1. **What is Data Mining? What is the need for preprocessing?**

Ans: Data mining also known as knowledge discovery from data is the extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amounts of data or databases or repositories.

Need for data preprocessing or preprocessing:

- It improves accuracy and reliability.
- Preprocessing data removes missing or inconsistent data values resulting from human or computer error, which can improve the accuracy and quality of a dataset, making it more reliable.

2. **What are the different types of attributes?**

Ans: Attribute (or dimensions, features, variables) is a data field, representing a characteristic or feature of a data object.

Types of attributes include:

- **Nominal:** categories, states or "name of things".
  Marital status, occupation, ID numbers or zip codes etc.
- **Binary:** Nominal attribute with only 2 states (0,1).
  There are two types:
  Symmetric binary and Asymmetric binary.
- **Ordinal:** values have meaningful order but magnitude between successive values is not known.
- **Numeric:** Quantitative
  Interval scaled: Measured on a scale of equal sized units.
  Ratio: inherent zero point. We can speak of values as being an order of magnitude larger than the unit of measurement.
- **Discrete:** has only finite or countably infinite set of values.
- **Continuous**: has real numbers as attribute values.

3. **What is KDD process?**

Ans: KDD is referred to as Knowledge Discovery in Database and is defined as a method of finding, transforming and refining meaningful data and patterns from a raw database in order to be utilized in different domains or applications.

1. **Data cleaning** (to remove noise and inconsistent data)
2. **Data integration** (where multiple data sources may be combined)[1]
3. **Data selection** (where data relevant to the analysis task are retrieved from the database)
4. **Data transformation** (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance)[2]
5. **Data mining** (an essential process where intelligent methods are applied in order to extract data patterns)
6. **Pattern evaluation** (to identify the truly interesting patterns representing knowledge based on some **interestingness measures**; Section 1.5)
7. **Knowledge presentation** (where visualization and knowledge representation techniques are used to present the mined knowledge to the user)

4. **What is Metadata in data ware house (DWH? What is data integration in DWH?**

Ans: Metadata is simply defined as data about data. The data that is used to represent other data is known as metadata. For example, the index of a book serves as a metadata for the contents in the book.

Data warehouse integration combines data from several sources into a single, unified warehouse. The data warehouse can be accessed by any department within an organization and the data can be easily structured into spreadsheets tables for research and analysis process.

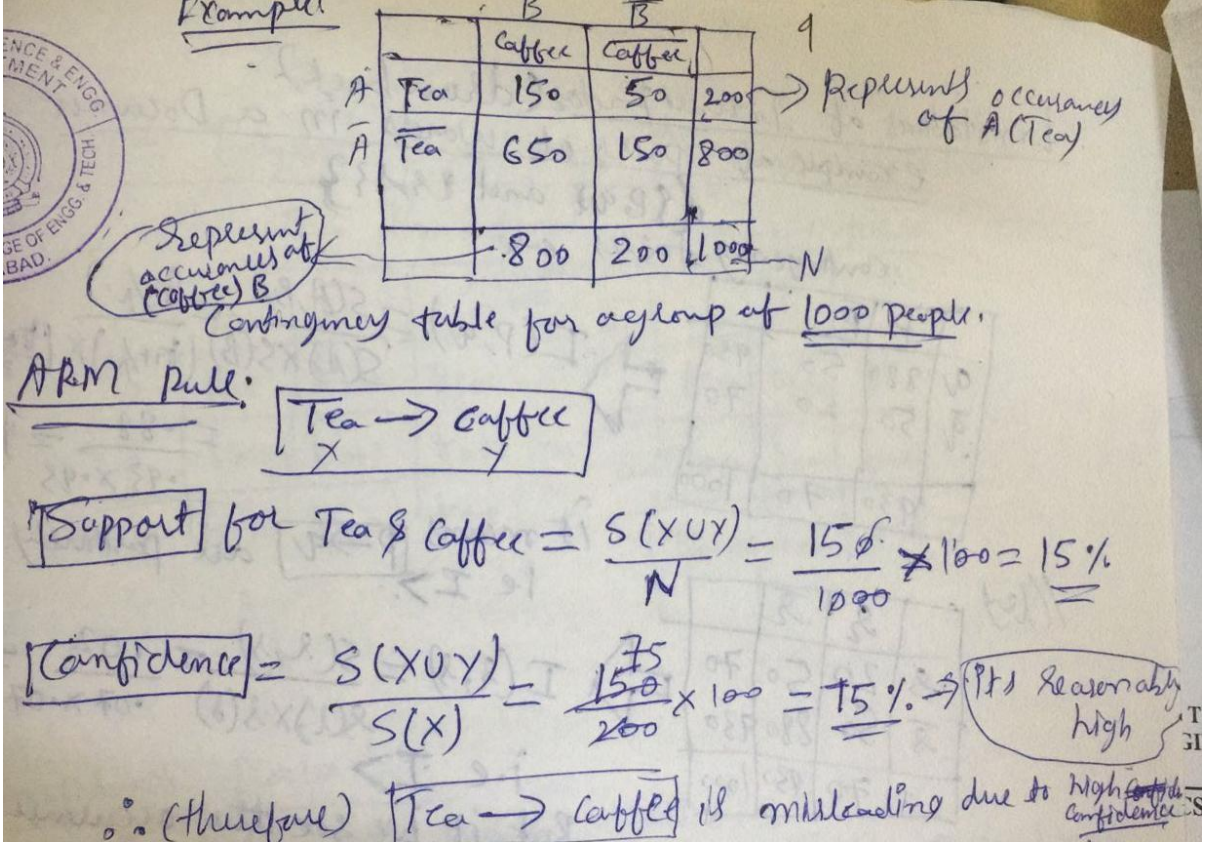
5. **Define Support and Confidence?**

Ans: Support measures the frequency of occurrence of a particular itemset in the dataset.

Support(A) = (Number of transactions containing A) / (Total number of transactions)

Confidence measures the strength of association between two itemsets, A and B.

Confidence(A $\rightarrow$ B) = (Number of transactions containing both A and B) / (Number of transactions containing A)



6. **What are examples of Classifier in Data Mining?**

Ans: Best classifiers for mining data or data mining are:

- SVM (Support Vector Machine)
- K-NN
- Gradient Boosting Classifier
- XG Classifier
- Random forest.
- Naïve Bayes.
- Decision tree

Part- B

8. **Write and Explain Apriori algorithm to find all frequent item sets and strong association rules for the following data base, where min-supp = 60 % and min-confidence = 80 %**

TID	ITEMS
T100	{K, A, D, B}
T200	{D, A, C, E, B}
T300	{C, A, B, E}
T400	{B, A, D}

Explaining the Apriori Algorithm ...

1: Find all large 1-itemsets
2: For (k = 2 ; while L_{k-1} is non-empty; k++)
3 {C_k = apriori-gen(L_{k-1})
4 For each c in C_k, initialise $c.count$ to zero
5 For all records r in the DB
6 {C_r = subset(C_k, r); For each c in C_r, $c.count$++ }
7 Set L_k := all c in C_k whose $count$ >= $minsup$

Here, we are simply scanning through the DB to count

the support for each of our candidates in C_k , throwing

out the ones without enough support, and the rest become

Ans: Given: Min - supp. = 60% & Min - conf. = 80%

TID	ITEMS
T100	{K, A, D, B}
T200	{D, A, C, E, B}
T300	{C, A, B, E}
T400	{B, A, D}

Creating one item sets and calculating supports:

	Frequency	support
A	4	$\frac{4}{4} = 100\%$
B	3	$\frac{3}{4} = 75\%$
C	2	$\frac{2}{4} = 50\%$
D	3	$\frac{3}{4} = 75\%$
E	2	$\frac{2}{4} = 50\%$
K	1	$\frac{1}{4} = 25\%$

as Min - support = 60%:

A	100%
B	75%
D	75%

creating (two-item) sets by calculating supports:

	Frequency	Support
AB	4	$\frac{4}{4} = 100\%$
BD	3	$\frac{3}{4} = 75\%$
AD	3	$\frac{3}{4} = 75\%$

Create (3-item) sets by calculating supports:

	Frequency	Support
ABD	3	$\frac{3}{4} = 75\%$

$\therefore I = \{A, B, D\}$

Listing association rules and calc confidence

$A \rightarrow BD$	$\frac{3}{4} = 75\%$	$A \rightarrow B$	$\frac{4}{4} = 100\%$
$B \rightarrow AD$	$\frac{3}{4} = 75\%$	$B \rightarrow A$	$\frac{4}{4} = 100\%$
$D \rightarrow AB$	$\frac{3}{3} = 100\%$	$A \rightarrow D$	$\frac{3}{4} = 75\%$
$AB \rightarrow D$	$\frac{3}{4} = 75\%$	$D \rightarrow A$	$\frac{3}{3} = 100\%$
$AD \rightarrow B$	$\frac{3}{3} = 100\%$	$B \rightarrow D$	$\frac{3}{4} = 75\%$
$BD \rightarrow A$	$\frac{3}{3} = 100\%$	$D \rightarrow B$	$\frac{3}{3} = 100\%$

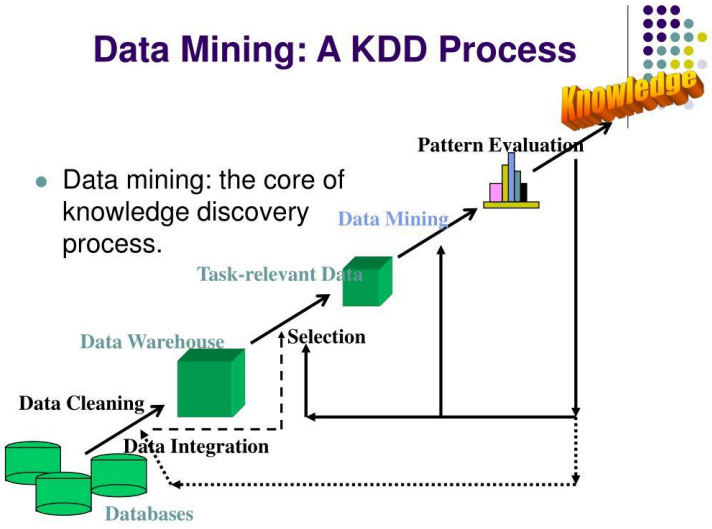
9. **Explain the Knowledge Discovery from Databases?**

Ans: KDD is referred to as Knowledge Discovery in Database and is defined as a method of finding, transforming and refining meaningful data and patterns from a raw database in order to be utilized in different domains or applications.

Some people don't differentiate data mining from knowledge discovery while others view data mining as an essential step in the process of knowledge discovery.

Here is the list of steps involved in the knowledge discovery process :

- **Data Cleaning** – Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.
- **Data Integration** – In this step, the processing of data from multiple heterogeneous sources of data and combining them coherently to retain a unified view of the information is done.
- **Data Selection** – It is defined as the process of determining the appropriate data type and source, as well as suitable instruments to collect data. Data selection precedes the actual practice of data collection.
- **Data Transformation** − In this step, data is transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations.
- **Data Mining** − In this step, intelligent methods are applied in order to extract data patterns.
- **Pattern Evaluation** − In this step, data patterns are evaluated.
- **Knowledge Presentation** − In this step, knowledge is represented.



10. Explain briefly about measuring data similarity and dissimilarity?

Similarity and Dissimilarity

- **Similarity**
  - Numerical measure of how alike two data objects are
  - Value is higher when objects are more alike
  - Often falls in the range [0,1]
- **Dissimilarity** (e.g., distance)
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- **Proximity** refers to a similarity or dissimilarity

- Data matrix
  - n data points with p dimensions
  - Two modes

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - n data points, but registers only the distance
  - A triangular matrix
  - Single mode

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

  - Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
  - Method 1: Simple matching
    - *m*: # of matches, *p*: total # of variables

$$d(i, j) = \frac{p - m}{p}$$

  - Method 2: Use a large number of binary attributes
    - creating a new binary attribute for each of the *M* nominal states

- A contingency table for binary data

Object *i* \ Object *j*	1	0	sum
1	q	r	$q+r$
0	s	t	$s+t$
sum	$q+s$	$r+t$	p

- Distance measure for symmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

- Distance measure for asymmetric binary variables:

$$d(i,j) = \frac{r+s}{q+r+s}$$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$sim_{Jaccard}(i,j) = \frac{q}{q+r+s}$$

- Note: Jaccard coefficient is the same as "coherence":

$$coherence(i,j) = \frac{sup(i,j)}{sup(i) + sup(j) - sup(i,j)} = \frac{q}{(q+r)+(q+s)-q}$$

Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Gender is a symmetric attribute
- The remaining attributes are asymmetric binary
- Let the values Y and P be 1, and the value N 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

11. What is Cosine Similarity?

Ans: Cosine similarity is a metric, helpful in determining how similar the data objects are irrespective of their size. We can measure the similarity between two sentences in python using cosine similarity. In cosine similarity, data objects in a dataset are treated as a vector.

- The formula to find the cosine similarity between two vectors is:

Cos(x,y) = x.y / ||x|| * ||y||

Where,

- x.y = product(dot) of the vectors x and y

- ||x|| and ||y|| = length of the two vectors x and y

- ||x|| * ||y|| = cross product of the two vectors x and y

- A **document** can be represented by thousands of attributes, each recording the *frequency* of a particular word (such as keywords) or phrase in the document.

Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*season*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- Other vector objects: gene features in micro-arrays, …
- Applications: information retrieval, biologic taxonomy, gene feature mapping, ...

Ex: Find the **similarity** between documents 1 and 2.

d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)
d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

$d_1 \bullet d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$

$||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

$||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{0.5} = (17)^{0.5} = 4.12$

$\cos(d_1, d_2) = 0.94$

12. What is minkowski distance measure?

- *Minkowski distance*: A popular distance measure

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and h is the order (the distance so defined is also called L-h norm)

- Properties
  - d(i, j) > 0 if i ≠ j, and d(i, i) = 0 (Positive definiteness)
  - d(i, j) = d(j, i) (Symmetry)
  - d(i, j) ≤ d(i, k) + d(k, j) (Triangle Inequality)
- A distance that satisfies these properties is a metric

Special Cases of Minkowski Distance

- $h = 1$: Manhattan (city block, L_1 norm) distance
  - E.g., the Hamming distance: the number of bits that are different between two binary vectors

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + \ldots + |x_{i_p} - x_{j_p}|$$

- $h = 2$: (L_2 norm) Euclidean distance

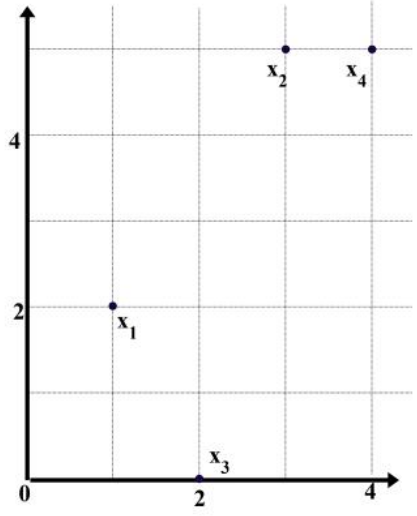
$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

- $h \to \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component (attribute) of the vectors

$$d(i, j) = \lim_{h \to \infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_{f}^{p} |x_{if} - x_{jf}|$$

Dissimilarity Matrices

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean (L_2)

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Supremum

L_∞	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0

13. **What is Sequential pattern mining, explain with an example**?

Ans:

Sequential Pattern Mining: Concepts and Primitives

"*What is sequential pattern mining?*" **Sequential pattern mining** is the mining of frequently occurring ordered events or subsequences as patterns. An example of a sequential pattern is "*Customers who buy a Canon digital camera are likely to buy an HP color printer within a month.*" For retail data, sequential patterns are useful for shelf placement and promotions. This industry, as well as telecommunications and other businesses, may also use sequential patterns for targeted marketing, customer retention, and many other tasks. Other areas in which sequential patterns can be applied include Web access pattern analysis, weather prediction, production processes, and network intrusion detection. Notice that most studies of sequential pattern mining concentrate on *categorical* (or *symbolic*) *patterns*, whereas numerical curve analysis usually belongs to the scope of trend analysis and forecasting in statistical time-series analysis, as discussed in Section 8.2.

The sequential pattern mining problem was first introduced by Agrawal and Srikant in 1995 [AS95] based on their study of customer purchase sequences, as follows: "*Given a set of sequences, where each sequence consists of a list of events (or elements) and each event consists of a set of items, and given a user-specified minimum support threshold of min_sup, sequential pattern mining finds all* **frequent** *subsequences, that is, the subsequences whose occurrence frequency in the set of sequences is no less than min_sup.*"

Let's establish some vocabulary for our discussion of sequential pattern mining. Let $I = \{I_1, I_2, \ldots, I_p\}$ be the set of all *items*. An **itemset** is a nonempty set of items. A **sequence** is an ordered list of **events**. A sequence s is denoted $\langle e_1 e_2 e_3 \cdots e_l \rangle$, where event e_1 occurs before e_2, which occurs before e_3, and so on. Event e_j is also called an **element** of s. In the case of customer purchase data, an event refers to a shopping trip in which a customer bought items at a certain store. The event is thus an itemset, that is, an unordered list of items that the customer purchased during the trip. The itemset (or event) is denoted $(x_1 x_2 \cdots x_q)$, where x_k is an item. For brevity, the brackets are omitted if an element has only one item, that is, element (x) is written as x. Suppose that a customer made several shopping trips to the store. These ordered events form a sequence for the customer. That is, the customer first bought the items in s_1, then later bought the items in s_2, and so on. An item can occur at most once in an event of a sequence, but can occur multiple times in different events of a sequence. The number of instances of items in a sequence is called the **length** of the sequence. A sequence with length l is called an l-**sequence**. A sequence $\alpha = \langle a_1 a_2 \cdots a_n \rangle$ is called a **subsequence** of another sequence $\beta = \langle b_1 b_2 \cdots b_m \rangle$, and β is a **supersequence** of α, denoted as $\alpha \sqsubseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < \cdots < j_n \leq m$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \ldots, a_n \subseteq b_{j_n}$. For example, if $\alpha = \langle (ab), d \rangle$ and $\beta = \langle (abc), (de) \rangle$, where a, b, c, d, and e are items, then α is a subsequence of β and β is a supersequence of α.

A **sequence database**, S, is a set of tuples, $\langle SID, s \rangle$, where SID is a *sequence_ID* and s is a sequence. For our example, S contains sequences for all customers of the store. A tuple $\langle SID, s \rangle$ is said to **contain** a sequence α, if α is a subsequence of s. The **support** of a sequence α in a sequence database S is the number of tuples in the database containing α, that is, $support_S(\alpha) = |\{\langle SID, s \rangle | (\langle SID, s \rangle \in S) \wedge (\alpha \sqsubseteq s)\}|$. It can be denoted as $support(\alpha)$ if the sequence database is clear from the context. Given a positive integer *min_sup* as the **minimum support threshold**, a sequence α is **frequent** in sequence database S if $support_S(\alpha) \geq min\_sup$. That is, for sequence α to be frequent, it must occur at least *min_sup* times in S. *A frequent sequence is called a* **sequential pattern**. A sequential pattern with length l is called an l-**pattern**. The following example illustrates these concepts.

Sequential patterns. Consider the sequence database, S, given in Table 8.1, which will be used in examples throughout this section. Let *min_sup* = 2. The set of *items* in the database is $\{a, b, c, d, e, f, g\}$. The database contains four sequences.

Let's look at *sequence* 1, which is $\langle a(abc)(ac)d(cf)\rangle$. It has five *events*, namely (a), (abc), (ac), (d), and (cf), which occur in the order listed. Items a and c each appear more than once in different events of the sequence. There are nine instances of items in sequence 1; therefore, it has a *length* of nine and is called a 9-*sequence*. Item a occurs three times in sequence 1 and so contributes three to the length of the sequence. However, the entire sequence contributes only one to the *support* of $\langle a\rangle$. Sequence $\langle a(bc)df\rangle$ is a *subsequence* of sequence 1 since the events of the former are each subsets of events in sequence 1, and the order of events is preserved. Consider subsequence $s = \langle(ab)c\rangle$. Looking at the sequence database, S, we see that sequences 1 and 3 are the only ones that *contain* the subsequence s. The support of s is thus 2, which satisfies minimum support.

A sequence database

Sequence_ID	Sequence
1	$\langle a(abc)(ac)d(cf)\rangle$
2	$\langle(ad)c(bc)(ae)\rangle$
3	$\langle(ef)(ab)(df)cb\rangle$
4	$\langle eg(af)cbc\rangle$

Therefore, s is frequent, and so we call it a *sequential pattern*. It is a 3-*pattern* since it is a sequential pattern of length three. ■

This model of sequential pattern mining is an abstraction of customer-shopping sequence analysis. Scalable methods for sequential pattern mining on such data are described in Section 8.3.2, which follows. Many other sequential pattern mining applications may not be covered by this model. For example, when analyzing Web clickstream sequences, gaps between clicks become important if one wants to predict what the next click might be. In DNA sequence analysis, *approximate* patterns become useful since DNA sequences may contain (symbol) insertions, deletions, and mutations. Such diverse requirements can be viewed as *constraint relaxation* or *enforcement*. In Section 8.3.3, we discuss how to extend the basic sequential mining model to *constrained* sequential pattern mining in order to handle these cases.

14. What are major issues in Data mining?

- *Mining different kinds of knowledge in databases:* Because different users can be interested in different kinds of knowledge, data mining should cover a wide spectrum of data analysis and knowledge discovery tasks, including data characterization, discrimination, association and correlation analysis, classification, prediction, clustering, outlier analysis, and evolution analysis (which includes trend and similarity analysis). These tasks may use the same database in different ways and require the development of numerous data mining techniques.
- *Interactive mining of knowledge at multiple levels of abstraction:* Because it is difficult to know exactly what can be discovered within a database, the data mining process should be *interactive*. For databases containing a huge amount of data, appropriate sampling techniques can first be applied to facilitate interactive data exploration. Interactive mining allows users to focus the search for patterns, providing and refining data mining requests based on returned results. Specifically, knowledge should be mined by drilling down, rolling up, and pivoting through the data space and knowledge space interactively, similar to what OLAP can do on data cubes. In this way, the user can interact with the data mining system to view data and discovered patterns at multiple granularities and from different angles.
- *Incorporation of background knowledge:* Background knowledge, or information regarding the domain under study, may be used to guide the discovery process and allow discovered patterns to be expressed in concise terms and at different levels of abstraction. Domain knowledge related to databases, such as integrity constraints and deduction rules, can help focus and speed up a data mining process, or judge the interestingness of discovered patterns.
- *Data mining query languages and ad hoc data mining:* Relational query languages (such as SQL) allow users to pose ad hoc queries for data retrieval. In a similar vein, high-level data mining query languages need to be developed to allow users to describe ad hoc data mining tasks by facilitating the specification of the relevant sets of data for analysis, the domain knowledge, the kinds of knowledge to be mined, and the conditions and constraints to be enforced on the discovered patterns. Such a language should be integrated with a database or data warehouse query language and optimized for efficient and flexible data mining.
- *Presentation and visualization of data mining results:* Discovered knowledge should be expressed in high-level languages, visual representations, or other expressive forms so that the knowledge can be easily understood and directly usable by humans. This is especially crucial if the data mining system is to be interactive. This requires the system to adopt expressive knowledge representation techniques, such as trees, tables, rules, graphs, charts, crosstabs, matrices, or curves.
- *Handling noisy or incomplete data:* The data stored in a database may reflect noise, exceptional cases, or incomplete data objects. When mining data regularities, these objects may confuse the process, causing the knowledge model constructed to overfit the data. As a result, the accuracy of the discovered patterns can be poor. Data cleaning methods and data analysis methods that can handle noise are required, as well as outlier mining methods for the discovery and analysis of exceptional cases.

- *Pattern evaluation—the interestingness problem:* A data mining system can uncover thousands of patterns. Many of the patterns discovered may be uninteresting to the given user, either because they represent common knowledge or lack novelty. Several challenges remain regarding the development of techniques to assess the interestingness of discovered patterns, particularly with regard to subjective measures that estimate the value of patterns with respect to a given user class, based on user beliefs or expectations. The use of interestingness measures or user-specified constraints to guide the discovery process and reduce the search space is another active area of research.

Performance issues: These include efficiency, scalability, and parallelization of data mining algorithms.

- *Efficiency and scalability of data mining algorithms:* To effectively extract information from a huge amount of data in databases, data mining algorithms must be efficient and scalable. In other words, the running time of a data mining algorithm must be predictable and acceptable in large databases. From a database perspective on knowledge discovery, efficiency and scalability are key issues in the implementation of data mining systems. Many of the issues discussed above under *mining methodology and user interaction* must also consider efficiency and scalability.
- *Parallel, distributed, and incremental mining algorithms:* The huge size of many databases, the wide distribution of data, and the computational complexity of some data mining methods are factors motivating the development of **parallel and distributed data mining algorithms**. Such algorithms divide the data into partitions, which are processed in parallel. The results from the partitions are then merged. Moreover, the high cost of some data mining processes promotes the need for **incremental** data mining algorithms that incorporate database updates without having to mine the entire data again "from scratch." Such algorithms perform knowledge modification incrementally to amend and strengthen what was previously discovered.