Chapter – 9

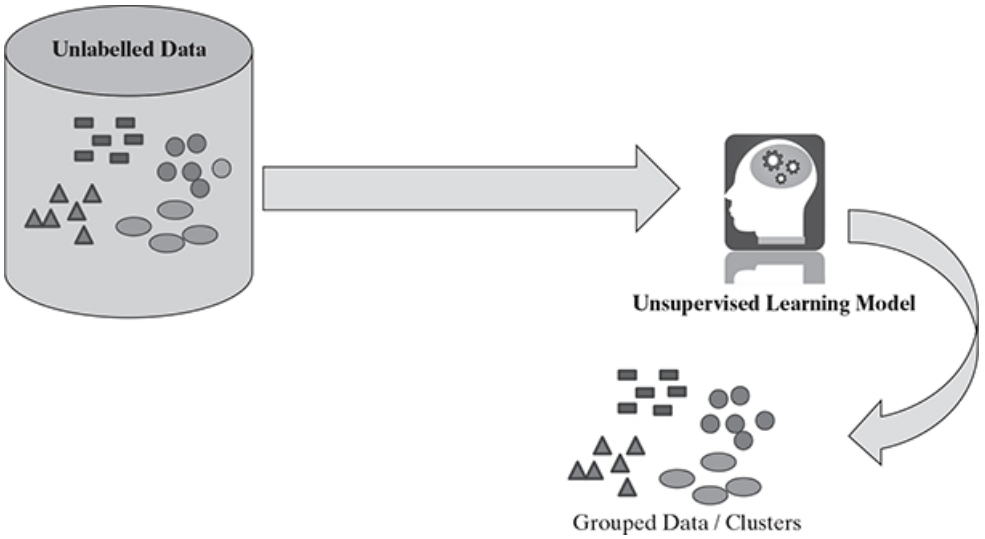
A study material for the students of GLS University
Compiled by Dr. Krupa Mehta

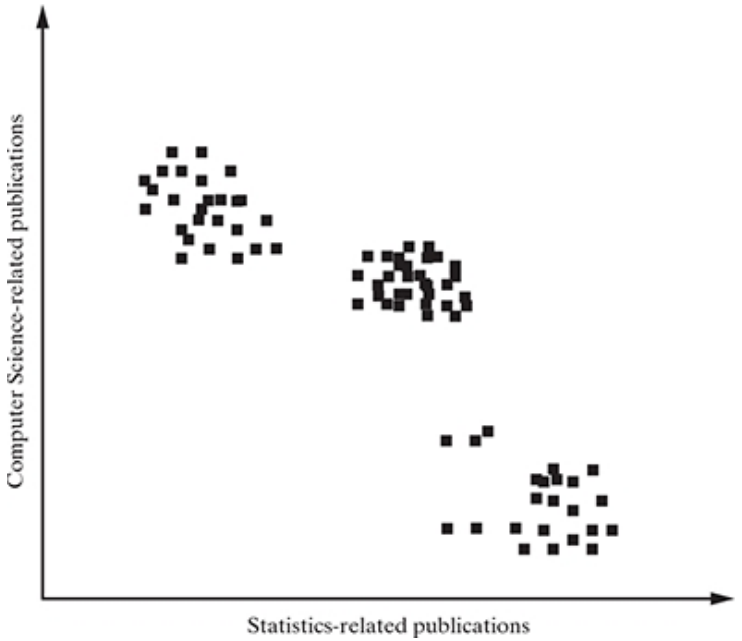
Clustering

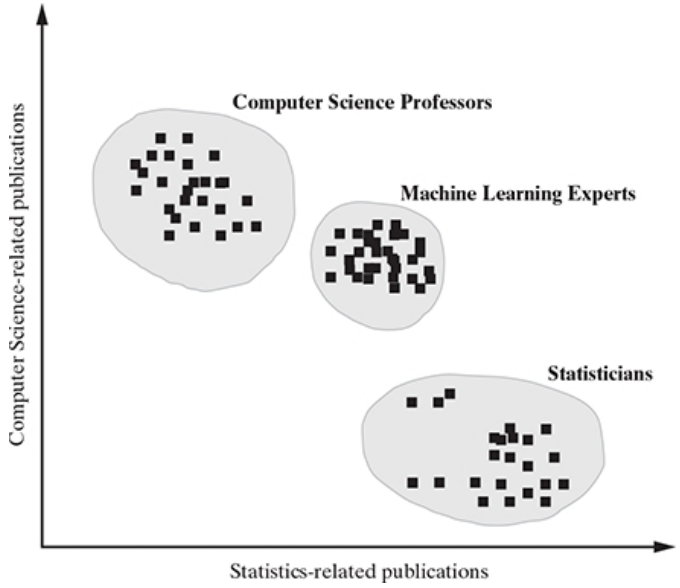
- broad set of techniques for finding subgroups, or clusters, in a data set on the basis of the characteristics
- on the basis of the characteristics of the objects within that data set
- objects within the group are similar
- but are different from the objects from the other groups

Clustering

- Applications
  - Text data mining
  - Customer segmentation
  - Anomaly checking
  - Data mining


Unlabelled Data
Unsupervised Learning Model
Grouped Data / Clusters


Computer Science-related publications
Statistics-related publications


Computer Science Professors
Machine Learning Experts
Statisticians
Computer Science-related publications
Statistics-related publications

Clustering

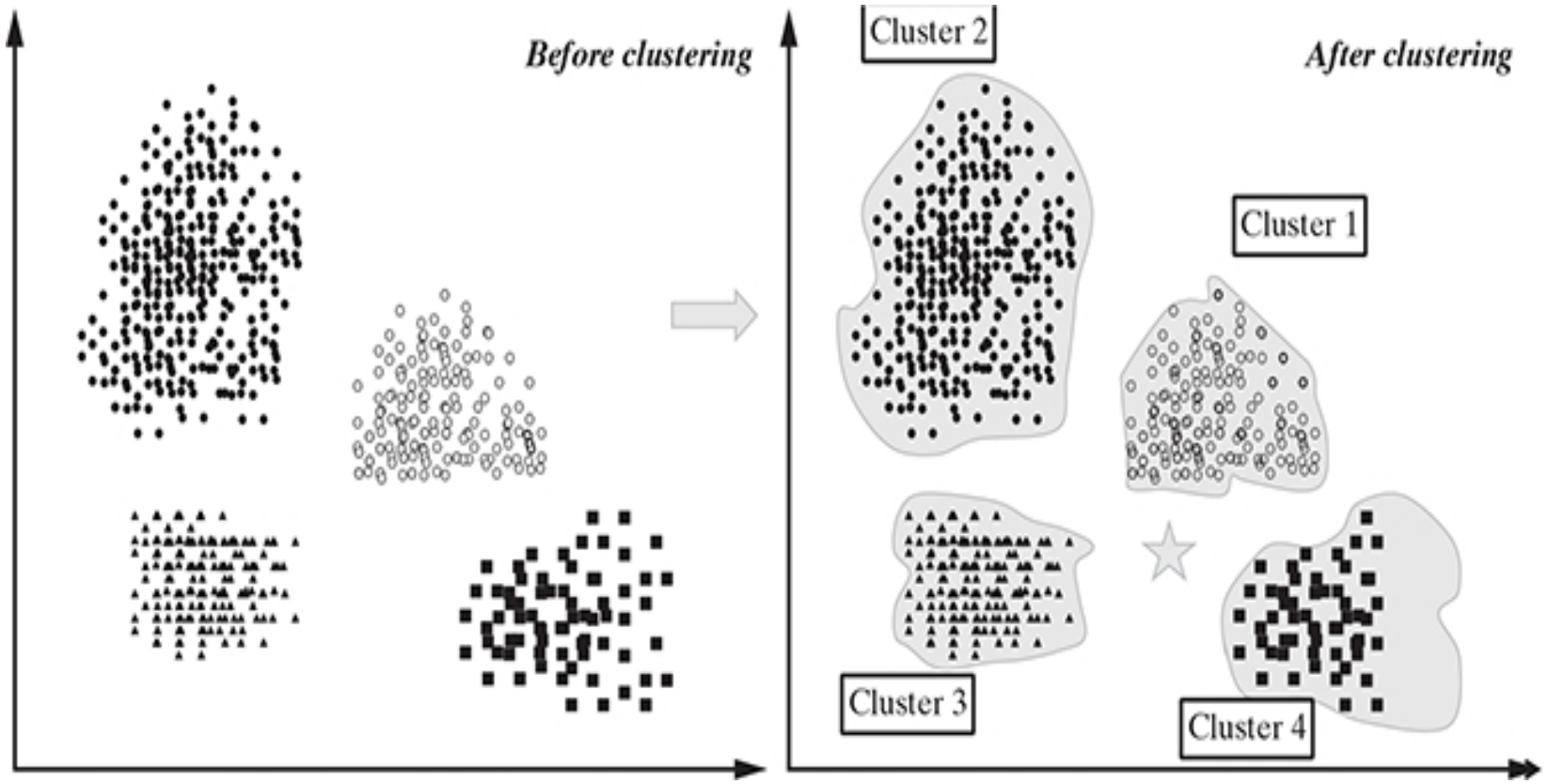
- Different types of clustering techniques

Method	Characteristics
Partitioning methods	• Uses mean or medoid (etc.) to represent cluster centre • Adopts distance-based approach to refine clusters • Finds mutually exclusive clusters of spherical or nearly spherical shape • Effective for data sets of small to medium size
Hierarchical methods	• Creates hierarchical or tree-like structure through decomposition or merger • Uses distance between the nearest or furthest points in neighbouring clusters as a guideline for refinement • Erroneous merges or splits cannot be corrected at subsequent levels
Density-based methods	• Useful for identifying arbitrarily shaped clusters • Guiding principle of cluster creation is the identification of dense regions of objects in space which are separated by low-density regions • May filter out outliers

Partitioning methods - K-means

- **Algorithm**
  - **Step 1:** Select K points in the data space and mark them as initial centroids
  - **loop**
  - **Step 2:** Assign each point in the data space to the nearest centroid to form K clusters
  - **Step 3:** Measure the distance of each point in the cluster from the centroid
  - **Step 4:** Calculate the Sum of Squared Error (SSE) to measure the quality of the clusters
  - **Step 5:** Identify the new centroid of each cluster on the basis of distance between points
  - **Step 6:** Repeat Steps 2 to 5 to refine until centroids do not change
  - **end loop**

K-means


Before clustering
Cluster 2
After clustering
Cluster 1
Cluster 3
Cluster 4

K-means

- Consider the data points P1(1,3) , P2(2,2) , P3(5,8) , P4(8,5) , P5(3,9) , P6(10,7) , P7(3,3) , P8(9,4) , P9(3,7)
- K = 3
- Assume initial cluster centers as P7(3,3), P9(3,7), P8(9,4) as C1, C2, C3
- Find the distance between data points and Centroids

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

K-means

- Calcualting the distance between data points and (C1,C2,C3)

ITERATION - 1

- C1P1 =>(3,3)(1,3) => sqrt[(1–3)²+(3–3)²] => sqrt[4] =>2
- C2P1 =>(3,7)(1,3)=> sqrt[(1–3)²+(3–7)²] => sqrt[20] =>4.5
- C3P1 =>(9,4)(1,3) => sqrt[(1–9)²+(3–4)²]=> sqrt[65] =>8.1
- C1P2 =>(3,3)(2,2) => sqrt[(2–3)²+(2–3)²] => sqrt[2] =>1.4
- C2P2 =>(3,7)(2,2)=> sqrt[(2–3)²+(2–7)²] => sqrt[26] =>5.1
- C3P2 =>(9,4)(2,2) => sqrt[(2–9)²+(2–4)²]=> sqrt[53] =>7.3
- C1P2 =>(3,3)(5,8) => sqrt[(5–3)²+(8–3)²] => sqrt[29] =>5.3
- C2P2 =>(3,7)(5,8)=> sqrt[(5–3)²+(8–7)²] => sqrt[5] =>2.2
- C3P2 =>(9,4)(5,8) => sqrt[(5–9)²+(8–4)²]=> sqrt[32] =>5.7

K-means

Data Points	Centroid (3,3)	Centroid (3,7)	Centroid (9,4)	Cluster
P1(1,3)	2	4.5	8.1	C1
P2(2,2)	1.4	5.1	7.3	C1
P3(5,8)	5.3	2.2	5.7	C2
P4(8,5)	5.4	5.4	5.1	C3
P5(3,9)	6	2	7.9	C2
P6(10,7)	8.1	7	3.2	C3
P7(3,3)	0	4	6.1	C1
P8(9,4)	6.1	6.7	0	C3
P9(3,7)	4	0	6.7	C2

Cluster 1 => P1(1,3) , P2(2,2) , P7(3,3)

Cluster 2 => P3(5,8) , P5(3,9) , P9(3,7)

Cluster 3 => P4(8,5) , P6(10,7) , P8(9,4)

K-means

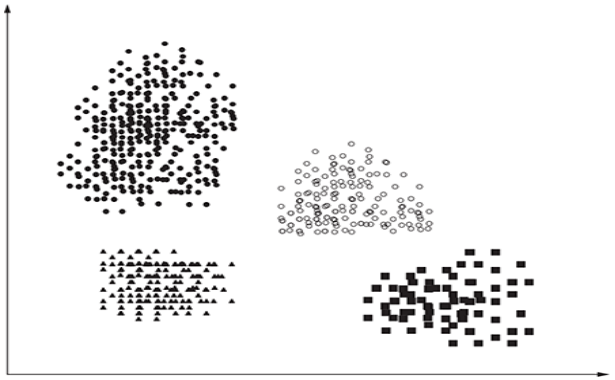
- Re-compute the new clusters and the new cluster center is computed by taking the mean of all the points contained in that particular cluster
  - New center of Cluster 1 => (1+2+3)/3 , (3+2+3)/3 => 2,2.7
  - New center of Cluster 2 => (5+3+3)/3 , (8+9+7)/3 => 3.7,8
  - New center of Cluster 3 => (8+10+9)/3 , (5+7+4)/3 => 9,5.3
- Calcualte the distance between data points and K (C1,C2,C3)
  - C1(2,2.7) , C2(3.7,8) , C3(9,5.3)
  - C1P1 =>(2,2.7)(1,3) => $\text{sqrt}[(1-2)^2+(3-2.7)^2]$ => sqrt[1.1] =>1.0
  - C2P1 =>(3.7,8)(1,3)=> $\text{sqrt}[(1-3.7)^2+(3-8)^2]$ => sqrt[32.29] =>4.5
  - C3P1 =>(9,5.3)(1,3) => $\text{sqrt}[(1-9)^2+(3-5.3)^2]$=> sqrt[69.29] =>8.3

K-means

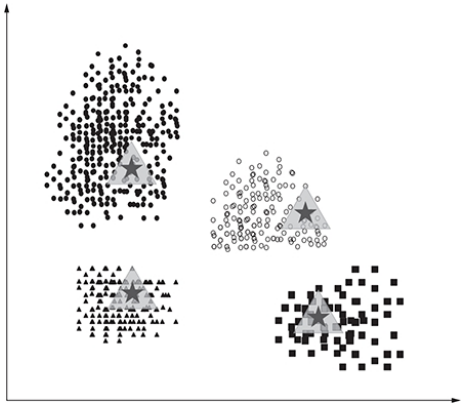
ITERATION - 2

Data Points	Centroid (2,2.7)	Centroid (3.7,8)	Centroid (9,5.3)	Cluster
P1(1,3)	1.0	4.5	8.3	C1
P2(2,2)	0.7	6.2	7.7	C1
P3(5,8)	6.1	1.3	4.8	C2
P4(8,5)	6.4	5.2	1.0	C3
P5(3,9)	6.4	1.2	7.0	C2
P6(10,7)	9.1	6.4	1.9	C3
P7(3,3)	1.0	5.0	6.4	C1
P8(9,4)	7.1	6.6	1.3	C3
P9(3,7)	4.4	1.2	6.2	C2

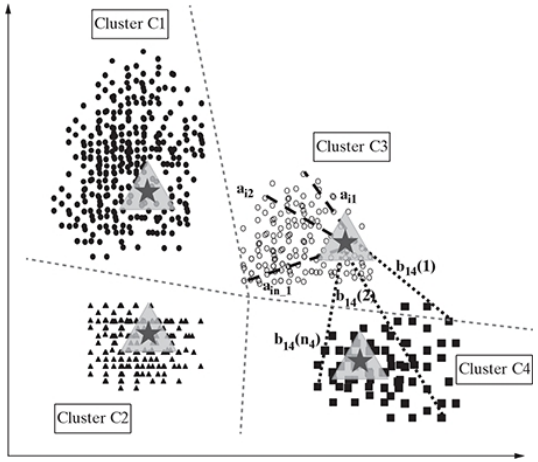
- Cluster 1 => P1(1,3) , P2(2,2) , P7(3,3)
- Cluster 2 => P3(5,8) , P5(3,9) , P9(3,7)
- Cluster 3 => P4(8,5) , P6(10,7) , P8(9,4)
- Center of Cluster 1 => (1+2+3)/3 , (3+2+3)/3 => 2,2.7
- Center of Cluster 2 => (5+3+3)/3 , (8+9+7)/3 => 3.7,8
- Center of Cluster 3 => (8+10+9)/3 , (5+7+4)/3 => 9,5.3



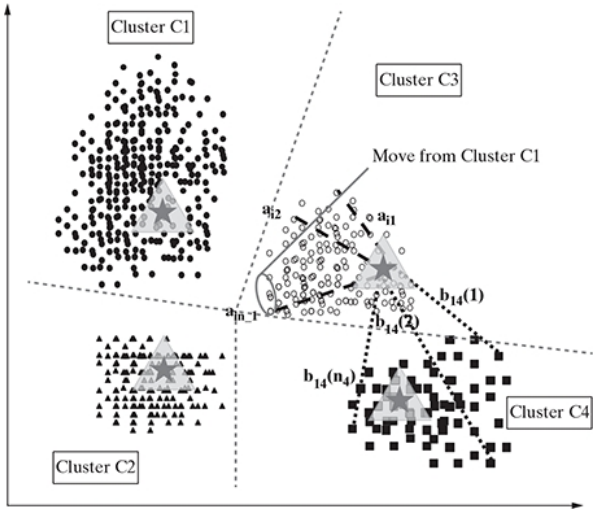
Clustering of data set



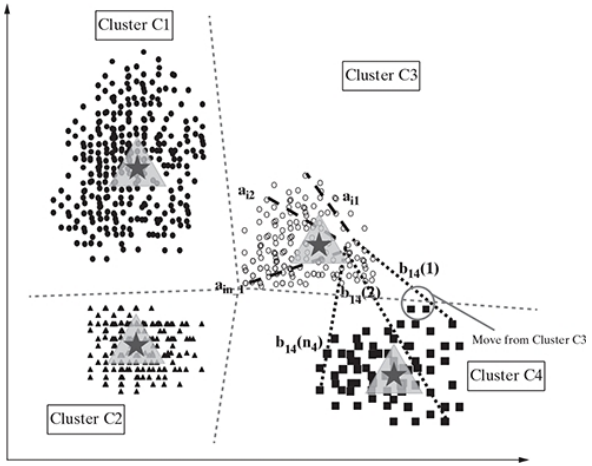
Clustering with initial centroids



Iteration 1



Iteration 2



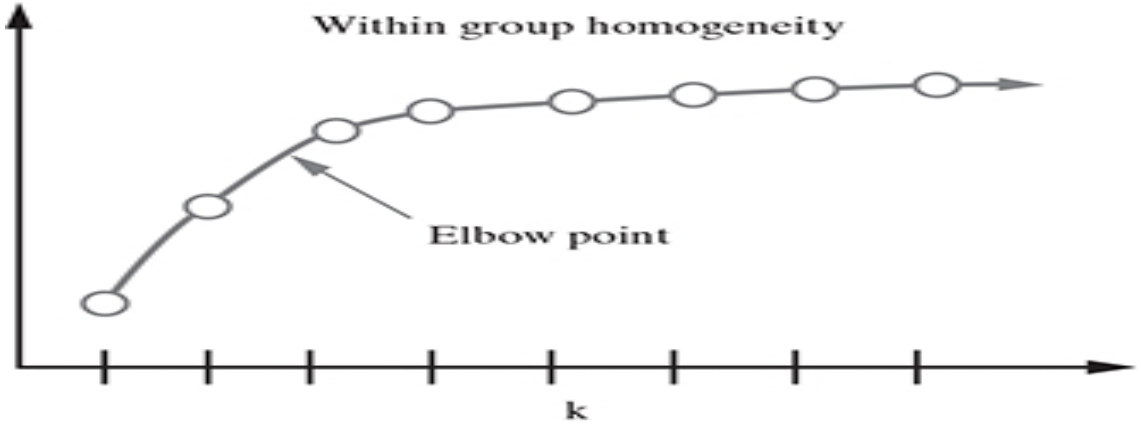
Iteration 3

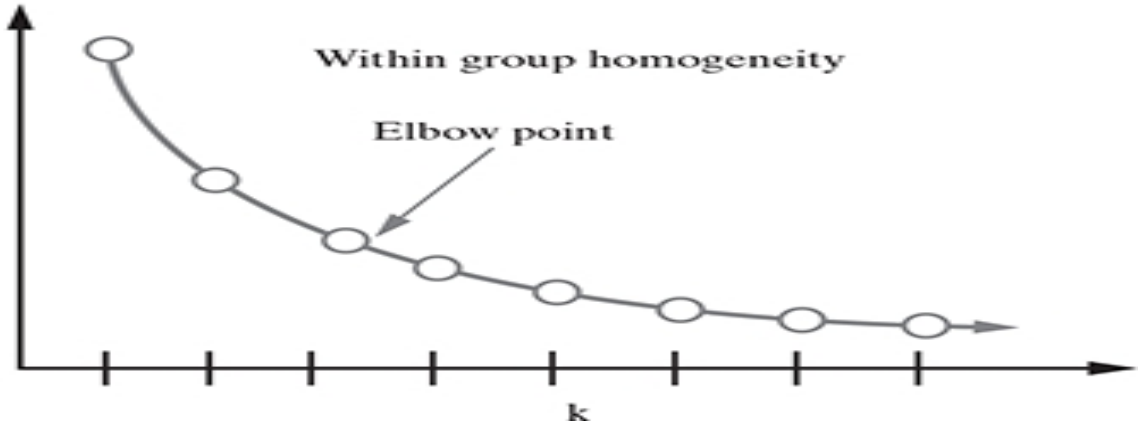
K-means

Strengths	Weaknesses
• The principle used for identifying the clusters is very simple and involves very less complexity of statistical terms • The algorithm is very flexible and thus can be adjusted for most scenarios and complexities • The performance and efficiency are very high and comparable to those of any sophisticated algorithm in term of dividing the data into useful clusters	• The algorithm involves an element of random chance and thus may not find the optimal set of cluster in some cases • The starting point of guessing the number natural clusters within the data requires some experience of the user, so that the final outcome is efficient

Elbow method

- ries to measure the homogeneity or heterogeneity within the cluster and for various values of ‘K’ and helps in arriving at the optimal ‘K’





K – Medoids

- k-means algorithm is sensitive to outliers K – Medoids solves this problem
- Let's consider data points: 1, 2, 3, 5, 9, 10, 11, and 25
- With K = 2, the initial clusters we arrived at are {1, 2, 3, 6} and {9, 10, 11, 25}
  - The mean of the cluster {1, 2, 3, 6} = 12/4 = 3
  - The mean of the cluster {9, 10, 11, 25} = 56/4 = 14
  - the SSE (Sum of Squared Error) within the clusters is

$$(1-3)^2 + (2-3)^2 + (3-3)^2 + (6-3)^2 + (9-14)^2 + (10-14)^2 + (12-14)^2 + (25-14)^2 = 179$$

K – Medoids

- If we compare this with the cluster {1, 2, 3, 6, 9} and {10,11, 25}
  - the mean of the cluster {1, 2, 3, 6, 9} = 21/5 = 4.2
  - the mean of the cluster {10, 11, 25} = 47/3 = 15.67
  - the SSE (Sum of Squared Error) within the clusters is

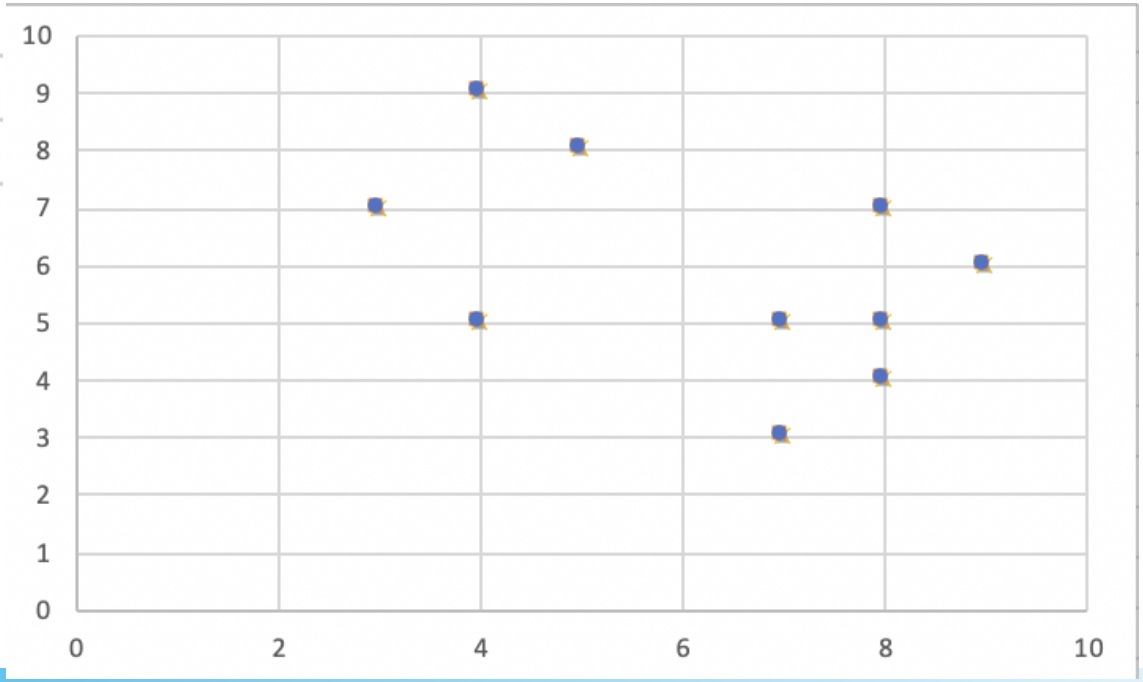
$$(1 - 4.2)^2 + (2 - 4.2)^2 + (3 - 4.2)^2 + (6 - 4.2)^2 + (9 - 4.2)^2$$
$$+ (10 - 15.67)^2 + (12 - 15.67)^2 + (25 - 15.67)^2 = 113.84$$

K – Medoids

- k-medoids method groups n objects in k clusters by minimizing the SSE
- k-medoids is less influenced by the outliers in the data
- One of the practical implementation of the k-medoids principle is the Partitioning Around Medoids (PAM) algorithm

K – Medoids

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	8	5
5	5	8
6	7	3
7	8	4
8	7	5
9	4	5



K – Medoids

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	2
1	3	7	3	7
2	4	9	4	8
3	9	6	6	2
4	**8**	**5**	-	-
5	5	8	4	6
6	7	3	5	3
7	8	4	5	1
8	7	5	3	1
9	**4**	**5**	-	-

Manhattan distance = |X1-X2| + |Y1-Y2|

K – Medoids

	X	Y	Dist. from C1	Dist. From C2	Cluster
0	8	7	6	2	C2
1	3	7	3	7	C1
2	4	9	4	8	C1
3	9	6	6	2	C2
4	**8**	**5**	-	-	-
5	5	8	4	6	C1
6	7	3	5	3	C2
7	8	4	5	1	C2
8	7	5	3	1	C2
9	**4**	**5**	-	-	-

Manhattan distance = |X1-X2| + |Y1-Y2|

(3 + 4 + 4) + (2 + 2 + 3 + 1 + 1) = 20

K – Medoids

	X	Y	Dissimilarity from C1	Dissimilarity from C2
0	8	7	6	3
1	3	7	3	8
2	4	9	4	9
3	9	6	6	3
4	8	5	4	1
5	5	8	4	7
6	7	3	5	2
7	**8**	**4**	-	-
8	7	5	3	2
9	**4**	**5**	-	-

K – Medoids

	X	Y	Dist. from C1	Dist. From C2	Cluster
0	8	7	6	3	C2
1	3	7	3	8	C1
2	4	9	4	9	C1
3	9	6	6	3	C2
4	8	5	4	1	C2
5	5	8	4	7	C1
6	7	3	5	2	C2
7	**8**	**4**	-	-	-
8	7	5	3	2	C2
9	**4**	**5**	-	-	-

Manhattan distance = |X1-X2| + |Y1-Y2|

(3 + 4 + 4) + (3 + 3 + 1 + 2 + 2) = 22

New Cost (22) > Previous Cost (20), Undo Swap

Partitioning Around Medoids (PAM)

Step 1: Randomly choose k points in the data set as the initial representative points

loop

Step 2: Assign each of the remaining points to the cluster which has the nearest representative point

Step 3: Randomly select a non-representative point o r in each cluster

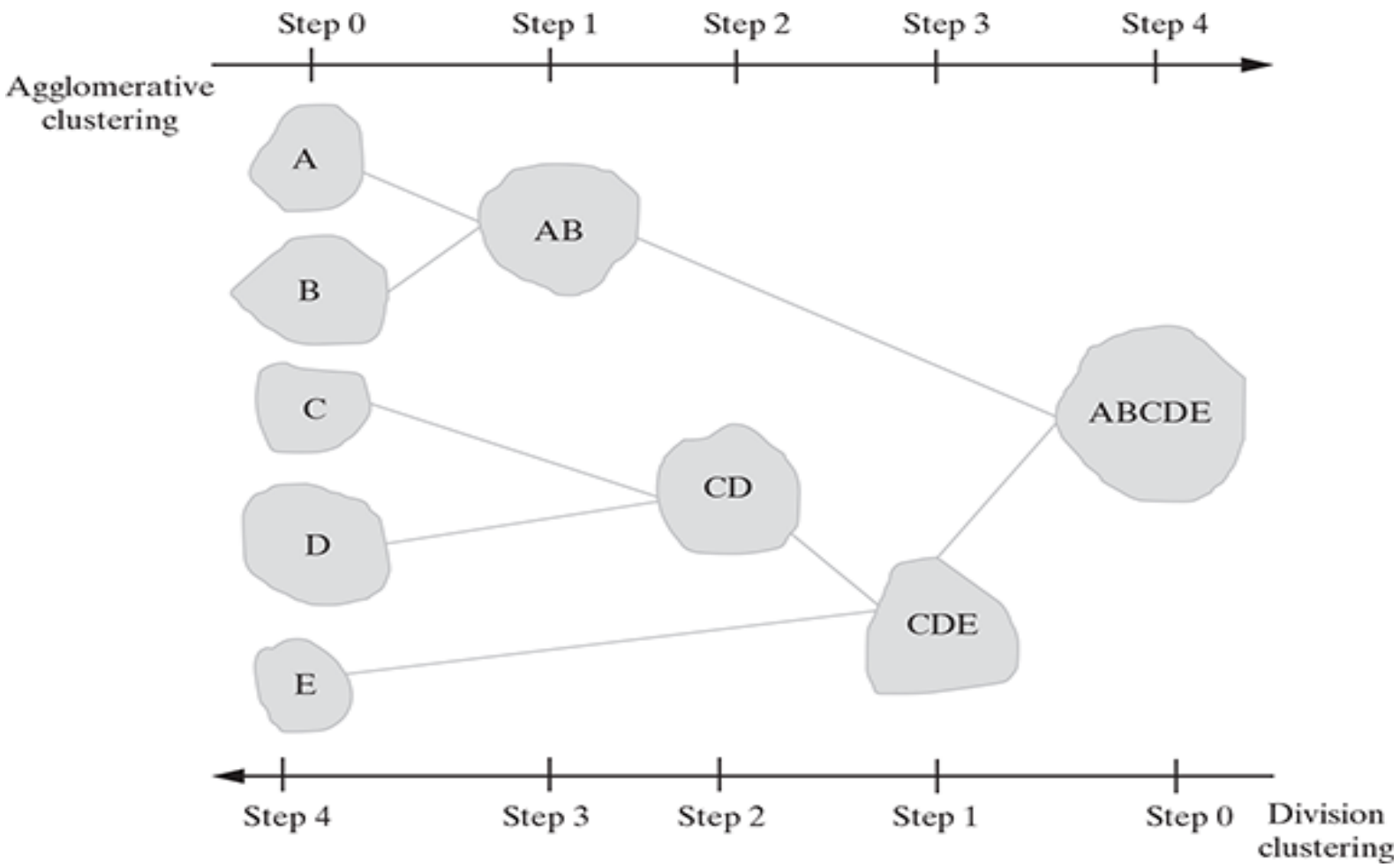
Step 4: Swap the representative point o j with o r and compute the new SSE after swapping

Step 5: If SSEnew < SSEold, then swap o j with o r to form the new set of k representative objects;

Step 6: Refine the k clusters on the basis of the nearest representative point. Logic continues until there is no change

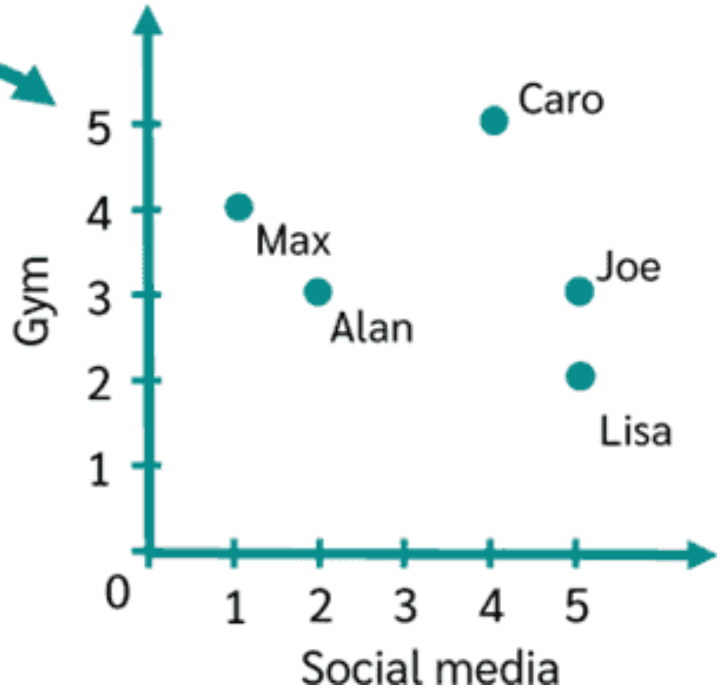
end loop

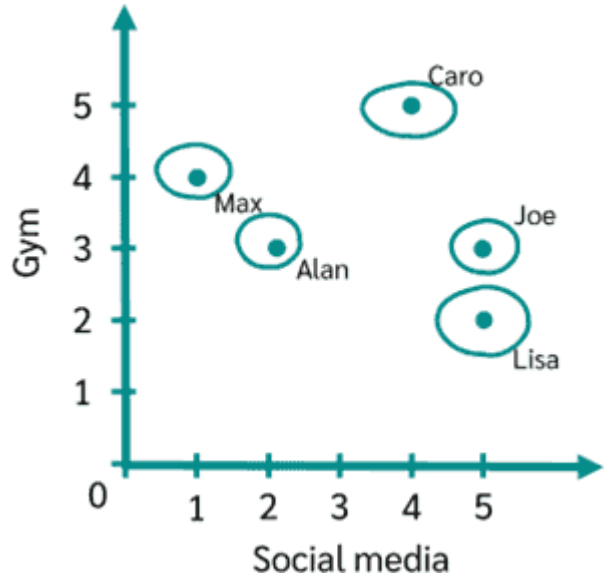
Hierarchical Clustering



Hierarchical Clustering

	Social media	Gym
Alan	2	3
Lisa	5	2
Joe	5	3
Max	1	4
Cora	4	5





Hierarchical Clustering

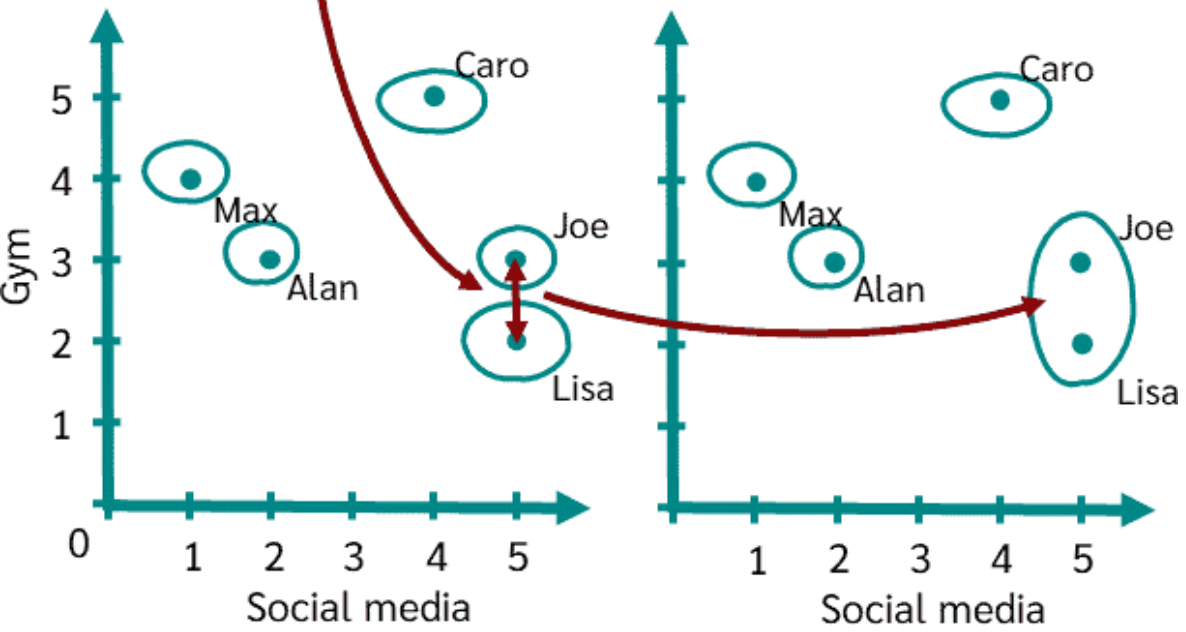
$$d = \sqrt{(5-2)^2+(2-3)^2} = 3.16$$

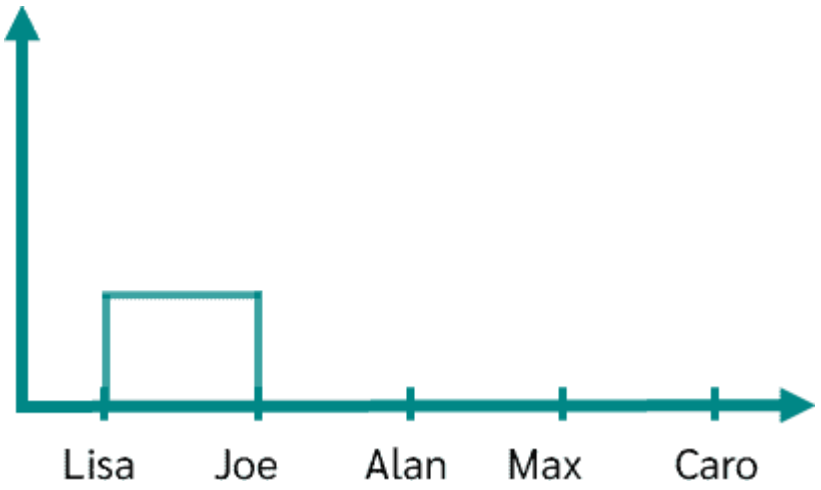
	Social media	Gym
Alan	2	3
Lisa	5	2
Joe	5	3
Max	1	4
Caro	4	5

	Alan	Lisa	Joe	Max	Caro
Alan	0				
Lisa	3.16	0			
Joe	3.00	1.00	0		
Max	1.41	4.47	4.12	0	
Caro	2.83	3.16	2.24	3.16	0

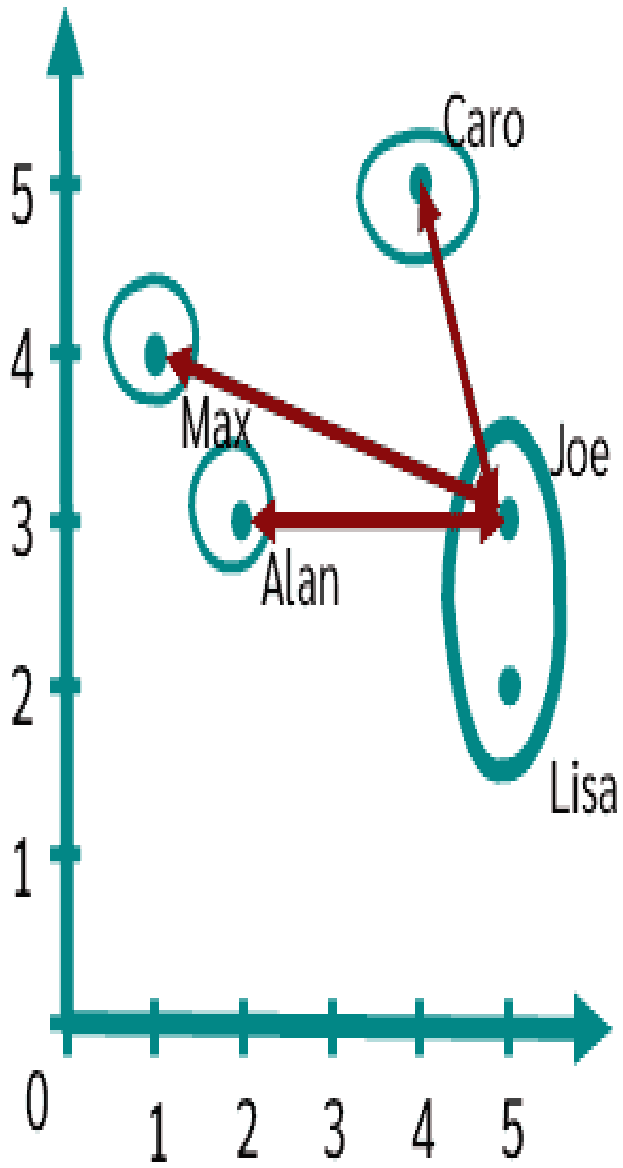
Hierarchical Clustering

	Alan	Lisa	Joe	Max	Caro
Alan	0				
Lisa	3.16	0			
Joe	3.00	1.00	0		
Max	1.41	4.47	4.12	0	
Caro	2.83	3.16	2.24	3.16	0





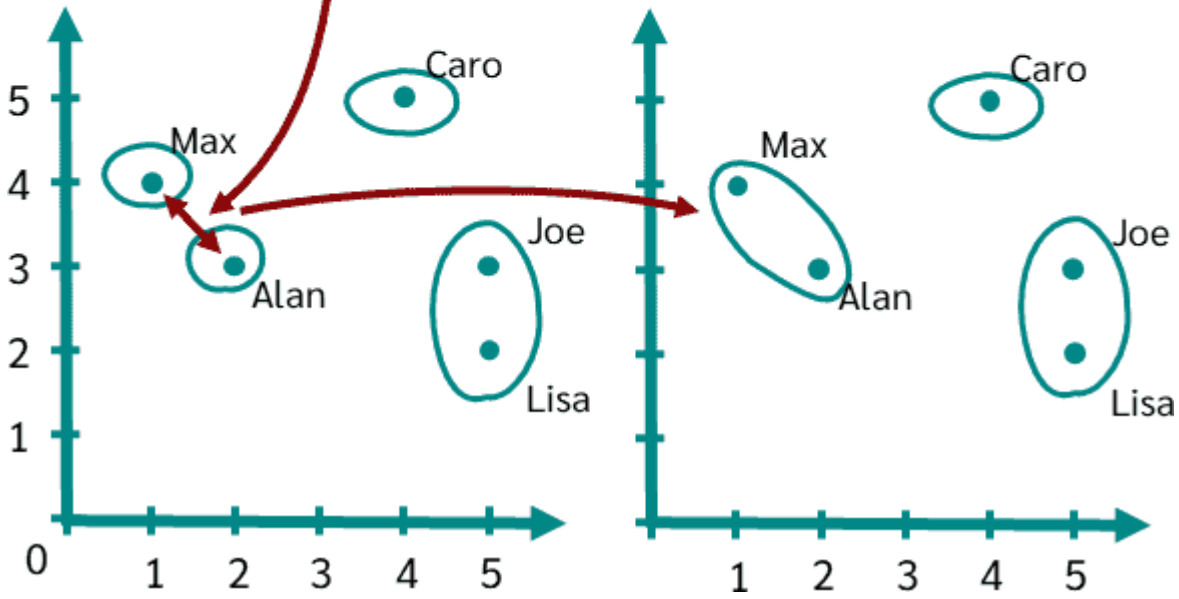
Hierarchical Clustering

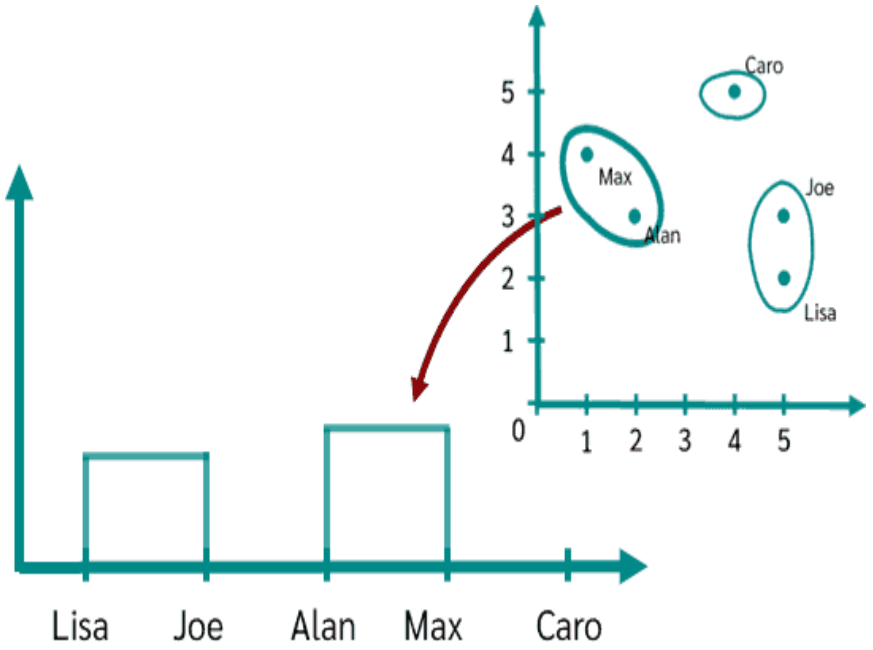


	Alan	Lisa, Joe	Max	Caro
Alan	0			
Lisa, Joe	3.00	0		
Max	1.41	4.12	0	
Cora	2.83	2.24	3.16	0

Hierarchical Clustering

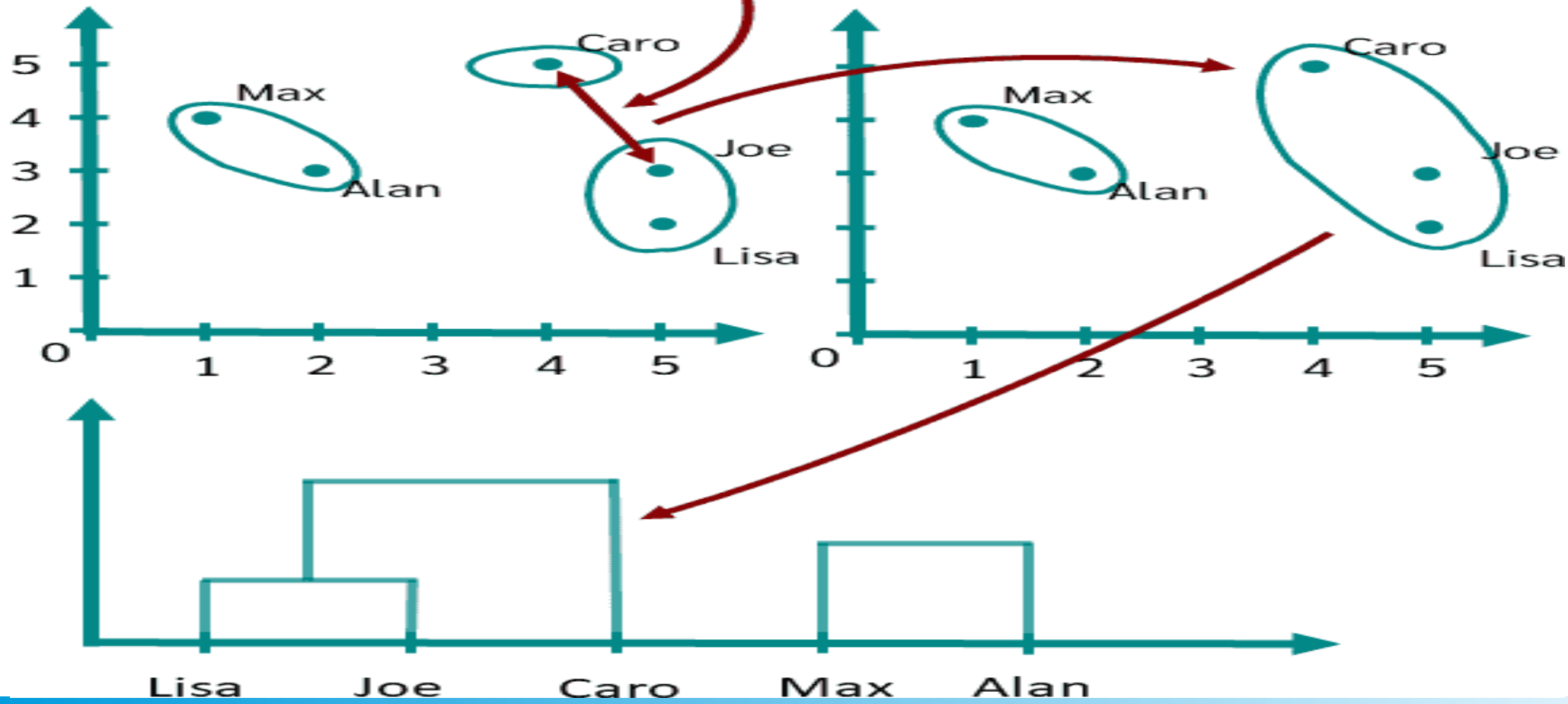
	Alan	Lisa, Joe	Max	Caro
Alan	0			
Lisa, Joe	3.00	0		
Max	1.41	4.12	0	
Caro	2.83	2.24	3.16	0



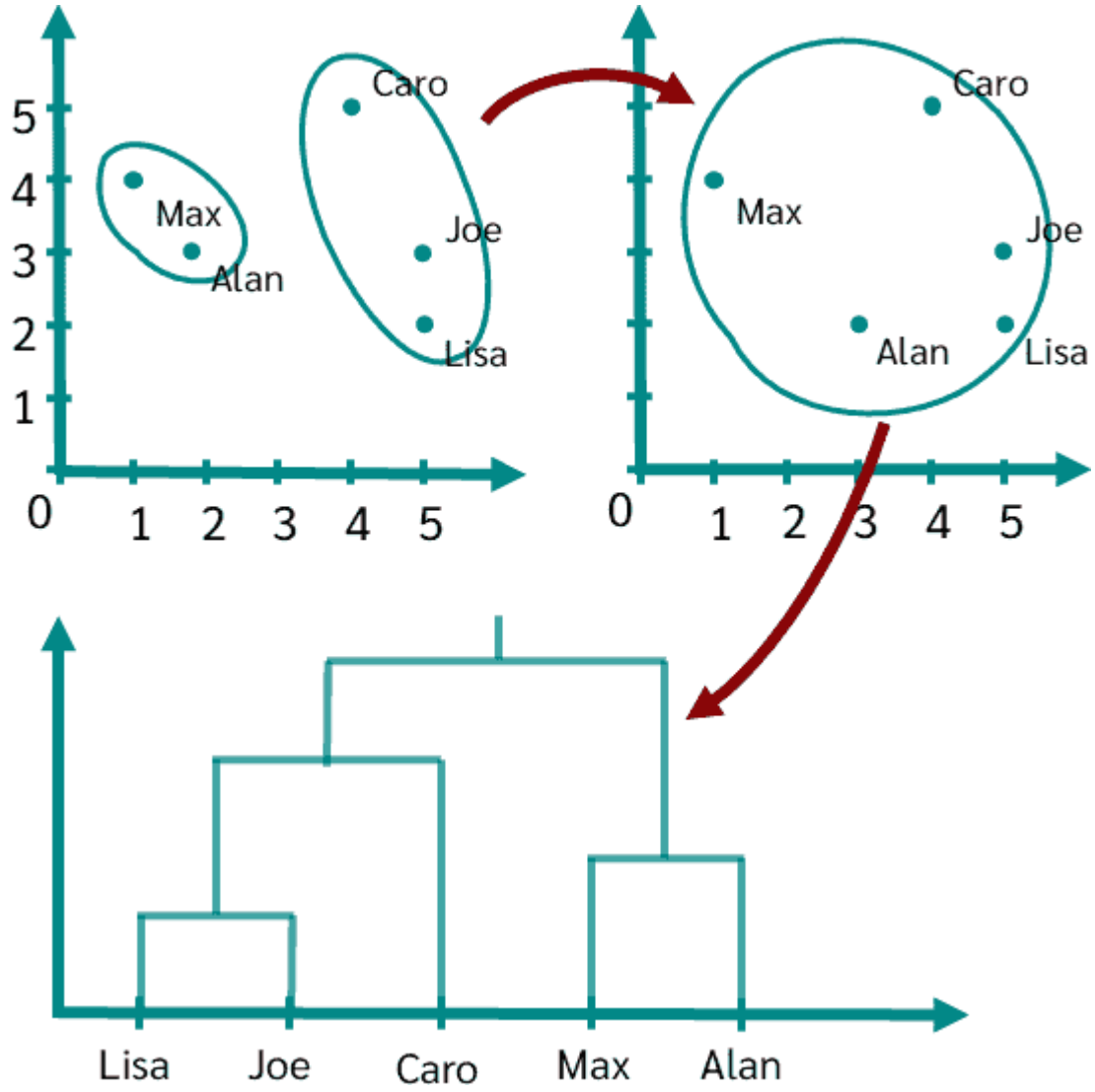


Hierarchical Clustering

	Lisa, Joe	Max, Alan	Caro
Lisa, Joe	0		
Max, Alan	3,00	0	
Caro	2,24	2,83	0



Hierarchical Clustering



FINDING PATTERN USING ASSOCIATION RULE

- **Association Analysis:** methodology to identify the interesting relationships hidden in large data sets
- Market Basket Analysis

- Definition of common terms
  - **Itemset:** A collection of zero or more items. {Bread, Milk, Egg} (three – itemset)

Transaction Number	Purchased Items
1	{Bread, Milk, Egg, Butter, Salt, Apple}
2	{Bread, Milk, Egg, Apple}
3	{Bread, Milk, Butter, Apple}
4	{Milk, Egg, Butter, Apple}
5	{Bread, Egg, Salt}
6	{Bread, Milk, Egg, Apple}

- Definition of common terms
  - **Support count:** the number of transactions in which a particular itemset is present $\sigma(X) = |\{t_i | X \subseteq t_i, t_i \varepsilon T\}|$
  - {Bread, Milk, Egg} = support count = 3

Association Rule

- The result of the market basket analysis is expressed as a set of association rules
- specify patterns of relationships among items
  - {Bread, Milk} →{Egg}
- Support denotes how often a rule is applicable to a given data set
  - low support may indicate that the rule has occurred by chance
- Confidence indicates how often the items in Y appear in transactions that contain X in a total transaction of N
  - measurement for reliability of the inference of a rule

$$\text{Support}, s(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{N}$$

$$\text{Confidence}, c(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{\sigma(X)}$$

Association Rule

$$\text{Confidence } c(\{\text{Bread, Milk}\} \rightarrow \{\text{Egg}\}) = \frac{\text{support count of } \{\text{Bread, Milk, Egg}\}}{\text{support count of } \{\text{Bread, Milk}\}}$$
$$= \frac{3}{4}$$
$$= 0.75$$

$$\text{Confidence } c(\{\text{Bread, Milk}\} \rightarrow \{\text{Egg}\}) = \frac{\text{support count of } \{\text{Bread, Milk, Egg}\}}{\text{support count of } \{\text{Bread, Milk}\}}$$
$$= \frac{3}{4}$$
$$= 0.75$$

$$\text{Support, } s(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{N}$$

$$\text{Confidence, } c(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{\sigma(X)}$$

$$\{\text{Egg}\} \rightarrow \{\text{Bread, Milk}\} = \frac{3}{5} = 0.6.$$

Transaction Number	Purchased Items
1	{Bread, Milk, Egg, Butter, Salt, Apple}
2	{Bread, Milk, Egg, Apple}
3	{Bread, Milk, Butter, Apple}
4	{Milk, Egg, Butter, Apple}
5	{Bread, Egg, Salt}
6	{Bread, Milk, Egg, Apple}

Association Rule

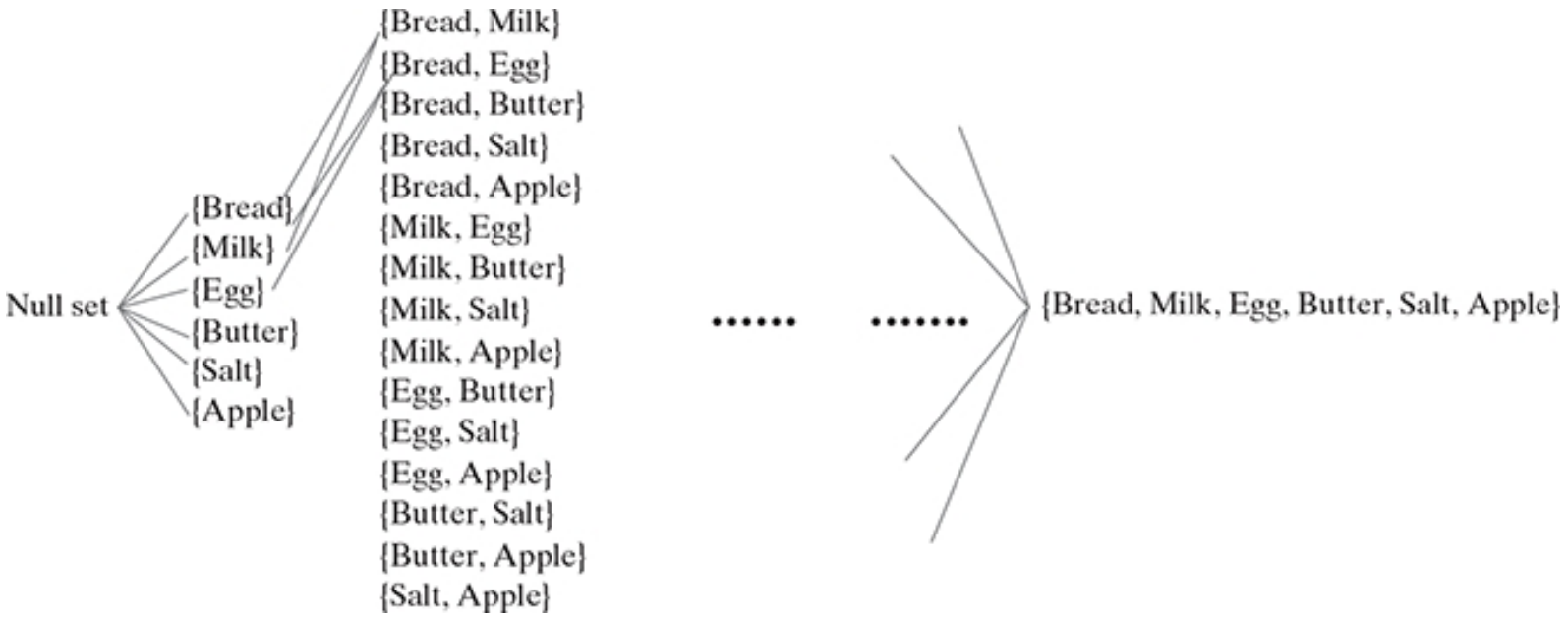
- developed in the context of Big Data and data science and are not used for prediction
- Used for unsupervised knowledge discovery in large databases, unlike the classification and numeric prediction algorithms

Association Rule

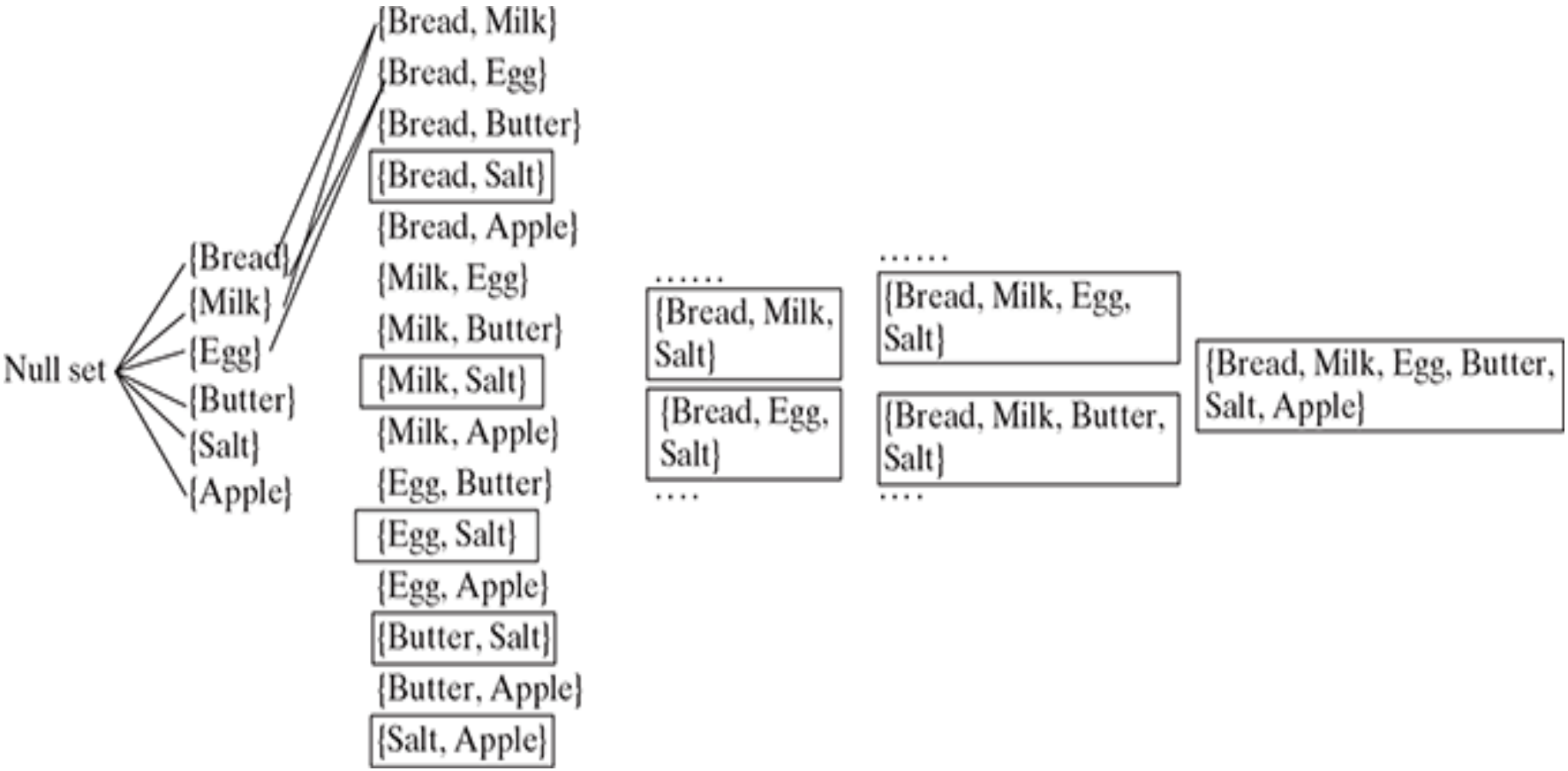
- Apriori Algorithm
  - most widely used algorithm to reduce the number of itemsets to search for the association rule
  - the algorithm utilizes a simple prior belief (i.e. a priori) about the properties of frequent itemsets
  - If an itemset is frequent, then all of its subsets must also be frequent
  - If an itemset is frequent, then all the supersets must be frequent too

Apriori Algorithm

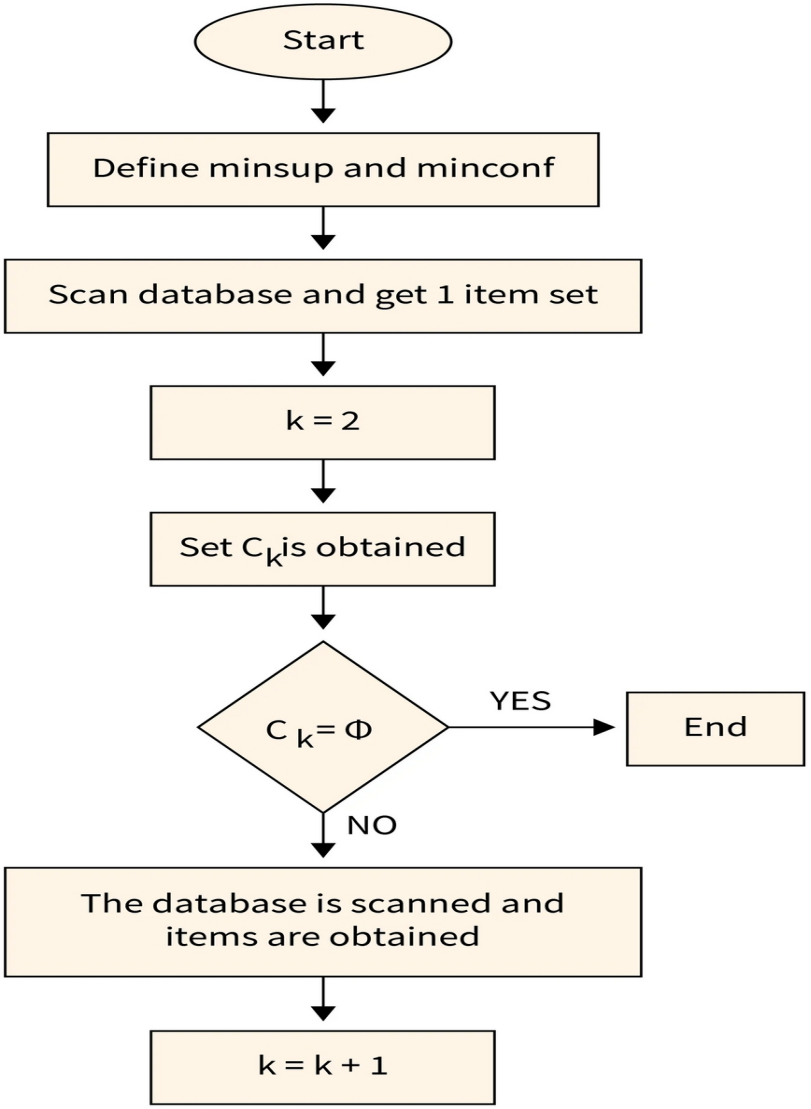
- 2^6 ways to create baskets or itemsets (including the null itemset)



Apriori Algorithm



Apriori Algorithm



Apriori Algorithm

Transaction ID	Items
T1	Hot Dogs, Buns, Ketchup
T2	Hot Dogs, Buns
T3	Hot Dogs, Coke, Chips
T4	Chips, Coke
T5	Chips, Ketchup
T6	Hot Dogs, Coke, Chips

Apriori Algorithm

Transaction ID	Items
T1	Hot Dogs, Buns, Ketchup
T2	Hot Dogs, Buns
T3	Hot Dogs, Coke, Chips
T4	Chips, Coke
T5	Chips, Ketchup
T6	Hot Dogs, Coke, Chips

Item set	Sup-count
Hot Dogs	4
Buns	2
Ketchup	2
Coke	3
Chips	4

Item set	Sup-count
Hot Dogs	4
Buns	2
Ketchup	2
Coke	3
Chips	4

- support threshold (s) = 33.33%)
- minimum confident threshold (c = 60%)

Item set	Sup-count
Hot Dogs, Buns	2
Hot Dogs, Ketchup	1
Hot Dogs, Coke	2
Hot Dogs, Chips	2
Buns, Ketchup	1
Buns, Coke	0
Buns, Chips	0
Ketchup, Coke	0
Ketchup, Chips	1
Coke, Chips	3

Item set	Sup-count
Hot Dogs, Buns	2
Hot Dogs, Coke	2
Hot Dogs, Chips	2
Coke, Chips	3

Item set	Sup-count
Hot Dogs, Buns, Coke	0
Hot Dogs, Buns, Chips	0
Hot Dogs, Coke, Chips	2

Item set	Sup-count
Hot Dogs, Coke, Chips	2

Apriori Algorithm

- [Hot Dogs ∧ Coke]=>[Chips]
  - //confidence = sup(Hot Dogs ∧ Coke ∧ Chips)/sup(Hot Dogs ∧ Coke) = 2/2*100=100% //Selected
- [Hot Dogs ∧ Chips]=>[Coke]
  - //confidence = sup(Hot Dogs ∧ Coke ∧ Chips)/sup(Hot Dogs ∧ Chips) = 2/2*100=100% //Selected
- [Coke ∧ Chips]=>[Hot Dogs]
  - //confidence = sup(Hot Dogs ∧ Coke ∧ Chips)/sup(Coke ∧ Chips) = 2/3*100=66.67% //Selected
- [Hot Dogs]=>[Coke ∧ Chips]
  - //confidence = sup(Hot Dogs ∧ Coke ∧ Chips)/sup(Hot Dogs) = 2/4*100=50% //Rejected
- [Coke]=>[Hot Dogs ∧ Chips]
  - //confidence = sup(Hot Dogs ∧ Coke ∧ Chips)/sup(Coke) = 2/3*100=66.67% //Selected
- [Chips]=>[Hot Dogs ∧ Coke]
  - //confidence = sup(Hot Dogs ∧ Coke ∧ Chips)/sup(Chips) = 2/4*100=50% //Rejected

Apriori Algorithm

Strengths	Weaknesses
• Provides reasonable accuracy while working with very large amounts of transactional data • Discovers rules that are easy to understand • Provides valuable insight into the unexpected knowledge in data sets, which is a key aspect of learning	• Not very accurate in the case the data set is small as the smaller occurrences of itemsets may not be due to chance • Some effort is involved to separate the insight from the common sense • In the case of widespread presence of random patterns, the principle can draw spurious conclusions

Reference

- Machine Learning by Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das published by Pearson