DATA ANALYTICS AND R PROGRAMMING (18MIT22C)

Prepared by Dr.M.Soranamageswari

UNIT-I: Introduction – Data – Types of Data – Data Mining Functionalities – Interestingness of Patterns – Classification of Data Mining Systems – Data Mining Task Primitives –Association rule mining: Mining Frequent Patterns, Associations and Correlations – Mining Methods – Mining various kinds of association rules. Pg(1-34, 227-259)

UNIT-II: Classification and Clustering: Classification and Prediction - Basic Concepts- Decision Tree Induction - Bayesian Classification – Rule Based Classification – Classification by Back Propagation Cluster Analysis - Types of Data – Categorization of Major Clustering Methods–K-means-Partitioning Methods – Hierarchical Methods – Clustering High Dimensional Data- Constraint Based Cluster Analysis – Outlier Analysis – Data Mining Applications.

UNIT-III: Big Data Analytics: Introduction to the Big Data Era – Description of Big Data – Industry Examples of Big Data – Descriptive power and predictive Pattern Matching – The Value of Data – Big Data Analytics – Architectures, Frameworks, and Tools – Big Data Analytics Methodology – Challenges – Big Data Analytics in Healthcare.

UNIT-IV: Getting Started with R- R Nuts and Bolts - Getting Data in and Out of R - Using Textual and Binary Formats for Storing Data- Interfaces to the Outside World- Subsetting R Objects - Vectorized Operations - Managing Data Frames with the dplyr package.

UNIT-V: Control Structures -Functions- Scoping Rules of R - Loop Functions- Debugging Tool in R- Profiling R Code- Simulation.

TEXT BOOKS

1. Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, Second
Edition, Elsevier, 2007. (Unit I and II)
2. Stephan Kudyba Foreword by Thomas H.Davenport, “Big Data, Mining, and Analytics”, CRC Press, 2015. (Unit III)
3. Roger D. Peng, “R Programming for Data Science” Lean Publishing, 2014. (Unit IV & V)

UNIT-I: Introduction – Data – Types of Data – Data Mining Functionalities – Interestingness of Patterns – Classification of Data Mining Systems – Data Mining Task Primitives –Association rule mining: Mining Frequent Patterns, Associations and Correlations – Mining Methods – Mining various kinds of association rules.

Introduction

- Data mining refers to extracting or “mining” knowledge from large amounts of data.
- Mining of gold from rocks or sand is referenced as gold mining rather than rock\sand mining.
- so data mining is named knowledge mining from data.
- Data mining is a process of finds small set of precious nuggets from a great deal of raw materials.
- Data mining is synonym of knowledge discovery from data or KDD.
- Data mining consist of an iterative sequence of the ~ing.

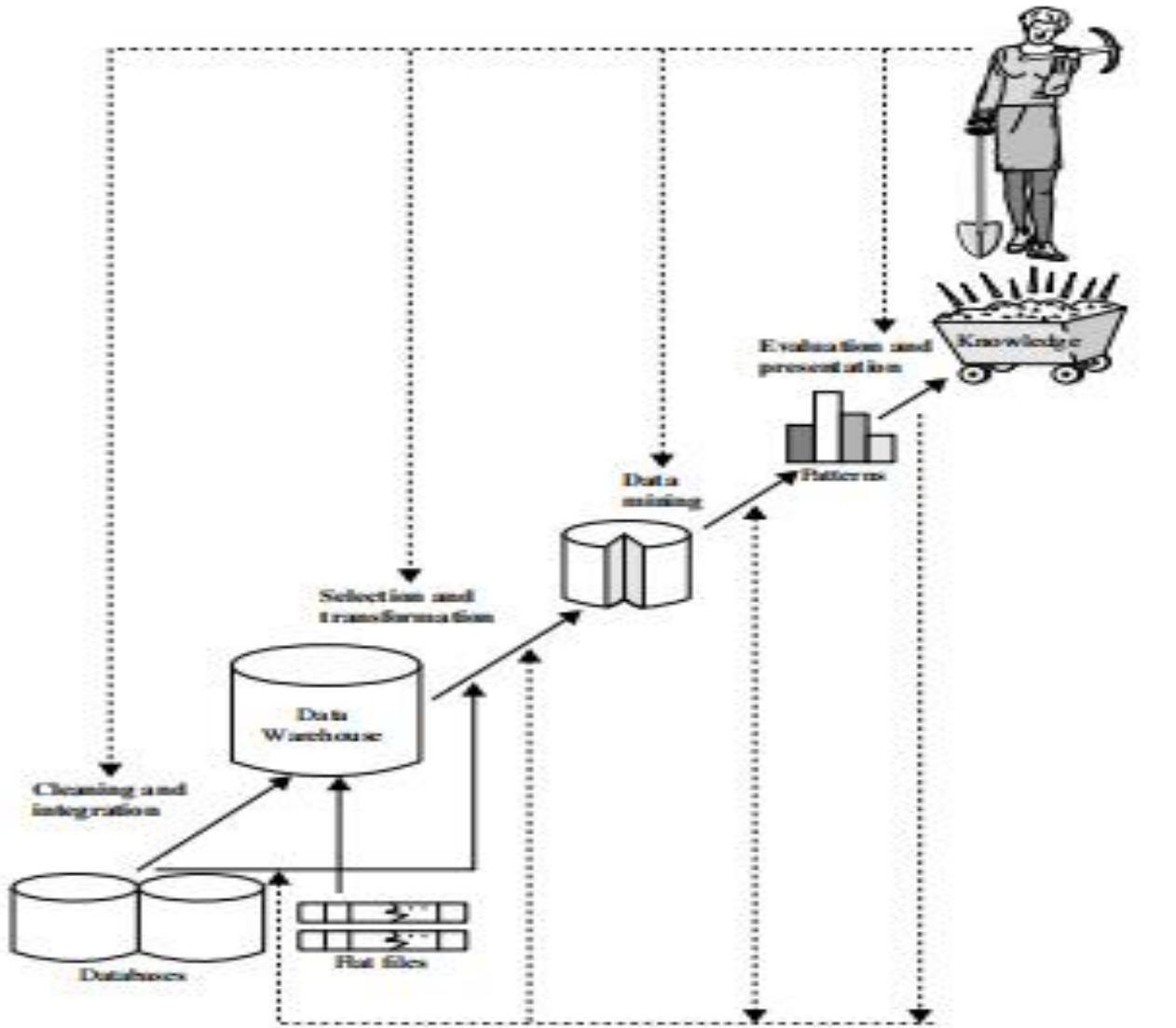

Figure 1.4 Data mining as a step in the process of knowledge discovery.

1.Data cleaning (to remove noise and inconsistent data)

2. Data integration (where multiple data sources may be combined)

3. Data selection (where data relevant to the analysis task are retrieved from the

database)

4. Data transformation (where data are transformed and consolidated into forms

appropriate for mining by performing summary or aggregation operations)

5. Data mining (an essential process where intelligent methods are applied to extract data patterns)

6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on interestingness measures)

7. Knowledge presentation (where visualization and knowledge representation techniques are used to present mined knowledge to users)

Data mining:

-The data mining step may interact with the user or a knowledge base.

Patterns:

-presented to the user and may be stored as new knowledge in the knowledge base.

Knowledge base:

This is the domain knowledge we used to guide the search /evaluate to interestingners or resulting patterns.

This knowledge include concept hierarchies used to organize attributes or attribute values into different levels of abstraction.

Used to describe data from multiple hetrogenious source.

Datamining Engine:

It consist of a set of functional modules. (i.e)such as characterization ,association&correlation,analysis,classification,prediction,cluster analysis,outlier analysis & evolution analysis.

Patterns Evaluation module:

- It employs interestingness measures
- Interact with data mining modules to focus the season towards interesting patterns.
- For efficient data mining this is used to push the evaluation of pattern.

User interface:

* This module communicates between users & the data mining system.

* This allow user to interact with the system by specifying a data mining query or task and exploratory data mining based on the intermediate data mining results.

* It allows the user to browse database & datawarewhouse schemes/ data structures, evaluate mined pattern & visualize the patterns in the different forms.

* Data mining can be an advanced stage of Online Analytics Processing

(OLAP).

* Data mining involve an integration of various technique from different disipline such as, database & data warehouse technology, statistics,machine learning, high performance computing, pattern recognition, neural network,etc.

1.2. Data mining data types:

(a) Relational data base (b) Transactional database

(c) Data warehouse

(d) Advanced data & Informatiom system & advanced applications.

(a) Relational database:

-A database system, also called a database management system (DBMS), consists of a collection of interrelated data, known as a database, and a set of software programs to manage and access the data.

- A relational database is a collection of tables, each of which is assigned a unique name.

- Each table consists of a set of attributes (columns or fields) and usually stores a large set of tuples (records or rows).

- Each tuple in a relational table represents an object identified by a unique key and described by a set of attribute values.

- A semantic data model, such as an entity-relationship (ER) data model, is often constructed for relational databases.

-An ER data model represents the database as a set of entities and their relationships.

Example 1.2 A relational database for AllElectronics

-The relation customer consists of a set of attributes describing the customer information, including a unique customer identity number (cust ID), customer name,address,age,occupation, annual income, credit information, and category. Similarly, each of the relations item, employee, and branch consists of a set of attributes describing the properties of these entities.

- Tables can also be used to represent the relationships between or among multiple entities.

In the example, these include purchases (customer purchases items, creating a sales transaction handled by an employee), items sold (lists items sold in a given transaction), and works at (employee works at a branch of AllElectronics).

customer (cust ID, name, address, age, occupation, annual income, credit information, category, . . .)

customer	(*cust_ID*, *name*, *address*, *age*, *occupation*, *annual_income*, *credit_information*, *category*, . . .)
item	(*item_ID*, *brand*, *category*, *type*, *price*, *place_made*, *supplier*, *cost*, . . .)
employee	(*empl_ID*, *name*, *category*, *group*, *salary*, *commission*, . . .)
branch	(*branch_ID*, *name*, *address*, . . .)
purchases	(*trans_ID*, *cust_ID*, *empl_ID*, *date*, *time*, *method_paid*, *amount*)
items_sold	(*trans_ID*, *item_ID*, *qty*)
works_at	(*empl_ID*, *branch_ID*)

Figure 1.5 Relational schema for a relational database, *AllElectronics*.

-Relational data can be accessed by database queries written in a relational query language (e.g., SQL).

-A given query is transformed into a set of relational operations, such as join, selection, and projection, and then optimized for efficient processing.

(b) Data warehouse:

- Suppose that AllElectronics is a successful international company with branches around the world.

- Each branch has its own set of databases.

- The president of AllElectronics has asked you to provide an analysis of the company's sales per item type per branch for the third quarter.

- This is a difficult task, particularly since the relevant data are spread out over several databases physically located at numerous sites.

- If AllElectronics had a data warehouse, this task would be easy.

Definition:

A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and usually residing at a single site.

-Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing.

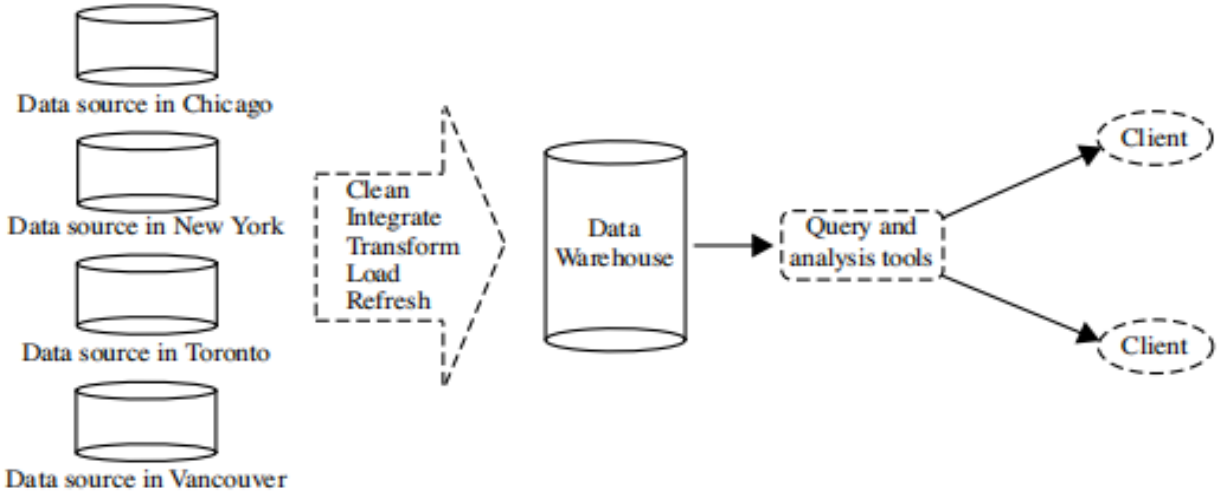

Figure 1.6 Typical framework of a data warehouse for *AllElectronics*.

-To facilitate decision making, the data in a data warehouse are organized around major subjects (e.g., customer, item, supplier, and activity).

- A data warehouse is usually modeled by a multidimensional data structure, called a data cube.

- By providing multidimensional data views and the precomputation of summarized data, data warehouse systems can provide inherent support for OLAP.

Ex:A data cube for AllElectronics:

The cube has three dimensions:

-address (with city values Chicago, New York, Toronto, Vancouver),

-time (with quarter values Q1, Q2, Q3, Q4), and

-item(with item type values home entertainment, computer, phone, security).

The aggregate value stored in each cell of the cube is sales amount (in thousands).

Difference between data warehouse and datamart:

Data warehouse:

- It Collects information about subjects that span an entire organization and its scope is enterprise wide.

Data mart:

- It is a dept. subset of datawarehouse.It focus on selected subset.Ex: itemwise scope is department –wide.

- Because of 3’D representation it is best suited for OLAP.

(c) Transactional database:

- In general, each record in a transactional database captures a transaction, such as a customer’s purchase, a flight booking, or a user’s clicks on a web page.

- A transaction typically includes a unique transaction identity number (trans ID) and a list of the items making up the transaction, such as the items purchased in the transaction.

-A transactional database may have additional tables, which contain other information related to the transactions, such as item description, information about the salesperson or the branch, and so on.

Ex:A transactional database for AllElectronics.

-Transactions can be stored in a table, with one record per transaction.

-From the relational database point of view, the sales table in the figure is a nested relation because the attribute list of item IDs contains a set of items.

trans_ID	*list_of_item_IDs*
T100	I1, I3, I8, I16
T200	I2, I8
...	...

Figure 1.8 Fragment of a transactional database for sales at *AllElectronics*.

(d) Advanced data & information system & advanced application:

-Besides relational database data, data warehouse data, and transaction data, there are many other kinds of data that have versatile forms and structures and rather different semantic meanings.

Such kinds of data can be seen in many applications:

1)time-related or sequence data (e.g., historical records, stock exchange data, and time-series and biological sequence data),

2)data streams (e.g., video surveillance and sensor data, which are continuously transmitted),

3)spatial data (e.g., maps), engineering design data (e.g., the design of buildings, system components, or integrated circuits),

4)hypertext and multimedia data (including text, image, video, and audio data),

5)graph and networked data (e.g., social and information networks), and

6)the Web (a huge, widely distributed information repository made available by the Internet).

-These applications bring about new challenges, like how to handle data carrying special structures (e.g., sequences, trees, graphs, and networks) and specific semantics (such as ordering, image, audio and video contents, and connectivity), and how to mine patterns that carry rich structures and semantics.

1.4 Data mining functionalities- what kind of pattern can be mined?

Data mining functionalities are used to specify the kinds of patterns to be found in data mining tasks. In general, such tasks can be classified into two categories: descriptive and predictive.

1) Descriptive mining tasks characterize properties of the data in a target data set.

2)Predictive mining tasks perform induction on the current data in order to make predictions.

https://youtu.be/6-Q59-ZD7ds

1.4.1 Class/Concept Description: Characterization and Discrimination

- Data entries can be associated with classes or concepts. For example, in the AllElectronics store, classes of items for sale include computers and printers, and concepts of customers include big Spenders and budget Spenders.
- These descriptions can be derived using

(1) data characterization, by summarizing the data of the class under study (often called the target class) in general terms, or

(2) data discrimination, by comparison of the target class with one or a set of comparative classes (often called the contrasting classes), or

(3) both data characterization and discrimination.

- . For example, to study the characteristics of software products with sales that increased by 10% in the previous year, the data related to such products can be collected by executing an SQL query on the sales database.
- The output of data characterization can be presented in various forms. Examplesinclude pie charts, bar charts, curves, multidimensional data cubes, and multidimensional tables, including crosstabs.

Data discrimination :

-It is a comparison of the general features of the target class data objects against the general features of objects from one or multiple contrasting classes.

-The target and contrasting classes can be specified by a user, and the corresponding data objects can be retrieved through database queries.

Ex: sales that increased by 10% last year against those with sales that decreased by at least 30% during the same period.

-The forms of output presentationare similar to those for characteristic descriptions.

Ex: 80% of the customers who frequently purchase computer products are between 20 and 40 years old and have a university education, whereas 60% of the customers who infrequently buy such products are either seniors.

1.4.2 Mining Frequent Patterns, Associations, and Correlations

- Frequent patterns:

There are many kinds of frequent patterns, including frequent itemsets, frequent subsequences (also known as sequential patterns), and frequent substructures. A frequent

- Frequent itemset:

Itemset typically refers to a set of items that often appear together in a transactional data set—for example, milk and bread, which are frequently bought together in grocery stores by many customers.

- Subsequences:

A frequently occurring subsequence, such as the pattern that customers, tend to purchase first a laptop, followed by a digital camera, and then a memory card, is a (frequent) sequential pattern. A substructure can refer to different structural forms (e.g., graphs, trees, or lattices) that may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a (frequent) structured pattern. Mining frequent patterns leads to the discovery of interesting associations and correlations within data.

https://youtu.be/QN3_wxqnSlw

-An example of such a rule, mined from the AllElectronics transactional database, is

buys(X, “computer”) ⇒ buys(X, “software”) [support = 1%,confidence = 50%]

- A 50% chance that she will buysoftware as well.

-A 1% support means that 1% of all the transactions under analysis show that computer and software are purchased together.

- This association rule involves a single attribute or predicate (i.e., buys) that repeats. Association rules that contain a single predicate are referred to as single-dimensional association rules. Dropping the predicate notation, the rule can be written simply as “computer ⇒ software [1%, 50%].”

- Suppose, instead, that we are given the AllElectronics relational database related to purchases. A data mining system may find association rules like

-age(X, “20..29”) ∧ income(X, “40K..49K”) ⇒ buys(X, “laptop”) [support = 2%, confidence = 60%].

1.4.3 Classification and Prediction:

- **Classification:**

- It is the process of finding a model (or function) that describes and distinguishes data classes or concepts. The model are derived based on the analysis of a set of training data (i.e., data objects for which the class labels are known). The model is used to predict the class label of objects for which the the class label is unknown.

- "How is the derived model presented?" The derived model may be represented in various forms, such as

1)classification rules (i.e., IF-THEN rules),

2)decision trees,

3)mathematical formulae, or

4)neural networks

-A **decision tree** is a flowchart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leafes represent classes or class distributions. Decision trees can easily

ge(X, “youth”) AND income(X, “high”)=> class(X, “A”)

age(X, “youth”) AND income(X, “low”)=> class(X, “B”)

age(X, “middle_aged”)=>class(X, “C”)

age(X, “senior”)=>class(X, “C”)

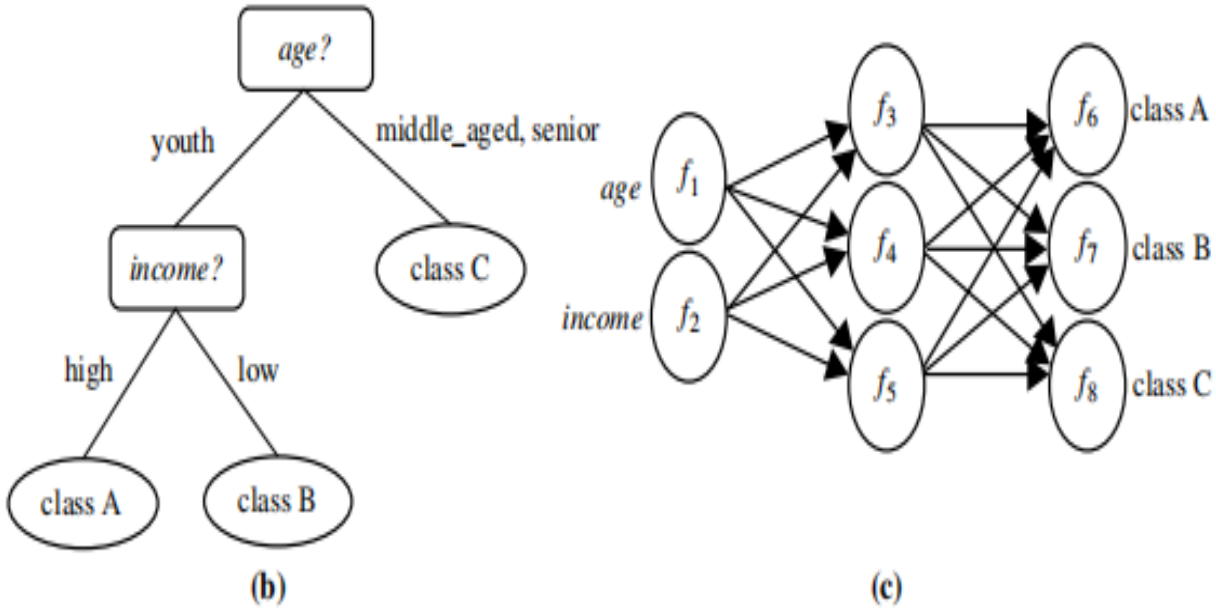

Figure 1.9 A classification model can be represented in various forms: (a) IF-THEN rules, (b) a decision tree, or (c) a neural network.

1.4.4 Cluster Analysis

- Unlike classification and regression, which analyze class-labeled (training) datasets, clustering analyzes data objects without consulting class labels. In many cases, class labeled data may simply not exist at the beginning. Clustering can be used to generate class labels for a group of data. The objects are clustered or grouped based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity.

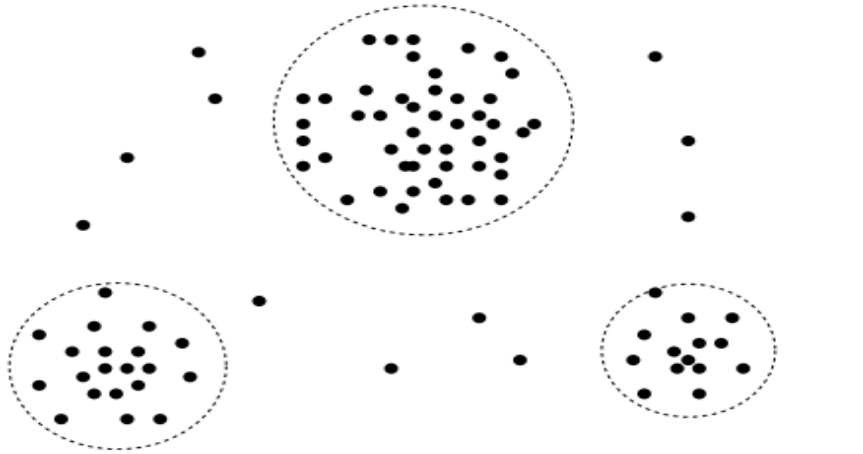

Figure 1.10 A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters.

1.4.5 Outlier Analysis

-A data set may contain objects that do not comply with the general behavior or model of the data. These data objects are outliers. Many data mining methods discard outliers as noise or exceptions. However, in some applications (e.g., fraud detection) the rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as outlier analysis or anomaly mining.

-Outliers may be detected using statistical tests that assume a distribution or probability model for the data, or using distance measures where objects that are remote from any other cluster are considered outliers. Rather than using statistical or distance measures, density-based methods may identify outliers in a local region, although they look normal from a global statistical distribution view.

1.4.6 Evolution analysis:

- Data evolution analysis describes & models regularities or trends for objects whose behaviours change overtime.

Ex: study of stock exchange to identify stock evolution

1.5 Interesting of patterns:

https://youtu.be/QQmrrQsfUDU

- A data mining system has the potential to generate thousands or even millions of patterns, or rules.

- You may ask, “Are all of the patterns interesting?” Typically, the answer is no—only a small fraction of the patterns potentially generated would actually be of interest to a given user.

-This raises some serious questions for data mining. You may wonder, “What makes a pattern interesting? Can a data mining system generate all of the interesting patterns? Or, Can the system generate only the interesting ones?” To answer the first question, a pattern is interesting if it is

(1) easily understood by humans,

(2) valid on new or test data with some degree of certainty,

(3) potentially useful, and

(4) novel.

-A pattern is also interesting if it validates a hypothesis that the usersought to confirm. An interesting pattern represents knowledge.

1)Pattern are interesting based on the structures of discovered patterns & the statistics in that.

Objective of association rules of the form x=>y (i.e) 1 % of transactions from a transaction database that given rule statistics the 2 measures are

(i)probability => p(x,y) where transaction contains both x<y (i.e) union of items x&y.

(ii) confidence => p(y/x) (i.e) the probability that a transaction containing X that also containing Y.

Ie.defines as

* support(x=>y)=p(xUy)
* confidence(x=>y)=p(y/x)

-but each interesting is association with a threshold which is controlled by user.

(i.e)confidence threshold is <50% . then it is uninteresting linewise.

1.6 Classification of data mining system:

https://youtu.be/Zj_csB0anJU

-Data mining is an interdisciplinary field. (i.e)set of disciplines including database system, statistics, machine learning, visualization & information system.

-Moreover depending on the data mining approach used techniques from other disciplines may be applied such as neural network, fuzzy & or rough set theory, knowledge, representation etc... for various applications like spatial data analysis, information retrivals image analysis etc.

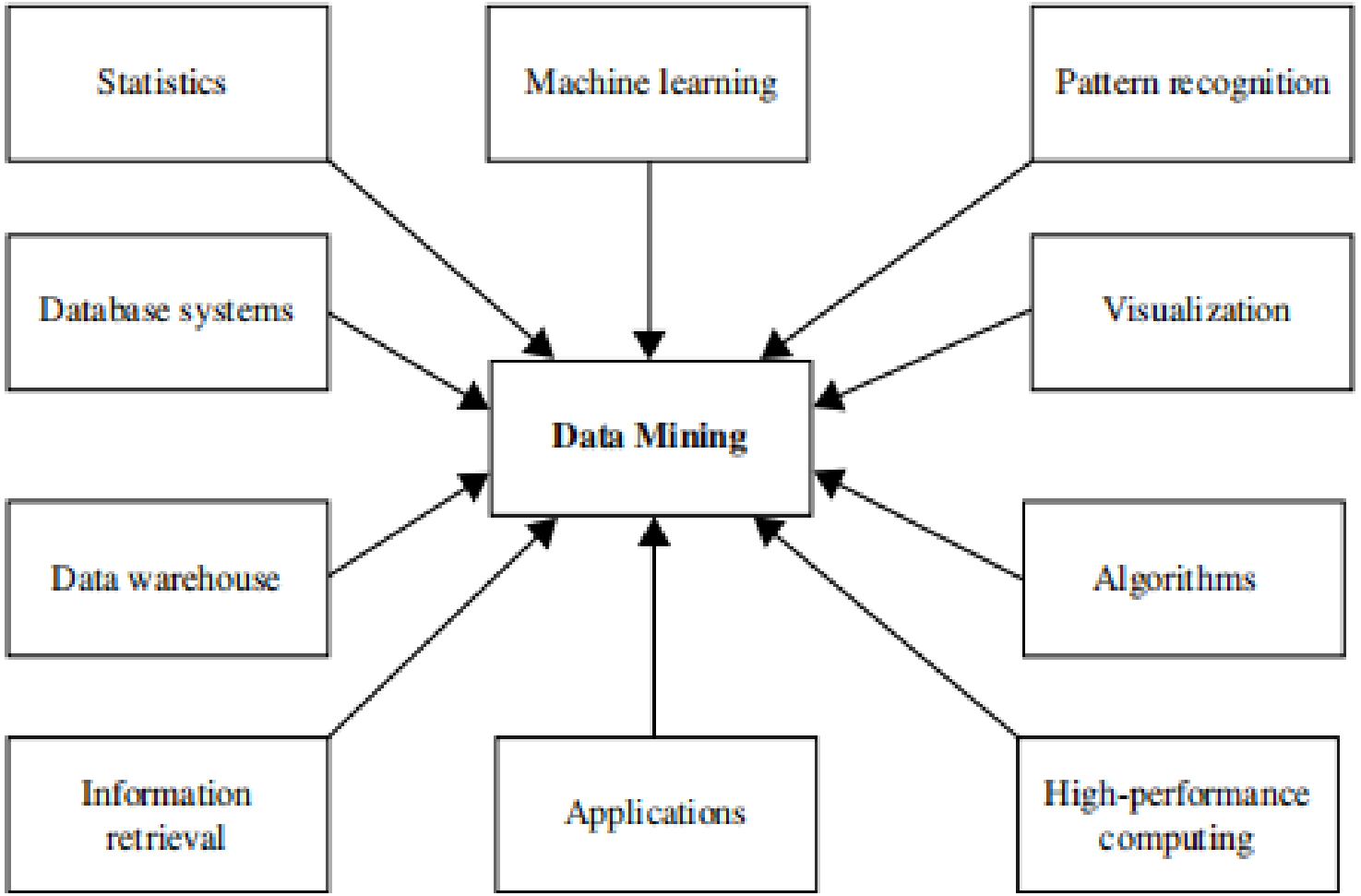

Figure 1.11 Data mining adopts techniques from many domains.

i. Classification according to the kinds of DB's mined:

-Data mining system can be classified according to the kinds of database mined such as data models/types of data or application involved.

ii. Classification according to the kinds of knowledge mined:

- Data mining system are categorized according to the kinds of knowledge they mine(i.e) based on data mining functionality such as:-

Characterization,discriminatio,association&correlation,analysis,classification,prediction,clustering outlier analysis & evolution analysis & also provide and/ or integrated data mining functionality.

1.7 Association rule mining:

Mining Frequent Patterns , Associations, and Correlations:

=> Frequent patterns are patterns (such as item sets, subsequences, or substructures) that appear in a data set frequently. For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set is a frequent item set.

=> A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern.

=> A substructure can refer to different structural forms, such as sub graphs , sub trees, or sub lattices, which may be combined with item sets or subsequences. If a substructure occurs frequently, it is called a (frequent) structured pattern.

- Finding such frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data.

-Moreover, it helps in data classification, clustering, and other data mining tasks as well.

-Thus, frequent pattern mining has become an important data mining task and a focused theme in data mining research.

Basic Concepts and a Road Map

-Frequent pattern mining searches for recurring relationships in a given data set. This section introduces the basic concepts of frequent pattern mining for the discovery of interesting associations and correlations between item sets in transactional and relational databases.

Market Basket Analysis: A Motivating Example

- Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets.

-With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases.

-The discovery of interesting correlation relationships among huge amounts of business transaction records can help in many business decision-making processes, such as catalog design, cross-marketing, and customer shopping behavior analysis.

- A typical example of frequent itemset mining is market basket analysis. This process analyzes customer buying habits by finding associations between the different items that customers place in their “shopping baskets”.

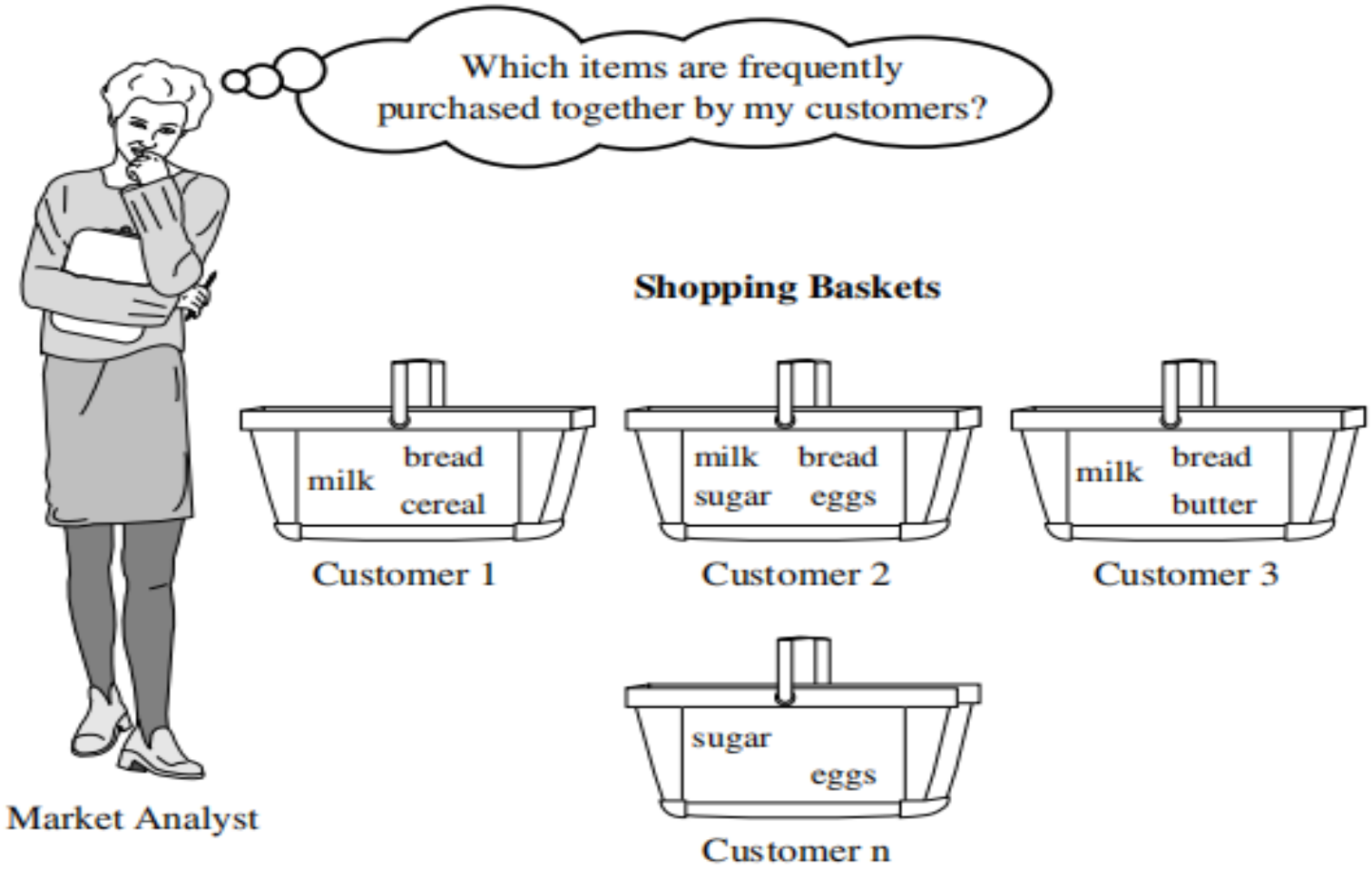

Figure 5.1 Market basket analysis.

Market basket analysis:

- Suppose, as manager of an AllElectronics branch, you would like to learn more about the buying habits of your customers.

-Specifically, you wonder, “Which groups or sets of items are customers likely to purchase on a given trip to the store?” To answer your question, market basket analysis may be performed on the retail data of customer transactions at your store.

-You can then use the results to plan marketing or advertising strategies, or in the design of a new catalog. For instance, market basket analysis may help you design different store layouts.

- In one strategy, items that are frequently purchased together can be placed in proximity in order to further encourage the sale of such items together. If customers who purchase computers also tend to buy antivirus software at the same time, then placing the hardware display close to the software display may help increase the sales of both items.

-For example, the information that customers who purchase

-computers also tend to buy antivirus software at the same time is represented in

Association Rule below:

- computer ⇒ antivirus software [support = 2%,confidence = 60%] (5.1)

Rule support and confidence are two measures of rule interestingness. They respectively reflect the usefulness and certainty of discovered rules.

Support:

A support of 2% for Association Rule (5.1) means that 2% of all the transactions under analysis show that computer and antivirus software are purchased together.

Confidence:

A confidence of 60% means that 60% of the customers who purchased a computer also bought the software.

1.8 Association and correlation:

Assosiation rules:

-Association rules to show association analysis is illustrated; this is a useful method to discover interesting relationships within a huge dataset.

- The relations can be represented in the form of association rules or frequent item sets [

-Association rule mining is to find the result rule set on a given dataset (the transaction data set or other sequence pattern-type dataset), a predefined minimum support count s, and a predefined confidence c, given any found rule, and is an association rule where ; X and Y are disjoint.

-The interesting thing about this rule is that it is measured by its support and confidence. Support means the frequency in which this rule appears in the dataset, and confidence means the probability of the appearance of Y when X is present.

- For association rules, the key measures of rule interestingness are rule support and confidence.

- Their relationship is given as follows: support_count(X) is the count of itemset in the dataset, contained X. As a convention, in support_count(X), in the confidence value and support count value are represented as a percentage between 0 and 100.

- The association rule is strong once and . The predefinedminimum support threshold is s, and c is the predefined minimum confidence threshold. The meaning of the found association rules should be explained with caution, especially when there is not enough to judge whether the rule implies causality. It only shows the co-occurrence of the prefix and postfix of the rule.

- The following are the different kinds of rules you can come across:

A rule is a Boolean association rule if it contains association of the presence of the item

A rule is a single-dimensional association if there is, at the most, only one dimension referred to in the rules

-A rule is a multidimensional association rule if there are at least two dimensions referred to in the rules

-A rule is a correlation-association rule if the relations or rules are measured by statistical correlation, which, once passed, leads to a correlation rule.

-A rule is a quantitative-association rule if at least one item or attribute contained in it is quantitative.

Correlation rules:

-In some situations, the support and confidence pairs are not sufficient to filter uninteresting association rules.

-In such a case, we will use support , count , confidence , and correlations to filter association rules.There are a lot of methods to calculate the correlation of an association rule, such as analyses, all-confidence analysis, and cosine.

For a k-itemset , define the all-confidence value of X as:

Lift(X->Y)=confidence(X->Y) /

P(Y)=P(XUY)/(P(X)P(Y))

1.9 **Mining Methods:**

you will learn methods for mining the simplest form of frequent patterns—single-dimensional, single-level, Boolean frequent itemsets, such as those discussed for market basket analysis.

Apriori algorithm for improved efficiency and scalabilitys methods for mining frequent itemsets that, unlike Apriori, do not involve the generation of “candidate” frequent item sets.

The Apriori Algorithm: Finding Frequent Itemsets Using Candidate Generation

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent item sets for Boolean association rules.

The name of the algorithm is based onthe fact that the algorithm uses prior knowledge of frequent itemset properties, as we shall see following.

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support.

The resulting set is denoted L1. Next, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of the database.

To improve the efficiency of the level-wise generation of frequent itemsets, an impor_x0002_tant property called the Apriori property, presented below, is used to reduce the search space. We will first describe this property, and then show an example illustrating its use.

Apriori property: All nonempty subsets of a frequent itemset must also be frequent

The join step:

To find Lk, a set of candidate k-itemsets is generated by joining Lkk 1 with itself. This set of candidates is denoted Ck. Let l1 and l2 be itemsets in Lkk 1.

This set of candidates is denoted Ck. Let l1 and l2 be itemsets in Lkk 1. The notation li[j] refers to the jth item in li (e.g., l1[kk 2] refers to the second to the last item in l1).

By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. For the (k k 1)-itemset, li , this means that the items are sorted such that li[1] < li[2] < ... < li[k k 1].

The join, Lkk 1 on Lkk 1, is performed, where members of Lkk 1 are joinable if their first (k k 2) items are in common. That is, members l1 and l2 of Lkk 1 are joined if (l1[1] = l2[1]) ∧ (l1[2] = l2[2]) ∧...∧(l1[kk 2] = l2[kk 2]) ∧(l1[kk 1] < l2[kk 1]).

The condition l1[kk 1] < l2 1]simply ensures that no duplica are generated. The resulting item formed by joining l1 and l2 is l1| l1[2],..., l1[k k 2], l1[k k 1], l2[k k 1]

Table 5.1 Transactional data for an *AllElectronics* branch.

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

The prune step:

-Ck is a superset of Lk, that is, its members may or may not be frequent,but all of the frequent k-itemsets are included in Ck.

-A scan of the database to determine the count of each candidate in Ck would result in the determination of Lk (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to Lk).

-Ck, however, can be huge, and so this could involve heavy computation. To reduce the size of Ck, the Apriori property is used as follows. Any (k k 1)-itemset that is not frequent cannot be a subset of a frequent k-itemset.

-Hence, if any (k k 1)-subset of a candidate k-itemset is not in Lkk 1, then the candidate cannot be frequent either and so can be removed from Ck. This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets.

Apriori. Let’s look at a concrete example, based on the AllElectronicstransaction database, D, of Table 5.1. There are nine transactions in this database, that is, |D| = 9.

1. In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, C1. The algorithm simply scans all of the transactions in order to count the number of occurrences of each item.

2. Suppose that the minimum support count required is 2, that is, min sup = 2. (Here, we are referring to absolute support because we are using a support count. The corre_x0002_sponding relative support is 2/9 = 22%). The set of frequent 1-itemsets, L1, can then be determined. It consists of the candidate 1-itemsets satisfying minimum support.In our example, all of the candidates in C1 satisfy minimum support.

3. To discover the set of frequent 2-itemsets, L2, the algorithm uses the join L1 on L1 to generate a candidate set of 2-itemsets, C2.8 C2 consists of f|L21| 2-itemsets. Note that no candidates are removed from C2 during the prune step because each subset of the candidates is also frequent.

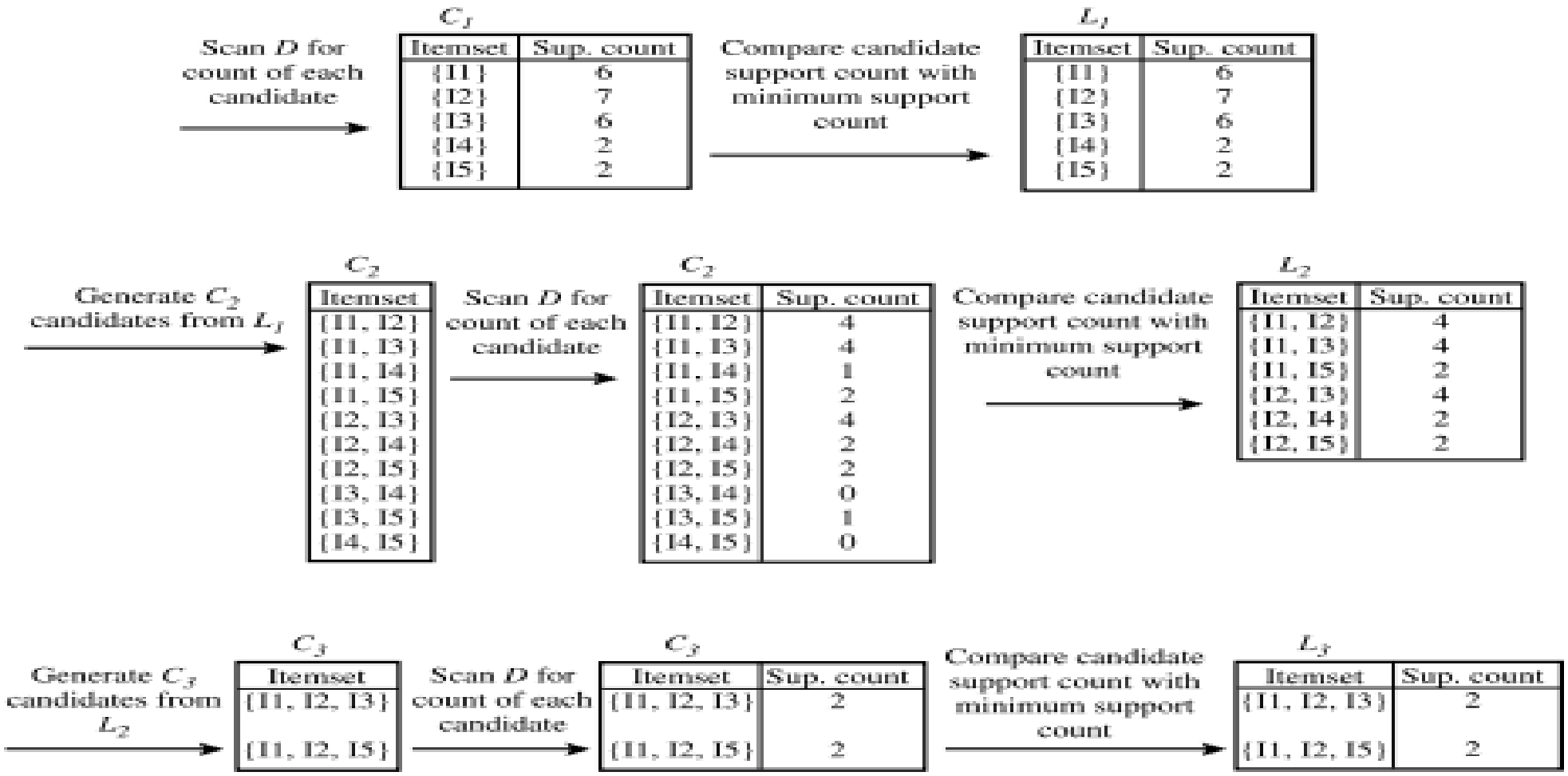

Figure 5.2 Generation of candidate itemsets and frequent itemsets, where the minimum support count is 2.

4. Next, the transactions in D are scanned and the support count of each candidate itemset inC2 is accumulated, as shown in the middle table of the second row in Figure 5.2.

5. The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate 2-itemsets in C2 having minimum support.

6. The generation of the set of candidate 3-itemsets,C3, is detailed in Figure 5.3. From the join step, we first getC3 = L2 on L2 = {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}.

Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that the four latter candidates cannot possibly be frequent. We therefore remove them from C3, thereby saving the effort of unnecessarily obtaining their counts during the subsequent scan of D to determine L3.

Note that when given a candidate k-itemset, we only need to check if its(kk 1)-subsets are frequent since the Apriori algorithm uses a level-wise search strategy. The resulting pruned version of C3 is shown in the first table of the bottom row of Figure 5.2.

7. The transactions in D are scanned in order to determine L3, consisting of those candidate 3-itemsets in C3 having minimum support.

(a) Join: $C_3 = L_2 \bowtie L_2 = \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\} \bowtie \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\} = \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}$.

(b) Prune using the Apriori property: All nonempty subsets of a frequent itemset must also be frequent. Do any of the candidates have a subset that is not frequent?

- The 2-item subsets of $\{I1, I2, I3\}$ are $\{I1, I2\}$, $\{I1, I3\}$, and $\{I2, I3\}$. All 2-item subsets of $\{I1, I2, I3\}$ are members of L_2. Therefore, keep $\{I1, I2, I3\}$ in C_3.
- The 2-item subsets of $\{I1, I2, I5\}$ are $\{I1, I2\}$, $\{I1, I5\}$, and $\{I2, I5\}$. All 2-item subsets of $\{I1, I2, I5\}$ are members of L_2. Therefore, keep $\{I1, I2, I5\}$ in C_3.
- The 2-item subsets of $\{I1, I3, I5\}$ are $\{I1, I3\}$, $\{I1, I5\}$, and $\{I3, I5\}$. $\{I3, I5\}$ is not a member of L_2, and so it is not frequent. Therefore, remove $\{I1, I3, I5\}$ from C_3.
- The 2-item subsets of $\{I2, I3, I4\}$ are $\{I2, I3\}$, $\{I2, I4\}$, and $\{I3, I4\}$. $\{I3, I4\}$ is not a member of L_2, and so it is not frequent. Therefore, remove $\{I2, I3, I4\}$ from C_3.
- The 2-item subsets of $\{I2, I3, I5\}$ are $\{I2, I3\}$, $\{I2, I5\}$, and $\{I3, I5\}$. $\{I3, I5\}$ is not a member of L_2, and so it is not frequent. Therefore, remove $\{I2, I3, I5\}$ from C_3.
- The 2-item subsets of $\{I2, I4, I5\}$ are $\{I2, I4\}$, $\{I2, I5\}$, and $\{I4, I5\}$. $\{I4, I5\}$ is not a member of L_2, and so it is not frequent. Therefore, remove $\{I2, I4, I5\}$ from C_3.

(c) Therefore, $C_3 = \{\{I1, I2, I3\}, \{I1, I2, I5\}\}$ after pruning.

Figure 5.3 Generation and pruning of candidate 3-itemsets, C_3, from L_2 using the Apriori property.

8. The algorithm uses L3 on L3 to generate a candidate set of 4-itemsets, C4. Although the join results in {{I1, I2, I3, I5}}, this itemset is pruned because its subset {{I2, I3,I5}} is not frequent. Thus, C4 = ϕ, and the algorithm terminates, having found all of the frequent item sets.

Algorithm: Apriori Find frequent itemvsets using an iterative level-wise approach based on candidate generation.

Input:

D, a database of transactions;

min sup, the minimum support count threshold.

Output: L, frequent item sets in D.

Method:

(1) L1 = find frequent 1-itemsets(D);

(2) for (k = 2;Lkk 1 6= ϕ;k++) {

(3) Ck = apriori gen(Lkk 1);

(4) for each transaction t ∈ D { // scan D for counts

(5) Ct = subset(Ck, t); // get the subsets of t that are candidates

(6) for each candidate c ∈ Ct

(7) c.count++;

(8) }

(9) Lk = {c ∈ Ck|c.count ≥ min sup}

(10) }

(11) return L = UkLk;

procedure apriori gen(Lkk 1:frequent (k k 1)-itemsets)

(1) for each itemset l1 ∈ Lkk 1

2) for each itemset l2 ∈ Lkk 1

(3) if (l1[1] = l2[1])∧(l1[2] = l2[2])∧...∧(l1[k k 2] = l2[k k 2])∧(l1[k k 1] < l2[k k 1]) then {

(4) c = l1 on l2; // join step: generate candidates

(5) if has infrequent subset(c, Lkk 1) then

(6) delete c; // prune step: remove unfruitful candidate

(7) else add c to Ck;

(8) }

(9) return Ck;

procedure has infrequent subset(c: candidate k-itemset;

Lkk 1: frequent (k k 1)-itemsets); // use prior knowledge

(1) for each (k k 1)-subset s of c

(2) if s 6∈ Lkk 1 then

(3) return TRUE;

(4) return FALSE;

https://youtu.be/Rd1OIzJoh88