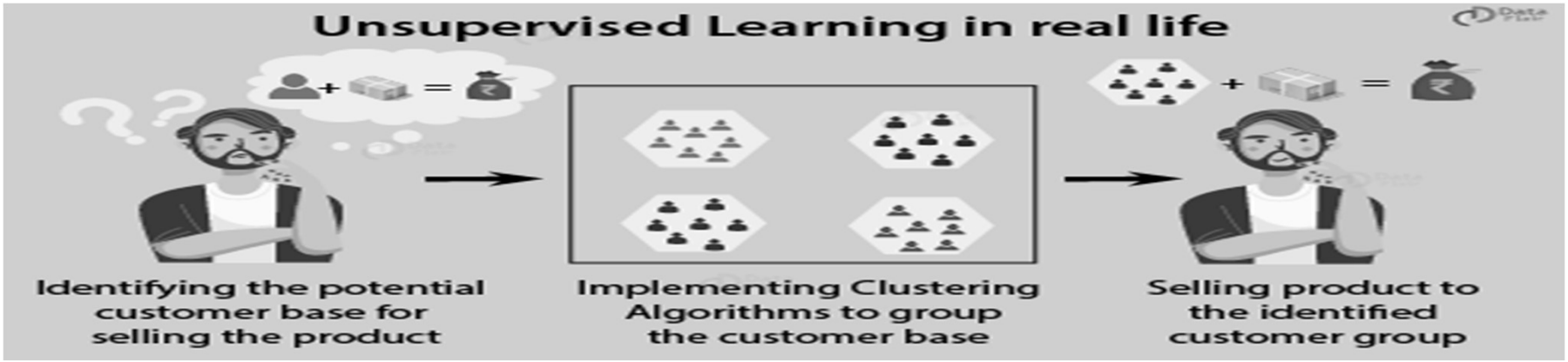
UNIT – 5 Unsupervised Learning

By : Prof. Nootan Padia, Marwadi University, Rajkot

Unsupervised learning

- In unsupervised learning, there is no labeled training data to learn from and no prediction to be made.
- In unsupervised learning, the objective is to take a dataset as input and try to find natural groupings or patterns within the data elements or records.



- Therefore, unsupervised learning is often termed as descriptive model and the process of unsupervised learning is referred as pattern discovery or knowledge discovery.
- One critical application of unsupervised learning is customer segmentation.
- Clustering is the main type of unsupervised learning.
- It intends to group or organize similar objects together.
- Other than clustering of data and getting a summarized view from it, one more variant of unsupervised learning is association analysis.
- It means – “What item goes with what”
- It means that there is a strong association of the event 'purchase of item ‘A' with the event 'purchase of item ‘B’ or 'purchase of item ‘C’.

SUPERVISED	UNSUPERVISED
This type of learning is used when you know how to classify a given data, or in other words classes or labels are available.	This type of learning is used when there is no idea about the class or label of a particular data. The model has to find pattern in the data.
Labelled training data is needed. Model is built based on training data.	Any unknown and unlabelled data set is given to the model as input and records are grouped.
The model performance can be evaluated based on how many misclassifications have been done based on a comparison between predicted and actual values.	Difficult to measure whether the model did something useful or interesting. Homogeneity of records grouped together is the only measure.
There are two types of supervised learning problems — classification and regression.	There are two types of unsupervised learning problems — clustering and association.
Simplest one to understand.	More difficult to understand and implement than supervised learning.
Standard algorithms include • Naïve Bayes • *k*-nearest neighbour (kNN) • Decision tree • Linear regression • Logistic regression • Support Vector Machine SVM), etc.	Standard algorithms are • *k*-means • Principal Component Analysis (PCA) • Self-organizing map (SOM) • Apriori algorithm • DBSCAN etc.
Practical applications include • Handwriting recognition • Stock market prediction • Disease prediction • Fraud detection, etc.	Practical applications include • Market basket analysis • Recommender systems • Customer segmentation, etc.

Application of Unsupervised Learning

- Segmentation of target consumer populations by an advertisement consulting agency on the basis of few dimensions such as demography, financial data, purchasing habits, etc. so that the advertisers can reach their target consumers efficiently.
- Anomaly or fraud detection in the banking sector by identifying the pattern of loan defaulters.
- Image processing and image segmentation such as face recognition, expression identification, etc.
- Grouping of important characteristics in genes to identify important influencers of new area of genetics

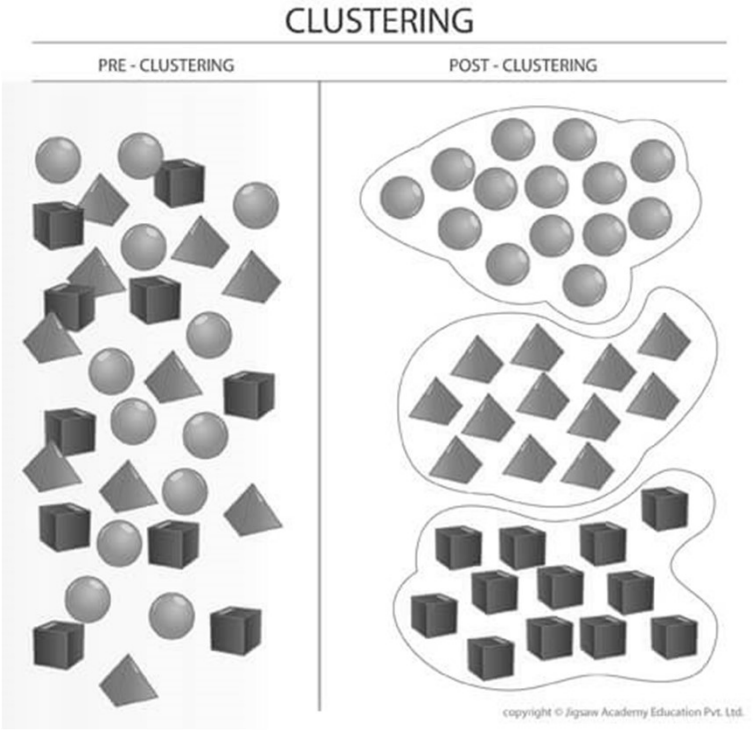
- Utilization by data scientists to reduce the dimensionalities in sample data to simplify modelling.
- Document clustering and identifying potential labelling options.

Today, unsupervised learning is used in many areas involving Artificial Intelligence(AI) and Machine Learning(ML).

Chat bots, self – driven cars, and many more recent innovations are results of the combination of unsupervised learning and supervised learning.

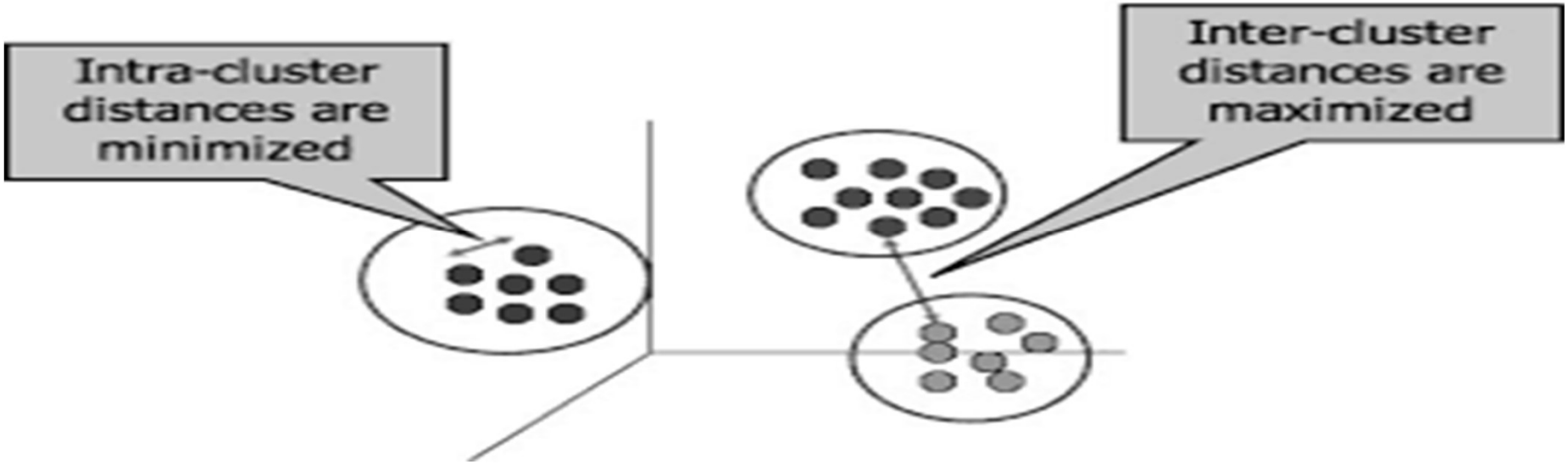
Clustering

Clustering



- Cluster is a group of objects that belongs to the same class.
- In other words, similar objects are grouped in one cluster and dissimilar objects are grouped in another cluster.
- Cluster analysis is used to form groups or clusters of similar records based on several measurements made on these records.
- Cluster analysis is also known as clustering or data segmentation.

- The effectiveness of clustering depends on how similar or related the objects within the group or how different or unrelated the objects in different groups are from each other.
- Measuring distance between two records (Intra – cluster distances)
- Measuring distance between two clusters (Inter – cluster distances)
-



Areas where cluster analysis is used effectively

- Text data mining : this includes tasks such as text categorization, text clustering, document summarization, concept extraction, sentiment analysis, and entity relation modelling.
- Customer segmentation : creating clusters of customers on the basis of parameters such as demographics, financial conditions, buying habits, etc. which can be used by retailers and advertisers to promote their products in the correct segment.
- Anomaly checking : checking of anomalous behaviours such as fraudulent bank transaction, unauthorized computer intrusion, suspicious movements on a radar scanner, etc.
- Data Mining : simplify the data mining task by grouping a large no. of features from an extremely large data set to make the analysis manageable.

Clustering as a machine learning task

- The primary driver of clustering knowledge is discovery rather than prediction.
- Why? – Because we may not even know what we are looking for before starting the clustering analysis.
- Therefore clustering is defined as an unsupervised learning task that automatically divides the data into clusters or groups of similar items.
- Clustering is somewhat different from the classification and numeric prediction which are part of supervised learning. – Unlabelled objects are given a cluster label which is inferred entirely from the relationship of attributes within the data.

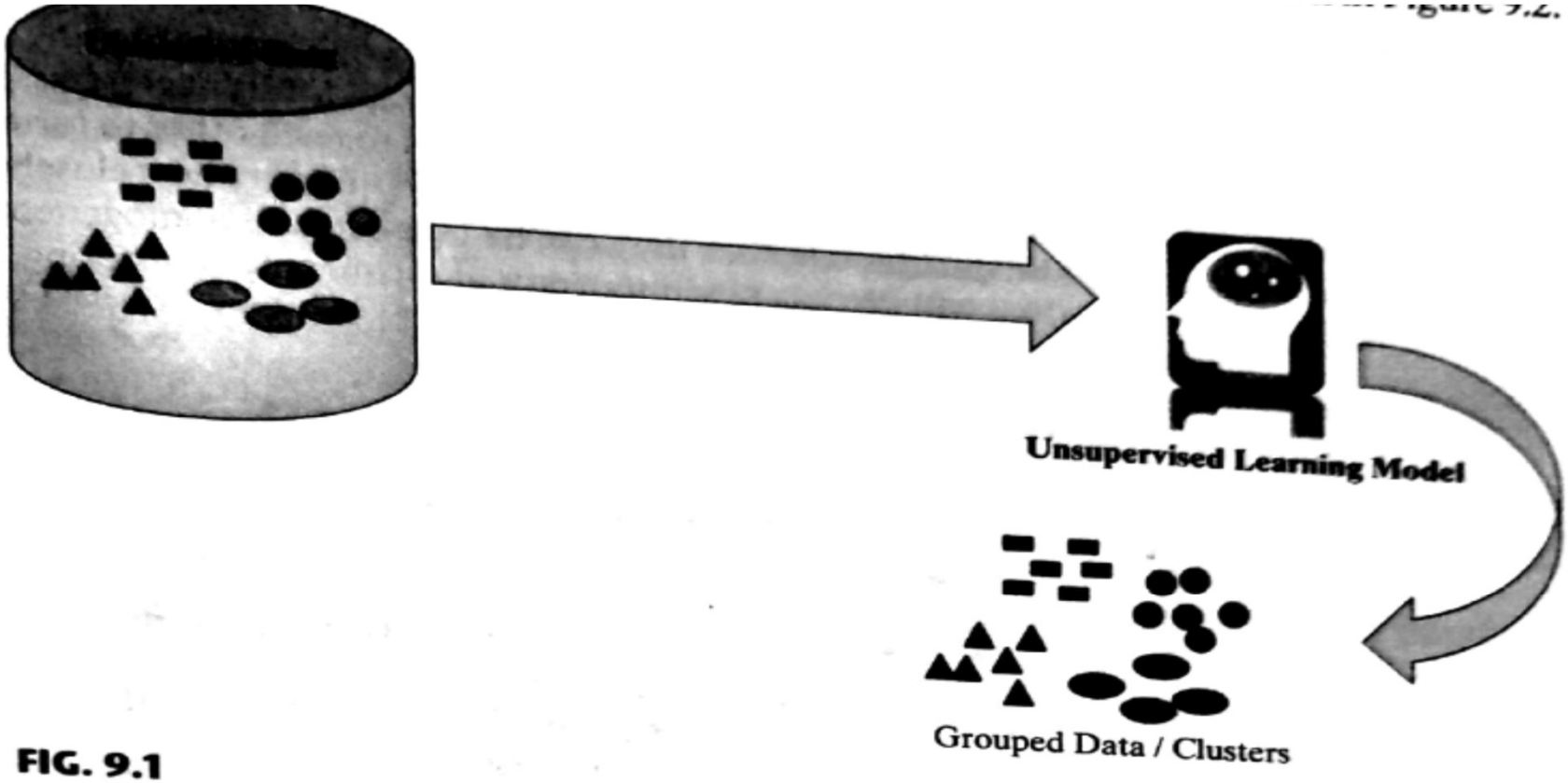


FIG. 9.1
Unsupervised learning – clustering

Example :

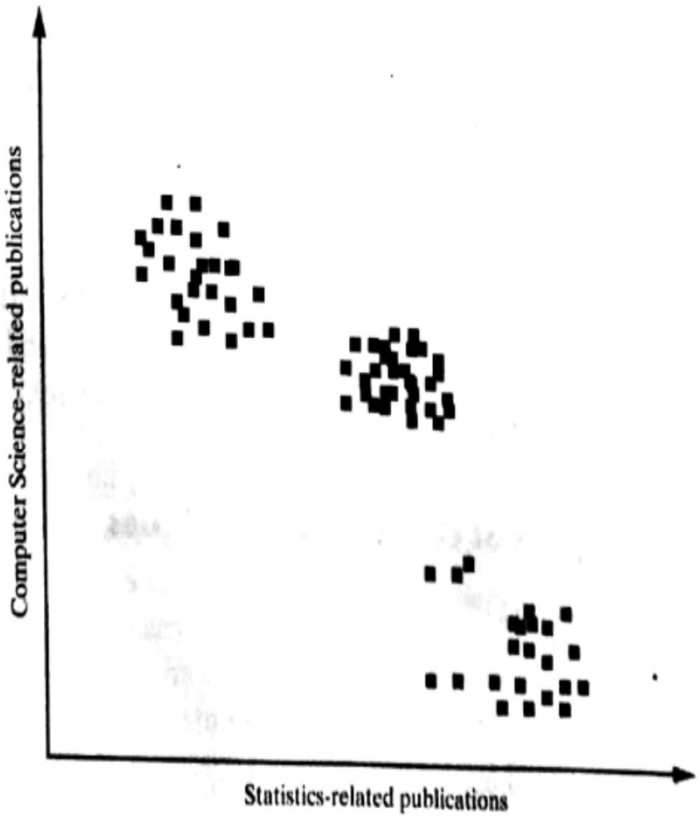


FIG. 9.2
Data set for the conference attendees

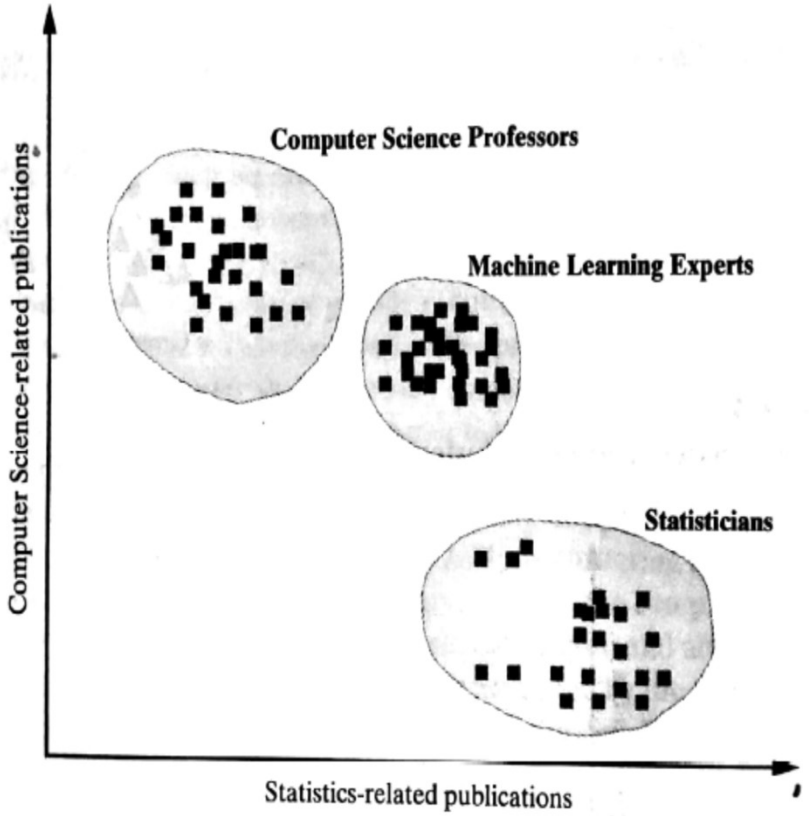


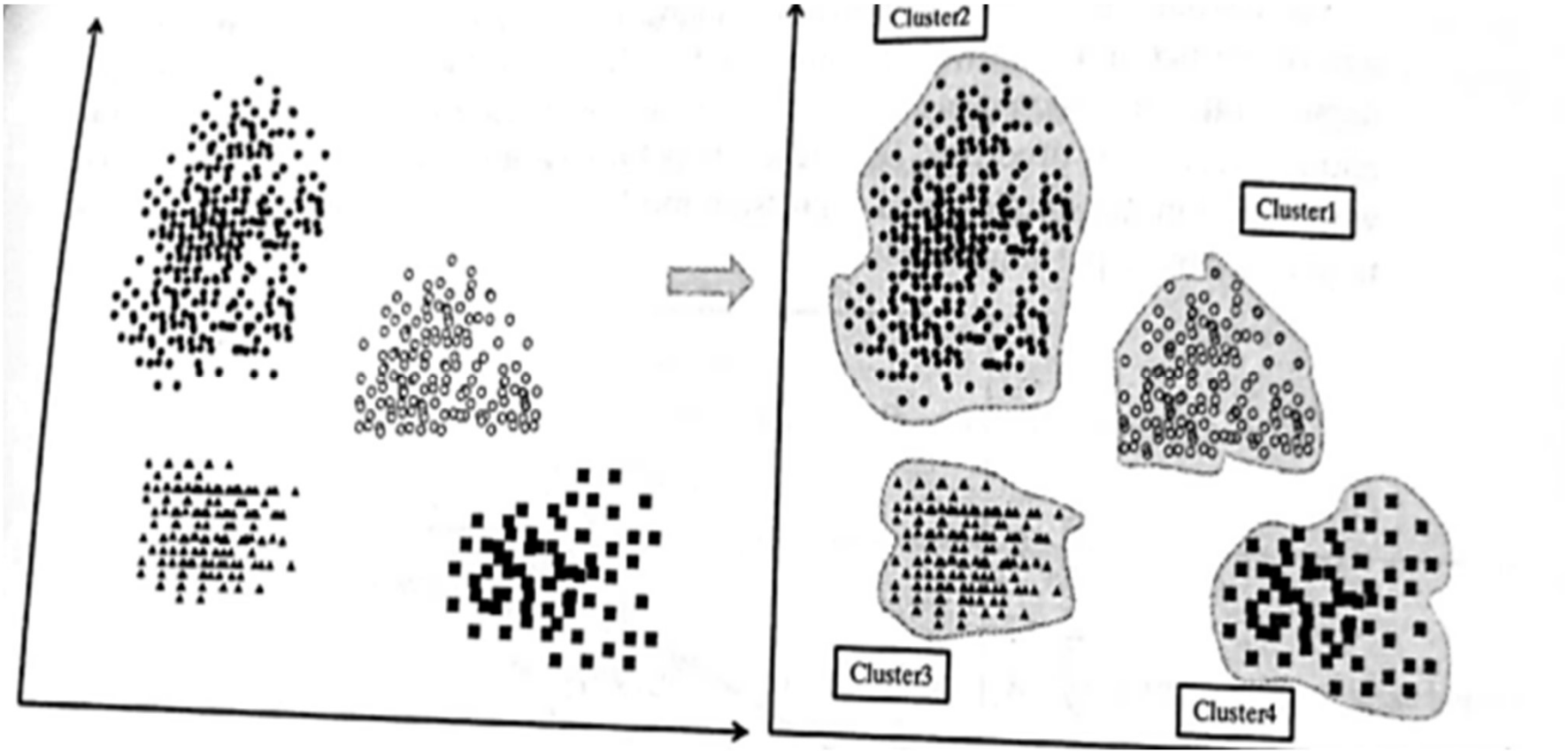
FIG. 9.3
Clusters for the conference attendees

Different types of clustering techniques

- Their approach towards creating the clusters, way to measure the quality of the clusters, and applicability are different.
- Three techniques :
  - Partitioning methods
  - Hierarchical methods
  - Density – based methods

Method	Characteristics
Partitioning Methods (Non – hierarchical)	• 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 ae separated by low – density regions • May filter out outliers

- Before clustering and After clustering



Partitioning Method

- Two most important algorithms for partitioning – based clustering are :
  - K – means
  - K - medoid

K – means – A centroid – based technique

- This is one of the oldest and most popularly used algorithm for clustering.
- The principle of the k – means algorithm is to assign each of the ‘n’ data points to one of the K clusters where ‘K’ is a user – defined parameter as the no. of clusters desired.
- The objective is to maximize the homogeneity within the clusters and also to maximize the differences between the clusters.
- The homogeneity and differences are measured in terms of the distance between the objects or points in the data set.

- 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 data points
    - Step – 6 : Repeat step – 2 to 5 to refine until centroids do not change
  - End loop

- Choosing the initial centroids :
  - Another key step for the K – Means algorithm is to choose the initial centroids properly.
  - One common practice is to choose random points in the data space on the basis of the no. of cluster requirement and refine the points as we move into the iterations.
  - But this often leads to higher squared error in the final clustering, thus resulting in sub – optimal clustering solution.
  - The assumption for selecting random centroids is that multiple subsequent runs will minimize the SSE and identify the optimal clusters.
  - But this is often not true on the basis of the spread of the data set and the no. of clusters sought.
  - So, one effective approach is to employ the hierarchical clustering technique on sample points from the data set and then arrive at sample K clusters.
  - The centroids of these initial K clusters are used as the initial centroids.
  - This approach is practical when the data set has small no. of points and K is relatively small.

- Recomputing cluster centroids :
  - The distance of the data point from its nearest centroid can also be calculated to minimize the distances to arrive at the refined centroid.
  - The Euclidean distance between two data points is measured as follows :

$$\text{dist}(x, y) = \sqrt{\sum_{1}^{n} (x_i - y_i)^2}$$

  - The measure of quality of clustering uses the SSE technique.

$$\text{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} \text{dist}\,(c_i, x)^2$$

  - Where dist() calculates the Euclidean distance between the centroid ci of the cluster Ci and the data points x in the cluster.
  - The summation of such distances over all the ‘K’ clusters gives the total sum of squared error.

- Choosing appropriate number of clusters :
  - One of the most important success factors in arriving the correct clustering is to start with the correct no. of cluster assumptions.
  - Different numbers of starting cluster lead to completely different types of data split.
  - It will always help if we have some prior knowledge about the no. of clusters and we start our k – means algorithm with that prior knowledge.
  - For e.g. , if we are clustering the data of the students of a university, it is always better to start with the no. of departments in that university.
  - For a small data set, sometimes a rule of thumb that is followed is

$$K = \sqrt{\frac{n}{2}}$$

Strengths and Weaknesses of 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

- This method tries to measure the homogeneity within the cluster and for various values of ‘K’ and helps in arriving at the optimal ‘K’.
- From fig. we can see the homogeneity will increase or heterogeneity will decrease with increasing ‘K’ as the no. of data points inside each cluster reduces with this increase.

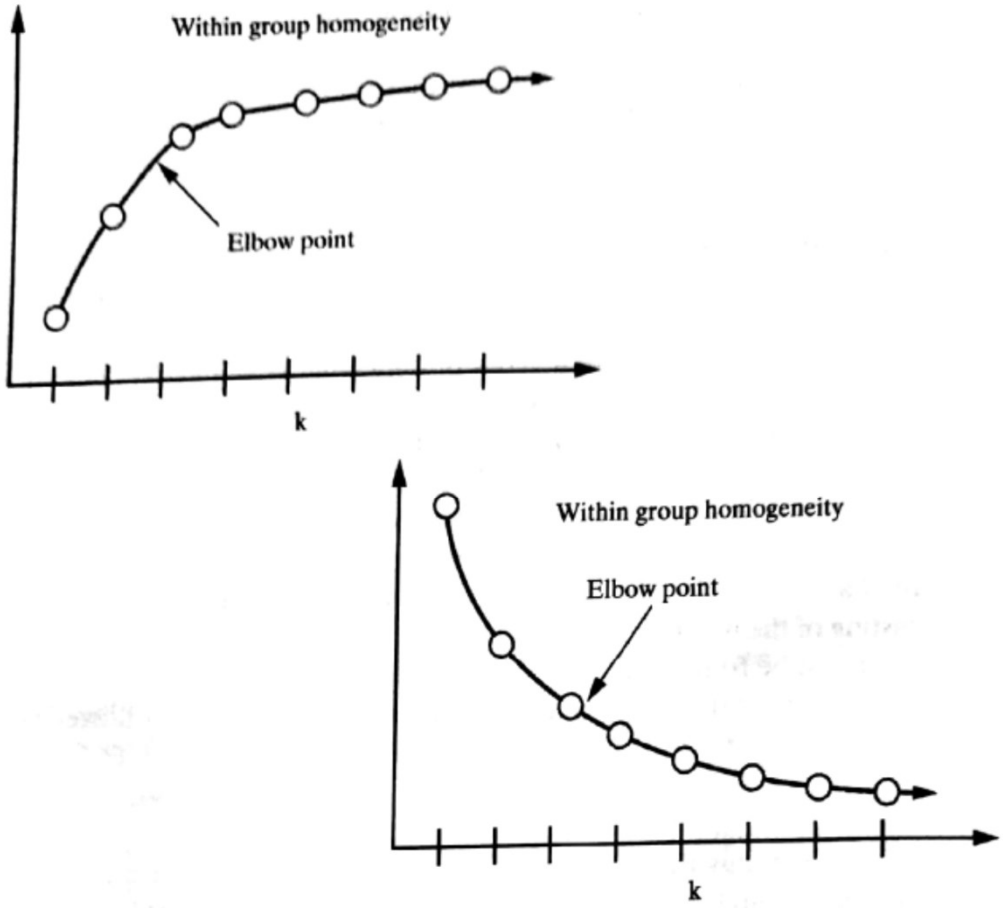


FIG. 9.5
Elbow point to determine the appropriate number of clusters

Non-Hierarchical clustering

(K-Means Algorithm)

Height	Weight	
150	50	→ centroid 1 → cluster 1
160	55	→ centroid 2 → cluster 2.
175	65	→ cluster 2 → (167.5, 60)
170	60	→ cluster 2 → (168.75, 60)
165	70	→ cluster 2

Assume

centroids and No. of clusters.

Suppose there are two clusters

c1	c2
(150, 50)	(160, 55)

Find the distance from centroid using Euclidean distance.

$$\sqrt{(x_{i2} - x_{i1})^2 + (x_{i2} - y_{i2})^2}$$

MM.DD.20XX

* Euclidean distance of record 3 from the centroids

$c1 = \sqrt{(175-150)^2 + (65-50)^2}$

$= \sqrt{625 + 225}$

$= \sqrt{850} = 29.15$

$c2 = \sqrt{(175-160)^2 + (65-55)^2}$

$= \sqrt{225 + 100} = \sqrt{325}$

$= 18.03$

Min. distance is 18.03 → cluster 2

* Find new centroid of cluster 2.

$= \left(\frac{160+175}{2}, \frac{55+65}{2}\right)$

$= (167.5, 60)$

* Euclidean distance of record 4 from the centroids.

$c1 = \sqrt{(170-150)^2 + (60-50)^2}$

$= 22.36$

$c2 = \sqrt{(170-167.5)^2 + (60-60)^2}$

$= 2.5$

distance is 2.5 → cluster 2

* Find the new centroid for cluster 2

$\left(\frac{167.5+170}{2}, \frac{60+60}{2}\right)$

$= (168.75, 60)$

* Euclidean distance of record 5 from centroids.

$c1 = \sqrt{(165-150)^2 + (70-50)^2}$

$= 25$

$c2 = \sqrt{(165-168.75)^2 + (70-60)^2}$

$= \sqrt{14.06 + 100}$

$= 10.67$

Min. distance is 10.67 → cluster 2.

c1 → {1}

c2 → {2, 3, 4, 5}

K – Medoids : a representative object – based technique

- A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.
- K – means algorithm is sensitive to outliers in the data set.
- E.g.
  - Data = 1,2,3,6,9,10,11,25.
  - Point 25 is the outlier.
  - K = 2
  - So, initial cluster {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 {9,10,12,25} = 56/4 = 14

So, the SSE within the clusters is

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

If we compare this with the cluster {1, 2, 3, 6, 9} and {10, 11, 25},

the mean of the cluster $\{1, 2, 3, 6, 9\} = \frac{21}{5} = 4.2,$

and the mean of the cluster $\{10, 12, 25\} = \frac{47}{3} = 15.67.$

So, the SSE within the clusters is

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

- Because SSE of the second clustering is lower, k-means tend to put point 9 in the same cluster with 1,2,3, and 6.
- But the 9 is logically nearer to points 10 and 11.
- This skewedness is introduced due to the outlier point 25, which shifts the mean away from the centre of the cluster.
- K – medoids provides a solution to this problem.
- Instead of considering the mean of the data points in the cluster, k – medoids considers k representative data points from the existing points in the data set as the centre of the clusters.
- It then assign the data points according to their distance from these centres to form k clusters.
- Note that the medoids in this case are actual data points or objects from the data set and not an imaginary point as in the case when the mean of the data sets within cluster is used as the centroid in the k-means technique.

- The SSE is calculated as :

$$\text{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i} \text{dist}\,(o_i, x)^2 \qquad (9.3)$$

where o_i is the representative point or object of cluster C_i.

- PAM (Partitioning Around Medoids) algorithm :

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

- In this algorithm, we replaced the current representative object with a non – representative object and checked if it improves the quality of clustering.
- In the iterative process, all possible replacements are attempted until the quality of clusters no longer improves.

If $o_1, \ldots, o_k$ are the current set of representative objects or medoids and there is a non-representative object o_r, then to determine whether o_r is a good replacement of o_j $(1 \leq j \leq k)$, the distance of each object x is calculated from its nearest medoid from the set $\{o_1, o_2, \ldots, o_{j-1}, o_r, o_{j+1}, \ldots, o_k\}$ and the SSE is calculated. If the SSE after replacing o_j with o_r decreases, it means that o_r represents the cluster better than o_j, and the data points in the set are reassigned according to the nearest medoids now.

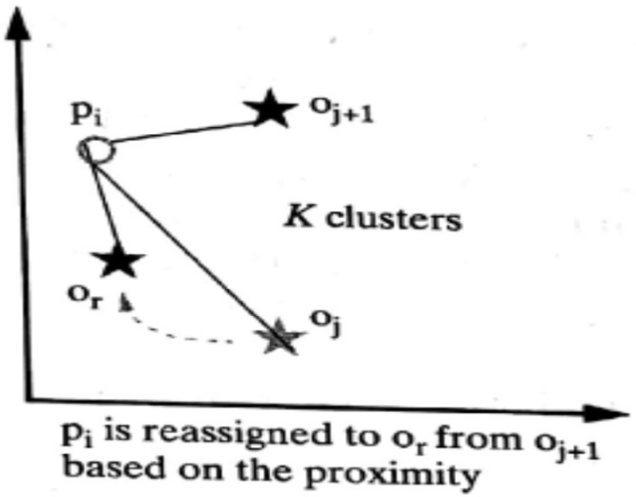


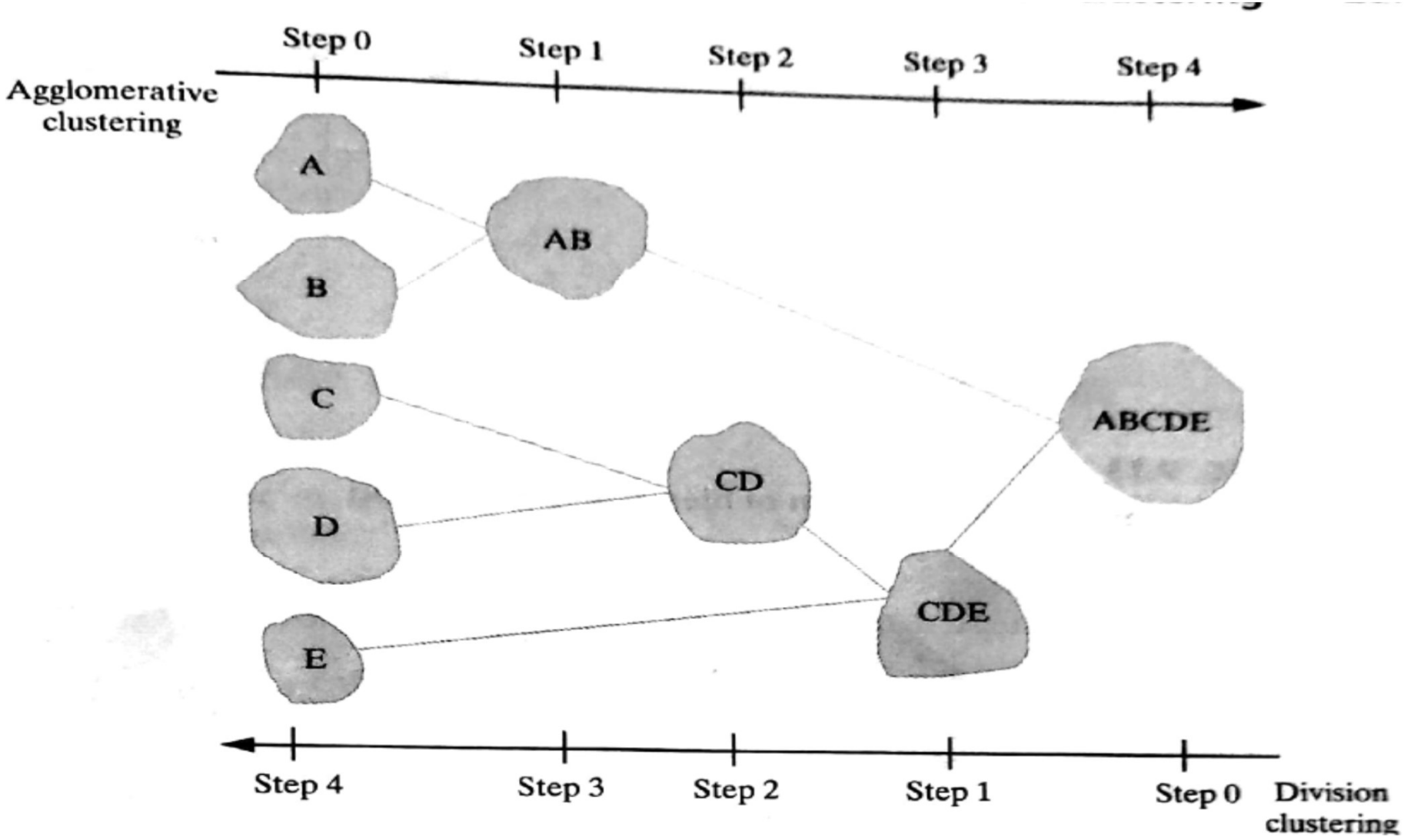
FIG. 9.11
PAM algorithm: Reassignment of points to different clusters

As shown in Figure 9.11, point p_i was belonging to the cluster with medoid o_{j+1} in the first iteration, but after o_j was replaced by o_r, it was found that p_i is nearest to the new random medoid and thus gets assigned to it. In this way, the clusters get refined after each medoid is replaced with a new non-representative medoid. Each time a reassignment is done, the SSE based on the new medoid is calculated. The difference between the SSE before and after the swap indicates whether or not the replacement is improving the quality of the clustering by bringing the most similar points together.

Hierarchical clustering

- The hierarchical clustering methods are used to group the data into hierarchy or tree – like structure.
- There are two main hierarchical clustering methods :
  - Agglomerative clustering
  - Divisive clustering
- The agglomerative hierarchical clustering method uses the bottom up strategy. It starts with each object forming its own cluster and then iteratively merges the clusters according to their similarity to form larger clusters. It terminates either when a certain clustering condition imposed by the user is achieved or all the clusters merge into a single cluster.

- The divisive hierarchical clustering method uses a top – down strategy. The starting point is the largest cluster with all the objects in it, and then, it is split recursively to form smaller and smaller clusters, thus forming the hierarchy. The end of iterations is achieved when the objects in the final clusters are sufficiently homogeneous to each other or the final clusters contain only one object or the user – defined clustering condition is achieved.
- In both the cases, it is important to select the split and merger points carefully.
- A dendrogram is a commonly used tree structure representation of step – by – step creation of hierarchical clustering.
- It shows how clusters are merged iteratively (in the case of agglomerative clustering) or split iteratively (in the case of divisive clustering).


Step 0
Step 1
Step 2
Step 3
Step 4
Agglomerative clustering
A
B
C
D
E
AB
CD
CDE
ABCDE
Step 4
Step 3
Step 2
Step 1
Step 0
Division clustering

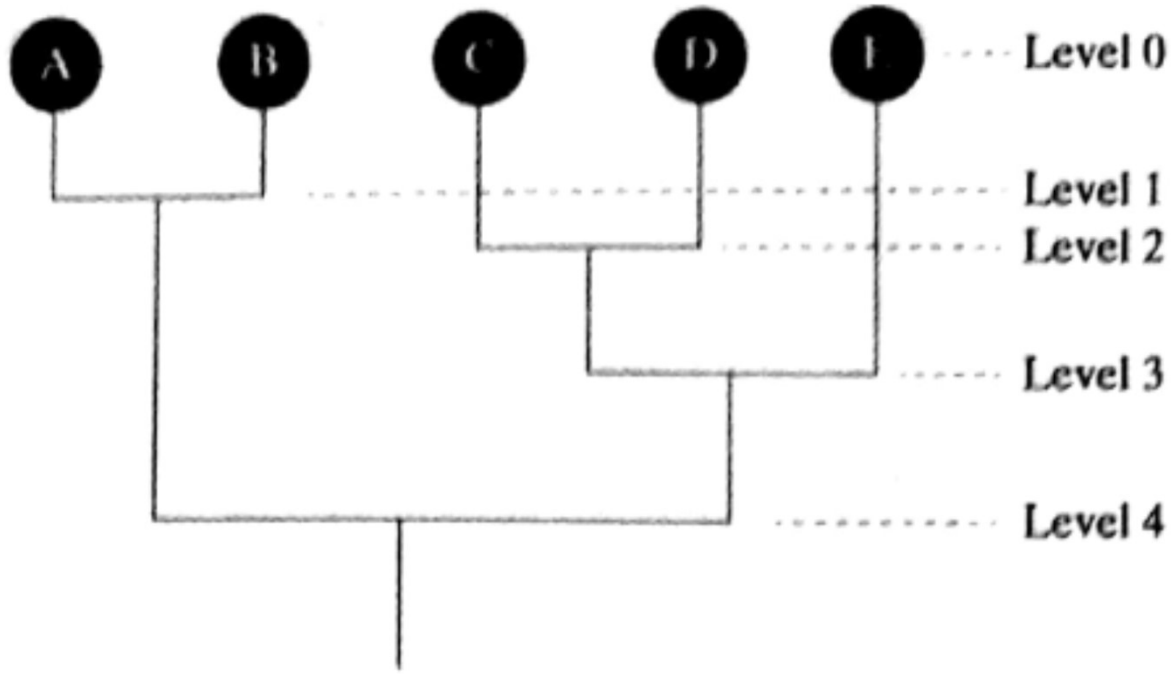


FIG. 9.13
Dendrogram representation of hierarchical clustering

One of the core measures of proximities between clusters is the distance between them. There are four standard methods to measure the distance between clusters:

Let C_i and C_j be the two clusters with n_i and n_j respectively. p_i and p_j represents the points in clusters C_i and C_j respectively. We will denote the mean of cluster C_i as m_i.

$$\text{Minimum distance } D_{\min}(C_i, C_j) = \min_{p_i \in C_i,\ p_j \in C_j} \{|p_i - p_j|\} \quad (9.4)$$

$$\text{Maximum distance } D_{\max}(C_i, C_j) = \max_{p_i \in C_i,\ p_j \in C_j} \{|p_i - p_j|\} \quad (9.5)$$

$$\text{Mean distance } D_{\text{mean}}(C_i, C_j) = \{|m_i - m_j|\} \quad (9.6)$$

$$\text{Average distance } D_{\text{avg}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p_i \in C_i,\ p_j \in C_j} |pi - pj| \quad (9.7)$$

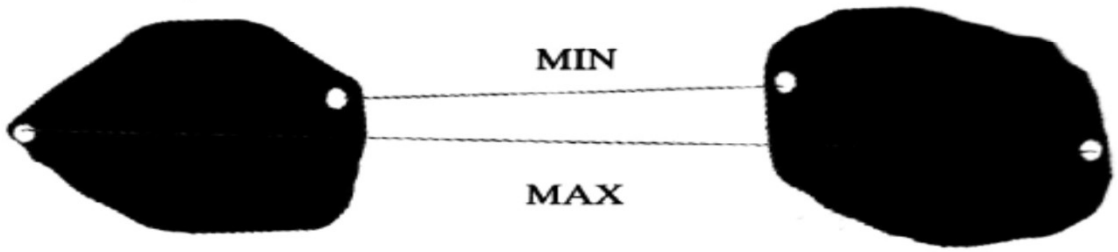


FIG. 9.14
Distance measure in algorithmic methods

- Note :
- Minimum distance = single linkage
- Maximum distance = complete linkage

Density – based methods - DBSCAN

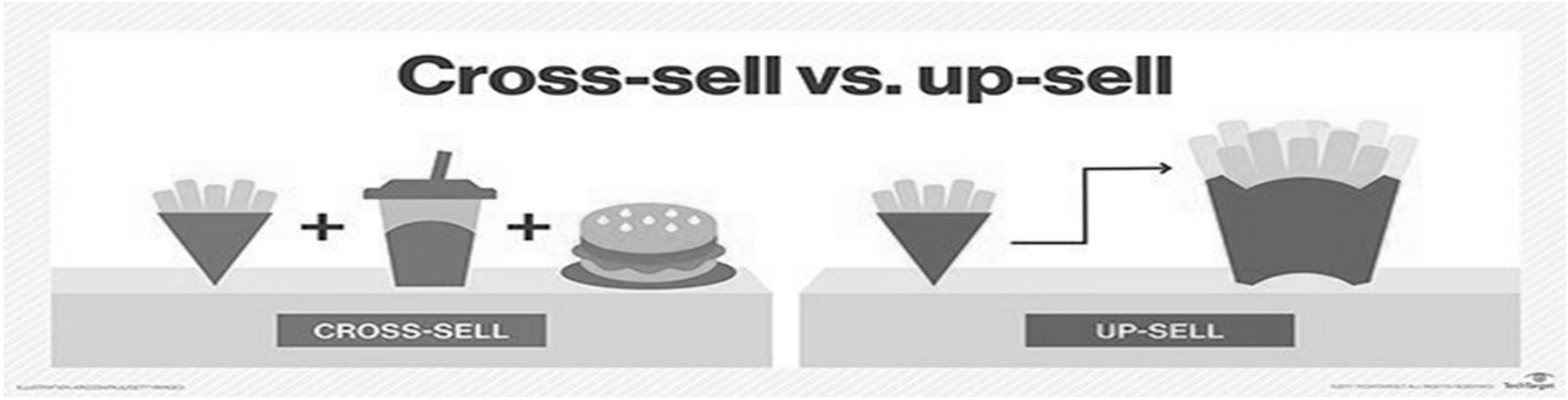
- You might have noticed that when we used the partitioning and hierarchical clustering methods, the resulting clusters are spherical or nearly spherical in nature.
- In the case of other shaped clusters such as S – shaped or uneven shaped clusters, the above two types of method do not provide accurate results.
- The density – based clustering approach provides a solution to identify clusters of arbitrary shapes.
- The principle is based on identifying the dense area and sparse area within the data set and then run the clustering algorithm.
- DBSCAN is one of the popular density – based algorithm which creates clusters by using connected regions with high density.

Finding Pattern Using Association Rule

- Association rule presents a methodology that is useful for identifying interesting relationships hidden in large data sets.
- A common application of this analysis is the Market Basket Analysis that retailers use for cross – selling.

Market basket analysis

- It allows retailers to identify the relationship between items which are more frequently bought together.
- E.g. conditioner with shampoo, Butter/cheese with bread, tongue cleaner with toothbrush and toothpaste etc.

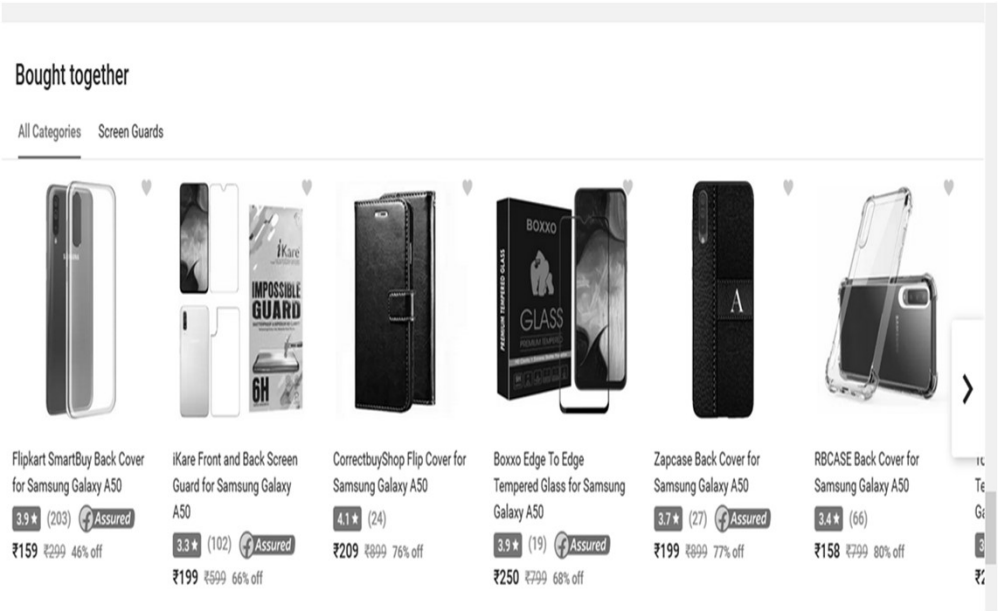


Association rule

- Association is a data mining function that discovers the probability of the co-occurrence of items in a collection.
- The relationship between co-occurring items are expressed as association rules.
- Strategy – “what goes with what”
- Also known as *affinity analysis or market basket analysis.*
- Association rules are heavily used in retail for learning about items that are purchased together, but they are also useful in other fields.
- For example, a medical researcher might want to learn what symptoms appear together. In law, word combinations that appear too often might indicate plagiarism.

- **Discovering Association Rules in Transaction Databases :**
  - Detailed information of customer transaction is stored in database.
  - So association between items can be found easily.
  - E.g. market basket analysis in supermarket
    - Collect data using barcode scanner
    - Market basket database consists large no. of transaction records.
    - Each record lists all items bought by a customer on a single-purchase transaction.
    - Managers are interested to know if certain groups of items are consistently purchased together.
    - They could use such information for making decisions on store layouts and item placement, for cross-selling, for promotions, for catalog design, and for identifying customer segments based on buying patterns.

- Association rules provide information of this type in the form of “if–then” statements.
- These rules are computed from the data; unlike the if–then rules of logic, association rules are probabilistic in nature.
- Association rules are commonly encountered in online *recommendation systems* (or *recommender systems*), where customers examining an item or items for possible purchase are shown other items that are often purchased in conjunction with the first item(s).



Here,

- Bought together – recommender system
- Buy together and save upto 20% - Product bundling

- Generating candidate rules :
  - Examine all possible rules between items in an if–then format, and select only those that are most likely to be indicators of true dependence.
  - We use the term *antecedent* to describe the IF part, and *consequent* to describe the THEN part.
  - Suppose item **A** is being bought by the customer, then the chances of item **B** being picked by the customer too under the same **Transaction ID** is found out.

  - There are two elements of these rules:
    - **Antecedent** (IF): This is an item/group of items that are typically found in the Itemsets or Datasets.
    - **Consequent** (THEN): This comes along as an item with an Antecedent/group of Antecedents.

- Apriori algorithm :
  - Apriori algorithm uses frequent itemsets to generate association rules.
  - It is based on the concept that a subset of a frequent itemset must also be a frequent itemset.
  - Frequent Itemset is an itemset whose support value is greater than a threshold value(minimum support).
  - Points : (The apriori algorithm for association rule learning)
    - Decide minimum support and minimum confidence
    - Generate frequent itemset
    - Selecting strong rules
    - Data format
    - **The Process of Rule Selection**

- Selecting Strong Rules :
- To build or prepare an association rule, there are three important matrices :
  1. Support
  2. Confidence
  3. Lift
- **Support:**
  - It gives the fraction of transactions which contains item A and B.
  - Basically Support tells us about the frequently bought items or the combination of items bought frequently.
  - We need to set a minimum support called Support Threshold for further analysis.
  - We only consider itemsets on or above the Support Threshold.

  Here,

  A,B = items

  N = total no. of transactions

$$Support = \frac{freq(A,B)}{N}$$

- **Confidence :**
  - It tells us how often the items A and B occur together, given the number times A occurs.
  - This is likelihood of an item being purchased given another item is purchased.

$$Confidence = \frac{freq(A,B)}{freq(A)}$$

  - Typically, when you work with the Apriori Algorithm, you define these terms accordingly.
  - **But how do you decide the value?** Honestly, there isn't a way to define these terms.
  - Suppose you've assigned the support value as 2. What this means is, until and unless the item/s frequency is not 2%, you will not consider that item/s for the Apriori algorithm.
  - This makes sense as considering items that are bought less frequently is a waste of time.
  - Now suppose, after filtering you still have around 5000 items left. Creating association rules for them is a practically impossible task for anyone. This is where the concept of lift comes into play.

- **Lift**.
  - This says how likely item Y is purchased when item X is already purchased, while controlling for how popular item Y is.
  - Lift indicates the strength of a rule over the random occurrence of A and B.
  - Lift value > 1 – Item B is likely to be bought when A is already bought.
  - **More the Lift more is the strength.**

$$Lift = \frac{Support}{Supp(A)\ x\ Supp(B)}$$

Transaction 1	🍎 🍺 🍚 🍗
Transaction 2	🍎 🍺 🍚
Transaction 3	🍎 🍺
Transaction 4	🍎 🍐
Transaction 5	🍼 🍺 🍚 🍗
Transaction 6	🍼 🍺 🍚
Transaction 7	🍼 🍺
Transaction 8	🍼 🍐

$$\text{Support}\{🍎\} = \frac{4}{8}$$

= 0.50

$$\text{Confidence}\{🍎 \rightarrow 🍺\} = \frac{\text{Support}\{🍎, 🍺\}}{\text{Support}\{🍎\}}$$

(3/8) / (4/8) = (3/4) = 0.75

$$Rule: X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

$$\text{Lift}\{🍎 \rightarrow 🍺\} = \frac{\text{Support}\{🍎, 🍺\}}{\text{Support}\{🍎\} \times \text{Support}\{🍺\}}$$

(3/8) / ((4/8)*(6/8)) = (3/8) / (3/8) = 1

E.g.

Market Basket Transaction Data

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}

- Consider the association rule {Bread,Milk} -> {Egg}, then find out its support and confidence.
- Also find out confidence of {Egg} -> {Bread,Milk}

- The rule with higher confidence is the strong rule.
- Here, when you calculate confidence–
  - {Bread,Milk} -> {Egg} = 0.75
  - {Egg} -> {Bread,Milk} =0.60
- So, we can say that rule {Bread,Milk} -> {Egg} is the strong rule.

Build the apriori principle rules

- The algorithm utilizes a simple prior belief (i.e. priori) about the properties of frequent itemsets :
  - If an itemset is frequent, then all of its subsets must also be frequent.
    - This principle significantly restricts the no. of itemsets to be searched for rule generation.
    - For e.g., if in a market basket analysis, it is found that an item like ‘salt’ is not so frequently bought along with the breakfast items, then it is fine to remove all the itemsets containing salt for rule generation as their contribution to the support and confidence of the rule will be insignificant.
  - If an itemset is frequent, then all the supersets must be frequent too.
    - These are very powerful principles which help in pruning the exponential search space based on the support measure and is known as support – based pruning.
    - The key property of the support measure used here is that the support for an itemset never exceeds the support for its subsets.
    - This is also known as anti-monotone property of the support measure.

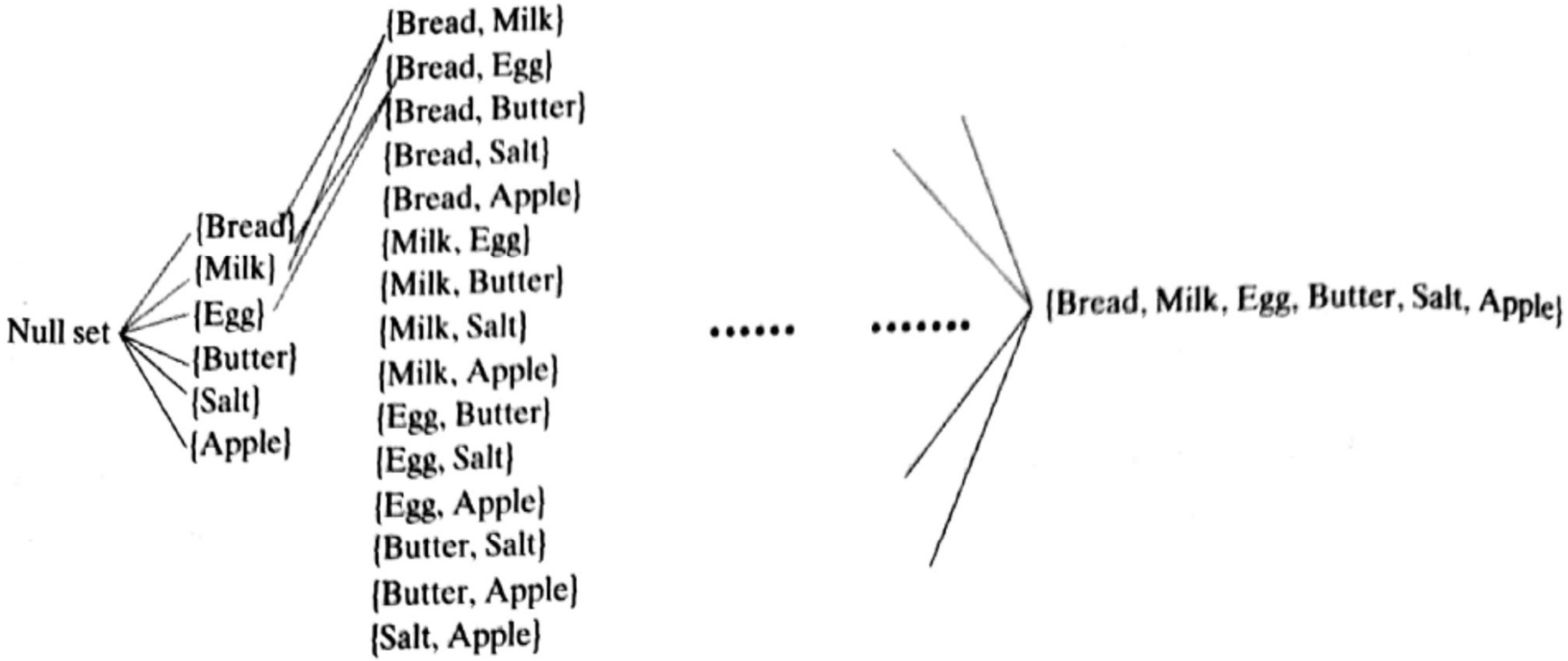


FIG. 9.15
Sixty-four ways to create itemsets from 6 items

Without applying any filtering logic, the brute-force approach would involve calculating the support count for each itemset in Figure 9.16. Thus, by comparing each item in the generated itemset with the actual transactions mentioned in **Table 9.3**, we can determine the support count of the itemset. For example, if {Bread, Milk} is present in any transactions in **Table 9.3**, then its support count will be incremented by 1. As we can understand, this is a very computation heavy activity, and as discussed earlier, many of the computations may get wasted at a later point of time because some itemsets will be found to be infrequent in the transactions. To get an idea of the total computations to be done, the number of comparisons to be done is $T \times N \times L$, where T is the number of transactions (6 in our case), N is the number of candidate itemsets (64 in our case), and L is the maximum transaction width (6 in our case).

Let us apply the Apriori principle on this data set to reduce the number of candidate itemsets (N). We could identify from the transaction **Table 9.3** that Salt is an infrequent item. So, by applying the Apriori principle, we can say that all the itemsets which are superset of Salt will be infrequent and thus can be discarded from comparison to discover the association rule as shown in Figure 9.16.

This approach reduces the computation effort for a good number of itemsets and will make our search process more efficient. Thus, in each such iteration, we can determine the support count of each itemset, and on the basis of the min support value fixed for our analysis, any itemset in the hierarchy that does not meet the min support criteria can be discarded to make the rule generation faster and easier.

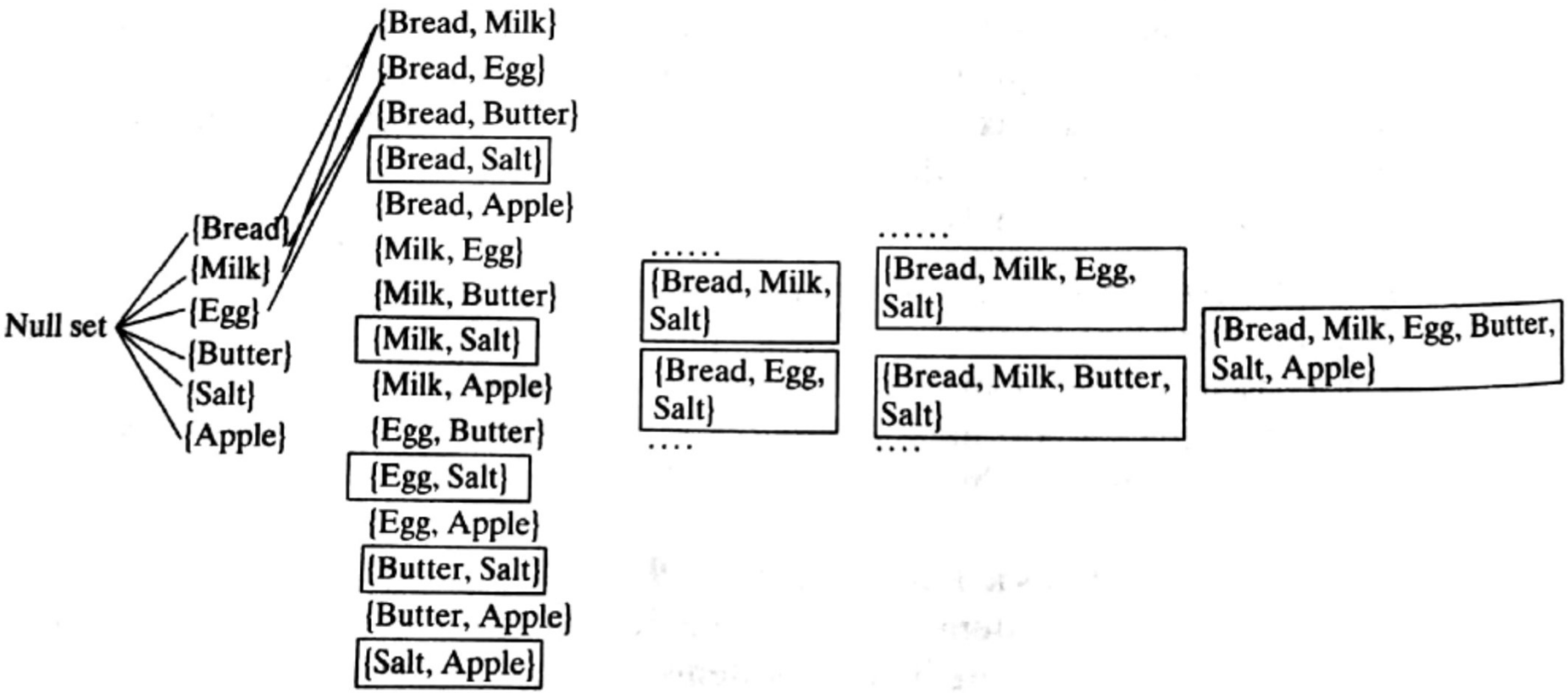


FIG. 9.16
Discarding the itemsets consisting of Salt

- The actual process of creating rules involves two phases :
  - **Identifying all itemsets that meet a minimum support threshold set for the analysis**
  - **Creating rules from these itemsets that meet a minimum confidence threshold which identifies the strong rules**

Strengths and Weaknesses

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