Unit 4

CONCEPT DESCRIPTION : CHARACTERIZATION AND COMPARISON

INTRODUCTION

From a data analysis point of view, data generalization is a form of descriptive data mining, which describes data in a concise and summarative manner and presents interesting general properties of the data. In this chapter, we look at descriptive data mining in greater detail.

4.1 What is Concept Description?

The simplest kind of descriptive data mining is concept description. A concept usually refers to a collection of data such as frequent_buyers, graduate_students, and so on. As a data mining task, concept description is not simple enumeration of the data. Instead, concept description generates description for characterization, and comparison of the data. It is sometimes called class description, when the concept to be described refers to a class of objects. Characterization provides a concise and succinct summarization of the given collection of data, while concept or class comparison provides description comparing two or more collections of data. Since concept description involves both characterization and comparison, we will study techniques for accomplishing each of these tasks.

Concept description has close ties with data generalization, given the large amount of data stored in data bases, it is useful to be able to describe concepts in concise and succinct terms at generalized at multiple levels of abstraction. Allowing data sets to be generalized at multiple levels of abstraction facilitates users in examining the general behavior of the data.

What are the differences between concept description and the online analytical processing? The fundamental differences between the two involve the following.

Complex data types and aggregation:

Data warehouses and OLAP tools are based on the multidimensional data model that views data in the form of a data cube, consisting of dimensions and measures. However the possible data types of the dimensions and measures for most commercial versions of these systems are restricted. Many current OLAP systems confine dimensions to nonnumeric data. Similarly measures in current OLAP systems apply only to numeric data. In contrast, for concept formation, the data base attributes can be of various data types, including numeric, nonnumeric, spatial, text or image.

Use-control versus automation:

On-line analytical processing in data warehouses is a purely user-controlled process. The selection of dimensions and the applications of OLAP operations, such as drill-down, roll-up, slicing, and dicing are directed and controlled by the users. Although the control in most OLAP systems is quite user-friendly, user do require to find a satisfactory description of the data, users may need to specify a long sequence of the OLAP operations. In contrast, concept description in data mining strives for more automated process that helps the user determine which dimensions should be included in the analysis, and the degree to which the given data set should be generalized in order to produce an interesting summarization of the data.

4.2 Data Generalization and Summarization-Based Characterization

Data generalization is a process that abstracts a large set of task-relevant data in the data base from a relatively low conceptual level to higher conceptual levels. Methods for the efficient and flexible generalization of large data sets can be categorized according to two approaches

1. Data cube approach and
2. The attribute induction approach.

4.2.1 Attribute-Oriented induction

The data cube approach can be considered as a data ware-house based, permutation –oriented, materialized-view approach. It performs off-line aggregation before an OLAP or data mining query is submitted for processing. On the other hand the attribute oriented induction approach, at east in its initial proposal, is a relational database query-oriented, generalization-based, online data analysis technique.

The general idea of attribute oriented induction is t first collect the task relevant data using a relational database query and then perform generalization based on the examination of the number of distinct values of each attribute in the relevant set on data. The generalization is performed based by either attribute removal or attribute generalization. Aggregation is performed by merging identical generalized tuples and accumulating their respective counts. This reduces the size of the generalized data set. The resulting generalized relation can be mapped into different forms for presentation to the user, such as charts or rules.

Example 4.1: Specifying a data mining query for characterization with DMQL:

Suppose if user want to describe the general characteristics of graduate students in Big university database, attributes are name, gender, major, birth_place, birth_date, residence, phone#, gpa

A Data mining query for this characterization can be expressed in the Data mining query language as follows:

```
use Big_University_DB
mine characteristics as "Science_Students"
in relevance to name, gender, major, birth_place, birth_date, residence, phone#, gpa
from student
where status in "graduate"
```

"What is the first step of attribute-oriented induction" first, data focusing should be performed prior to attribute_oriented induction. The data are collected based on the information provided in the data mining query. Since a data mining query is usually relevant to only a portion of the database, selecting the relevant set of data not only makes mining more efficient, but also derives more meaningful results than mining on the entire database.

Specifying the set of relevant attributes may be difficult for the user. A user may select only a few attributes that he feels may be important, while missing others that could also play a role in the description.

At the other extreme, a user may introduce too many attributes by specifying all of the possible attributes with the clause **"in relevance to *"**. In this case, all of the attributes in the relation specified by the **from** clause would be included in the analysis.

"What does the Where status in "graduate" clause mean?" This **where** clause implies that a concept hierarchy exists for the attribute status, Such a concept hierarchy organizes primitive-level data values for status, such as "M.Sc.", "M.A.", "M.B.A." , "Ph.D.", "B.Sc.", "B.A.", into higher conceptual levels such as *"graduate"* and *"under graduate"*.

Example4.2: Transforming a data mining query to a relational query

The data mining query presented in Example4.1 is transformed into the following relational query for the collection of the task-relevant set of data:

```
use_Big_University_DB
Select name, gender, major, birth_place, birth_date, residence, phone#, gpa
from student
where status in {"Msc", "MBA", "PhD" }
```

The transformed query is executed against the relational database, Big-university_DB, and returns the data shown in table 4.1. This table is called the (task_relevant) **initial working relation.**

Table 4.1 Initial working relation: a collection of task_relevant data

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone#	GPA
Jim Woodman	M	CS	Vancouver, BC, Canada	8-12-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83

Now the data is ready for attribute_oriented induction, how is attribute-oriented induction performed?

The essential operation of attribute_oriented induction is data generalization, which can be performed in either of two ways on the initial working relation;

1. Attribute Removal
2. Attribute generalization

Attribute removal is based on the following rule : if there is a large set of the distinct values for an attribute of the initial working relation, but either there is no generalization operator on the attribute, or its higher-level concepts are expressed in terms of other attributes, then the attribute should be removed form the working relation.

Attribute generalization is based on the following rule: if there is a large set of attributes in the initial working relation and there exists a set of generalization operators on the attribute, then a generalization operator should be selected an applied to the attribute. Use of a generalization operator to generalize an attribute value within a tuple, or rule, In the working relation will make the rule cover more of the original data tuples, thus generalizing the concept it represents. This corresponds to the generalization rule known as climbing generalization trees in learning form examples, or concept tree ascension.

There are many possible ways to control a generalization process. We will describe two common approaches.

The first technique is called ***attribute generalization threshold control,*** either sets one generalization threshold for all of the attributes, or sets one threshold fro each attribute. If the number of distinct values in an attribute is greater than the attribute threshold, further attribute removal or attribute generalization should performed. Data mining systems should typically have a default attribute threshold value ranging from 2 to 8 and should allow experts and users to modify the threshold values as well.

The second technique called ***generalized relation threshold control*** sets a threshold for the generalized relation. If the number if tuples in the generalized relation is greater than the threshold, further generalization should be performed. Otherwise no further generalization should be performed.

These two techniques can be applied in sequence: first apply the attribute threshold control technique to generalize each attribute, and then apply relation threshold control to further reduce the size of the generalized relation.

Example 4.3: Attribute-oriented induction: The attribute-oriented induction is performed on the initial working relation of table 4.1. For each attribute of the relation, the generalized proceeds as follows:

1. **name:** Since there is large number of distinct values for name and there is no generalized operation defined on it, this attribute is removed.
2. **gender:** Since there are only two distinct values for gender this attribute is retained and no generalization is performed on it.
3. **major:** Suppose that the attribute gereralization threshold value set to 5 and there are over 20 distinct values for major in the initial working relation. By attribute generalization and attribute generalization control, major is generalized by climbing the given hierarchy.
4. **birth-place:** This attribute has a large number of distinct values therefore we would like to generalize it. Suppose that a concept hierarchy for birth-place defined as city<state<country. If the number of distinct values for country in the initial working relation is greater than the attribute generalized threshold, then birth-place should be removed since even though a generalized operator exists for it, the generalization threshold should not be satisfied. If instead, the number of distinct values for country is less than the attribute generalization threshold, then the birth-place should be generalized to birth-country.

Table 4.2 A generalized relation obtained by attribute_oriented induction on the data of Table 4.1.

Gender	Major	Birth_region	Age_range	Residence	GPA	Count
M	Science	Canada	20-25	Richmond	Very-good	16
F	Science	Foreign	25-30	Burnaby	Excellent	22
...	...	...	...	...	...	...

1. **birth-date:** Suppose that a hierarchy exists that can generalize birth-date to age, and age to age-range and that the number of age ranges is small with respect to the attribute generalization threshold. Generalization of birth-date should be take place.
2. **residence:** Suppose that residence is defined by the attributes number,street,residence-city,residence-state,and residence-country. The number of distinct values for number and street will likely be very high, since these concepts are quite low level. The attributes number and street should therefore be removed, so that residence is then generalized to residende-city,which contains fewer distinct values.
3. **phone#:** As with the attribute name above, this attribute contains too many distinct values and should therefore be removed in generalization.
4. **gpa:** Suppose that a concept hierarchy exists for gpa that groups values fro grade point average into numerical intervals, which in turn are grouped into descriptive values. Therefore attribute can be generalized.

4.2.2 Efficient Implementation of Attribute-Oriented induction

How is attribute-oriented induction actually implemented? The general algorithm.

Algorithm: Attribute-oriented-induction, mining generalized characteristics in a relational database given a users data mining request.

INPUT:

i) DB, a relational database
ii) DM query, a data mining query
iii) a-list, list of attributes
iv) Gen (a$_i$), a set of concept hierarchies or generalization operators on attribute a$_i$.
v) a-gen-thresh (a$_i$), attribute generalization thresholds for each a$_i$.

OUTPUT:

P, a Prime-generalized-relation.

METHOD:

1. W ⟵ get-task-relevant-data(DM Query, DB)
2. Prepare-for-generalization(W);
   a) Scan W and collect the distinct values for each attribute a$_i$.
   b) For each attribute a$_i$, determine whether a$_i$ should be removed and if not, compute its min-desired level L$_i$ based on its given or default attribute threshold and determine the mapping pairs(v, v') where
      v- distinct value of a$_i$ in W
      v' – its corresponding generalized value at level L$_i$
3. P ⟵ generalization(W)
   The prime-generalized-relation, p is derived by replacing each value v in W. While accumulating count and computing any other aggregate values.

This can be implemented using either of 2 ways.

a) For each generalized tuple, insert the tuple into a sorted Prime relation P by a Binary Search: If the tuple is already in P, simply increase its count and other aggregate values. Otherwise, insert it into P.
b) The Prime relation can be coded as an m-dimensional array, where m is no. of attributes in P.

The insertion of a generalized tuple is performed by measure aggregation in the corresponding array element.

Efficiency of this algorithm is analyzed as follows:

1. Step 1 of the algorithm is essentially a relational query to collect the task relevant data into working relation, W. Its processing efficiency depends on the query processing methods used. This step is expected to have a good performance.
2. Step 2 collects statistics on the working relation. This requires scanning the relation at most once. The cost for computing the minimum desired level and determining the mapping pairs(v, v') for each attribute is dependent on the number of distinct values for each attribute and is smaller than n, the number of tuples in the initial relation.
3. Step 3 derives Prime relation P. This is performed by inserting generalized tuples into p. There are a total of n tuples in w and p tuples in P. For each tuple t in W, we substitute its attribute values based on the derived mapping-pairs. This results in a generalized tuple t'.

The overall time complexity is O(n) for all of the generalized tuples.

The data cube implementation of attribute oriented induction can be performed in two ways.

Construct a data cube on the fly for the given data mining query:

This method constructs a data cube dynamically based on the task-relevant set of data. This is desirable if either the task-relevant data set is too specific to match any predefined data cube, or it is not very large. Since such data cube is computed only after the query is submitted, the major motivation for constructing such a data cube is to facilitate efficient drill-drown analysis. With such a data cube, drilling down below the level of the prime relation will simply require retrieving data from the cube, or performing minor generalization from the primitive-level data.

Use a predefined data cube: DMP

An alternative method is to construct a data cube before a data mining query is posed to the system, and use this predefined cube fro subsequent data mining. This is desirable if the granularity of the task-relevant data can match that of the predefined data cube and the set of task-relevant data is quite large. Since such a data cube is pre computed, if facilitates attribute relevance analysis, attribute oriented induction, slicing and dicing, roll-up, and drill down.

4.2.3 Presentation of the Derived Generalization

Attribute-oriented induction generates one or a set of generalized descriptions. How can these descriptions be visualized? The descriptor can be presented to the user in a number of different ways. Generalized descriptions resulting from a attribute oriented induction are most commonly displayed in the form of a generalized relation.

Example 4.4 : Suppose that attribute_oriented induction was performed on a *sales* relation of the All Electronics database, resulting in the generalized description of Table 4.3 for sales in 1999.

The description is shown in the form of a generalized relation.

Descriptions can also be visualized in the from of cross-tabulations or cross-tabs. In a two dimensional cross tab, each row represents a value from an attribute, and each column represents a value from another attribute. In an n-dimensional cross tab the columns may represent the values of more than one attribute, with subtotals shown for attribute-value groupings. It is easy to map directly from a data cube structure to a cross tab.

Example 4.5 : The generalized relation shown in Table 4.3 can be transformed into 3-D cross-tabulation shown in Table 4.4.

Table 4.3 A generalized relation for the sales in 1999.

Location	Item	Sales (in million dollars)	Count (in thousands)
Asia	TV	15	300
Europe	TV	12	250
North_America	TV	28	450
Asia	Computer	120	1000
Europe	Computer	150	1200
North_America	Computer	200	1800

Table 4.4 A crosstab for the sales in 1999.

Location \ Item	TV		Computer		Both-items	
	Sales	Count	Sales	Count	Sales	Count
Asia	15	300	120	1000	135	1300
Europe	12	250	150	1200	162	1450
North_America	28	450	200	1800	228	2250
All_regions	45	1000	470	4000	525	5000

Generalized data can be presented graphically using bar charts, pie charts, and curves. Visualization with graphs us popular in data analysis. Such graphs and curves can represent 2-D or 3-D data.

Example 4.6 : The sales data of the cross tab shown in Table 4.4 can be transformed into the bar chart representation of figure 4.2 and the pie chart, representation of figure 4.3

Finally a 3-D generalized relation or cross tab can be represented by a 3-D data cube. Such a 3-D cube view is an attractive tool for cube browsing.

Example 4.7 : Consider the data cube for the dimensions *item*, *location*, and *cost*. The *size* of a cell represents the count of the corresponding cell, while the *brightness* of the cell can be used to represent another measure of the cell, such as *sum (sales)*. Pivoting, drilling and slicing-and-dicing operations can be performed on the data cube browser by mouse clicking.

A generalized relation may also be represented in the form of logic rules. Typically, each generalized tuple represents a rule disjunct. Since data in a large data base usually span a diverse range of distributions, a single generalization tuple is unlikely to cover, or represent, 100% of the initial working relation tuples, or cases.

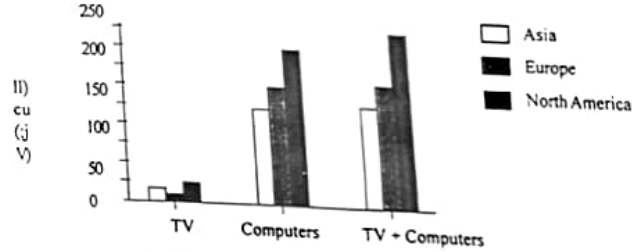

Figure 4.2 Bar chart representation of the sales in 1999.

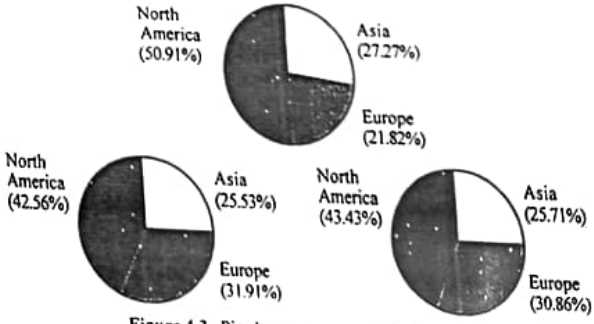

Figure 4.3 Pie chart representation of the sales in 1999.

Thus quantitative information, such as the percentage of data tuples that satisfy the left-hand and right-hand side of the rule, should be associated with each rule. A logic rule that is associated with quantitative information is called **a quantitative rule.**

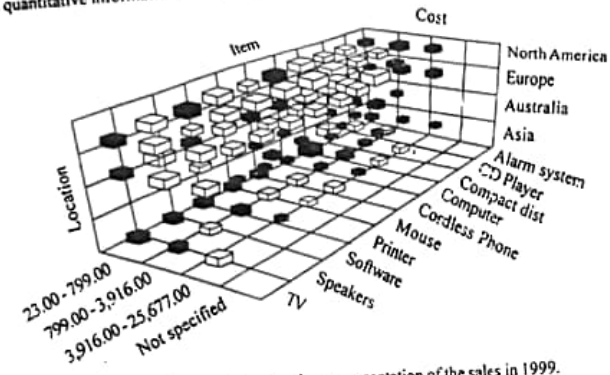

Figure 4.4 A 3-D cube view representation of the sales in 1999.

To define a quantitative characteristic rule, we introduce the **t-weight** as an interestingness measure that describes the typicality of each *disjunct* in the rule, or of each tuple in the corresponding generalized relation. The measure is defined as follows. Let the class of objects that is to be characterized be called the target class. Let qa be a generalized tuple describing the target class. The t-weight for qa is the percentage of tuples of the target class from the initial working relation that are covered by qa. Formally we have

$$\text{t-weight} = \text{count}(qa) / \sum_{i=1}^{n} \text{count}(q_i), \quad \text{---------- (4.1)}$$

where n is the no. of tuples for the target class in the generalized relation; $q_1, \ldots, q_n$ are tuples for the target class in the generalized relation; and q_a is in $q1, \ldots, qn$. Obviously the range for the t-weight is [0.0, 1.0] or [0%, 100%].

A quantitative characteristic rule can be represented either in logic from by associating the corresponding t-weight value with each *disjunct* covering the target class or in the relational table or crosstab form by changing the count values in these tables for tuples of the target class to the corresponding t-weight values.

Each disjunct of a quantitative characteristic rule represents a condition. In general, the disjunction of these conditions forms a necessary condition of the target class, since the condition is derived based on all of the cases of the target class; that is, all tuples of the target class must satisfy this condition. However the rule may not be a sufficient condition of the target class, since a tuple satisfying the same condition could belong to another class. Therefore, the rule should be expressed in the form

$$\forall X, \text{target\_class}(X) \Leftrightarrow$$
$$\text{Condition}_1(X)\,[t : w_1, d : w'_1] \vee \ldots \vee \text{Condition}_n(X)\,[t : w_n, d : w'_n] \quad \text{---------- (4.2)}$$

Example 4.8 : The cross tab shown in Table 4.4 can be transformed into logic rule form. Let the target class be the set of computer items. The corresponding characteristic rule, in logic form is

$$\forall X, \text{item}(X) = \text{"computer"} \Rightarrow (\text{location}(X) = \text{"Asia"})\,[t : 25.00\%] \vee (\text{Location}(X) = \text{"Europe"})\,[t : 30.00\%] \vee (\text{location}(X) = \text{"North America"})\,[t : 45.00\%] \quad \text{---------- (4.3)}$$

The first t-weight value of 25.00% is obtained by 1000, the value corresponding to the count slot for ("Asia, computer"), divided by 4000, the value corresponding to count slot for ("all_regions, computer"). The t-weights of the other two disjuncts were similarly derived.

How can the t-weight and interestingness measures in general be used by the data mining system to display only the concept description that it objectively evaluates as interesting? A threshold can be set for this purpose. For example, if the t-weight of a generalized tuple is lower than the threshold, then the tuple is considered to represent only a negligible portion of the data base and can therefore be ignored as uninteresting. Ignoring such negligible tuples does not mean that they should be removed from the intermediate results since they may be contribute to subsequent further exploration of the data by the user via interactive rolling op or drilling down of other dimensions and levels of abstraction. Such a threshold may be referred as a significance threshold or support threshold where the latter is popularly used in association rule mining.

4.3 Analytical Characterization: Analysis of Attribute Relevance

"What if I am not sure which attribute to include for class characterization and class comparison? I may end up specifying too many attributes, which could slow down the system considerably." Measures of attribute relevance analysis can be used to help identify irrelevant or weakly relevant attributes that can be excluded from the concept description process. The incorporation of this preprocessing step into class characterization or comparison is referred to as analytical characterization or analytical comparison, respectively. This section describes a general method of attribute relevance analysis and its integration with attribute-oriented induction.

4.3.1 Why to Perform Attribute Relevance Analysis?

The first limitation of class characterization for multidimensional data analysis in data warehouses and OLAP tools is the handling of complex objects. The second limitation is the lack of an automated generalization process: the user must explicitly tell the system which dimensions should be included in the class characterization and to how high a level each dimension should be generalized. Actually, each step of generalization or specialization on any dimension must be specified by the user.

Usually, it is not difficult for a user to instruct a data mining system regarding how high a level each dimension should be generalized. For example, users can set attribute generalization thresholds for this, or specify which level a given dimension should reach, such as with the command "generalize dimension location to the country level". Even without explicit user instruction, a default value such as 2 to 8 can be set by the data mining system, which would allow each dimension to be generalized

to a level that contains only 2 to 8 distinct values. If the user is not satisfied with the current level of generalization, she can specify dimension on which drill-down or roll-up operations should be applied.

It is nontrivial, however, for users to determine which dimensions should be included in the analysis of class characteristics. Data relations often contain 50 to 100 attributes, and a user may have little knowledge regarding which attributes or dimensions should be selected for effective data mining. A user may include too few attributes in the analysis, causing the resulting mined description to be incomplete. On the other hand, a user may introduce too many attributes for analysis (e.g., by indicating "**in relevance to** *", which includes all the attributes in the specified relations).

Methods should be introduced to perform attribute (or dimension) relevance analysis in order to filter out statistically irrelevant or weakly relevant attributes, and retain or even rank the most relevant attributes for the descriptive mining task at hand. Class characterization that includes such analysis of attribute/dimension relevance called **analytical characterization**. Class comparison that includes such analysis is called **analytical comparison**.

Intuitively, an attribute or dimension is considered *highly relevant* with respect to a given class if it is likely that the values of the attribute or dimension may be used to distinguish the class from others. For example, it is unlikely that the color of an automobile can be used to distinguish expensive for cheap cars, but the model, make, style, and number of cylinders are likely to be more relevant attributes. Moreover, even within the same dimension, different levels of concepts may have dramatically different powers for distinguishing a class from others. For example, in the *birth_date* dimension, *birth_day* and *birth_month* are unlikely to be relevant to the *salary* of employees. However, the *birth_decade* (i.e., age interval) may be highly relevant to the *salary* of employees. This implies that the analysis of dimension relevance should be performed at *multilevels of abstraction*, and only the most relevant levels of a dimension should be included in the analysis.

Above we said that attribute/dimension relevance is evaluated based on the ability of the attribute/dimension to distinguish objects of a class from others. When mining a class comparison (or discrimination), the target class and the contrasting classes are explicitly given in the mining query. The relevance analysis should be performed by comparison of these classes, as we shall see below. However, when mining class characteristics, there is only one class to be characterized. That is, no contrasting class is specified. It is therefore not obvious what the contrasting class should be for use in the relevance analysis. In this case, typically, the contrasting class is taken to be the set of *comparable data in the database that excludes the set of data to be characterized*. For example, to characterize graduate students, the contrasting class can be composed of the set of undergraduate students.

4.3.2 Methods of Attribute Relevance Analysis

There have been many studies in machine learning, statistics, fuzzy and rough set theories, and so on, on attribute relevance analysis. The general idea behind attribute relevance analysis is to compute some measure that is used to quantify the relevance of an attribute with respect to a given class or concept. Such measures include information gain, the Gini index, uncertainty, and correlation coefficients.

Here we introduce a method that integrates an *information gain* analysis technique (such as presented in the ID3 and C4.5 algorithms for learning decision trees) with a dimension-based analysis method. The resulting method removes the less informative attributes, collecting the more informative ones for use in concept description analysis.

"How does the information gain calculation work?" Let S be a set of *training samples*, where the class label of each sample is known. Each sample is in fact a tuple. One attribute is used to determine the class of the training samples. For instance, the attribute status can be used to define the class label of each sample as either "*graduate*" or "*undergraduate*". Suppose that there are m classes. Let S contain s_i samples of class C_i, for i = 1,...,m. An arbitrary sample belongs to class C_i with probability s_i / S, where S is the total number of samples in set S. The **expected information** needed to classify a given sample is

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{S} \log_2 \frac{s_i}{S} \qquad \text{---------- (4.4)}$$

Attribute A with values $\{a_1, a_2, \ldots, a_v\}$ can be used to partition S into the subsets $\{S_1, S_2, \ldots, S_v\}$, where S_j contains those samples in S that have value a_j of A. Let S_j contains s_{ij} samples of class C_i. The expected information based on this partitioning by A is known as the **entropy** of A. It is the weighted average:

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{S} I(s_{1j}, \ldots, s_{mj}) \qquad \text{---------- (4.5)}$$

The **information gain** obtained by this partitioning on A is defined by

$$\text{Gain}(A) = I(s_1, s_2, \ldots, s_m) - E(A) \qquad \text{---------- (4.6)}$$

In this approach to relevance analysis, we compute the information gain for each of the attributes defining the samples in S. The attribute with the highest information gain is considered the most discriminating of the given set. By computing the information gain for each attribute, we therefore obtain a ranking of the attributes. This ranking can be used for relevance analysis to select the attributes to be used in concept description.

Attribute relevance analysis for concept description is performed as follows:

1. **Data Collection:** Collect data for both the target class and the contrasting class by query processing. For class comparison, both the *target class* and the *contrasting class* are provided by the user in the data mining query. For class characterization, the *target class* is the class to be characterized, whereas the *contrasting class* is the set of comparable data that are not in the target class.

2. **Preliminary relevance analysis using conservative AOI:** This step identifies a set of dimensions and attributes on which the selected relevance measures is to be applied. Since different levels of a dimension may have dramatically different relevance with respect to given class, each attribute defining the conceptual levels of the dimension should be included in the relevance analysis in principle. Attribute-Oriented Induction (AOI) can be used to perform some preliminary relevance on the data by removing or generalizing attributes

having a very large number of distinct values (such as *name* and *phone*). Such attributes are unlikely to be found useful for concept description. To be conservative, the AOI performed here should employ attribute generalization thresholds that are set reasonably large so as to allow more (but not all) attributes to be considered in further relevance analysis by the selected measure (Step 3 below). The relation obtained by such an application of AOI is called the candidate relation of the mining task.

3. **Remove irrelevant and weakly relevant attributes using the selected relevance analysis measure:** Evaluate each attribute in the candidate relation using the selected relevance analysis measure. The relevance measure used in this step may be built into the data mining system or provided by the user. For example, the information gain measure described above may be used. The attributes are then sorted (i.e., ranked) according to their computed relevance to the data mining task. Attributes that are not relevant or are weakly relevant to the task are then removed. A threshold may be set to define "weakly relevant." This step results in an initial target class working relation and an initial contrasting class working relation.

4. **Generate the concept description using AOI:** Perform AOI using a less conservative set of attribute generalization thresholds. If the descriptive mining task is class characterization, only the initial target class working relation is included here. If the descriptive mining task is class comparison, both the initial target class working relation and the initial contrasting class working relation are included.

The complexity of this procedure is similar to the algorithm in figure 4.1 since the induction process is performed twice, that is, in preliminary relevance analysis (Step 2) and on the initial working relation (Step 4). The statistics used in attribute relevance analysis with the selected measure (Step 3) may be collected during the scanning of the database in Step 2.

4.3.3 Analytical Characterization: An Example

If the mined concept descriptions involve many attributes, analytical characterization should be performed. This procedure first removes irrelevant or weakly relevant attributes prior to performing generalization. Let's examine an example of such an analytical mining process.

Example 4.9: Suppose that we would like to mine the general characteristics describing graduate students at *Big-University* using analytical characterization. Given are the attributes *name, gender, major, birth_place, birth_date, phone#,* and *gpa.*

"How is the analytical characterization performed?" In Step 1, the target class data are collected, consisting of the set of graduate students. Data for a contrasting class are also required in order to perform relevance analysis. This is taken to be the set of undergraduate students.

In Step 2, preliminary relevance analysis is performed via attribute removal and attribute generalization by applying attribute-oriented induction with conservative attribute generalization thresholds. Similar to example 4.3, the attributes *name* and *phone#* are removed because their number of distinct values exceeds their respective attribute analytical thresholds. Also as in example 4.3, concept hierarchies are used to generalize *birth_place* to *birth_country*, and *birth_date* to *age_range*. The attributes *major* and *gpa* are also generalized to

Table 4.5 Candidate relation obtained for analytical characterization: target class (graduate students).

gender	major	birth_country	age_range	gpa	count
M	Science	Canada	20-25	Very_good	16
F	Science	Foreign	25-30	Excellent	22
M	Engineering	Foreign	25-30	Excellent	18
F	Science	Foreign	25-30	Excellent	25
M	Science	Canada	20-25	Excellent	21
F	Engineering	Canada	20-25	Excellent	18

Table 4.6 Candidate relations obtained for analytical characterization: contrasting class (undergraduate students).

gender	major	birth_country	age_range	gpa	count
M	Science	Foreign	<20	Very_good	18
F	Business	Canada	<20	Fair	20
M	Business	Canada	<20	Fair	22
F	Science	Canada	20-25	Fair	24
M	Engineering	Foreign	20-25	Very_good	22
F	Engineering	Canada	<20	Excellent	24

higher abstraction levels using the concept hierarchies described in Example 4.3. Hence, the attributes remaining for the candidate relation are *gender, major, birth_country, age_range,* and *gpa.* The resulting relation is shown in tables 4.5 and 4.6.

In Step 3, the attributes in the candidate relation are evaluated using the selected relevance analysis measure, such as information gain. Let C_1 correspond to the class *graduate* and C_2 correspond to the class *undergraduate.* There are 120 samples of class *graduate* and 130 samples of class *undergraduate.* To compute the information gain of each attribute, we first use Equation (4.4) to compute the expected information needed to classify a given sample:

$$I(s_1, s_2) = I(120,130) = -120/250 \log_2 120/250 - 130/250 \log_2 130/250 = 0.9988$$

Next, we need to compute the entropy of each attribute. Let's try the attribute *major.* We need to look at the distribution of *graduate* and *undergraduate* students for each value of *major.* We compute the expected information for each of these distributions.

For *major* = "*Science*":

$s_{11} = 84 \quad s_{21} = 40 \quad I(s_{11}, s_{21}) = 0.9183$

For *major* = "*Engineering*":

$s_{12} = 36 \quad s_{22} = 46 \quad I(s_{12}, s_{22}) = 0.9892$

For *major* = "*Business*":

$s_{13} = 0 \quad s_{23} = 42 \quad I(s_{13}, s_{23}) = 0$

Using Equation (4.5), the expected information needed to classify a given sample if the samples are partitioned according to *major* is

$$E(major) = 126/250\ I(s_{11}, s_{21}) + 82/250\ I(s_{12}, s_{22}) + 42/250\ I(s_{13}, s_{23}) = 0.7873.$$

Hence, the gain in information from such a partitioning would be

$$Gain(major) = I(s_1, s_2) - E(major) = 0.2115.$$

Similarly, we can compute the information gain for each of the remaining attributes. The information gain for each attribute, sorted in increasing order, is 0.0003 for *gender*, 0.0407 for *birth_country*, and 0.2115 for *major*, 0.4990 for *gpa*, and 0.5971 for *age_range*. Suppose that we use an attribute relevance threshold of 0.1 to identify weakly relevant attributes. The information gain of the attributes *gender* and *birth_country* are below threshold, and therefore considered weakly relevant. Thus, they are removed. The contrasting class is also removed, resulting in he initial target class working relation.

In Step 4, attribute-oriented induction is applied to the initial target class working relation, following the algorithm is Figure 4.1.

4.4 Mining Class Comparisons: Discriminating between Different Classes

In many applications, users may not be interested in having a single class (or concept) described or characterized, but rather would prefer to mine a description that compares or distinguishes one class (or concept) from other comparable classes (or concepts). Class discrimination or comparison (hereafter referred to as class comparison) mines descriptions that distinguish a target class from its contrasting classes. Notice that the target and contrasting classes must be comparable in the sense that they share similar dimensions and attributes. For example, the three class's person, address, and item are not comparable. However, the sales in the last three years are comparable classes, and so are computer science students versus physics students.

Our discussions on class characterization in the previous sections handle multilevel data summarization and characterization in a single class. The techniques developed can be extended to handle class comparison across several comparable classes. For example, the attribute generalization process described for class characterization can be modified so that the generalization is performed synchronously among all the classes compared. This allows the attributes in all of the classes to be generalized to the same levels of abstraction.

Suppose, for instance, that we are given the All Electronics data for sales in 1998 and sales in 1999 and would like to compare these two classes. Consider the dimension location with abstractions at the city, province_or_state, and country levels. Each class of data should be generalized to the same location level. That is, they are synchronously all generalized to either the city level, or the province_or_state level, or the country level. Ideally, this is more useful than comparing, say, the sales in Vancouver in 1998with the sales in the United States in 1999 (i.e., where each set of sales data is generalized to a different level). The users, however, should have the option to overwrite such an automated, synchronous comparison with their own choices, when preferred.

4.4.1 Class Comparison Methods and Implementations

"How is class comparison performed?" in general, the procedure is as follows:

1. **Data collection :** The set of relevant data in the database is collected by query processing and is partitioned respectively into a target class and one or a set of contrasting class(es).
2. **Dimension relevance analysis :** If there are many dimensions and analytical comparison is desired, then dimension relevance analysis should be performed on these classes as described in section 4.3, and only the highly relevant dimensions are included in the further analysis.
3. **Synchronous generalization :** Generalization is performed on the target class to the level controlled by a user-or expert-specified dimension threshold, which results in a prime target class relation/cuboids. The concepts in the contrasting class (es) are generalized to the same level as those in the prime target class relation/cuboids, forming the prime contrasting class (es) relation/cuboids.
4. **Presentation of the derived comparison :** The resulting class comparison description can be visualized in the form of tables, graphs, and rules. This presentation usually includes a "contrasting" measure (such as count %) that reflects the comparison between the target and contrasting classes. The user can adjust the comparison description by applying drill-down, roll-up, and other OLAP operations to the target and contrasting classes, as desired.

The above discussion outlines a general algorithm for mining analytical comparisons in databases. In comparison with analytical characterization, the above algorithm involves synchronous generalization of the target class with the contrasting classes so that classes are simultaneously compared at the same levels of abstraction.

"Can class comparison mining be implemented efficiently using data cube techniques?" yes- the procedure is similar to the implementation for mining data characterizations discussed in section 4.2.2. A flag can be used to indicate whether or not a tuple represents a target or contrasting class, where this flag is viewed as an additional dimension in the data cube. Since all of the other dimensions of the target and contrasting classes share the same portion of the cube, the synchronous generalization and specialization are realized automatically by rolling up and drilling down in the cube.

The following example mines a class comparison describing the graduate students and the undergraduate students at Big-University.

Example 4.10: Mining a class comparison. Suppose that you would like to compare the general properties between the graduate students and the undergraduate students at Big-University, given the attributes name, gender, major, birth_place, birth_date, residence, phone#, and gpa.

This data mining task can be expressed in DMQL as follows:

```
use Big_University_DB
mine comparison as "grad_vs_undergrad_students"
in relevance to name, gender, major, birth_place, birth_date, residence, phone#, gpa
for "graduate_students"
where status in "graduate"
versus "undergraduate_students"
where status in "undergraduate"
analyze count%
from student
```

Let's see how this typical example of a data mining query for mining comparison descriptions can be processed.

First, the query is transformed into two relational queries that collect two sets of task-relevant data: one for the initial target class working relation, and the other for the initial contrasting class working relation, as shown in tables 4.7 and 4.8. This can also be viewed as the construction of a data cube, where the status {graduate, undergraduate} serves as one dimension and the other attributes from the remaining dimensions.

Second, dimension relevance analysis is performed on the two classes of data. After this analysis, irrelevant or weakly relevant dimensions, such as name, gender, birth_place, residence, and phone#, are removed from the resulting classes. Only the highly relevant attributes are included in the subsequent analysis.

Third, synchronous generalization is performed: Generalization is performed on the target class to the levels controlled by user-or expert-specified dimension thresholds, forming the prime target class relation/cuboid. The contrasting class is generalized to the same levels as those in the prime target class relation/cuboid, forming the prime contrasting class(es) relation/cuboid, as presented in tables 4.9 and 4.10. In comparison with undergraduate students, graduate students tend to be older and have a higher GPA, in general.

Table 4.7 Initial target class working relation (graduate students).

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Jim Woodman	M	CS	Vancouver, BC, Canada	8-12-76	3511 Main St., Richmond	587-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83
.....	.....	.....	.....	.....	.....	.....	.....

Table 4.8 Initial contrasting class working relation (Undergraduate students).

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Bob Schumann	M	Chemistry	Calgary, Alt, Canada	10-1-78	2642 Halifax St., Burnaby	294-4291	2.96
Amy Eau	F	Biology	Golden, BC, Canada	30-3-76	463 Sunset Cres., Vancover	681-5417	3.52
.....	.....	.....	.....	.....	.....	.....	.....

Table 4.9 Prime generalized relation for the target class (graduate students)

Major	Age_range	Gpa	Count%
Science	21-25	Good	5.53%
Science	26-30	Good	5.02%
Science	Over_30	Very_good	5.86%
...	...	...	...
Business	Over_30	Excellent	4.68%

Finally, the resulting class comparison is presented in the form of tables, graphs, and/or rules. This visualization includes a contrasting measure (such as count %) that compares between the target class and the contrasting class. For example, 5.02% of the graduate students majoring in Science are between 26 and 30 years of age and have a "good" GPA, while only 2.32% of undergraduates have these same characteristics. Drilling and other OLAP operations may be performed on the target and contrasting classes as deemed necessary by the user in order to adjust the abstraction levels of the final description.

Table 4.10 Prime generalized relation for the contrasting class (under graduate students).

Major	Age_range	Gpa	Count%
Science	16-20	Fair	5.53%
Science	16-20	Good	4.53%
...	...	...	...
Science	26-30	Good	2.32%
...	...	...	...
Business	Over_30	Excellent	0.68%

4.4.2 Presentation of Class Comparison Descriptions

"How can class comparison descriptions be visualized?" As with class characterizations, class comparisons can be presented to the user in various forms, including generalized relations, crosstabs, bar charts, pie charts, curves, and rules. With the exception of logic rules, these forms are used in the same way for characterization as for comparison. In this section, we discuss the visualization of class comparisons in the form of discriminant rules.

As is similar with characterization description, the discriminative features of the target and contrasting classes of a comparison description can be described quantitatively by a quantitative discriminant rule, which associates a statistical interestingnesss measure, d-weight, with each generalized tuple in the description.

Let q_a be a generalized tuple, and C_j be the target class, where q_a covers some tuples of the target class. Note that it is possible that q_a also covers some tuples of the contrasting classes, particularly since we are dealing with a comparison description. The d-weight for q_a is the ratio of the number of tuples from the initial target class working relation that are covered by qa to the total number of tuples in both the initial target class and contrasting class working relations that are covered by q_a. Formally, the d-weight of q_a for the class C_j is defined as

$$d-weight = \frac{count(q_a \in C_j)}{\sum_{i=1}^{m} count(q_a \in C_i)} \quad \text{---------- (4.7)}$$

Where m is the total number of the target and contrasting classes, C_j is in $\{C_1,.....,C_m\}$, and count $(q_a \in C_j)$ is the number of tuples of class C_j that are covered by q_a. The range for the d-weight is [0.0,1.0] (or[0%,100%]).

A high d-weight in the target class indicates that the concept represented by the generalized tuple is primarily derived from the target class, whereas a low d-weight implies that the concept is primarily derived from the contrasting classes. A threshold can be set to control the display of interesting tuples based on the d-weight or other measures used, as described in section 4.2.3.

Table 4.11 Count Distribution between graduate and undergraduate students for a generalized tuple.

Status	Major	Age_range	Gpa	Count
Graduate	Science	21-25	Good	90
Undergraduate	Science	21-25	Good	210

Example 4.11: In Example 4.10, suppose that the count distribution for the generalized tuple major= "Science" and age_range= "21...25" and gpa="good" from Tables 4.9 and 4.10 is as shown in Table 4.11.

The d-weight for the given generalized tuple is 90/(90+210) = 30% with respect to the target class, and 210/(90+210) = 70% with respect to the contrasting class. That is, if a student majoring in Science is 21 to 25 years old and has a "good" gpa, then based on the data, there is a 30% probability that she is a graduate student, versus a 70% probability that she is an undergraduate student. Similarly, the d-weights for the other generalized tuples in Tables 4.9 and 4.10 can be derived.

A quantitative discriminant rule for the target class of a given comparison description is written in the form

$$\forall X,\ target\_class(X) \Leftarrow condition(X)\ \ [d : d\_weight] \quad \text{---------- (4.8)}$$

Where the condition is formed by a generalized tuple of the description. This is different from rules obtained in class characterization where the arrow of implication is from left to right.

Example 4.12: Based on the generalized tuple and count distribution in Example 4.11, a quantitative discriminant rule for the target class graduate-student can be written as follows:

$$\forall X, graduate\_student(X) \Leftarrow major(X) = \text{"Science"}$$
$$\wedge\ age\_range(X) = \text{"25 - 30"} \wedge gpa(X) = \text{"good"}\ [d : 30\%]$$

---------- (4.9)

Notice that a discriminant rule provides a sufficient condition, but not a necessary one, for an object (or tuple) to be in the target class. For example, Rule (4.9) implies that if X satisfies the condition, then the probability that X is a graduate student is 30%. However, it does not imply the probability that X meets the condition; given that X is a graduate student. This is because although the tuples that meet the condition are in the target class, other tuples that do not necessarily satisfy this condition may also be in the target class, since the rule may not cover all of the examples of the target class in the database. Therefore, the condition is sufficient, but not necessary.

Table 4.12 A crosstab for the total number (count) of TVs and computers sold in thousands in 1999.

Location/Item	TV	Computer	Both items
Europe	80	240	320
North_America	120	560	680
Both_regions	200	800	1000

4.4.3 Class Description: Presentation of Both Characterization and Comparison

"Since class characterization and class comparison are two aspects forming a class description, can we present both in the same table and in the same rule?" Actually, as long as we have a clear understanding of the meaning of the t-weight and d-weight measures and can interpret them correctly, there is no additional difficulty in presenting both aspects in the same table. Let's examine an example of expressing both class characterization and class discrimination in the same cross tab.

Example 4.13: Let Table 4.12 be a cross tab showing the total number (in thousands) of TVs and computers sold at All Electronics in 1999.

Let Europe be the target class and North-America be the contrasting class. The t-weights and d-weights of the sales distribution between the two classes are presented in Table 4.13. According to the table, the t-weight of a generalized tuple or object (e.g., item = "TV") for a given class (e.g., the target class Europe) shows how typical the tuple is of the given class (e.g., what proportion of these sales in Europe are for TVs?). The d-weight of a tuple shoes how distinctive the tuple is in the given (target or contrasting) class in comparison with its rival class (e.g., how do the TV sales in Europe compare with those in North America?).

Table 4.13 The same crosstab as in Table 4.12, but here the t-weight and d-weight values associated with each class are shown.

Location/item	TV			Computer			Both_items		
	Count	*t-wt*	*d-wt*	*Count*	*t-wt*	*d-wt*	*Count*	*t-wt*	*d-wt*
Europe	80	25%	40%	240	75%	30%	320	100%	32%
N_Am	120	17.65%	60%	560	82.35%	70%	680	100%	68%
Both_regions	200	20%	100%	800	80%	100%	1000	100%	100%

For example, the t-weight for "(Europe, TV)" is 25% because the number of TVs sold in Europe (80,000) represents only 25% of the European sales for both items (320,000). The d-weight for "(Europe, TV)" is 40% because the number of TVs sold in Europe (80,000) represents 40% of the number of TVs sold in both the target and the contrasting classes of Europe and North America, respectively (which is 200,000).

Notice that the count measure in the cross tab of table 4.13 obeys the general property of a cross tab (i.e., the count values per row and per column, when totaled, match the corresponding totals in the both_items and both_regions slots, respectively, for count). However, this property is not observed by the t-weight and d-weight measures. This is because the semantic meaning of each of these measures is different from that of count, as we explained in example 4.13.

"Can a quantitative characteristic rule and a quantitative discriminant rule be expressed together in the form of one rule?" The answer is yes-a quantitative characteristic rule and a quantitative descriminant rule for the same class can be combined to form a quantitative description rule for the class, which displays the t-weight and d-weights associated with the corresponding characteristics and discriminant rules. To see how this is done, let's quickly review how quantitative characteristic and descrininant rules are expressed.

As discussed in Section 4.2.3, a quantitative characteristic rule provides a necessary condition for the given target class since it presents a probability measurement for each property that can occur in the target class. Such a rule is of the form

$$\forall X, \text{target\_class}(X) \Rightarrow \text{condition}_1(X)\,[t{:}w1] \vee \ldots\ldots \vee \text{condition}_m(X)[t{:}w_m] \quad \text{---------(4.10)}$$

Where each condition represents a property of the target class. The rule indicates that if X is in the target_class, the probability that X satisfies condition i, is the value of the t-weight, w_i, where i is in {1,....,m}.

As previously discussed in Section 4.4.1, a quantitative discriminant rule provides a sufficient condition for the target class since it presents a quantitative measurement of the properties that occur in the target class versus those that occur in the contrasting classes. Such a rule is of the form

$$\forall X, \text{target\_class}(X) \Leftarrow \text{condition}_1(X)\,[d{:}w_1] \vee \ldots\ldots \vee \text{condition}_m(X)\,[d{:}\,w_m].$$

The rule indicates that if X satisfies condition i, there is a probability of Wi (the d-weight value) that x is in the target_class, where i is in {1,....,m}.

A quantitative characteristic rule and a quantitative discriminant rule for a given class can be combined as follows to form a **quantitative description rule:** (1) For each condition, show both the associated t-weight and d-weight, and (2) a bidirectional arrow should be used between the given class and the conditions. That is, a quantitative description rule is of the form

$$\forall X, \text{target\_class}(X) \Leftrightarrow \text{condition}_1(X)\,[t{:}\,w_1, d{:}w'_1] \vee \ldots\ldots \vee \text{condition}_m(X)\,[t{:}w_m, d{:}w'_m]. \quad \text{------- (4.11)}$$

This form indicates that for i from 1 to m, if X is in the target_calss, there is a probability of Wi that X satisfies condition i; and if X satisfies condition i, there is a probability of Wi' that X is in the target_class.

Example 4.14: It is straightforward to transform the cross tab of Table 4.13 in Example 4.13 into a class description rule for the target class, Europe, is

$$\forall X, \text{location}(X) = \text{"Europe"} \Leftrightarrow (\text{item}(X) = \text{"TV"})\,[t{:}25\%, d{:}40\%] \vee$$
$$(\text{item}(X) = \text{"computer"})\,[t{:}75\%, d{:}30\%] \quad \text{----------(4.12)}$$

The rule states that for the sales of TVs and computers at All Electronics in 1999, if the sale of one of these items occurred in Europe, then the probability of the item being a TV is 25%, while that of being a computer is 75%. On the other hand, if we compare the sales of these items in Europe and North America, then 40% of the TVs were sold in Europe (and therefore we can deduce that 60% of the TVs were sold in North Ame rica). Furthermore, regarding computer sales, 30% of these sales took place in Europe.

4.5 Mining Descriptive Statistical Measures in Large Databases

Earlier in this chapter, we discussed class description in terms of popular measures such as *count, sum,* and *average.* Relational database systems provide five built-in aggregate functions: *count(), sum(), avg(), max(),* and *min().* These functions can also be computed efficiently (in incremental and distributed manners) in data cubes. Thus, there is no problem in including these aggregate functions as basic measures in he descriptive mining of multidimensional data.

For many data mining tasks, however, users would like to learn more data characteristics regarding both central tendency and data dispersion. Measures of central tendency include *mean*, *mode*, and *midrange*, while measures of data dispersion include *quartiles*, outliers, and *variance*. These descriptive statistics are of great help in understanding the distribution of data. Such measures have been studied extensively in the statistical literature. From the data mining point of view, we need to examine how they can be computed efficiently in large multidimensional databases.

4.5.1 Measuring the Central Tendency

The most common and mist effective numerical measure of the "center" of a set of data is the (*arithmetic*) *mean*. Let $x_1, x_2, \ldots, x_n$ be a set of n values or observations. The **mean** of this set of values is

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{(4.13)}$$

Sometimes each value x_i in a set may be associated with a waight W_i, for i =1,...n. The weights reflect the significance, importance, or occurence frequency attached to their respective values. In this case, we can empute

$$\bar{x} = \frac{\sum_{i=1}^{n} W_i x_i}{\sum_{i=1}^{n} W_i} \qquad \text{(4.14)}$$

This is called the **weighted arithmetic mean** or the **weighted average.**

A measure was defined as *algebraic* if it can be computed from distributive aggregate measures. Since avg() can be computed by sum() / count(), where both sum() and count() are distributive aggregate measures in the sense that they can be computed in a distributive manner, then avg() is an algebraic measure. One can verify that the weighted average is also an algebraic measure.

Although the mean is the single most useful quantity that we us to describe a set if data, it is not the only, or even always the best, way of measuring the center of a set of data. For skewed data, a better measure of the center of data is the *median*, M. suppose that the values forming a given set of data are in numerical order. The median is the middle value of the ordered set if the number of values n is an odd number; otherwise (i.e., if n is even), it is the average of the middle two values.

Based on the categorization of measures the median is neither a distributive measure nor an algebraic measure – it is a holistic measure in the sense that it cannot be computed by partitioning a set of values arbitrarily into smaller subsets, computing their medians independently, and merging the median values of each subset. On the contrary, count(), sum(), max(), and min() can be computed in this manner (being distributive measures) and are therefore easier to compute than the median.

Although it is not easy to compute the exact median value in a large database, an appropriate median can be efficiently computed. For example, for grouped data, the median, obtained by interpolation, is given by

$$median = L_1 + \left(\frac{n/2 - (\sum f)l}{f_{median}}\right)c \qquad \text{(4.15)}$$

where L_1 is the lower class boundary of (i.e., lowest value for) the class containing the median, n is the number of values in the data, $(\sum f)l$ list the sum of the frequencies of all of the classes that are lower than the median class, f_{median} is the frequency of the median class, and c is the size of the median class interval.

Another measure of central tendency is the mode. The **mode** for a set of data is the value that occurs most frequently in the set. It is possible for the greater frequency to correspond to several different values, which results in more than one mode. Data sets with one, two, or three modes are respectively called **uni-modal, bimodal,** and **trimodal.** At the other extreme, if each data value occurs only once, then there is no mode.

For unimodal frequency curves that are moderately skewed (asymmetrical), we have the following empirical relation:

$$Mean - mode = 3 \times (mean - median) \qquad \text{(4.16)}$$

This implies that the mode for unimodal frequency curves that are moderately skewed can easily be computed if the mean and median values are known.

The **midrange**, that is, *the average of the largest and smallest values in a dataset*, can be used to measure the central tendency of the set of data. It is trivial to compute the midrange using the SQL aggregate functions, max() and min().

4.5.2 Measuring the Dispersion of Data

The degree to which numeric data tend to spread is called the **dispersion**, or **variance** of the data. The most common measures of data dispersion are the *five-number summary* (based on quartiles), the *interquartile range*, and the *standard deviation*. The plotting of *boxplots* (which show outlier values) also serves as a useful graphical method.

Quartiles, Outliers, and Boxplots

The **k^{th} percentile** of a set of data in numerical order is the value x having the property that k percent of the data entries lie at or below x. values at or below the *median* M (discussed in the previous subsection) correspond to the 50^{th} percentile.

The most commonly used percentiles other than the median are **quartiles. The first quartile,** denoted by Q_1, is the 25^{th} percentile; the **third percentile,** denoted by Q_3, is the 75^{th} percentile. The quartiles, including the median, give some indication of the center, spread, and shape of a distribution. The distance between the first and third quartiles is a simple measure of spread that gives the range covered by the middle half of the data. This distance is called the **interquartile range** (IQR) and is defined as

$$IQR = Q_3 - Q_1 \qquad \text{(4.17)}$$

We should be aware that no single numerical measure of spread, such as IQR is very useful for describing skewed distributions. The spreads of two sides of a skewed distribution are unequal. Therefore, it is more informative to also provide the two quartiles Q_1 and Q_3, along with the median, M. one common rule of thumb for identifying suspected outliers is to single out values falling at least 1:5 X IQR above the third quartile or below the first quartile.

Because Q_1, M, and Q_3 contain no information about the endpoints (e.g., tails) of the data, a fuller summary of the shape of a distribution can be obtained by providing the highest and lowest values as well. This known as the *five-number summary*. The **five-number summary** of a distribution consists of the median M, the quartiles Q_1 and Q_3, and the smallest and largest individual observations written in the order *Minimum*, Q_1, M, Q_3, *Maximum*.

A popularly used visual representation of a distribution is the **boxplot**. In a boxplot:

- Typically, the ends of the box are at the quartiles, so that the box length is the interquartile range, IQR.
- The median is marked by a line within the box.
- Two lines (called *whiskers*) outside the box extend to the smallest (*Minimum*) and largest (*Maximum*) observations.

When dealing with a moderate number of observations, it is worthwhile to plot potential outliers individually. To do this in a boxplot, the whiskers are extended to the extreme high and low observations only if these values are less than 1.5 X IQR beyond the quartiles. Otherwise, the whiskers terminate at the most extreme observations occurring within 1.5 X IQR of the quartiles. The remaining cases are plotted individually. Boxplots can be used in the comparisons of several sets of compatible data. Figure 4.5 shows boxplots for unit price data for items sold at four branches of *AllElectronics* during a given time period. For branch 1, we see that the median price of items sold is \$80, Q_1 is \$60, and Q_3 is \$100.

Based on similar reasoning as in our analysis of the median in Section 4.5.1, we can conclude that Q_1 and Q_3 are holistic measures, as is IQR. The efficient computation of boxplots or even *approximate boxplots* is interesting regarding the mining of large data sets.

Variance and Standard Deviation

The **variance** of n observations $x_1, x_2, \ldots, x_n$ is

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right] \quad \text{---------- (4.1}$$

The **standard deviation** s is the square root of the variance s^2.

The basic properties of the standard deviation s as a measure of spread are

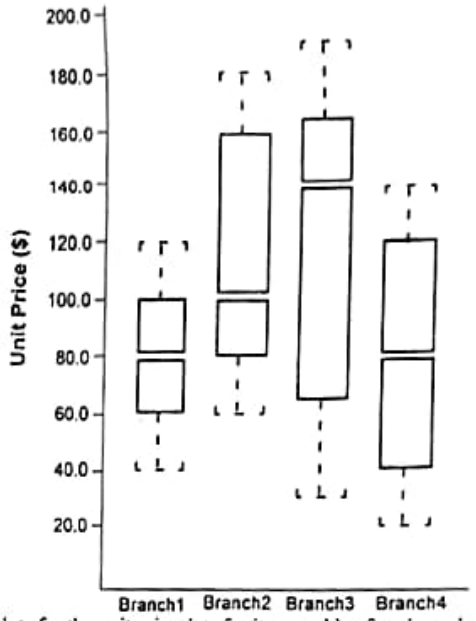

Figure 4.5 Boxplots for the unit price data for items sold at four branches of AllElectronics during a give time period.

- s measures spread about the mean and should be used only when the mean is chosen as the measure of center.
- $s = 0$ only when there is no spread, that is, when all observations have the same value. Otherwise $s > 0$.

Notice that variance and standard deviation are algebraic measures because n (which is count() in SQL), Σx_i (which is the sum() of x_i), and Σx_i^2 (which is the sum() of x_i^2) can be computed in any partition and then merged to feed into the algebraic equation (4.18). Thus the computation of the two measures is scalable in large databases.

Table 4.14 A set of unit price data for items sold at a branch of *AllElectronics*.

Unit Price ($)	Number of items sold
40	275
43	300
47	250
..	..
74	360
75	515
78	540
..	..
115	320
117	270
120	350

4.5.3 Graph Displays of Basic Statistical Class Descriptions

Aside from the bar charts, pie charts, and line graphs discussed earlier in this chapter, there are also a few additional popularly used graphs for the display of data summaries and distributions. These include *histograms*, *quantile plots*, *q-q plots*, *scatter plots*, and *loess curves*.

Plotting **histograms**, or **frequency histograms**, is a univariate graphical method. A histogram consists of a set of rectangles that reflect the counts or frequencies of the classes present in the given data. The base of each rectangle is on the horizontal axis, centered at a "class" mark, and the base length is equal to the class width. Typically, the class width is uniform, with classes being defined as the values of a categoric attribute, or equiwidth ranges of a discretized continuous attribute. In these cases, the height of each rectangle is equal to the count or relative frequency of the class it represents, and the histogram is generally referred to as a **bar chart**. Alternatively, classes for a continuous attribute may be defined by range of nonuniform width. In this case, for a given class, the class width is equal to the range width, and height of the rectangle is the class density (i.e., the count or relative frequency of the class, divided by the class width).

Figure 4.6 shows a histogram for the data set of table 4.14, where classes are defined by equiwidth ranges representing $20 increments and the frequency is the number of items sold.

Histograms are at least a century old and are a widely

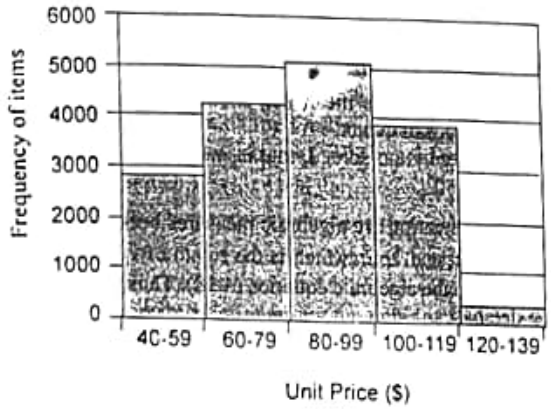

Figure 4.6 A histogram for the data set of Table 4.14.

used univariate graphical method. However, they may not be as effective as the quantile plot, q-q plot, and boxplot methods for comparing groups of univariate observations.

A quantile plot is a simple and effective way to have a first look at a univariate data distribution. First, it displays all of the data (allowing the user to access both the overall behavior and unusual occurrences). Second, it plots quantile information. The mechanism used in this step is slightly different from the percentile computation. Let $x_{(i)}$, for i = 1 to n, be the data sorted in increasing order so that $x_{(1)}$ is the smallest observation and $x_{(n)}$ is the largest. Each observation $x_{(i)}$ is paired with

percentage, f_i, which indicates that approximately $100 f_i$% of the data are below or equal to the value $x_{(i)}$. We say "approximately" because there may not be a value with exactly a fraction f_i of the data below or equal to $x_{(i)}$. Note that the 0.25 quantile corresponds to quartile Q_1, the 0.50 quantile is the median, and the 0.75 quantile is Q_3. Let

$$f_i = (i - 0.5) / n .$$

These numbers increase in equal steps of 1/n, ranging from 1/2n (which is slightly above zero) to 1 – 1/2n (which is slightly below one). On a quantile plot, $x_{(i)}$ is graphed against f_i. This allows us to compare different distributions based on their quantiles. For example, given the quantile plots of sales data for two different time periods, we can compare their Q_1, median, Q_3, and other f_i values at a glance. Figure 4.7 shows a quantile plot for the *unit price* data of Table 4.14.

A **quantile-quantile plot**, or **q-q plot**, graphs the quantiles of one univariate distribution against the corresponding quantiles of another. It is a powerful visualization tool in that it allows the user to view whether there is a shift in going from one distribution to another.

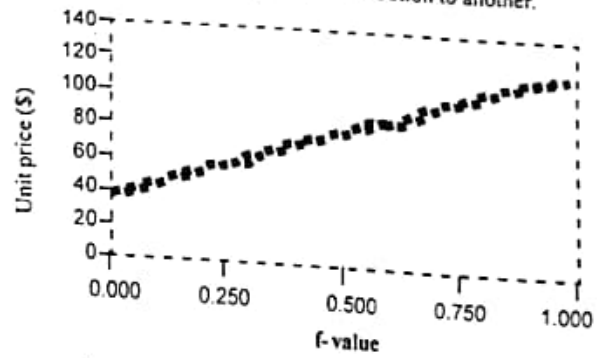

Figure 4.7 A quantile plot for the unit price data of Table 4.14.

Suppose that we have two sets of observations for the variable *unit price*, taken from two different branch locations. Let $x_{(1)}, \ldots, x_{(m)}$ be the data from the first branch, and $y_{(1)}, \ldots, y_{(n)}$ be the data from the second, where each data set is sorted in increasing order. If m = n (i.e., the number of points in each set is the same), then we simply plot $y_{(i)}$ against $x_{(i)}$, where $y_{(i)}$ and $x_{(i)}$ are both $(i - 0.5)$ quantiles of their respective data sets. If m < n (i.e., the second branch has fewer observations than the first), there can be only m points on the q-q plot. Here, $y_{(i)}$ is the $(i - 0.5) / m$ quantile of the y data, which is plotted against the $(i - 0.5) / m$ quantile of the x data. This computation typically involves interpolation.

Figure 4.8 shows a quantile-quantile plot for unit price data of items sold at two different branches of AllElectronics during a given time period. The lowest point in the left corner corresponds to the same quantile, 0.03, for each data set. (To aid in comparison, we also show a straight line that represents the case of when, for each given quantile, the unit price at each branch is the same. In addition, the darker point corresponds to the data for Q_1, the median, and Q_3, respectively.) For example, we see that at this quantile, the unit price of items sold at branch 1 was slightly less than that

at branch 2. In other words, 3% of items sold at branch 1 were less than or equal to $40, while 3% of items at branch 2 were less than or equal to $42. at the highest quantile, we see that the unit price of items at branch 2 was slightly less than that at branch 1. In general, we note that there is a shift in the distribution of branch 1 with respect to branch 2 in that the unit prices of items sold at branch 1 tend to be lower than those at branch 2.

A scatter plot is one of the most effective graphical methods for determining if there appears to be a relationship, pattern, or trend between two quantitative variables. To construct a scatter plot, each pair of values is treated as a pair of coordinates in an algebraic sense and plotted as points in the plane.

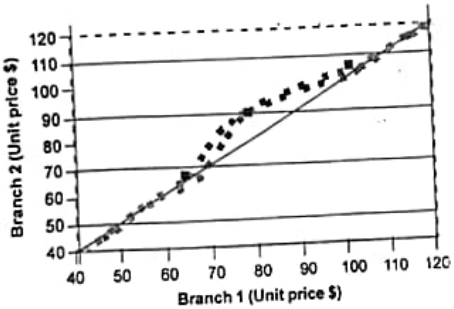

Figure 4.8 A quantile-quantile plot for unit price data from two different branches.

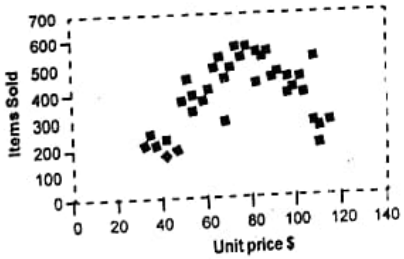

Figure 4.9 A scatter plot for the data set of Table 4.14.

The scatter plot is a useful exploratory method for providing a first look at bivariate data to see how they are distributed throughout the plane, for example, and to see clusters of points, outliers, and so forth. Figure 4.9 shows a scatter plot for the set of data in table 4.14.

A loess curve is another important exploratory graphic aid that add a smooth curve to a scatter plot in order to provide better perception of the pattern of dependence. The word *loess* is short for "local regression." Figure 4.10 shows a loess curve for the set of data in Table 4.14.

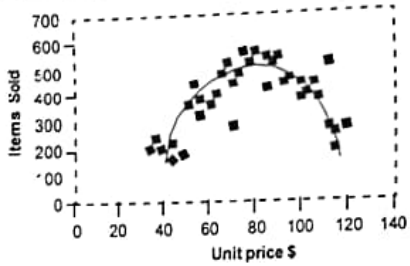

Figure 4.10 A loess curve for the data set of Table 4.14.

To fit a loess curve, values need to be set for two parameters – α, a smoothing parameter, and λ, the degree of the polynomials that are fitted by the regression. While α can be any positive number (typical values are between ¼ and a), λ can be 1 or 2. The goal in choosing α is to produce a fit that is as smooth as possible without unduly distorting the underlying pattern in the data. The curve becomes smoother as α increases. There may be some lack of fit, however, indicating possible "missing" data patterns. If α is very small, the underlying pattern is tracked, yet over fitting of the data may occur, where local "wiggles" in the curve may not be supported by the data. If the underlying pattern of the data has a "gentle" curvature with no local maxima and minima, then local linear fitting is usually sufficient (λ = 1). However, if there are local maxima or minima, then local quadratic fitting (λ = 2) typically does a better job of following the pattern of the data and maintaining local smoothness.

Summary

The simplest kind of descriptive data mining is concept description. Concept description has close ties with data generalization, given the large amount of data stored in data bases, it is useful to be able to describe concepts in concise and succinct terms at generalized at multiple levels of abstraction.

Data generalization is a process that abstracts a large set of task-relevant data in the data base from a relatively low conceptual level to higher conceptual levels. Methods for the efficient and flexible generalization of large data sets can be categorized according to two approaches viz., data cube approach and the attribute induction approach. The data cube approach can be considered as a data ware-house based, permutation –oriented, materialized-view approach. Attribute-oriented induction generates one or a set of generalized descriptions.

Measures of attribute relevance analysis can be used to help identify irrelevant or weakly relevant attributes that can be excluded from the concept description process. The incorporation of

this preprocessing step into class characterization or comparison is referred to as analytical characterization or analytical comparison, respectively.

In many applications, users may not be interested in having a single class but rather would prefer to mine a description that compares or distinguishes one class from other comparable classes. Class discrimination or comparison mines descriptions that distinguish a target class from its contrasting classes. The class comparison performed by sequence of steps like Data collection, Dimension relevance analysis, Synchronous generalization, and Presentation of the derived comparison. The class comparison mining be implemented efficiently using data cube techniques.

For many data mining tasks, however, users would like to learn more data characteristics regarding both central tendency and data dispersion. Measures of central tendency include *mean*, *mode*, and *midrange*, while measures of data dispersion include *quartiles*, outliers, and *variance*. From the data mining point of view, such measured are examined and are computed efficiently in large multidimensional databases.

Exercises:

1. For class characterization, what are the major differences between a data cube-based implementation and a relational implementation such as attribute-oriented induction? Discuss which method is most efficient and under what conditions this is so.
2. Discuss why analytical characterization is needed and how it can be performed. Compare the result of to induction methods: (1) with relevance analysis and (2) without relevance analysis?
3. Give three additional commonly used statistical measures for the characterization of data dispersion, and discuss how they can be computed efficiently in large databases.
4. Give a generalized relation R derived from a database DB, suppose that a set DB of tuples needs to be deleted from DB. Outline an incremental updating procedure for applying the necessary deletions to R.
5. Out line a data cube-based incremental algorithm for mining analytical class comparisions.
6. Outline a method for (1) parallel and (2) distributed mining of statistical measures of data dispersion in a data cube environment.
7. Suppose that the following table is derived by *attribute-oriented induction.*

Class	birth-place	count
	Canada	180
Programmer	others	120
	Canada	20
DBA	others	80

(a) Transform the table into a cross tab showing the associated t-weights and d-weights.

(b) Map the class *Programmer* into a (bidirectional) *quantitative descriptive rule*, for example, ∀X, *Programmer(X)* <=> *(birth-place(X) = "Canada" ∧ . . .)*
[t: x%, d: y%]] . . . ∨ (. .)[t: w%, d: z%].

8. Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13,15,16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

(a) What is the *mean* of the data? What is the *median*?

(b) What is the *mode* of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc;).

(c) What is the *midrange* of the data?

(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data.

(e) Give the *five-number summary* of the data.

(f) Show a *box plot* of the data.

(g) How is a *quantile-quantile plot* different from a *quantile plot*?

Objective :

1. **_____ hierarchies for numeric attributes can be constructed automatically based on data distribution analysis.**
   a. **Concept**
   b. Discretization
   c. Tree
   d. Index

2. **_____ techniques can be used to reduce the number of values for a given continuous attribute, by dividing the range of the attribute into intervals.**
   a. Concept hierarchy
   b. **Discretization**
   c. Tree-based
   d. Index

3. **An information-based measure called ____ can be used to recursively partition the values of a numeric attribute A, resulting in a hierarchical discretization.**
   a. **Entropy**
   b. Cluster
   c. Binning
   d. Segmentation

4. The kinds of knowledge include
   a. Image analysis
   b. Query process
   c. **Association**
   d. Multimedia analysis

5. Which of the following is a simplicity measure?
   a. Rule strength
   b. Rule quality
   c. Rule reliability
   d. **Rule length**

6. _____ hierarchies can be used to refine or enrich schema defined hierarchies. When the two types of hierarchies are combined.
   a. Schema
   b. **Set-grouping**
   c. Operation-derived
   d. rule-based

7. _____ are those that contribute new information or increased performance to the given pattern set.
   a. Utility patterns
   b. Certainty patterns
   c. **Novelty pattern**
   d. Simplicity patterns

8. Certainty factor is also known as
   a. Rule length
   b. Noise threshold
   c. Minable view
   d. **Rule strength**

9. Which of the following primitive specifies the data mining functions to be performed?
   a. Task-relevant data
   b. **The kind of knowledge to be mined**
   c. Background knowledge
   d. Interestingness measures

10. _____ may be used to guide the mining process or, after discovery to evaluate the discovered patterns.
    a. Task-relevant data
    b. The kind of knowledge to be mined
    c. Background knowledge
    d. **Interestingness measures**

11. A _____ hierarchy is a total or partial order among attributes in the database schema.
    a. Schema
    b. Set-grouping
    c. Operation-derived
    d. rule-based

12. _____ hierarchies include the decoding of information encoded strings information extraction from complex data objects and data clustering.
    a. Rule-based
    b. **Operation-derived**
    c. Schema
    d. Set grouping

13. Which of the following clause is the task-irrelevant data primitive?
    a. In relevance to
    b. Use for warehouse
    c. **Analysis**
    d. Order by

14. Mining with the use of _____, allows additional flexibility for ad hoc rule mining.
    a. Image patterns
    b. Data patterns
    c. Information patterns
    d. **Meta patterns**

15. Which of the following clause lists the attributes or dimensions for exploration
    a. Order by
    b. group by
    c. having
    d. **in relevance to**

16. Which of the following clause uses the meta pattern?
    a. Analyze
    b. In relevance to
    c. **Matching**
    d. Use data warehouse

17. Which of the following clause is used for discrimination?
    a. Mine characteristics
    b. Mine discriminant
    c. Mine association
    d. Mine comparison

18. DMQL expansion is
    a. Data Modeling Queue Level
    b. Design Modeling Query language
    c. **Data Mining Query Language**
    d. Data &Meta data Query Language

19. The _____ clause, when used for characterization, specific aggregate measures, such as count, sum or count
    a. Use database
    b. **Analyze**
    c. Matching
    d. Use hierarchy

20. Which of the following clause specifies the condition by which groups of data are considered relevant?
    a. **Having**
    b. Group by
    c. Order by
    d. analyze

21. CRISP-DM addresses an issue as
    a. Mapping from datamining problems to business issues
    b. Capturing and misunderstanding the data
    c. Disintegrating datamining results within the business context
    d. **Deploying and maintaining data mining results**

22. Which of the following data mining language uses SQL-like syntax and serves as rule generation queries for mining association rules.
    a. **MINE RULE operator**
    b. RULE MINE operator
    c. DATA MINE operator
    d. DWH operator

23. Which of the following is not a data mining language?
    a. DMQL
    b. MSQL
    c. **PSQL**
    d. OLE DB for

24. System of schema hierarchy is
    a. **textbf{Define hierarchy} location-hierarchy textbf{on} address textbf{as} [street, city, country]**
    b. textbf{Define hierarchy} location-hierarchy textbf{as} address textbf{on} [street, city, country]
    c. textbf{Define hierarchy} location-hierarchy textbf{from} address textbf{to} [street, city, country]
    d. textbf{Define hierarchy }location-hierarchy textbf{for} address textbf{all} [street, city, country]

25. Which of the following is a data mining query language
    a. PSQL
    b. QSQL
    c. **MSQL**
    d. RSQL

26. Which of the following provides a concise and succinct summarization of the given collection of data?
    a. Comparison
    b. **Characterization**
    c. Summarization
    d. Aggregation

27. _____ data mining describes the data set in a concise and summerative manner and presents interesting general properties of the data.
    a. **Descriptive**
    b. Predictive
    c. Active
    d. Constructive

28. _____ data mining analyzes the data in order to construct one or a set of models and attempts to predict the behavior of new data sets.
    a. Descriptive
    b. **Predictive**
    c. Active
    d. Constructive

29. Attribute removal is based on the following rule: If there is a large set of distinct values for an attribute of the initial working relation but,
    a. There is generalization operator on the attribute
    b. There is no generalization operand on the attribute
    c. **There is no generalization operator on the attribute**
    d. There is no aggregation operator on the attribute

30. On-line analysis processing in data warehouses is a purely-controlled process
    a. Machine
    b. database
    c. Developer
    d. **User**

31. Which of the following approach is used to control generalization process?
    a. **Generalized relation threshold control**
    b. Generalized class threshold control
    c. Generalized dimension threshold control
    d. Generalized query threshold control

32. Many current OLAP systems confine dimensions to _____ data
    a. Numeric
    b. **Non numeric**
    c. Meta
    d. Summarized

33. _____ is a process that abstracts a large set of task-relevant data in a database from a relatively low conceptual level to higher conceptual levels.
    a. Data realization
    b. Data characterization
    c. Data summarization
    d. **Data generalization**

34. The _____ approach can be considered as a data warehouse-based pre-computation-oriented, material-view approach.
    a. Object-oriented induction
    b. **Data cube**
    c. Attribute-oriented induction
    d. Data square

35. Which of the following approach is a relational database query-oriented, generalization-based, on-line data analysis technique?
    a. **Attribute-oriented induction**
    b. object-oriented approach
    c. Data cube
    d. Data square

36. _____ performs off-line aggregation before an OLAP or Data mining query is submitted for processing.
    a. Object-oriented induction
    b. **Data cube**
    c. Attribute-oriented induction
    d. Data square

37. How can the t-weight and interestingness measures in general be used by the data mining system to display only the concept descriptions that it objectively evaluates as interesting?
    a. **By threshold**
    b. By generalization
    c. By comparison
    d. By characterization

38. The data cube implementation of attribute-oriented induction can be performed by
    a. Using defined data cube
    b. **Using a predefined data cube**
    c. Using a generalized data cube
    d. Using a quantified data cube

39. A _____ can be represented by a 3-D data cube.
    a. **Cross-tab**
    b. Bar chart
    c. pie chart
    d. Flow chart

40. Step one of the attribute-oriented-induction algorithm is essentially a relational query to collect the task relevant data into the _____.
    a. Prime relation
    b. Secondary relation
    c. **Working relation**
    d. Analyzing relation

41. Which of the following relation collects the statistics of attribute-oriented-induction algorithm?
    a. **Working relation**
    b. Prime relation
    c. Secondary relation
    d. Analyzing relation

42. Descriptions can also be visualized in the form of _____.
    a. Cross-relations
    b. Cross-checks
    c. Cross-boards
    d. **Cross-tabs**

43. Step three of attribute-oriented-induction derives the _____ relation.
    a. Working
    b. **Prime**
    c. Secondary
    d. Analysing

44. The _____ as an interestingness measure that describes the typically of each disjoint in the rule, or of each tuple in the corresponding generalized relation.
    a. Quantitative rule
    b. Quantitative characteristic rule
    c. c-weight
    d. **t-weight**

45. The information gain is obtained by
    a. Expected information + entropy
    b. Entropy - Expected information
    c. **Expected information - entropy**
    d. Entropy + Expected information

46. Class comparison is also called as
    a. composition
    b. aggregation
    c. **discrimination**
    d. characterization

47. _____ can be used to perform some preliminary relevance analysis on the data by removing or generalizing attributes having a very large number of distinct values.
   a. Object-oriented induction
   b. Attribute-oriented induction
   c. Batch-oriented induction
   d. Class-oriented induction

48. Class characterization that includes the analysis of attribute/dimensions relevance is called _____.
   a. Analytical comparison
   b. Analytical measurement
   c. Analytical characterization
   d. Analytical difference

49. _____ irrelevant and weakly relevant attributes using the selected relevance analysis measure.
   a. Insert
   b. Update
   c. Modify
   d. Remove

50. The _____ class is the class to be characterized
   a. base
   b. target
   c. contrasting
   d. sub

51. The _____ class is the set of comparable data that are not in the target class.
   a. base
   b. target
   c. contrasting
   d. sub

52. Generalization is performed on the _____ to the level controlled by a user or expert-specified dimension threshold, which results in a _____
   a. Target class, Prime target class relation
   b. Contrasting class, Prime contrasting class relation
   c. Target class, Secondary target class relation
   d. Contrasting class, Secondary contrasting class relation

53. Can class comparison mining be implemented efficiently using data cube techniques?
   a. yes
   b. no
   c. limited
   d. difficult

54. Class discrimination is also called as
   a. class comparison
   b. class hierarchy
   c. class aggregation
   d. class concept

55. The set of relevant data in the database is collected by query processed and is partitioned respectively into a target class and one or a set of _____ class(es)
   a. discrimination
   b. contrasting
   c. comparable
   d. target

56. A _____ d-weight in the target class indicates that the concept represented by the generalized tuple is primarily derived from the target class
   a. Low
   b. High
   c. Average
   d. Middle

57. A _____ d-weight implies that the concept is primarily derived from the contrasting class
   a. Low
   b. High
   c. Average
   d. Middle

58. In d-weight, d stands for
   a. divide
   b. dead
   c. discrimination
   d. degree

59. Inter quartile is defined as
   a. First quartile -Third quartile
   b. First quartile + Third quartile
   c. Third quartile + First quartile
   d. Third quartile - First quartile

60. The most commonly used percentiles other the median are _____
   a. Outliers
   b. Boxplots
   c. Quartiles
   d. Modes

61. A popularly used visual representation of a distribution is the _____
    a. Boxplot
    b. Outlier
    c. Quartile
    d. Histogram

62. Dispersion is also called as
    a. Mean
    b. Variance
    c. Median
    d. mode

63. Which of the following is central tendency measure?
    a. Outliers
    b. Variance
    c. Quartiles
    d. Mode

64. Which of the following is a data dispersion measure?
    a. Mean
    b. Variance
    c. Mode
    d. Median

65. The average of the largest and smallest values in a data set is called as
    a. Median
    b. Mean
    c. Mid range
    d. Mode

66. The _____ for a set of data is the value that occurs most frequently in the set.
    a. Median
    b. Mean
    c. Mid range
    d. Mode

67. Which of the following is not central tendency measure?
    a. Variance
    b. Mean
    c. Median
    d. Mode

68. A _____ is one of the most effective graphical methods or trend between two quantitative variables.
    a. q-q plot
    b. scatter plot
    c. quantile plot
    d. q-q-q plot

69. A _____ is another important exploratory graphic aid that adds a smooth curve to a scatter plot in order to provide better perception of the pattern of dependence.
    a. Loess curve
    b. Scatter curve
    c. Bar chat
    d. Quantile plot

70. Histograms are also called as _____ histograms.
    a. frequency
    b. variance
    c. quartile
    d. outlier

71. The word loess is short for
    a. Load compression
    b. Local compression
    c. Load refression
    d. Local refression

72. A _____ consists of a set of rectangles that reflect the counts of the classes present in the given data.
    a. Quartile plot
    b. q-q plot
    c. Histogram
    d. Loess curves

73. A _____ is a simple and effective way to have a first look at an unvariate data distribution.
    a. q-q plot
    b. scatter plot
    c. histogram
    d. quantile plot

74. A _____, groups the quantiles of one univariate distribution against the corresponding quantiles of another.
    a. quantile plot
    b. q-q-q plot
    c. q-q plot
    d. Scatter plot

Walchand College of Engg. LIBRARY