INTRODUCTION TO ARTIFICIAL INTELLIGENCE

INTRODUCTION TO **MACHINE LEARNING** & **UNSUPERVISED LEARNING – CLUSTERING**

Machine Learning Part

- Consists of 4 lectures
- We will learn about the most widely used ML algorithms for:
  - Unsupervised learning
  - Supervised learning
  - Reinforcement learning

- The ML part is not assessed in the coursework, but it will be assessed in the exam!

Teaching team

- Lecturer: Lela (Theodora)Koulouri
  - lela.koulouri@kcl.ac.uk
  - Office hours: **Monday**, **1-3pm**
    - Room: **S1.12**
  - https://lela.youcanbook.me/

- Email me for any personal/individual issue…

The Discussion Board on KEATS

- For any question, use the **Discussion Board** first.
- Give opportunity for peers to think about, and to provide a answer
- Everyone can read and benefit from the discussion
- We always respond within 48 hours

Teaching Assistants

- Small group tutorials in weeks 24, 26, 28 (and 31)

Wiktor Piotrowski

Okkes (Emre) Savas

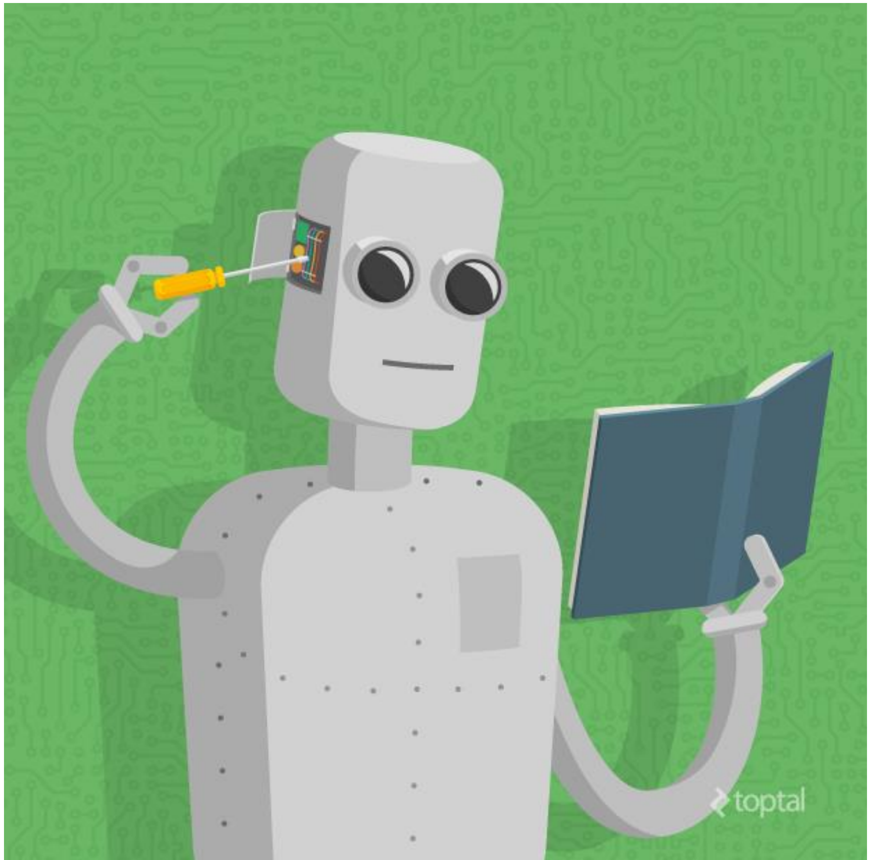
Stefan Sarkadi

Lecture outline

- What is Machine Learning (ML)
  - Why we need it, where we use it
- Distinguish between different types of ML:
  - Unsupervised Learning
  - Supervised Learning
  - Reinforcement Learning
- Unsupervised learning: Clustering
  - Similarity and distance metrics
  - K-means clustering
  - Hierarchical clustering
  - Strengths and Weaknesses of each algorithm
- Unsupervised learning: Association Rules
- Material covered over 2 weeks

Machine learning

- What is machine learning?



http://www.toptal.com

Machine learning

- "The field of study that gives computers the ability to **learn without** being explicitly **programmed**."

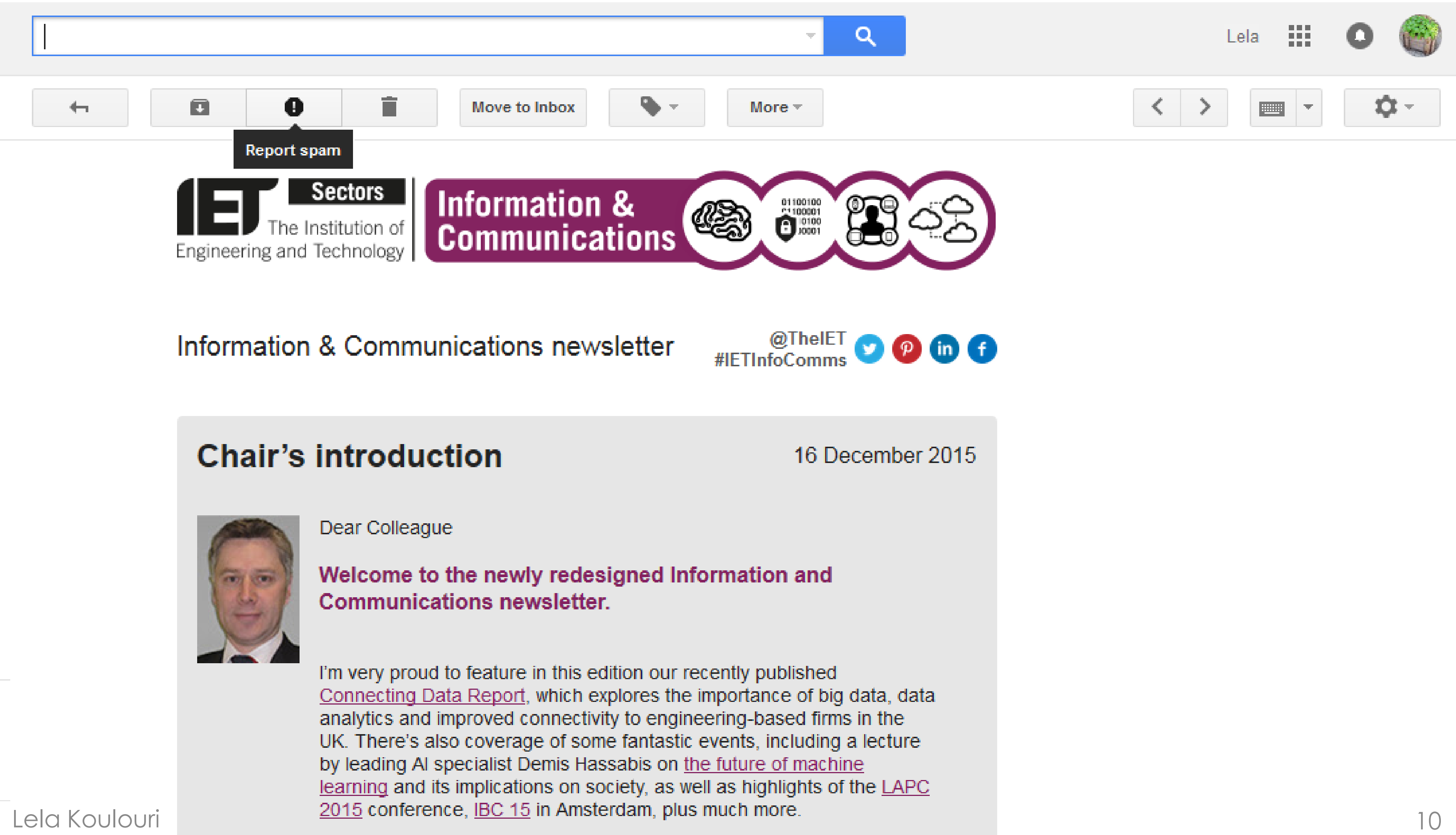
Arthur Samuel, 1959

Machine learning

"A computer program is said to **learn** from ***experience E*** with respect to some ***task T*** and some ***performance measure P***, if its performance on ***T***, as measured by ***P***, improves with experience ***E***."

Tom Mitchell, 1998

Machine learning



Quick quiz

"A computer program is said to **learn** from **experience *E*** with respect to some **task *T*** and some **performance measure *P***, if its performance on ***T***, as measured by ***P***, improves with experience ***E***."

Your email software watches which email you mark as 'spam' or not. Based on that, it learns how to better filter spam. What is ***T***, ***E***, and ***P*** in this scenario?

1. Classifying emails as spam or not spam.
2. Watching you label emails as spam or not spam.
3. The number (or ratio) of emails correctly classified as spam/not spam.
4. None of the above - this is not a machine learning problem.

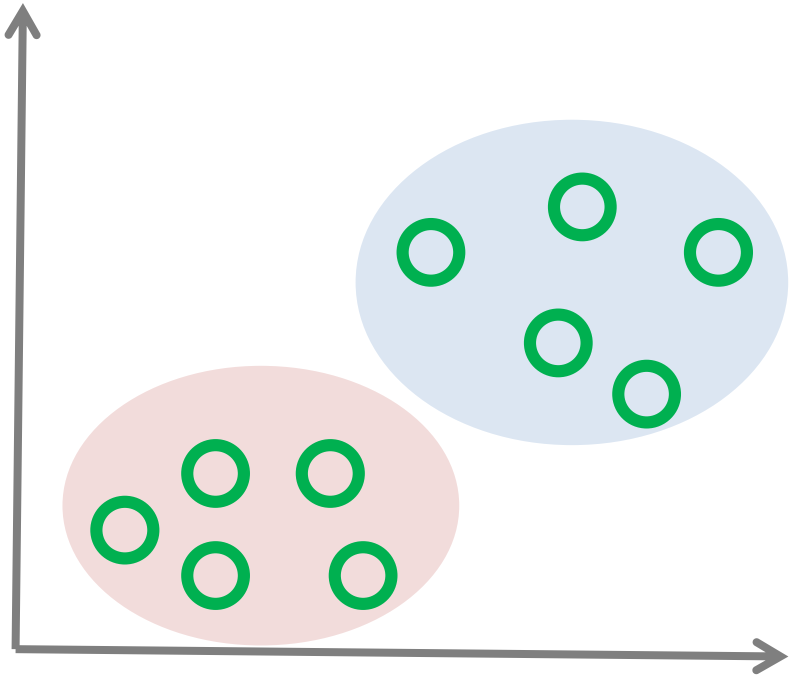
Some other machine learning problems

- Is this cancer?
- What is the market value of this house?
- Which of these people are good friends with each other?
- What will this customer buy?
- Will this person like this movie?
- Who is this person?
- What did you say?
- Will this rocket engine explode?
- How do you fly this thing?

Machine learning types

- **Supervised learning:** The program is 'trained' on a given set of examples. It learns how to reach an accurate conclusion when given new data.
  - **We teach** the computer how to do something.
- **Unsupervised learning:** The program is given a bunch of data and must discover patterns and relationships in them.
  - We let the computer **learn** something **by itself.**
- **Reinforcement learning:** The program learns from the consequences of its actions (reward or punishment), rather than from being explicitly taught and it selects its actions on basis of its past experiences (exploitation) and also by new choices (exploration).

Unsupervised learning



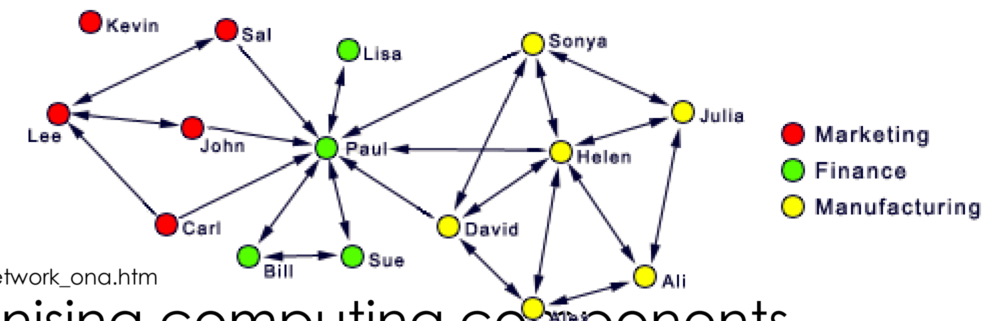
No labels
We have to find some structure in the data set
An unsupervised algorithm may decide that the data belongs in two clusters (groups)
This algorithm is referred to as **Clustering** algorithm

Unsupervised learning: Clustering

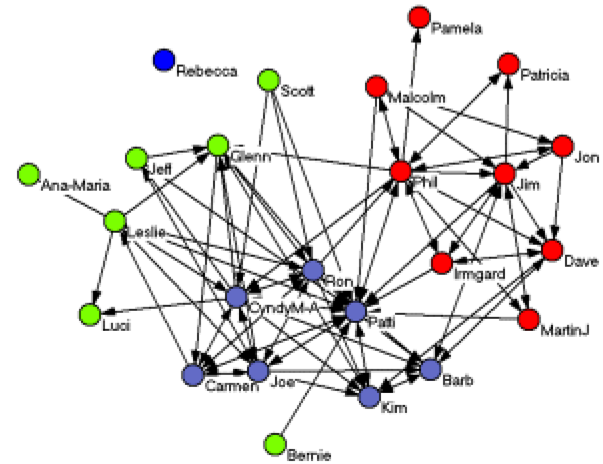
- Classic form of unsupervised Learning
- *'What data goes with what?'*
- Method for grouping items of a similar 'type'
- Applications include:
  - Customers who make similar purchases in Amazon
  - Categorising web pages
  - Grouping genes that work together to perform a specific biological function
  - Grouping code (OO Classes) that work together to perform a similar function

Unsupervised learning: other applications

- Social network analysis

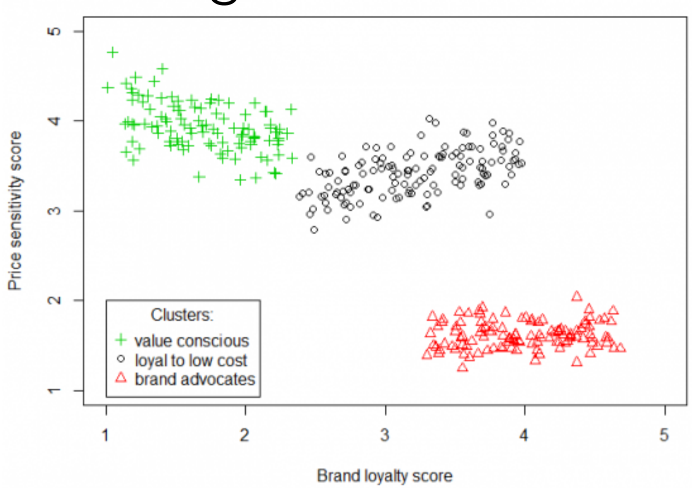


http://www.robcross.org/network_ona.htm

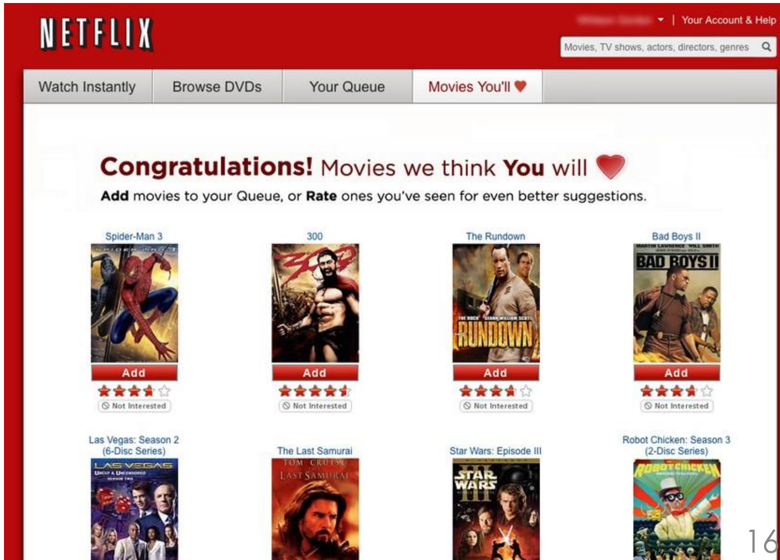


http://www.byeday.net/sna/

- Organising computing components
- Recommender systems
- Market segmentation

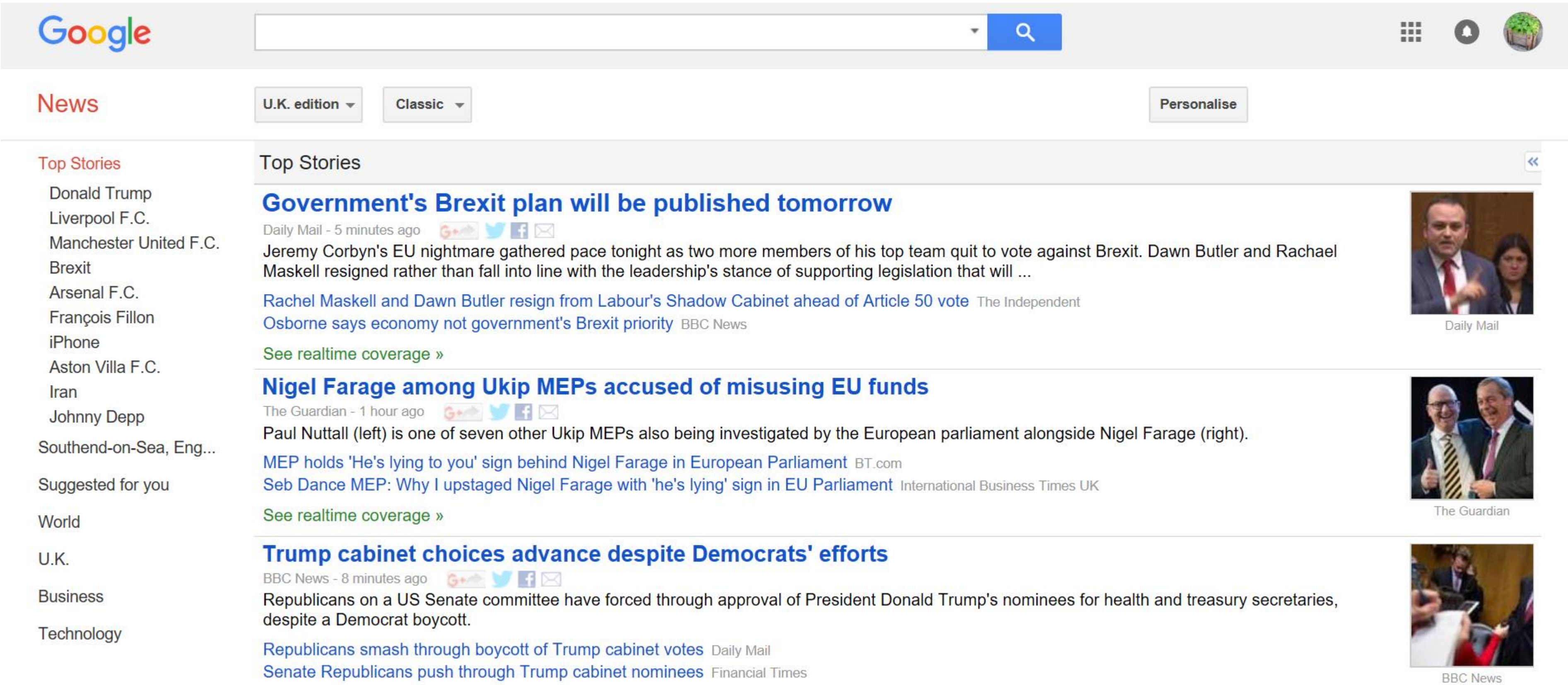


http://www.select-statistics.co.uk/article/blog-post/customer-segmentation

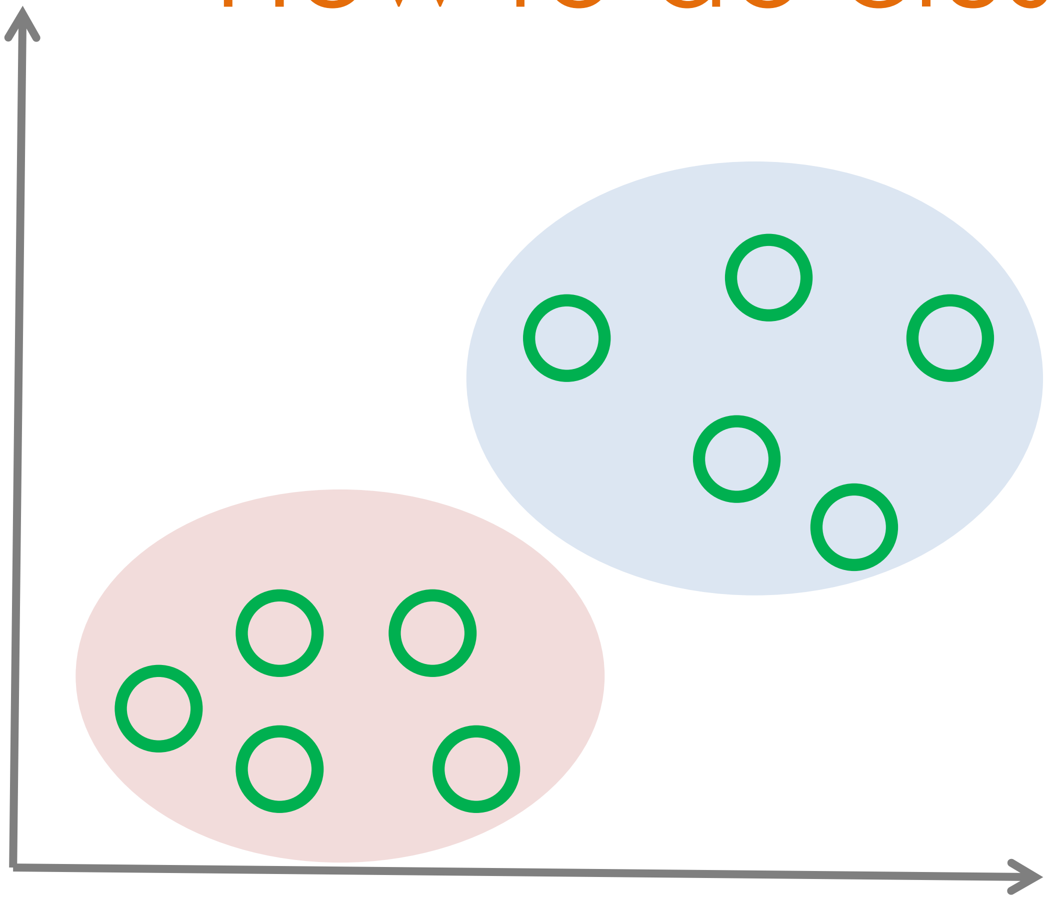


Unsupervised learning: example

- https://news.google.co.uk/



How to do clustering?



How to do clustering? **Similarity and Distance**

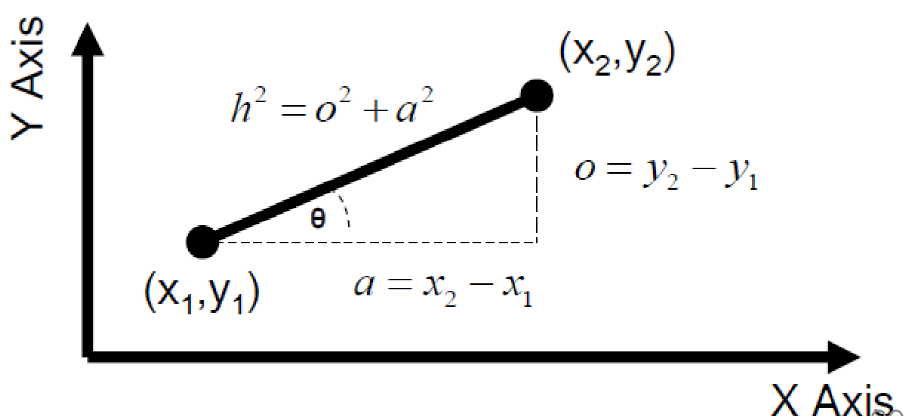
- Clusters are formed by ***similar*** patterns
- How do we define ***similar***?
- One of the commonly adopted similarity metric is **distance**
- A general definition of **distance** (between A and B):

Euclidean distance: b = 2
Manhattan distance: b = 1

$$d = \left[\sum_{i=1}^{d} \left| x_i^A - x_i^B \right|^b \right]^{\frac{1}{b}}$$

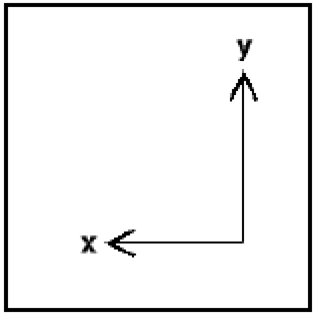
Euclidean distance

- The **shortest** distance between two points
- In the two dimensional case, this is the length of the hypotenuse of the right angled triangle constructed between two points (p_1 and p_2) (Pythagoras' Theorem)
- In a plane with p_1 at (x_1, y_1) and p_2 at (x_2, y_2), it is $\sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)}$

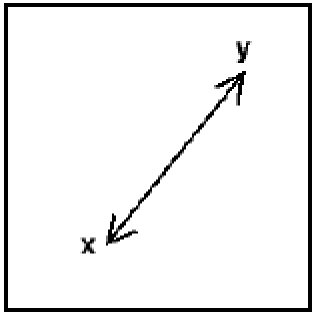


Manhattan distance

- The distance between two points (p_1 and p_2) measured along axes at right angles.
- In a plane with p_1 at (x_1, y_1) and p_2 at (x_2, y_2), it is $|x_1 - x_2| + |y_1 - y_2|$



Manhattan



Euclidean

Quick Quiz

- For $\mathbf{p_1}$ at (x_1, y_1) and $\mathbf{p_2}$ at (x_2, y_2)
- The **Manhattan** distance between p_1 and p_2 is:

$|\mathbf{x_1} - \mathbf{x_2}| + |\mathbf{y_1} - \mathbf{y_2}|$

- The **Euclidean** distance between p_1 and p_2 is:

$\sqrt{((\mathbf{x_1} - \mathbf{x_2})^2 + (\mathbf{y_1} - \mathbf{y_2})^2)}$

- Calculate the distances of point A1(2, 10) from point A4(5, 8) and point A7(1, 2)

		A4(5, 8)	A7(1, 2)
Manhattan	**A1(2, 10)**		
Euclidean	**A1(2, 10)**		

Quick Quiz

- For $\mathbf{p_1}$ at (x_1, y_1) and $\mathbf{p_2}$ at (x_2, y_2)
- The **Manhattan** distance between p_1 and p_2 is:

$|\mathbf{x_1 - x_2}| + |\mathbf{y_1 - y_2}|$

- The **Euclidean** distance between p_1 and p_2 is:

$\sqrt{(\mathbf{(x_1 - x_2)^2 + (y_1 - y_2)^2})}$

- Calculate the distances of point A1(2, 10) from point A4(5, 8) and point A7(1, 2)

		A4(5, 8)	**A7(1, 2)**
Manhattan	**A1(2, 10)**	$\|2 - 5\| + \|10 - 8\| = 3 + 2 = \mathbf{5}$	$\|2 - 1\| + \|10 - 2\| = 1 + 8 = \mathbf{9}$
Euclidean	**A1(2, 10)**	$\sqrt{((2 - 5)^2 + (10 - 8)^2)} = \sqrt{(3^2 + 2^2)} = \sqrt{13} \approx \mathbf{3.605}$	$\sqrt{((2 - 1)^2 + (10 - 2)^2)} = \sqrt{(1^2 + 8^2)} = \sqrt{65} \approx \mathbf{8.062}$

Quick Quiz

- Calculate the **Mean** of these 3 points:
- point A1(2, 10), point A4(5, 8) and point A7(1, 2).

- For x = (2+5+1)/3 ≈ 2.67
- For y = (10+8+2)/3 ≈ 6.67

The **mean** is represented by the point (2.67, 6.67).

If A1, A4 and A7 were a group of points, we would say that this point is the **centre/mean** of the group (aka **centroid**!).

Clustering algorithms

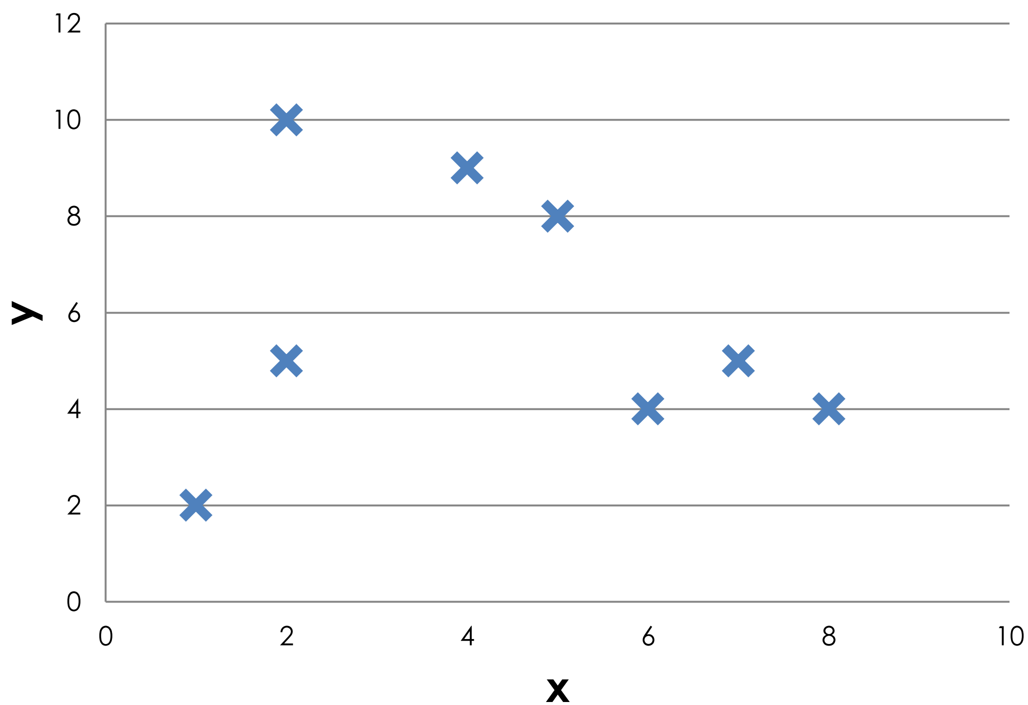
- 'NP–Hard Problem', the number of possible ways to cluster items increases exponentially
- So we need efficient ways to find good clusters using the combination of a metric and a heuristic search
- There are hundreds of approaches!
- But some are more famous. We will look at two of the most popular
  - **K-means clustering** *(today)*
  - **Hierarchical clustering** *(today or next time)*

K-Means algorithm

0. **Select** *K* points as initial **cluster means** (they are called **cluster centroids**)
1. **Assign** each point to the cluster with the *closest* centroid.
2. When all points have been assigned, **recalculate** the positions of the *K* centroids.
3. **Repeat** Steps 1 and 2 until the centroids no longer change.

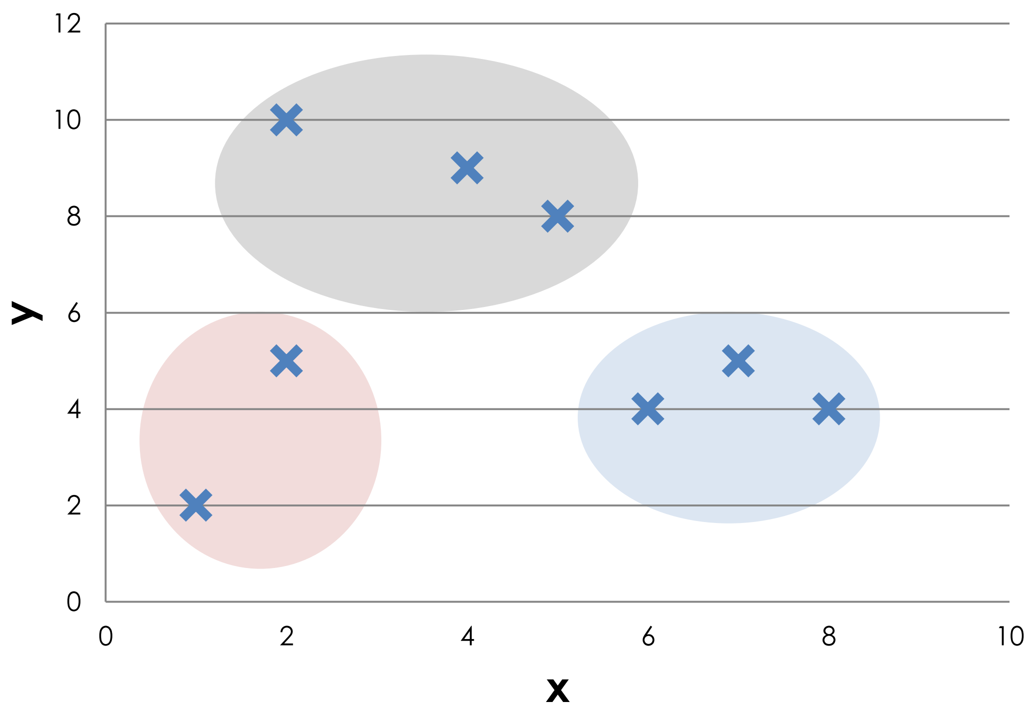
K-Means clustering example

- Cluster the following eight points (with (x, y) representing locations) into **three clusters (*K* = 3):**
- A1(2, 10) A2(2, 5) A3(8, 4) A4(5, 8) A5(7, 5) A6(6, 4) A7(1, 2) A8(4, 9).



K-Means clustering example: desired output

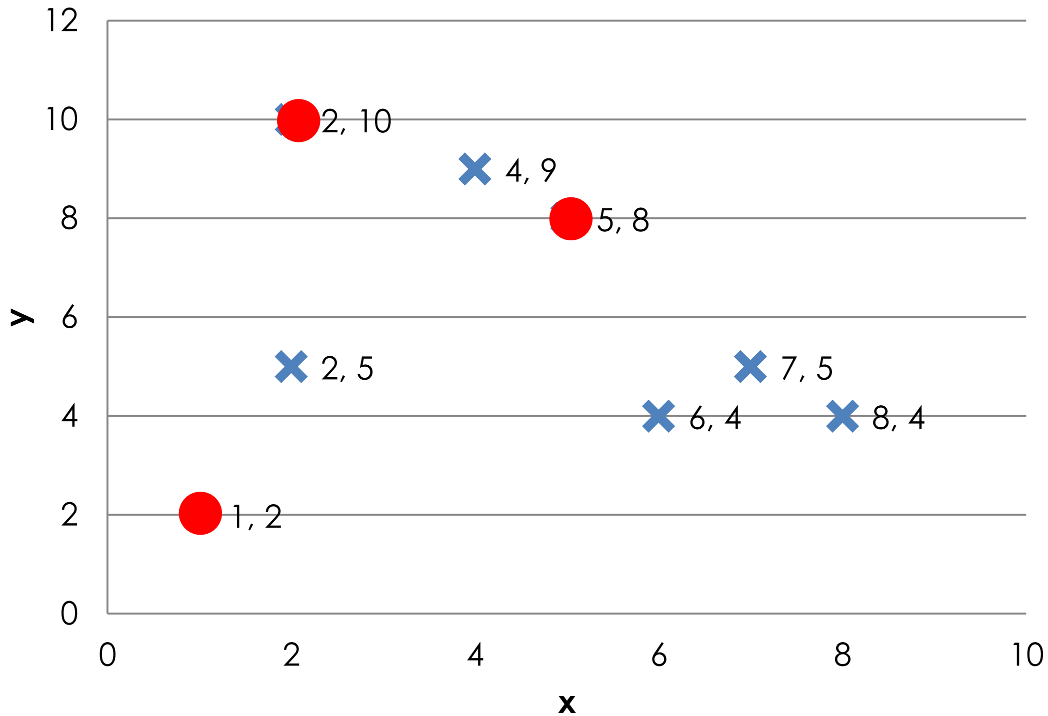
- Cluster the following eight points (with (x, y) representing locations) into **three clusters:**
- A1(2, 10) A2(2, 5) A3(8, 4) A4(5, 8) A5(7, 5) A6(6, 4) A7(1, 2) A8(4, 9).



K-Means algorithm: step 0

- *Randomly* select the initial cluster centroids.

A1(2, 10), A4(5, 8) and A7(1, 2).



Iteration 1 – step 1

Step 1: Assign each point to the group with the closest centroid:

– 'Closest'? So calculate the **distances** of each point from these centroids (Means)

In this example, I will use Manhattan distance:
$|x_1 - x_2| + |y_1 - y_2|$

		(2, 10)	(5, 8)	(1, 2)
	Point	**Mean 1**	**Mean 2**	**Mean 3**
A1	(2, 10)	0	5	9
A2	(2, 5)			
A3	(8, 4)			
A4	(5, 8)			
A5	(7, 5)			
A6	(6, 4)			
A7	(1, 2)			
A8	(4, 9)			

Iteration 1 – step 1

Step 1: Assign each point to group with the closest centroid:

– 'Closest'? So calculate the **distances** of each point from these centroids (Means)

		(2, 10)	(5, 8)	(1, 2)
	Point	**Mean 1**	**Mean 2**	**Mean 3**
A1	(2, 10)	0	5	9
A2	(2, 5)	5	6	4
A3	(8, 4)	12	7	9
A4	(5, 8)	5	0	10
A5	(7, 5)	10	5	9
A6	(6, 4)	10	5	7
A7	(1, 2)	9	10	0
A8	(4, 9)	3	2	10

Iteration 1 –

which cluster should the point **A1 (2, 10)** be placed in? The one where the point has the shortest distance to the mean – that is mean 1 (cluster 1), since the distance is 0.

Step 1: Assign each point to the group that has the closest centroid:

- Calculate the **distances** of each point from these centroids (Means)
- Identify the centroid to which the point has the shortest distance.
- Place the point in that centroid's cluster

		(2, 10)	(5, 8)	(1, 2)	
	Point	**Mean 1**	**Mean 2**	**Mean 3**	**Cluster**
A1	(2, 10)	0	5	9	?
A2	(2, 5)	5	6	4	
A3	(8, 4)	12	7	9	
A4	(5, 8)	5	0	10	
A5	(7, 5)	10	5	9	
A6	(6, 4)	10	5	7	
A7	(1, 2)	9	10	0	
A8	(4, 9)	3	2	10	

Iteration 1 –

which cluster should the point **A2 (2, 5)** be placed in? The one, where the point has the shortest distance to the mean – that is mean 3 (cluster 3), since the distance is 4.

Step 1: Assign each point to the group that has the closest centroid:

- Calculate the **distances** of each point from these centroids (Means)
- Identify the centroid to which the point has the shortest distance.
- Place the point in that centroid's cluster

		(2, 10)	(5, 8)	(1, 2)	
	Point	**Mean 1**	**Mean 2**	**Mean 3**	**Cluster**
A1	(2, 10)	0	5	9	**1**
A2	(2, 5)	5	6	4	?
A3	(8, 4)	12	7	9	
A4	(5, 8)	5	0	10	
A5	(7, 5)	10	5	9	
A6	(6, 4)	10	5	7	
A7	(1, 2)	9	10	0	
A8	(4, 9)	3	2	10	

Iteration 1 – step 1

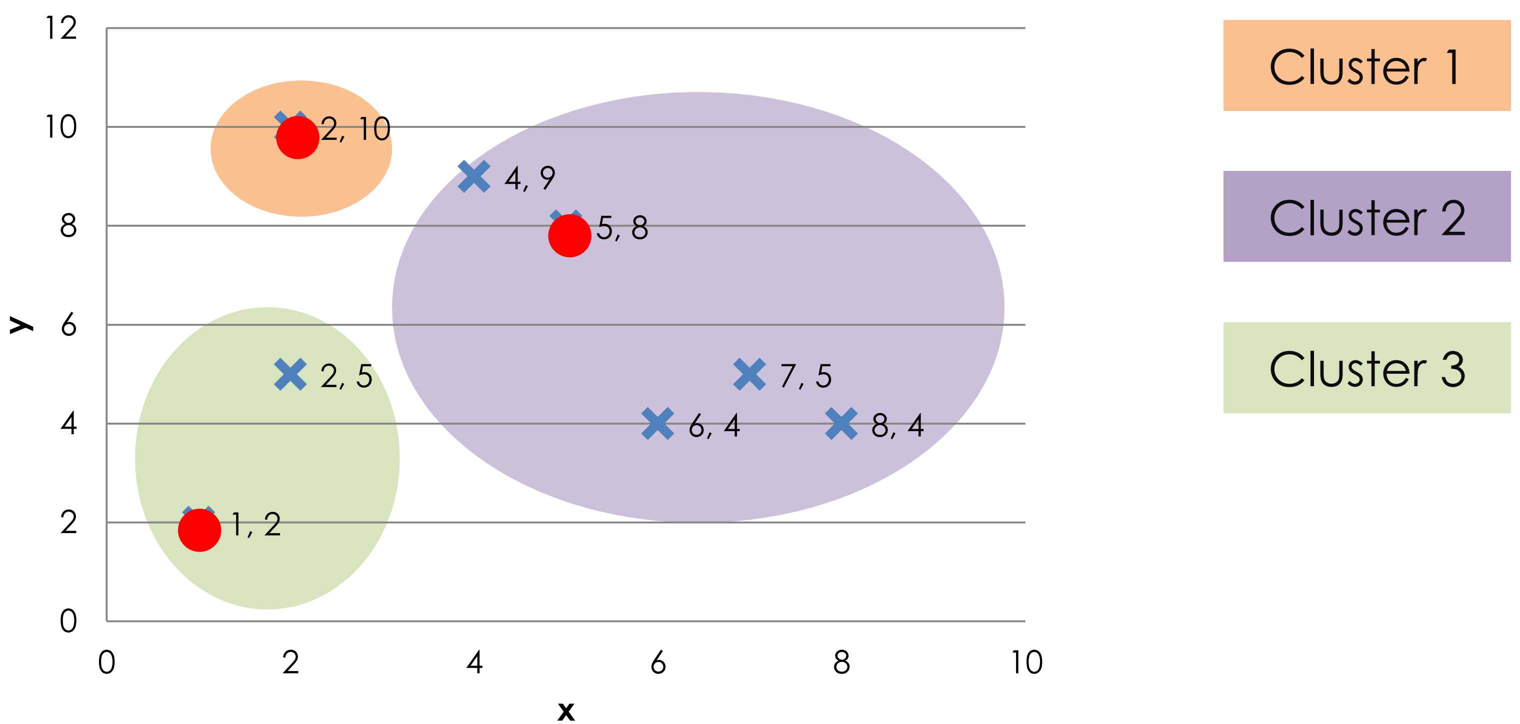
Step 1: Assign each point to the group that has the closest centroid:

- Calculate the **distances** of each point from these centroids (Means)
- Identify the centroid to which the point has the shortest distance.
- Place the point in that centroid's cluster

		(2, 10)	(5, 8)	(1, 2)	
	Point	**Mean 1**	**Mean 2**	**Mean 3**	**Cluster**
A1	(2, 10)	0	5	9	**1**
A2	(2, 5)	5	6	4	**3**
A3	(8, 4)	12	7	9	**2**
A4	(5, 8)	5	0	10	**2**
A5	(7, 5)	10	5	9	**2**
A6	(6, 4)	10	5	7	**2**
A7	(1, 2)	9	10	0	**3**
A8	(4, 9)	3	2	10	**2**

Iteration 1 clusters

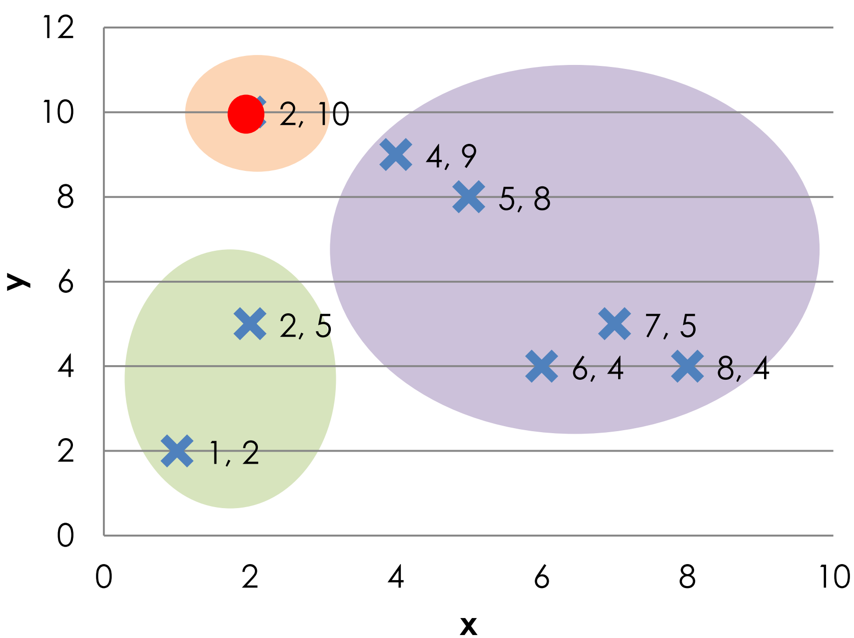
- The first iteration of the algorithm produces these clusters:



Iteration 1 – step 2

Step 2: When all points have been assigned, recalculate the positions of the K centroids.

– Recalculate the **means of the clusters**



Cluster 1
(2, 10)

Cluster 2
(8, 4)
(5, 8)
(7, 5)
(6, 4)
(4, 9)

Cluster 3
(2, 5)
(1, 2)

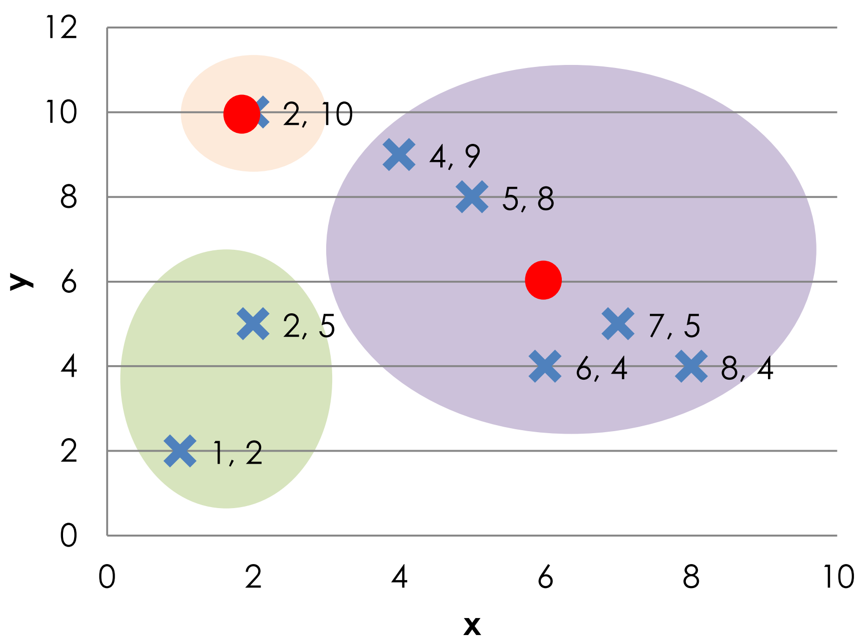
For **Cluster 1**: we only have one point A1(2, 10), which was the **old mean**, so the cluster centroid remains the same

Iteration 1 – step 2

Step 2: When all points have been assigned, recalculate the positions of the K centroids.

– Recalculate the **means of the clusters**

Cluster 1	**Cluster 2**	**Cluster 3**
(2, 10)	(8, 4)	(2, 5)
	(5, 8)	(1, 2)
	(7, 5)	
	(6, 4)	
	(4, 9)	



For **Cluster 2**, we have x= 8,5,7,6,4; y=4,8,5,4,9.

New mean of x = (8+5+7+6+4)/5 =6

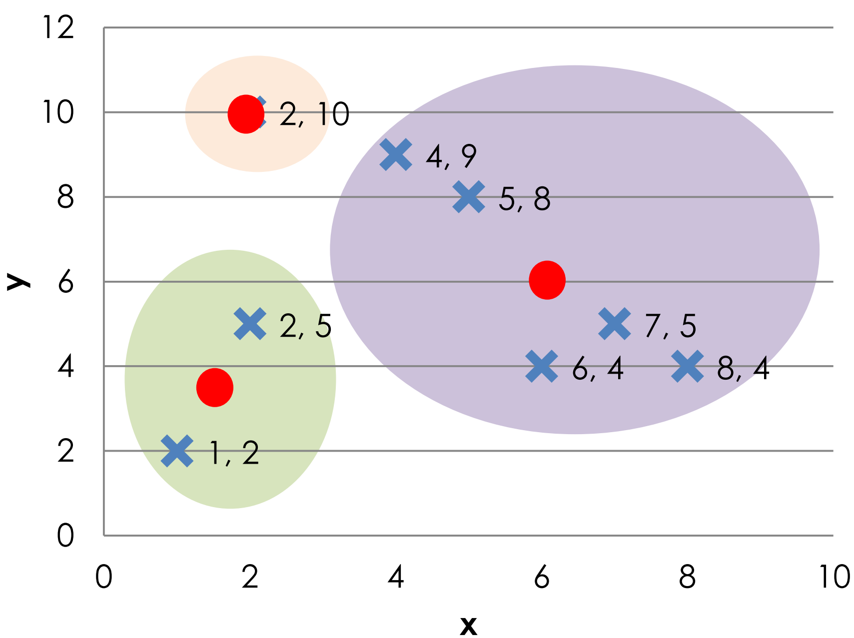
New mean of y= (4+8+5+4+9)/5 =6

Hence our new mean = (6, 6)

Iteration 1 – step 2

Step 2: When all points have been assigned, recalculate the positions of the K centroids.

– Recalculate the **means of the clusters**



Cluster 1
(2, 10)

Cluster 2
(8, 4)
(5, 8)
(7, 5)
(6, 4)
(4, 9)

Cluster 3
(2, 5)
(1, 2)

For **Cluster 3**, we have x= 2,1; y=5,2.
New mean of x is (2+1)/2 = 1.5
New mean of y=(5+2)/2) = 3.5
Hence our mean = (1.5, 3.5)

Iteration 2

- But we don’t stop here, we **repeat** steps 1 and 2…

Iteration 2 – step 1

Step 1: Assign each point to the group with the closest centroid:

– Calculate the **distances** of each point from these centroids (means)

These are the new means!

		(2, 10)	(6, 6)	(1.5, 3.5)
	Point	**Mean 1**	**Mean 2**	**Mean 3**
A1	(2, 10)			
A2	(2, 5)			
A3	(8, 4)			
A4	(5, 8)			
A5	(7, 5)			
A6	(6, 4)			
A7	(1, 2)			
A8	(4, 9)			

Iteration 2 – step 1

Step 1: Assign each point to the group with the closest centroid:

– Calculate the **distances** of each point from these centroids (means)

		(2, 10)	(6, 6)	(1.5, 3.5)
	Point	**Mean 1**	**Mean 2**	**Mean 3**
A1	(2, 10)	0	8	7
A2	(2, 5)	5	5	2
A3	(8, 4)	12	4	7
A4	(5, 8)	5	3	8
A5	(7, 5)	10	2	7
A6	(6, 4)	10	2	5
A7	(1, 2)	9	9	2
A8	(4, 9)	3	5	8

Iteration 2 – step 1

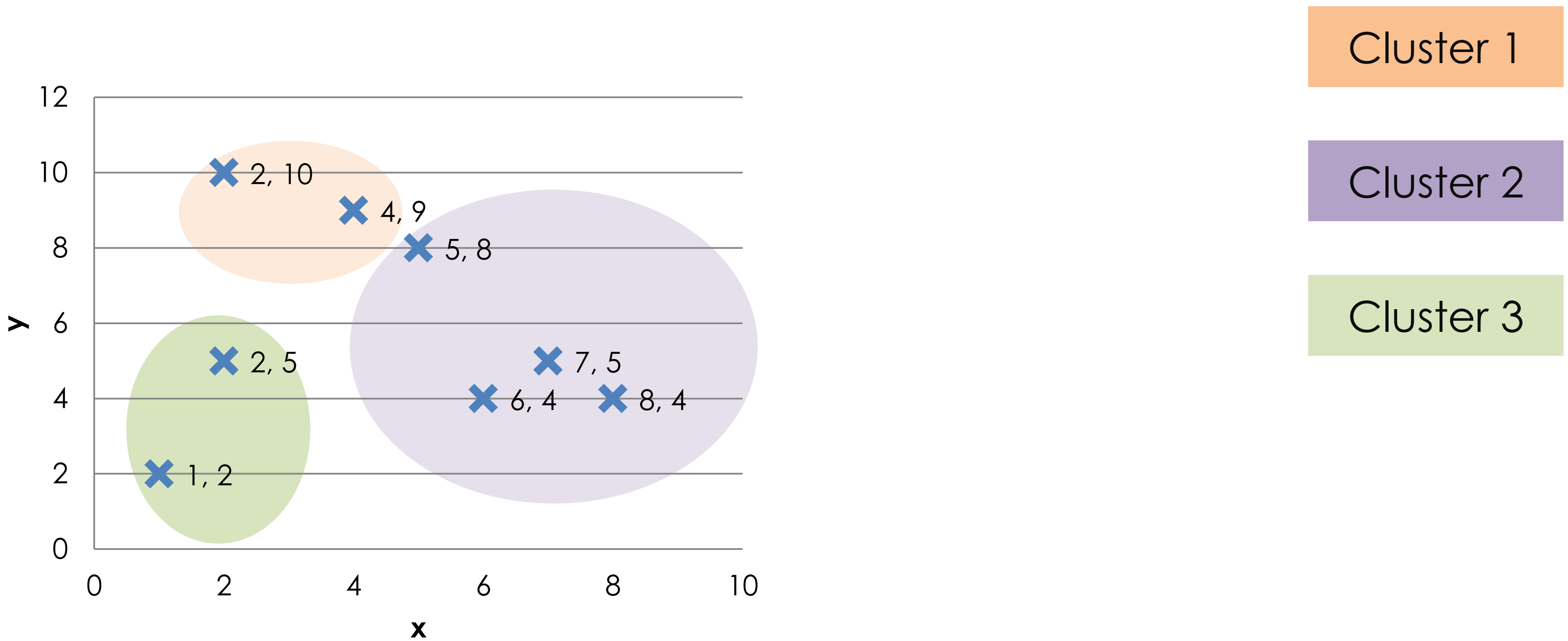
Step 1: Assign each point to the group that has the closest centroid:

- Calculate the **distances** of each point from these centroids (Means)
- Identify the centroid to which the point has the shortest distance.
- Place the point in that centroid's cluster

		(2, 10)	(6, 6)	(1.5, 3.5)	
	Point	**Mean 1**	**Mean 2**	**Mean 3**	**Cluster**
A1	(2, 10)	0	5	9	**1**
A2	(2, 5)	5	6	4	**3**
A3	(8, 4)	12	7	9	**2**
A4	(5, 8)	5	0	10	**2**
A5	(7, 5)	10	5	9	**2**
A6	(6, 4)	10	5	7	**2**
A7	(1, 2)	9	10	0	**3**
A8	(4, 9)	3	2	10	**1**

Iteration 2 clusters

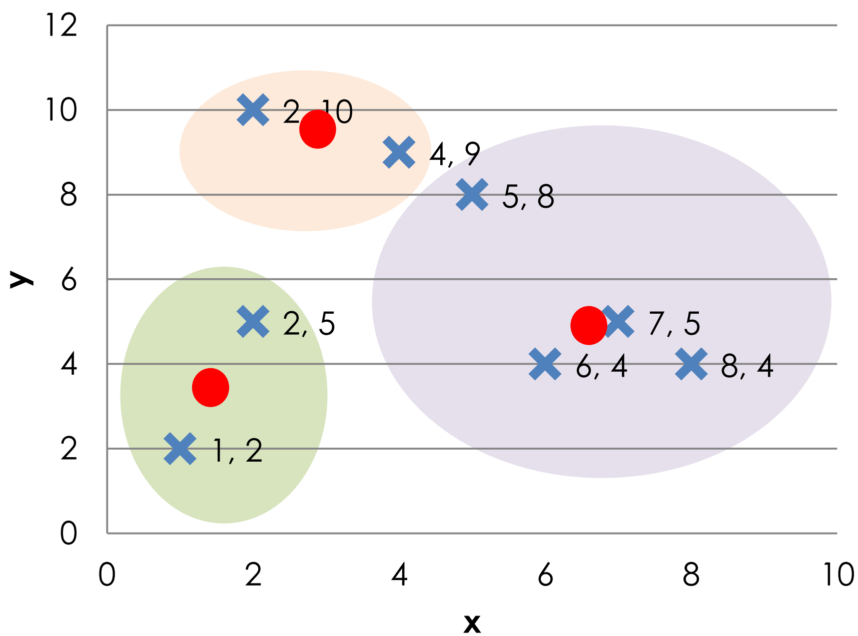
- The second iteration of the algorithm produces these clusters



Iteration 2 – step 2

Step 2: When all points have been assigned, recalculate the positions of the K centroids.

– Recalculate the **means of the clusters**



Cluster 1
(2, 10)
(4, 9)

Cluster 2
(5, 8)
(7, 5)
(6, 4)
(8, 4)

Cluster 3
(2, 5)
(1,2)

For Cluster 1:
The new means is (3, 9.5)
For Cluster 2:
The new means is (6.5, 5.25)
For Cluster 3:
The new means is (1.5, 3.5)

Iteration 3 – step 1

Step 1: Assign each point to the group with the closest centroid:

– Calculate the **distances** of each point from these centroids (Means)

		(3, 9.5)	(6.5, 5.25)	(1.5, 3.5)
	Point	**Mean 1**	**Mean 2**	**Mean 3**
A1	(2, 10)	1.5	9.25	7
A2	(2, 5)	5.5	4.75	2
A3	(8, 4)	10.5	2.75	7
A4	(5, 8)	3.5	4.25	8
A5	(7, 5)	8.5	0.75	7
A6	(6, 4)	8.5	1.75	5
A7	(1, 2)	9.5	8.75	2
A8	(4, 9)	1.5	6.25	8

Iteration 3

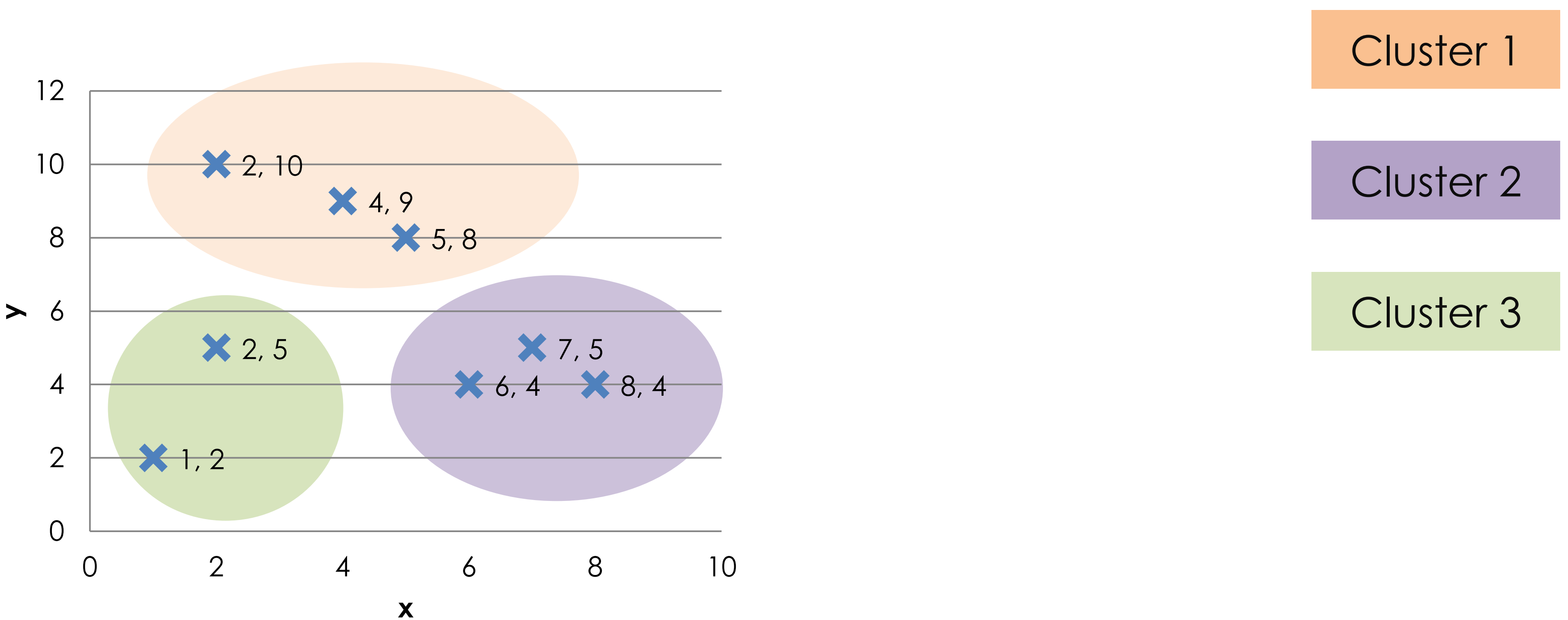
Step 1: Assign each point to the group with the closest centroid:

- Calculate the **distances** of each point from these centroids (Means)
- Identify the centroid to which the point has the shortest distance.
- Place the point in that centroid's cluster

		(3, 9.5)	(6.5, 5.25)	(1.5, 3.5)	
	Point	**Mean 1**	**Mean 2**	**Mean 3**	**Cluster**
A1	(2, 10)	1.5	9.25	7	**1**
A2	(2, 5)	5.5	4.75	2	**3**
A3	(8, 4)	10.5	2.75	7	**2**
A4	(5, 8)	3.5	4.25	8	**1**
A5	(7, 5)	8.5	0.75	7	**2**
A6	(6, 4)	8.5	1.75	5	**2**
A7	(1, 2)	9.5	8.75	2	**3**
A8	(4, 9)	1.5	6.25	8	**1**

Iteration 3 clusters

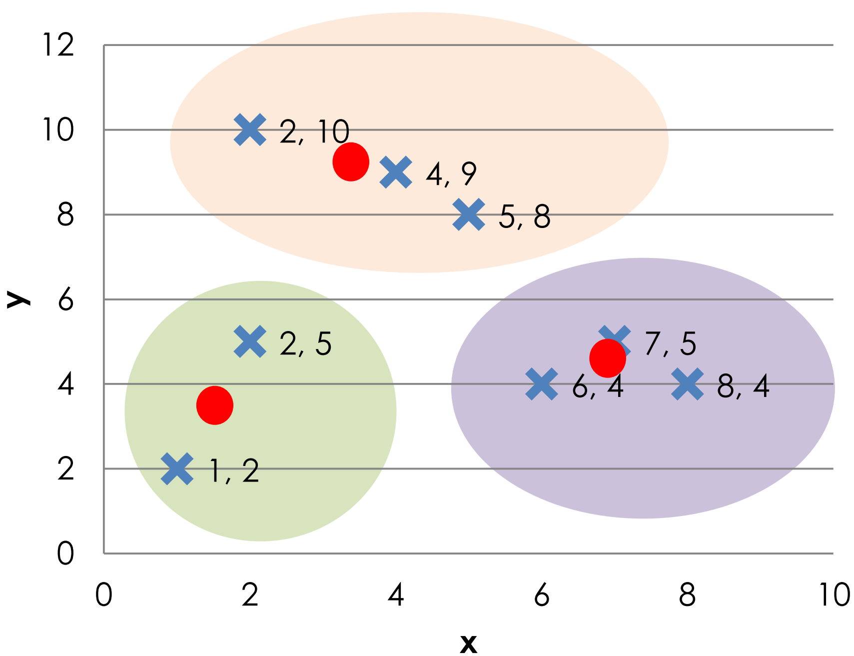
- The third iteration of the algorithm produces these clusters



Iteration 3 – step 2

Step 2: When all points have been assigned, recalculate the positions of the K centroids.

– Recalculate the **means of the clusters**



Cluster 1
(2, 10)
(4, 9)
(5, 8)

Cluster 2
(7, 5)
(6, 4)
(8, 4)

Cluster 3
(2, 5)
(1,2)

For Cluster 1:
The new means is (3.667, 9)

For Cluster 2:
The new means is (7, 4.33)

For Cluster 3:
The new means is (1.5, 3.5)

Iteration 4?

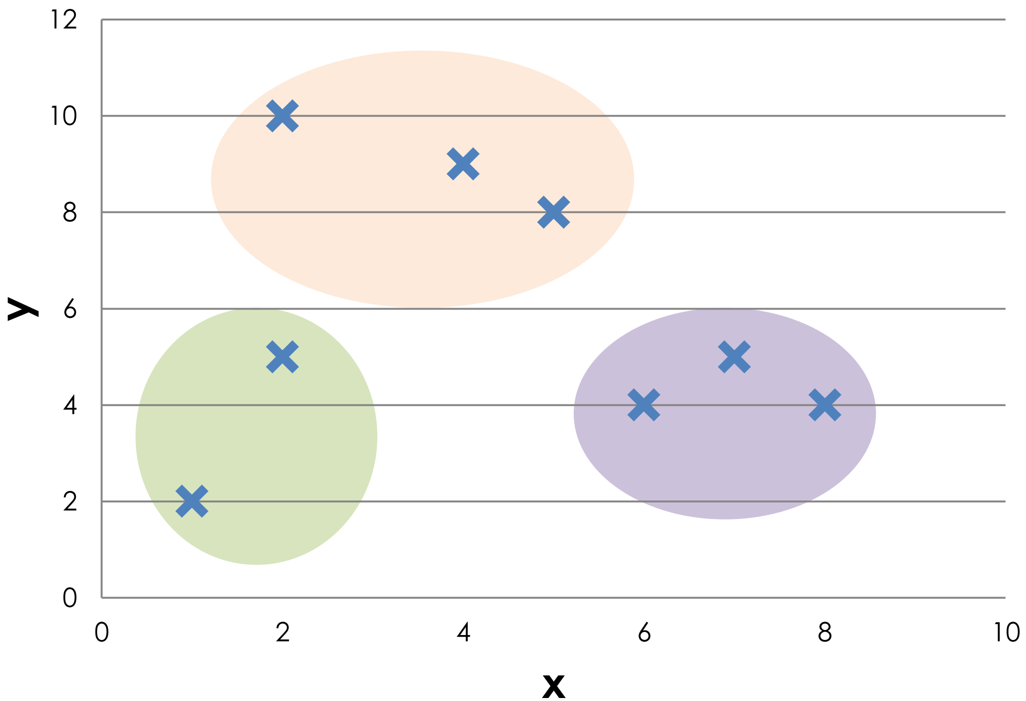
- If we keep going and repeat steps 1 and 2, we notice that there is **no change** in the cluster assignments, so we can **stop**.

		(3.67, 9)	(7, 4.33)	(1.5, 3.5)	
	Point	**Mean 1**	**Mean 2**	**Mean 3**	**Cluster**
A1	(2, 10)	2.67	10.67	7	**1**
A2	(2, 5)	5.67	5.67	2	**3**
A3	(8, 4)	9.33	1.33	7	**2**
A4	(5, 8)	2.33	5.67	8	**1**
A5	(7, 5)	7.33	0.67	7	**2**
A6	(6, 4)	7.33	1.33	5	**2**
A7	(1, 2)	9.67	8.33	2	**3**
A8	(4, 9)	0.33	7.67	8	**1**

Same clusters as in the previous iteration!

K-means algorithm

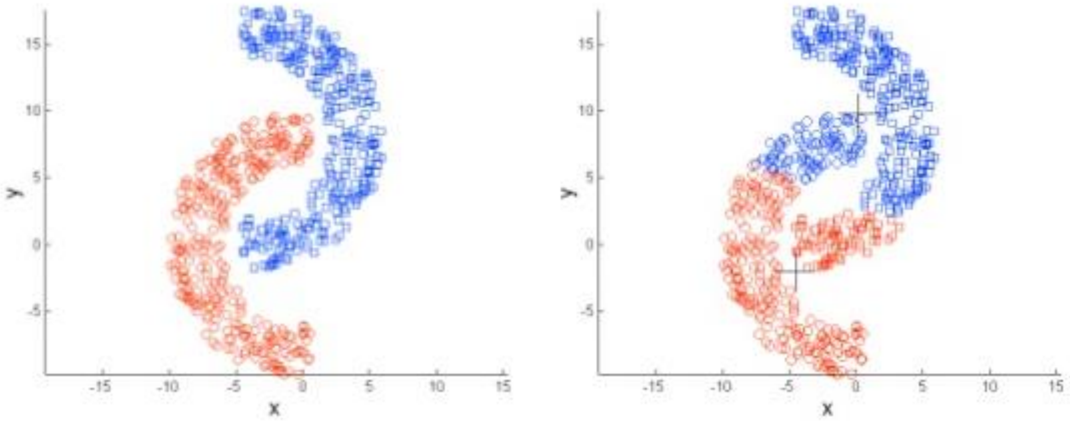
- So this is the final output of the clustering algorithm



K-means Strengths & Weaknesses

Advantages

- Simple
- Can be used for a variety of data types
- Efficient



Original Points K-means (2 Clusters)

Disadvantages

- It cannot handle non-globular data or data that do not have a centre
- Need to specify k in advance (in most cases)
- It has trouble clustering data that contains outliers
  - But outlier detection/removal can help

Hierarchical (agglomerative) clustering

- Hierarchical clustering results in a series of clustering results
- The results start off with each object in their own cluster and end with all of the objects in the same cluster
- The intermediate clusters are created by a series of **merges**
- The resultant tree like structure is called a 'dendrogram'

- The idea is to build a binary tree of the data that successively merges similar groups of points
- Visualizing this tree provides a useful summary of the data

Hierarchical (agglomerative) clustering - Algorithm

0. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances between the clusters the same as the distances between the items they contain.

1. Find the **closest** pair of clusters and **merge** them into a single cluster, so that now you have one cluster less.
2. **Recalculate** distances between the new cluster and each of the old clusters. *
3. **Repeat** steps 1 and 2 until all items are clustered into a **single** cluster of size N.

* Step 2 can be done in many ways (see slide later)

Hierarchical (agglomerative) clustering - Example

- This table shows the distances in km between Italian cities.
  - Using Euclidean distance
    - Upper triangle contains the same values so just for convenience I will not show it

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

Iteration 1 - step 1

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Which is the closest pair (step 1)?

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

Iteration 1 - step 1

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Milan and Turin are the closest (138 km), so I will merge (cluster) MI and TO together

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

Hierarchical (agglomerative) clustering - Example

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Milan and Torino are the closest, so I will merge MI and TO together

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI					0

Iteration 1- step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Recalculate distances between the new cluster and each of the old clusters.

Looking at the distances between the TO/MI cluster (new cluster) and the all other cities (old clusters), I find whether the **shortest** distance to each city is from TO or from MI, and add that number.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	?	?	?	?	0

Iteration 1 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Looking at the distances between the TO/MI cluster (new cluster) and the all other cities (old clusters), I find whether the **shortest** distance to each city is from TO or from MI, and add that number. In other words:

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

To BA, is the shortest distance from TO or from MI?

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	?				0

Iteration 1 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Looking at the distances between the TO/MI cluster (new cluster) and the all other cities (old clusters), I find whether the **shortest** distance to each city is from TO or from MI, and add that number.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	**877**	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	**996**	400	138	869	669	0

To BA, from TO the distance is 996, but from MI it is 877, so I add 877!

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877				0

Iteration 1 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Looking at the distances between the TO/MI cluster (new cluster) and the all other cities (old clusters), I find whether the **shortest** distance to each city is from TO or from MI, and add that number.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

To FI, is the shortest distance from TO or from MI?

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	?			0

Iteration 1 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Looking at the distances between the TO/MI cluster (new cluster) and the all other cities (old clusters), I find whether the **shortest** distance to each city is from TO or from MI, and add that number.

To FI, from TO the distance is 400, but from MI it is 295, so I add 295!

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

→

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	295			0

Iteration 1 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Looking at the distances between the TO/MI cluster (new cluster) and the all other cities (old clusters), I find whether the **shortest** distance to each city is from TO or from MI, and add that number.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

To NA, is the shortest distance from TO or from MI?

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	295	?		0

Iteration 1 - step 2

BA: Bari FI: Florence MI: Milan NA: Naples RM: Rome TO: Turin

Looking at the distances between the TO/MI cluster (new cluster) and the all other cities (old clusters), I find whether the **shortest** distance to each city is from TO or from MI, and add that number.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

To RM, is the shortest distance from TO or from MI?

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	295	754	?	0

Iteration 1 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Recalculate distances between the new cluster and each of the old clusters.

Looking at the distances between the TO/MI cluster (new cluster) and the all other cities (old clusters), I find whether the **shortest** distance to each city is from TO or from MI, and add that number.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

Step 2 is done!

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	295	754	564	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 2 - step 1

And we repeat!

Which is the closest pair now?

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	295	754	564	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 2 - step 1

NA and RM at 219 is the closest pair

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	295	754	564	0

Iteration 2 - step 1

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Next new cluster is NA/RM

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	295	754	564	0

	BA	FI	NA/RM	TO/MI
BA	0			
FI	662	0		
NA/RM			0	
TO/MI	877	295		0

Iteration 2 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

I recalculate distances between the new cluster and the old ones, picking the shortest distance from RM or NA

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	877	295	754	564	0

To BA, is the shortest distance from NA or from RM?

	BA	FI	NA/ RM	TO/ MI
BA	0			
FI	662	0		
NA/RM	?	?	0	
TO/MI	877	295	?	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 2 - step 2

I recalculate distances between the new cluster and the old ones, picking the shortest distance from RM or NA

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	**255**	468	0		
RM	**412**	268	219	0	
TO/MI	887	295	754	564	0

To BA, the distance from NA is 255, and from RM it is 412, so I add 255.

	BA	FI	NA/ RM	TO/ MI
BA	0			
FI	662	0		
NA/RM	255	?	0	
TO/MI	877	295	?	0

Iteration 2 - step 2

I recalculate distances between the new cluster and the old ones, picking the shortest distance from RM or NA.

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	**468**	0		
RM	412	**268**	219	0	
TO/MI	887	295	**754**	**564**	0

To FI and to TO/MI, the shortest distance is from RM (268 and 564).
Step 2 is done.

	BA	FI	NA/ RM	TO/ MI
BA	0			
FI	662	0		
NA/RM	255	268	0	
TO/MI	877	295	564	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 3 - step 1

The closest pair is BA and NA/RM at 255

	BA	FI	NA/RM	TO/MI
BA	0			
FI	662	0		
NA/RM	255	268	0	
TO/MI	877	295	564	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 3 - step 1

Next cluster is BA/NA/RM

	BA	FI	NA/RM	TO/MI
BA	0			
FI	662	0		
NA/RM	255	268	0	
TO/MI	877	295	564	0

→

	FI	BA/NA/RM	TO/MI
FI	0		
BA/NA/RM		0	
TO/MI	295		0

BA: Bari FI: Florence MI: Milan NA: Naples RM: Rome TO: Turin

Iteration 3 - step 2

I recalculate distances between the new cluster and the old ones, picking the shortest distance from BA or from NA/RM.

	BA	FI	NA/RM	TO/MI
BA	0			
FI	662	0		
NA/RM	255	268	0	
TO/MI	877	295	564	0

To FI, is the shortest distance from BA or from NA/RM?

→

	FI	BA/NA/RM	TO/MI
FI	0		
BA/NA/RM	?	0	
TO/MI	295	?	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 3 - step 2

I recalculate distances between the new cluster and the old ones, picking the shortest distance from BA or from NA/RM.

	BA	FI	NA/RM	TO/MI
BA	0			
FI	662	0		
NA/RM	255	268	0	
TO/MI	877	295	564	0

To TO/MI, is the shortest distance from BA or from NA/RM?

	FI	BA/NA/RM	TO/MI
FI	0		
BA/NA/RM	268	0	
TO/MI	295	?	0

Iteration 3 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

I recalculate distances between the new cluster and the old ones, picking the shortest distance from BA or from NA/RM.

Step 2 is done

	BA	FI	NA/RM	TO/MI
BA	0			
FI	662	0		
NA/RM	255	268	0	
TO/MI	877	295	564	0

→

	FI	BA/NA/RM	TO/MI
FI	0		
BA/NA/RM	268	0	
TO/MI	295	564	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 4 - step 1

FI and BA/NA/RM are the closest at 268

	FI	BA/NA/RM	TO/MI
FI	0		
BA/NA/RM	268	0	
TO/MI	295	564	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 4 - step 1

Next cluster is FI/BA/NA/RM

	FI	BA/NA/RM	TO/MI
FI	0		
BA/NA/RM	268	0	
TO/MI	295	564	0

	FI/BA/NA/RM	TO/MI
FI/BA/NA/RM	0	
TO/MI		0

Iteration 4 - step 2

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Picking shortest distance (from FI or from BA/NA/RM)

	FI	BA/NA/RM	TO/MI
FI	0		
BA/NA/RM	268	0	
TO/MI	295	564	0

To TO/MI, is the shortest distance from FI or from BA/NA/RM?

→

	FI/BA/NA/RM	TO/MI
FI/BA/NA/RM	0	
TO/MI	?	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 4 - step 2

Picking shortest distance (from FI or from BA/NA/RM)

	FI	BA/NA/RM	TO/MI
FI	0		
BA/NA/RM	268	0	
TO/MI	295	564	0

→

	FI/BA/NA/RM	TO/MI
FI/BA/NA/RM	0	
TO/MI	295	0

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Iteration 5

Finally we merge the two clusters TO/MI and FI/BA/NA/RM at 295

	FI/BA/NA/RM	TO/MI
FI/BA/NA/RM	0	
TO/MI	295	0

Iteration 5

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Finally we merge the two clusters TO/MI and FI/BA/NA/RM at 295, and algorithm terminates (all all items are clustered into a **single** cluster of size N).

	FI/BA/NA/RM	TO/MI
FI/BA/NA/RM	0	
TO/MI	295	0

→

	TO/MI/FI/BA/NA/RM
TO/MI/FI/BA/NA/RM	0

Hierarchical (agglomerative) clustering - Example

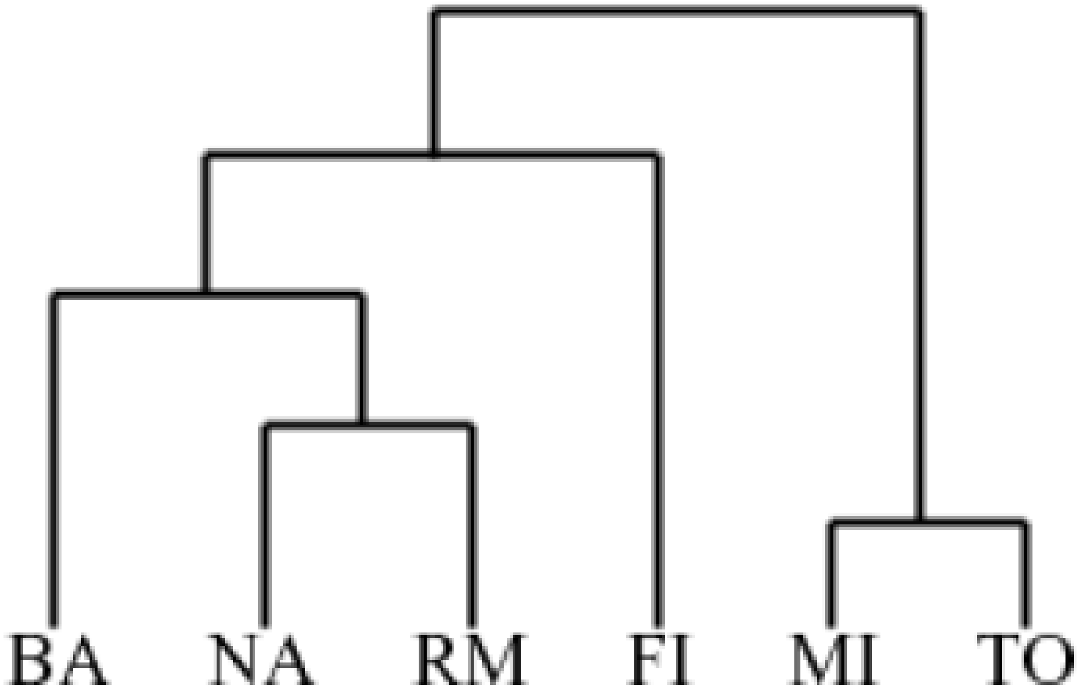
- So successively we merged:
  1. TO and MI
  2. NA and RM
  3. BA and NA/RM
  4. FI and BA/NA/RM
  5. TO/MI and FI/BA/NA/RM

BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

Resulting dendrogram

- Shows the merges

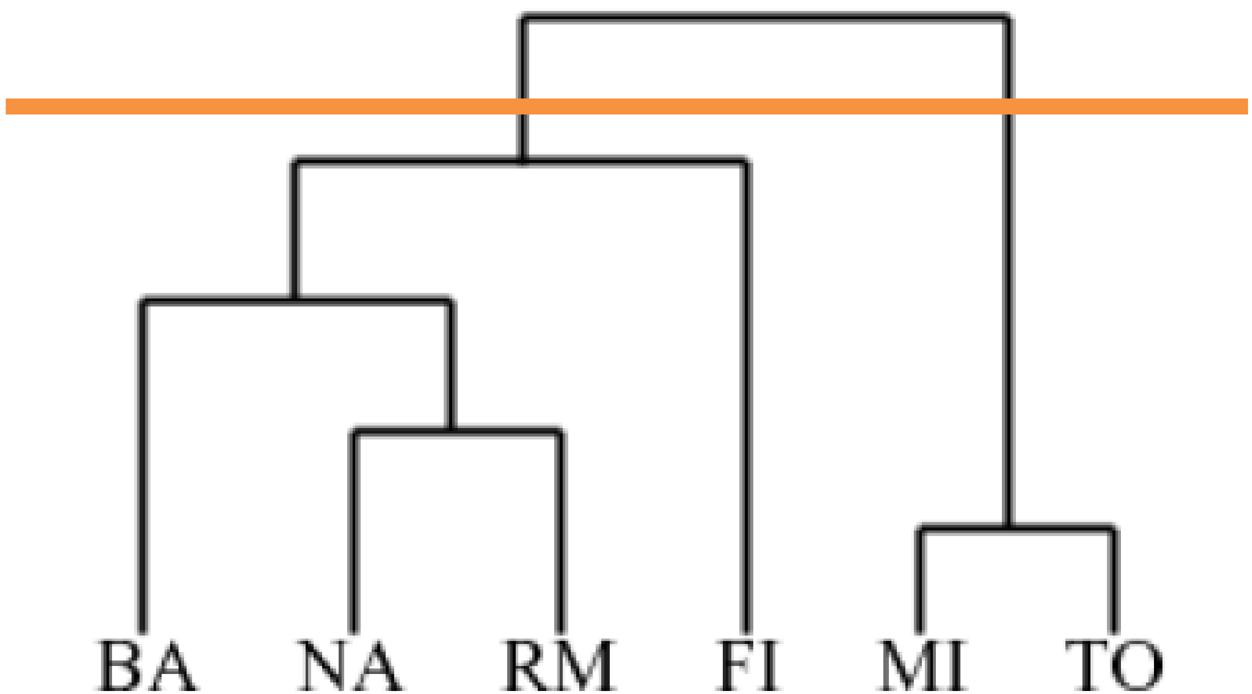
1. TO and MI
2. NA and RM
3. BA and NA/RM
4. FI and BA/NA/RM
5. TO/MI and FI/BA/NA/RM



BA: Bari
FI: Florence
MI: Milan
NA: Naples
RM: Rome
TO: Turin

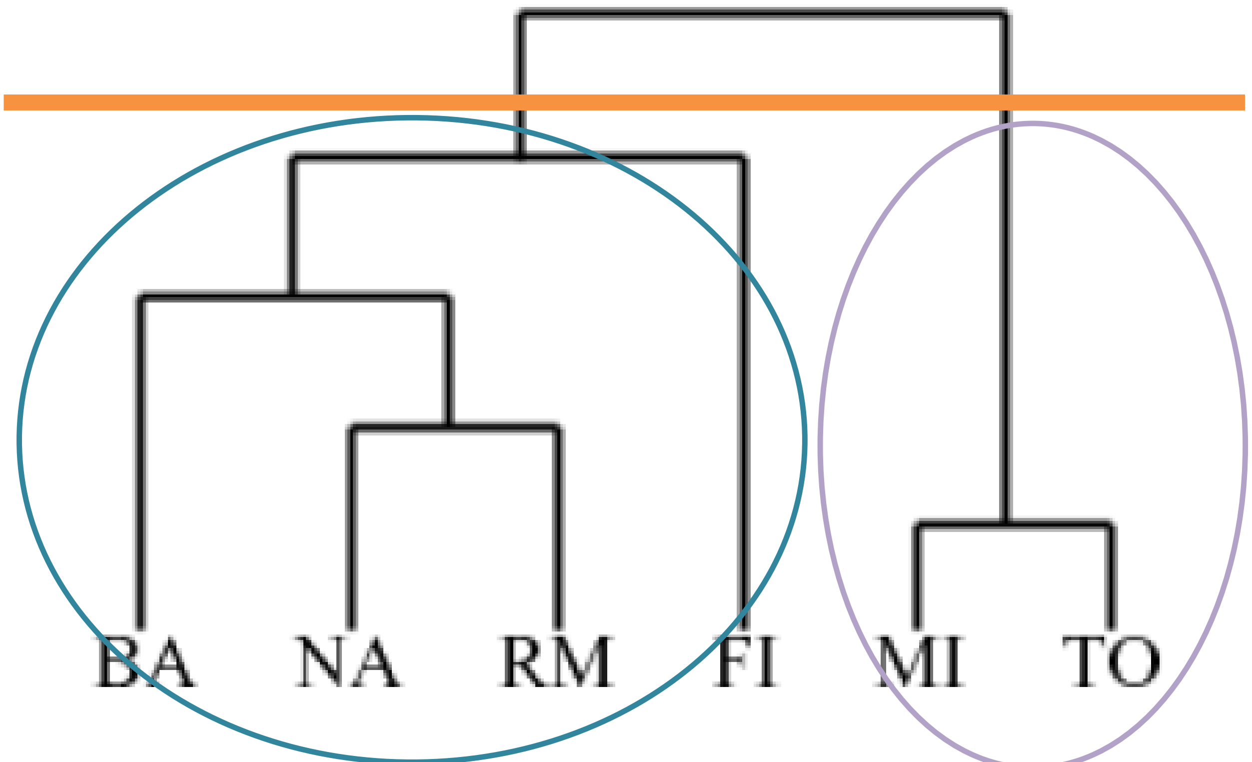
Determining clusters

- Clustering obtained by **cutting** the dendrogram at a desired level.
- How many clusters are there?
- Two clusters!



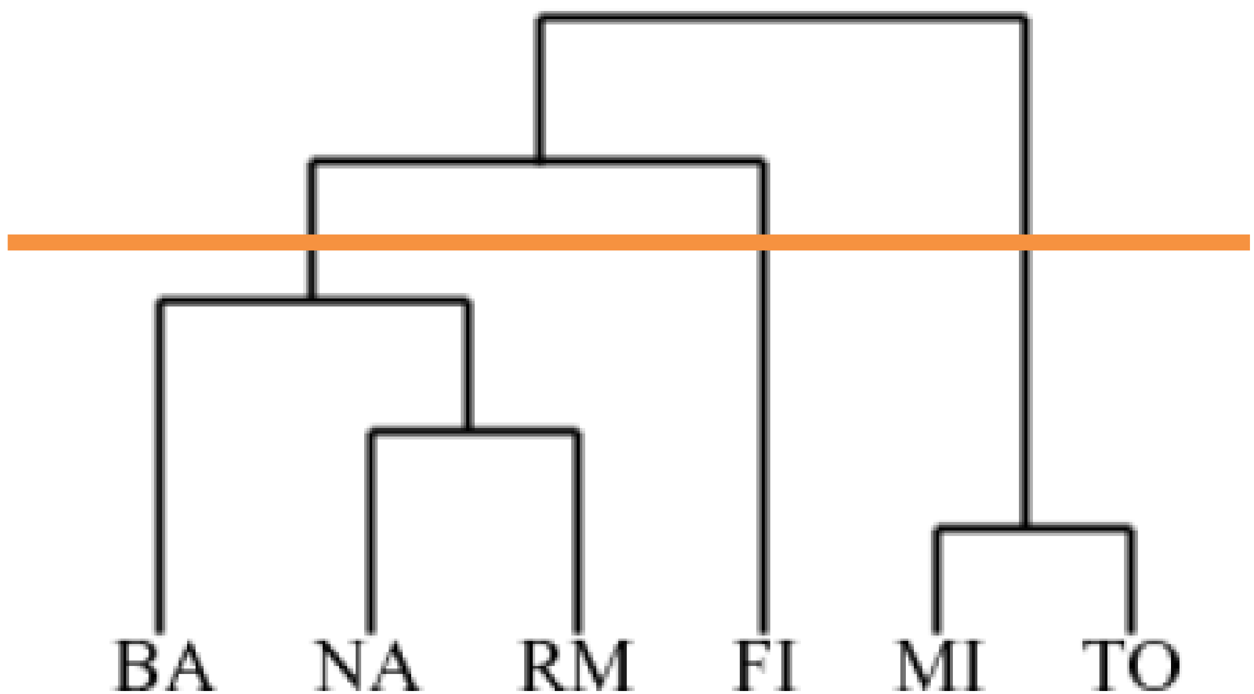
Determining clusters

- Clustering obtained by **cutting** the dendrogram at a desired level.
- How many clusters are there?
- Two clusters!



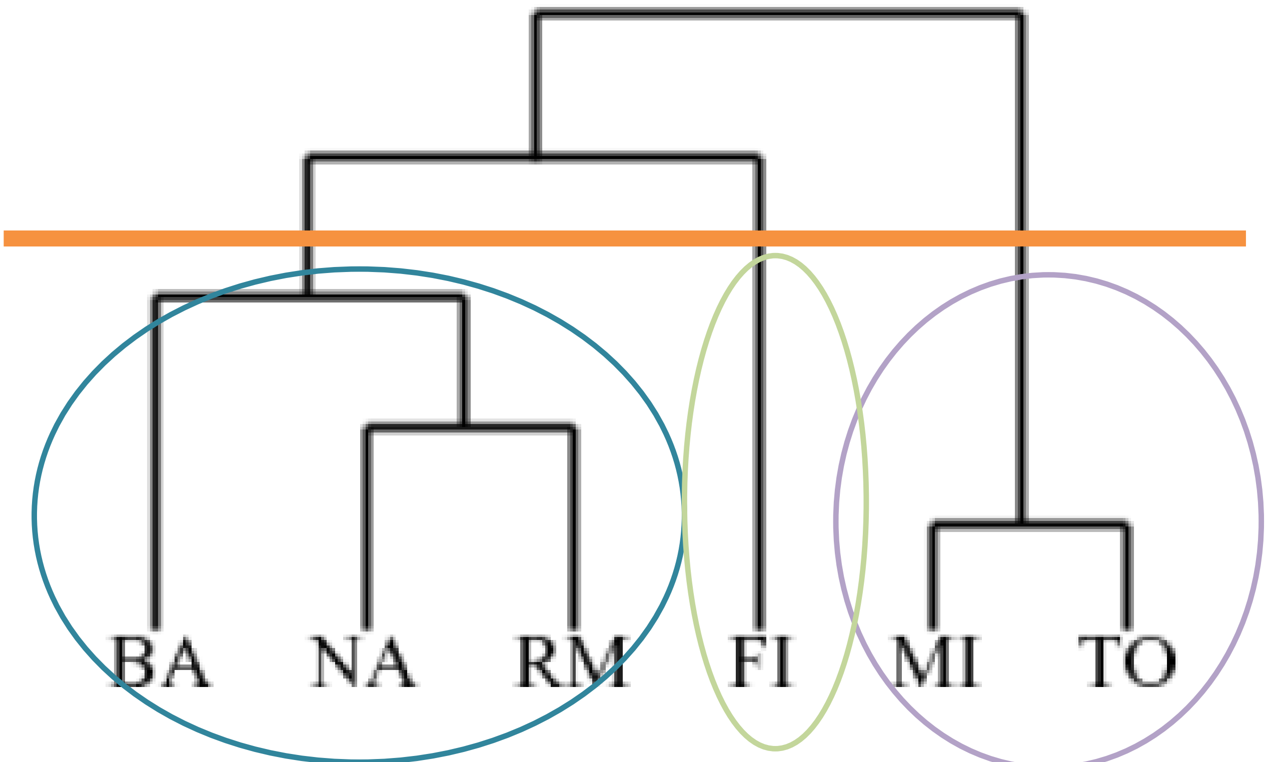
Determining clusters

- Clustering obtained by cutting the dendrogram at a desired level.
- How many clusters are there now?
- Three (one of which has a single element)



Determining clusters

- Clustering obtained by cutting the dendrogram at a desired level.
- How many clusters are there now?
- Three (one of which has a single element)



Variations

- In the example, we used the **shortest** distance between elements of each cluster (Step 2 in the algorithm)
- This is called **single-linkage**.
- We could have used the **longest** distance (called **complete-linkage**)
- Or the average distance

- Or even other ways…

Variations based on distance criterion

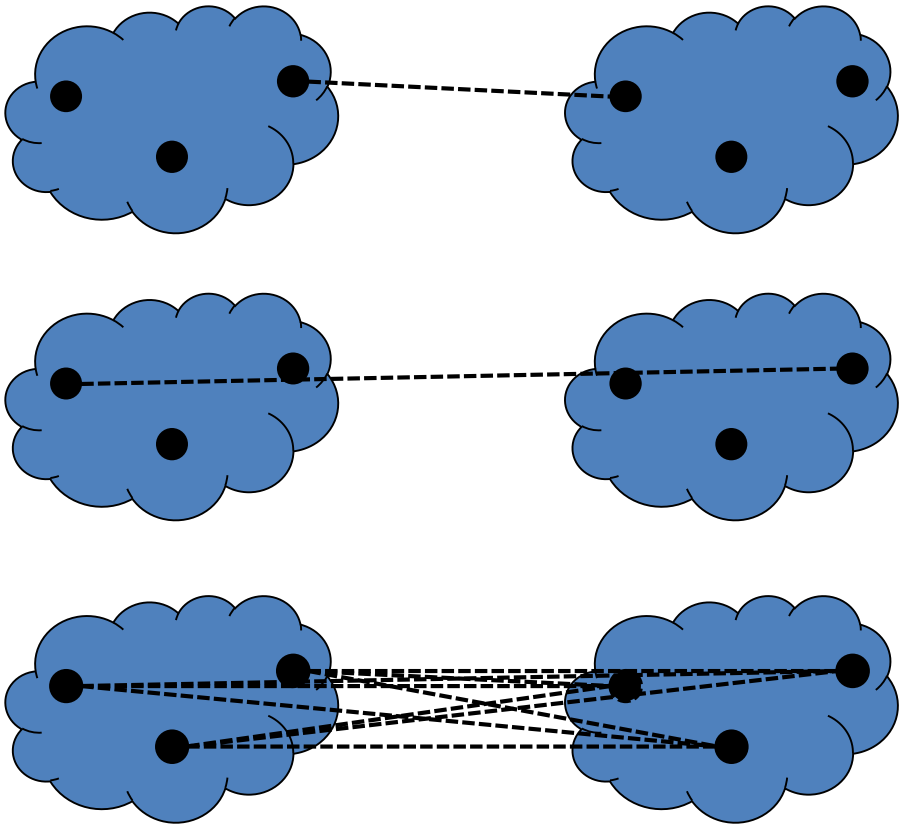
Linkage	Description
Single	The shortest distance between any two pairs from the two clusters (one from each) being compared/measured
Average	The average distance between pairs
Complete	The longest distance between any two pairs from the two clusters (one from each) being compared/measured

Variations based on distance criterion

Single Linkage

Complete Linkage

Average Linkage



How step 2 would have been performed had I used **complete** linkage

Recalculate distances between the new cluster and each of the old clusters.

Looking at the distances between all cities (old clusters) and the TO/MI cluster (new cluster), I find whether the **LONGEST** distance to each city is from TO or from MI, and add that number.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

→

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	?	?	?	?	0

How step 2 would have been performed had I used **complete** linkage

Looking at the distances between all cities (old clusters) and the TO/MI cluster (new cluster), I find whether the **longest** distance to each city is from TO or from MI, and add that number.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

→

To BA, is the longest distance from TO or from MI?

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	?				0

How step 2 would have been performed had I used complete linkage

Looking at the distances between all cities (old clusters) and the TO/MI cluster (new cluster), I find whether the **LONGEST** distance is from TO or from MI, and add that.

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

To BA, from TO the distance is 996, and from MI is 877, so I add 996!

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	996				0

How step 2 would have been performed had I used **average** linkage

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

To BA, from TO the distance is 996, and from MI is 877, so I add …

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI					0

How step 2 would have been performed had I used **average** linkage

	BA	FI	MI	NA	RM	TO
BA	0					
FI	662	0				
MI	877	295	0			
NA	255	468	754	0		
RM	412	268	564	219	0	
TO	996	400	138	869	669	0

To BA, from TO the distance is 996, and from MI is 877, so I add **(877 + 996)/2 = 936.5!**

	BA	FI	NA	RM	TO/MI
BA	0				
FI	662	0			
NA	255	468	0		
RM	412	268	219	0	
TO/MI	936.5				0

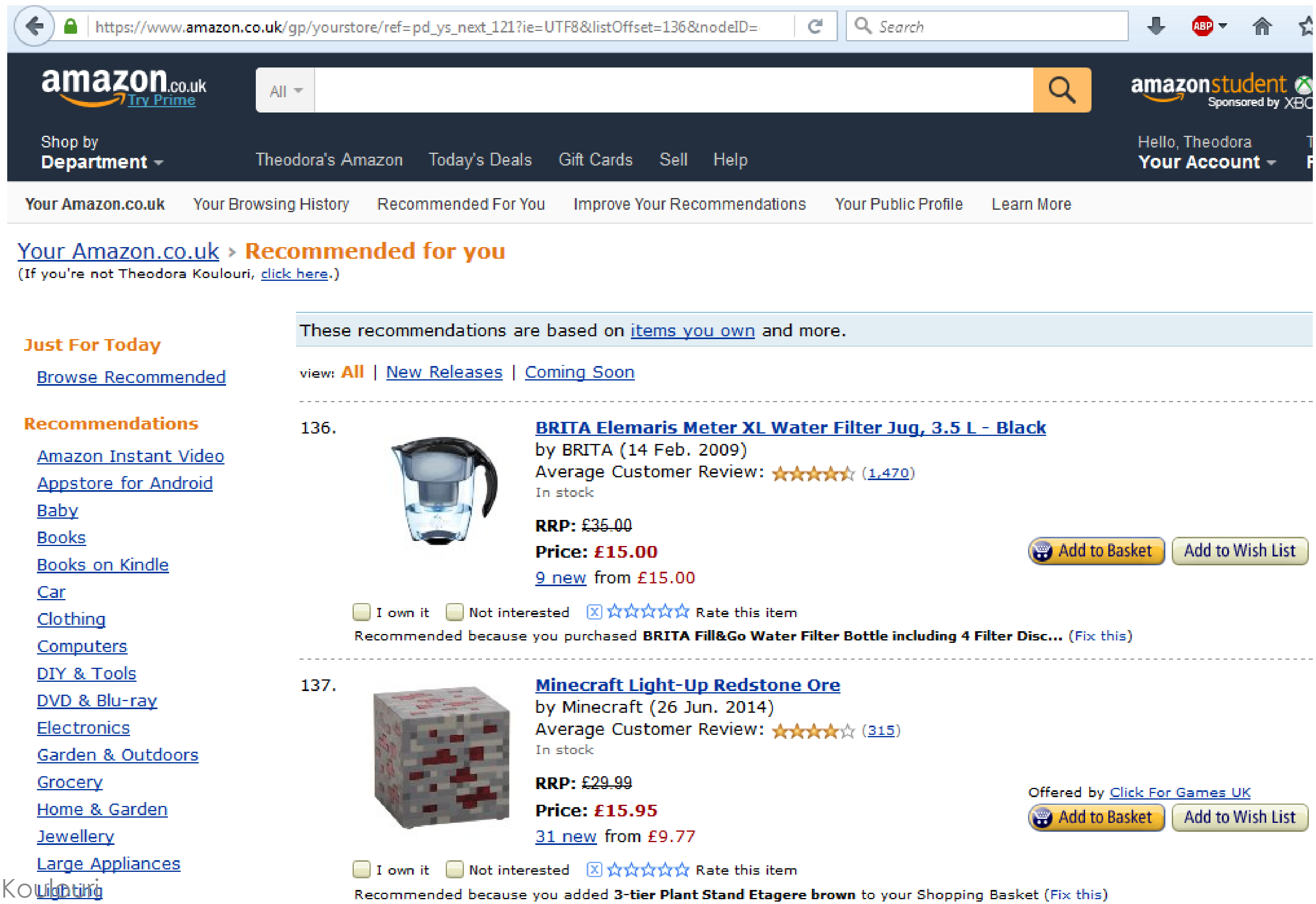
Pros and cons of hierarchical clustering

- Advantages
  - Hierarchies of similar objects are produced, which is informative for particular applications
  - Do not require information regarding number of clusters. Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level.
- Disadvantages
  - Not efficient
  - Once a decision to combine two clusters is made, it cannot be undone.
  - Use of different distance metrics for measuring distances between clusters may generate different results(single, complete, average linkage)

Example

- In 1992, consulting group Teradata prepared an analysis of 1.2 million market baskets from about 25 Osco Drug stores.
- The analysis "discovered that between 5:00 and 7:00 p.m. consumers bought beer and diapers".

Unsupervised learning: Association rules



Unsupervised learning: Association rules

ASSOCIATION RULE:
`if {X} then {Y}`
$X \rightarrow Y$

Association Rules are selected based on **Support** and **Confidence**:

Let X be an item-set, $X \rightarrow Y$ an association rule and T a set of transactions of a given database.

- **Support**
  - The support value of X with respect to T is defined as the **proportion of transactions which contains the item-set X.**
- **Confidence**
  - The confidence value of a rule, $X \rightarrow Y$ with respect to T is **the proportion of transactions that contains X which also contains Y.**

The goal of association rule mining is to find all rules that have support and confidence above some threshold

Example – Association rules

- The tables contains 5 market basket transactions
  - Of course in real-life the dataset can consist of trillions of transactions
- Calculate the **support** of the itemset {Milk, Diapers, Beer}
  - proportion of transactions that contain this itemset
  - 2/5 = 0.4

Trans ID	**Items**
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Cheesy puffs
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Cheesy puffs

Example – Association rules

- The tables contains 5 market basket transactions
  - Of course in real-life the dataset can consist of trillions of transactions
- Calculate the **confidence** of the association rule {Milk, Diapers} → {Beer}
  - proportion of transactions that contain {Milk, Diapers} which also contains {Beer}
  - (2/5)/(3/5) ≈ 0.67

Trans ID	**Items**
1	Bread, Milk
2	Bread, Diapers, Beer, Eggs
3	Milk, Diapers, Beer, Cheesy puffs
4	Bread, Milk, Diapers, Beer
5	Bread, Milk, Diapers, Cheesy puffs

Software implementations of clustering algorithms

- R *
- Weka *
- SciPy and scikit-learn python libraries *
- MATLAB
- SPSS
- SAS
- STATA
- …

* Free software/OS licence

BUT DO NOT TRUST IMPLEMENTATIONS BLINDLY, ML is not a black box to throw in some data and to get some results out!

Summary

- Machine Learning and different types of learning
- Unsupervised learning (Clustering and Association Rules)
- Clustering: K-means and Hierarchical clustering

Additional resources

Reading:

- Tan et al. 'Introduction to Data Mining'
  http://www-users.cs.umn.edu/~kumar/dmbook/index.php
- Chapter 8 – Cluster Analysis (k-means and hierarchical clustering)
- Chapter 6 – Association Analysis

Free download

Watching:

- Andrew Ng's Stanford Machine Learning lectures.

Brilliant module – covers everything in ML!