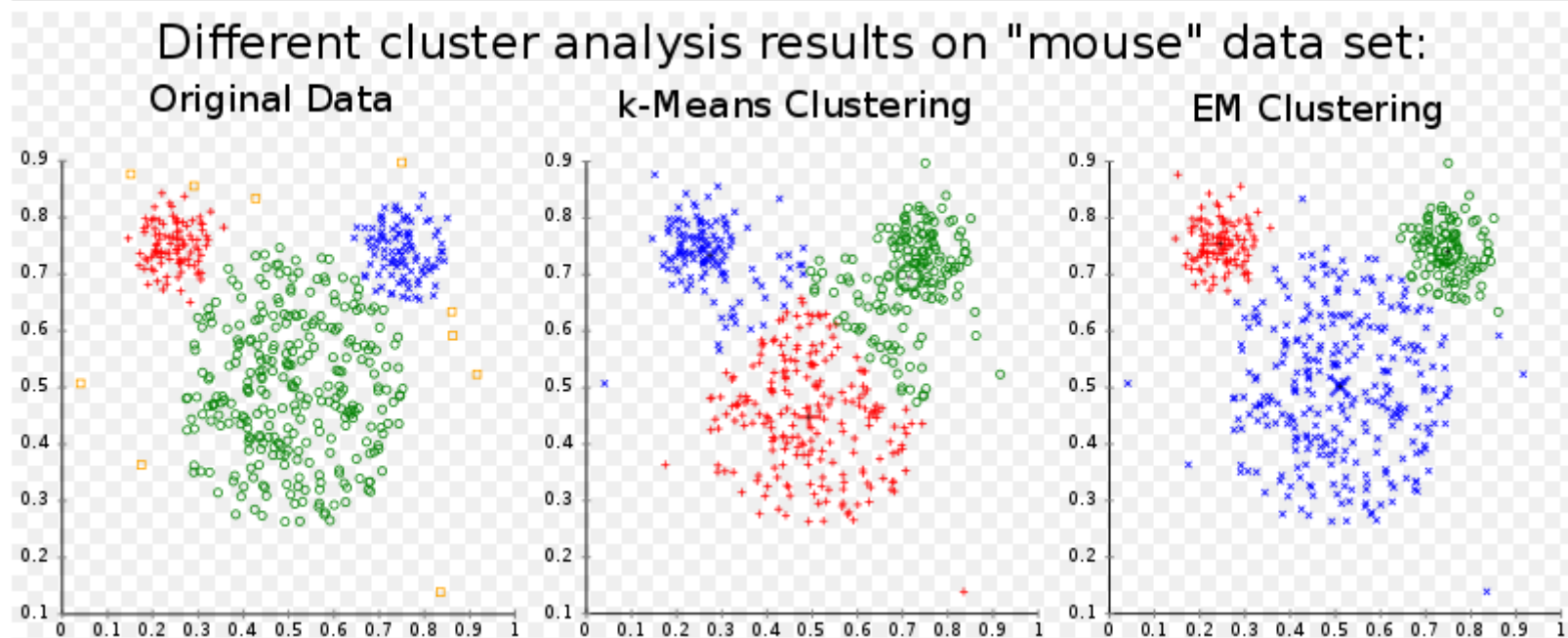


Iteration 25



Comments on the *EM*

- K-Means is a special form of EM
- EM algorithm maintains probabilistic assignments to clusters, instead of deterministic assignments, and multivariate Gaussian distributions instead of means
- Does not tends to build clusters of equal size



Source: http://en.wikipedia.org/wiki/K-means_algorithm

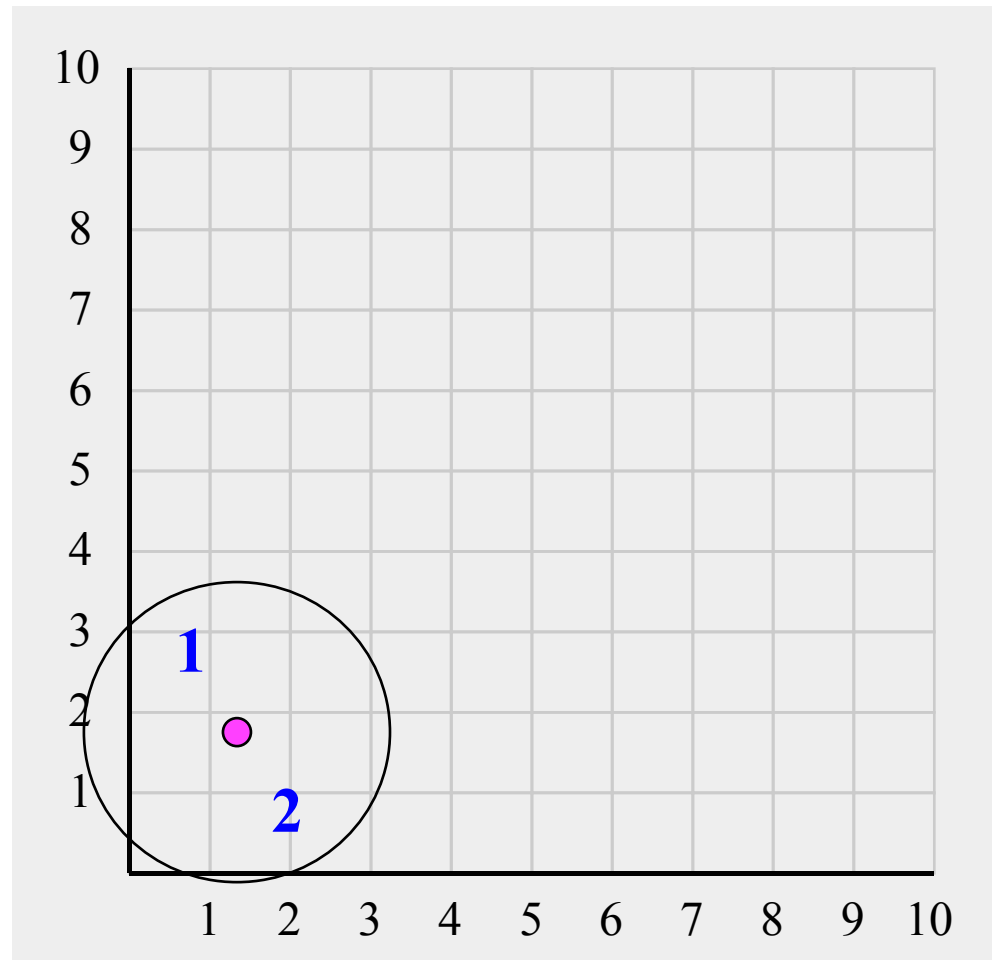
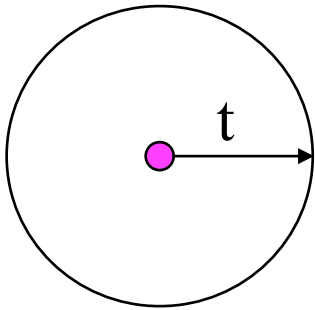
What happens if the data is streaming...

Nearest Neighbor Clustering

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

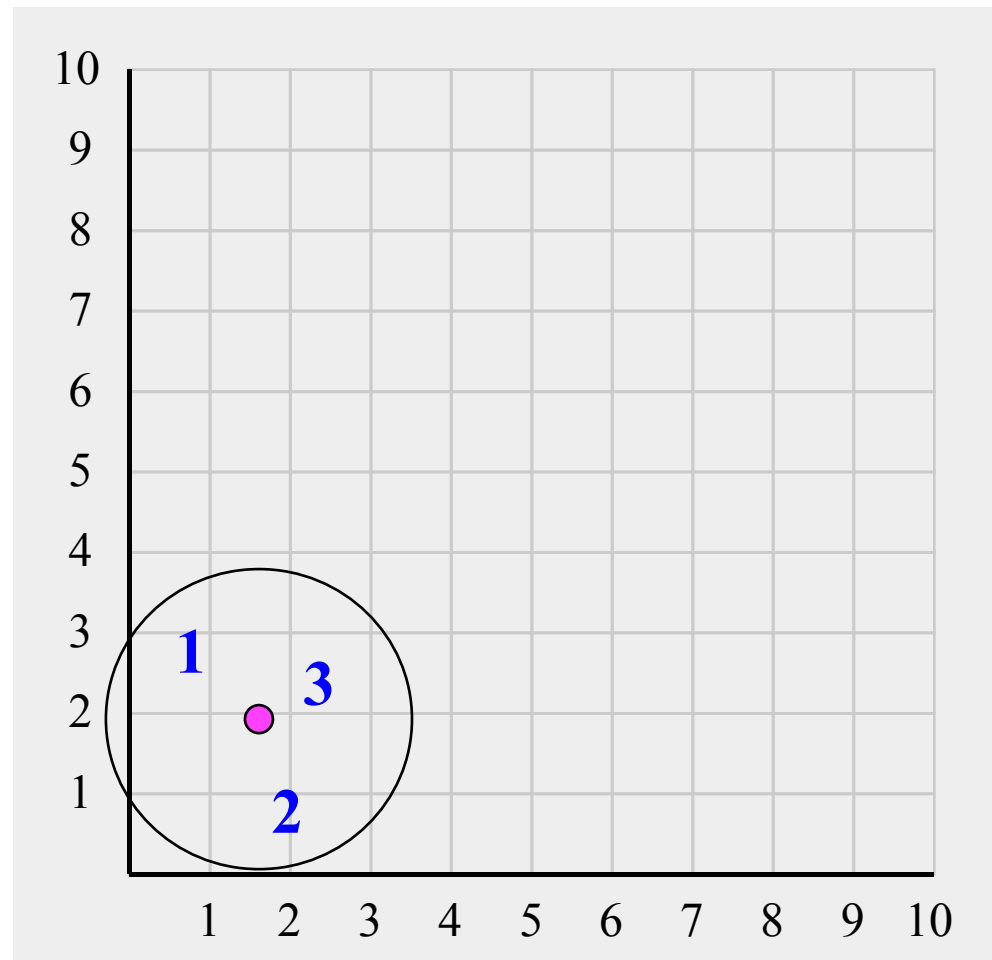
- Items are iteratively merged into the existing clusters that are closest.
- Incremental
- Threshold, t , used to determine if items are added to existing clusters or a new cluster is created.

Threshold t



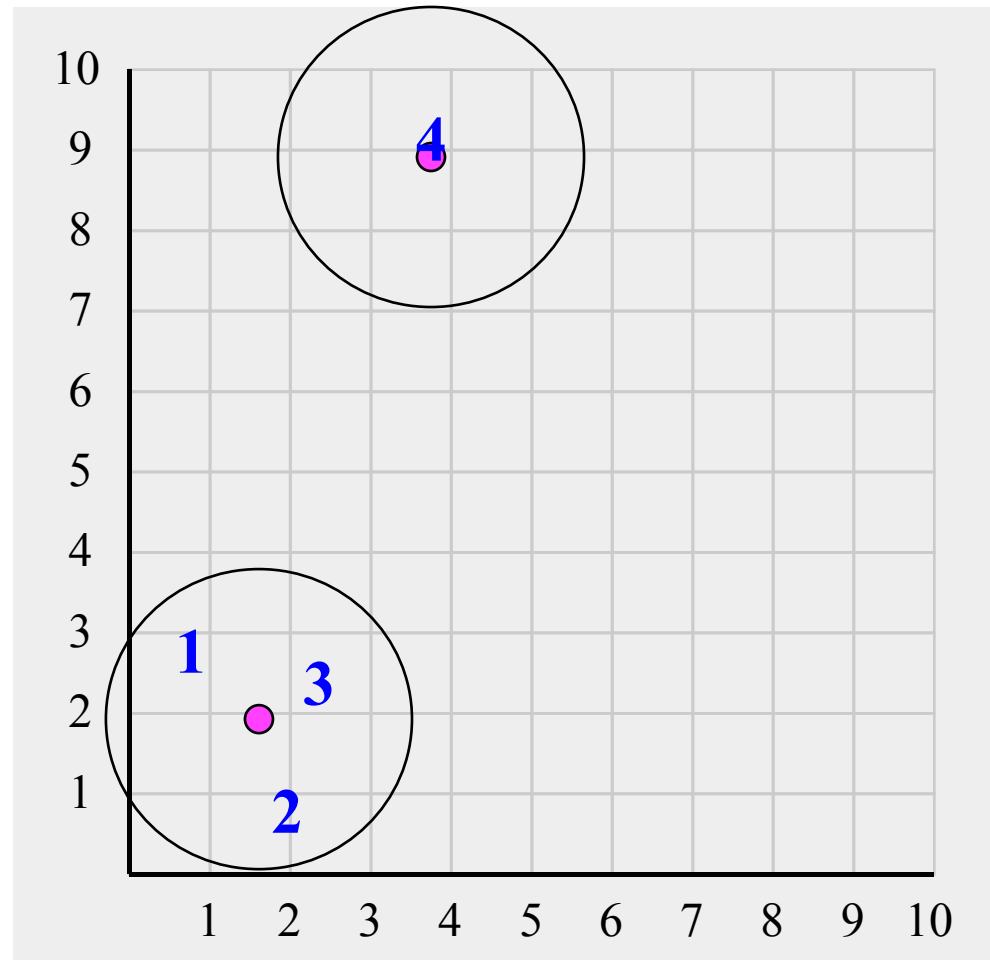
New data point arrives...

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..

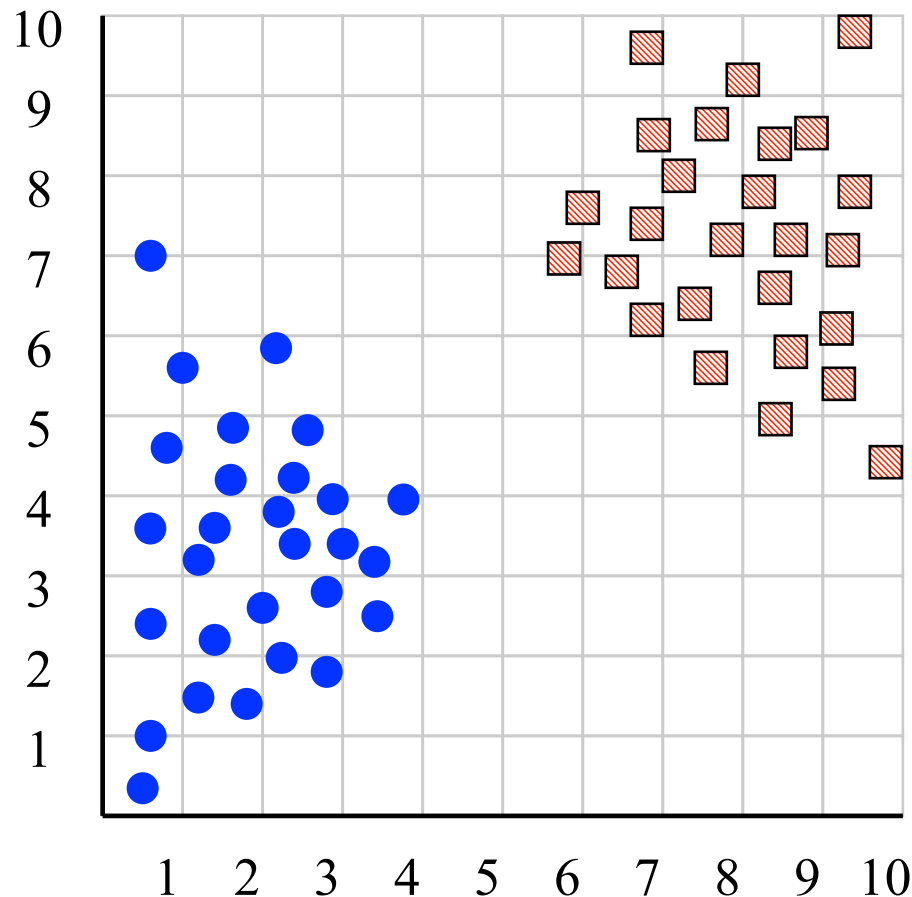


Algorithm is highly order dependent...

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.



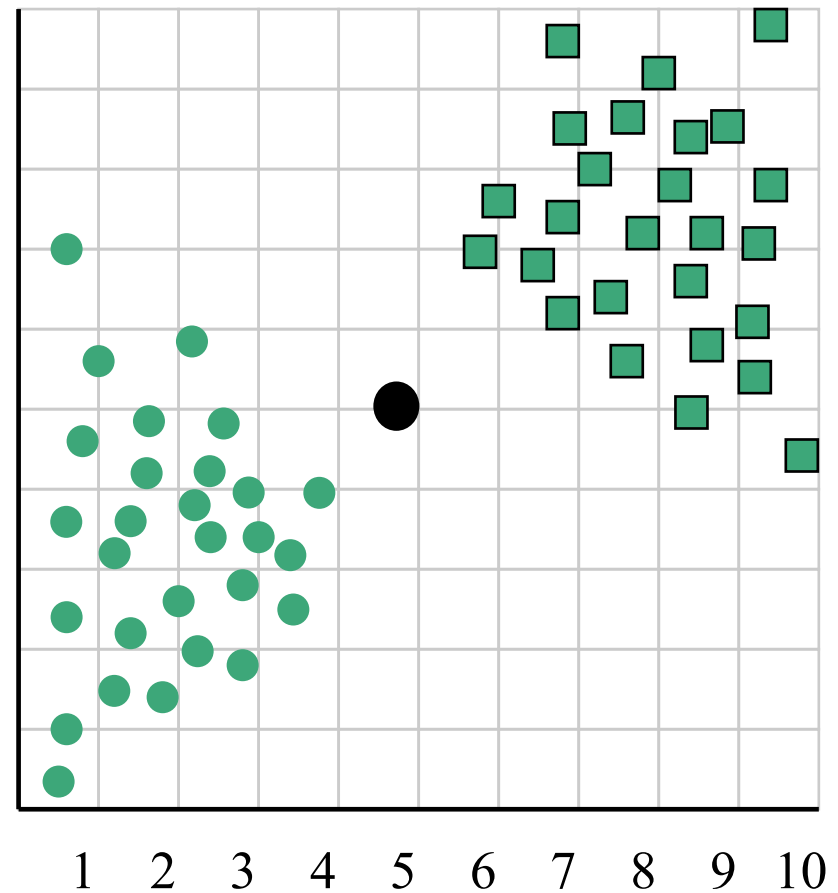
For our example, we will use the familiar **katydid**/**grasshopper** dataset.

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.

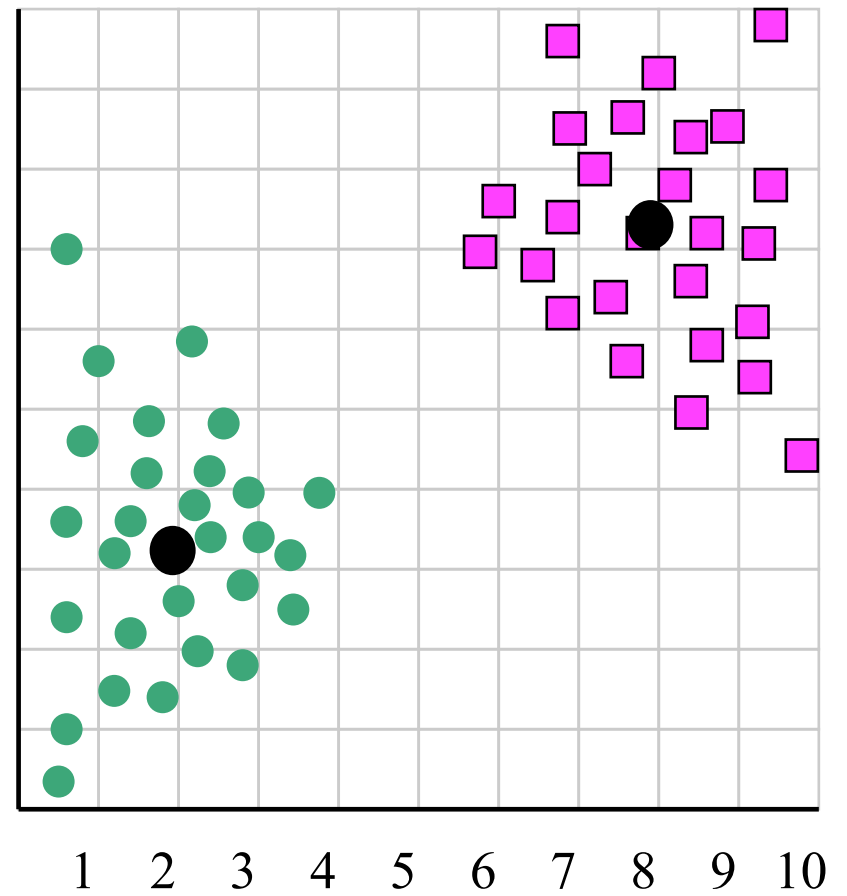
When $k = 1$, the objective function is 873.0

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

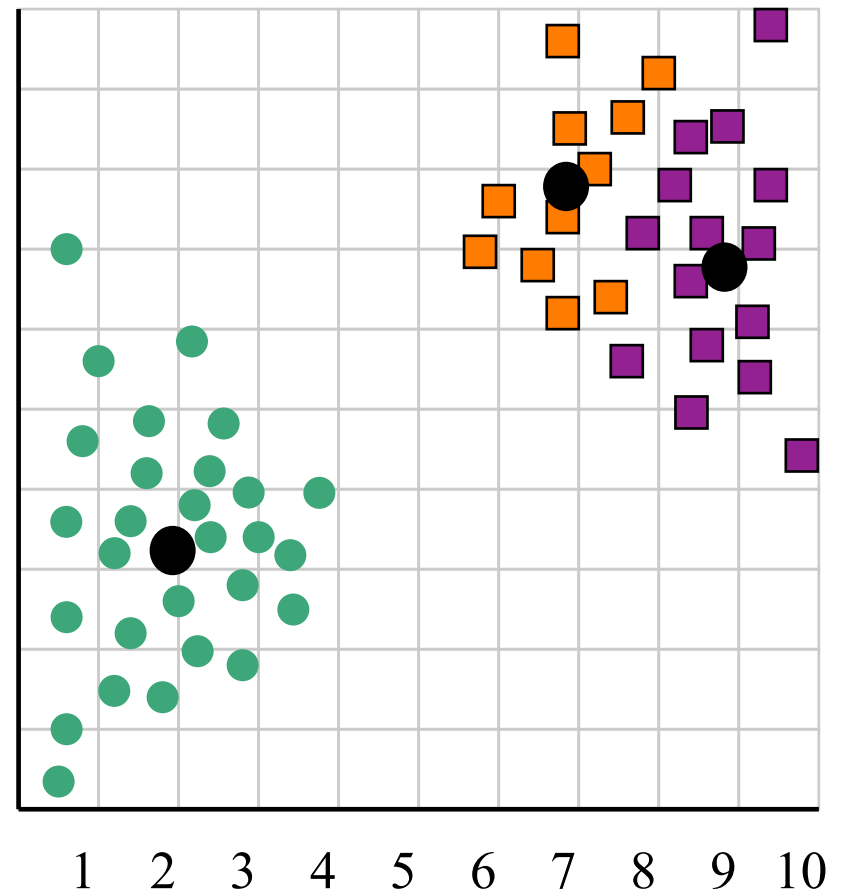
$$se_K = \sum_{j=1}^k se_{K_j}$$



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

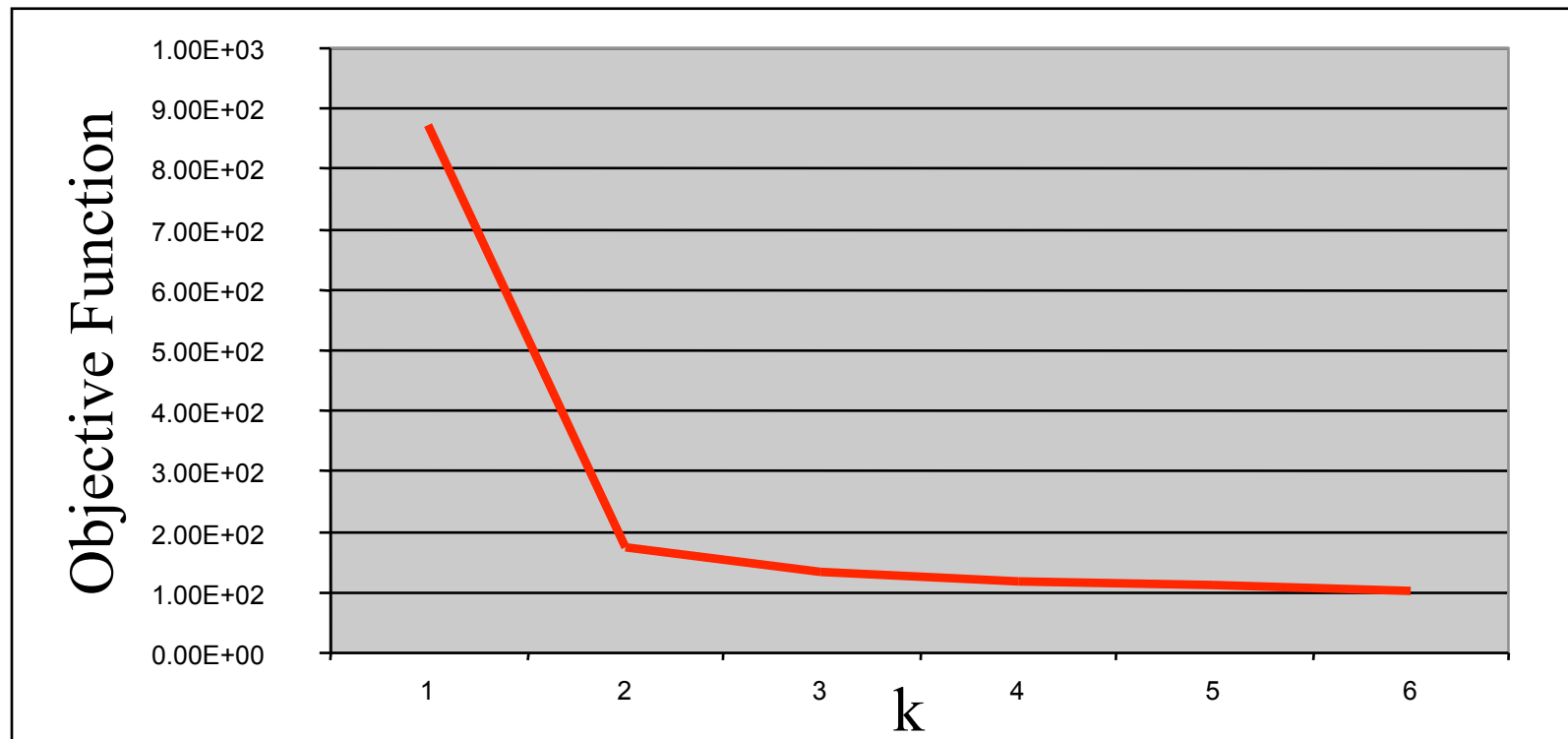


When $k = 3$, the objective function is 133.6



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