

### Data Preprocessing

Today's real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources. **Low-quality data will lead to low-quality mining results.** "How can the data be preprocessed in order to help improve the quality of the data and, consequently, of the mining results? How can the data be preprocessed so as to improve the efficiency and ease of the mining process?"

There are several data preprocessing techniques.

**Data cleaning** can be applied to remove noise and correct inconsistencies in data.

**Data integration** merges data from multiple sources into a coherent data store such as a data warehouse.

**Data reduction** can reduce data size by, for instance, aggregating, eliminating redundant features, or clustering.

**Data transformations** (e.g., normalization) may be applied, where data are scaled to fall within a smaller range like 0.0 to 1.0. This can improve the accuracy and efficiency of mining algorithms involving distance measurements.

These techniques are not mutually exclusive; they may work together. For example, data cleaning can involve transformations to correct wrong data, such as by transforming all entries for a date field to a common format.

Data processing techniques, when applied before mining, can substantially **improve the overall quality of the patterns mined and/or the time required for the actual mining.**

Data Quality: Why Preprocess the Data?

Data have quality if they satisfy the requirements of the intended use. There are many factors comprising data quality, including accuracy, completeness, consistency, timeliness, believability, and interpretability.

### Major Tasks in Data Preprocessing

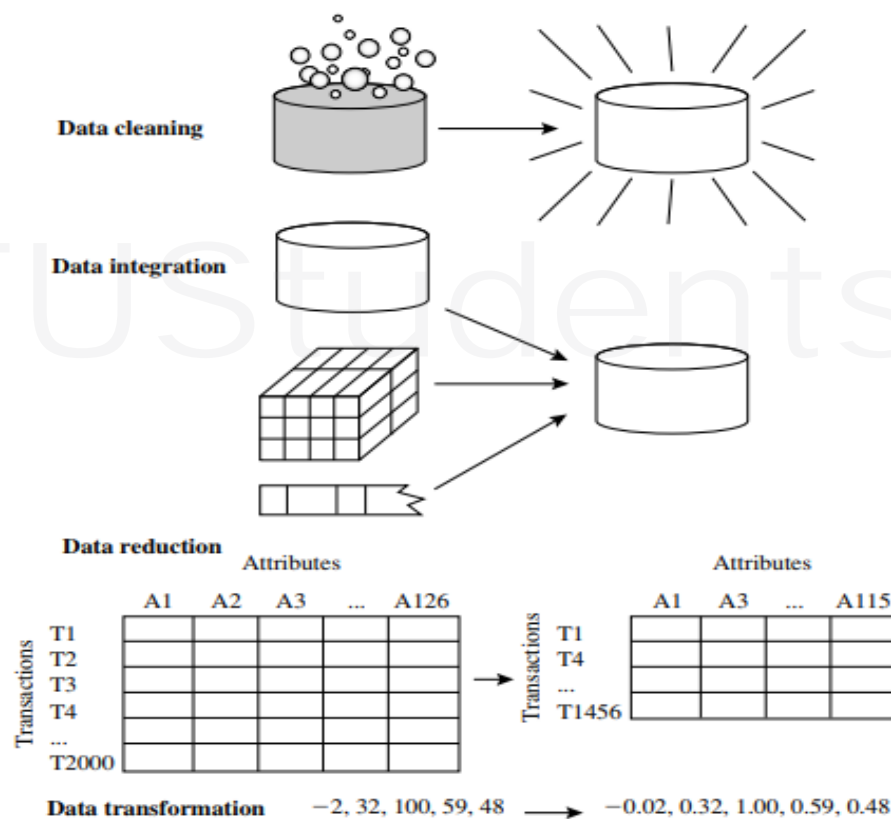
We look at the major steps involved in data preprocessing, namely, data cleaning, data integration, data reduction, and data transformation.

**Data cleaning routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies.** Dirty data can cause confusion in the mining procedure, resulting in unreliable output. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust. Instead, they may

concentrate on avoiding overfitting the data to the function being modeled. Therefore, a useful **preprocessing step is to run your data through some data cleaning routines.**

Discretization and concept hierarchy generation can also be useful, where raw data values for attributes are replaced by ranges or higher conceptual levels. For example, raw values for age may be replaced by higher-level concepts, such as youth, adult, or senior.

Discretization and concept hierarchy generation are powerful tools for data mining in that they allow data mining at multiple abstraction levels. **Normalization, data discretization, and concept hierarchy generation are forms of data transformation.** You soon realize such data transformation operations are additional data preprocessing procedures that would contribute toward the success of the mining process.



Forms of data preprocessing.

Data reduction obtains a reduced representation of the data set that is much smaller in volume, yet produces the same (or almost the same) analytical results. Data reduction strategies include data aggregation (building a data cube for example,) dimension reduction ( eg. Removing irrelevant attributes through correlation analysis), data compression ( e.g. using encoding schemes such as minimum length encoding or wavelets) and numerosity reduction( e.g “replacing” the data by alternative, smaller representations such as clusters or parametric models)

In dimensionality reduction, data encoding schemes are applied so as to obtain a reduced or “compressed” representation of the original data. Examples include data compression techniques (e.g., wavelet transforms and principal components analysis), attribute subset selection (e.g., removing irrelevant attributes), and attribute construction (e.g., where a small set of more useful attributes is derived from the original set).

In numerosity reduction, the data are replaced by alternative, **smaller representations using parametric models (e.g., regression or log-linear models) or nonparametric models (e.g., histograms, clusters, sampling, or data aggregation).**

In summary, real-world data tend to be dirty, incomplete, and inconsistent. Data preprocessing techniques can improve data quality, thereby helping to improve the accuracy and efficiency of the subsequent mining process. Data preprocessing is an important step in the knowledge discovery process, because quality decisions must be based on quality data. Detecting data anomalies, rectifying them early, and reducing the data to be analyzed can lead to huge payoffs for decision making.

### **Data Cleaning**

Real-world data tend to be incomplete, noisy, and inconsistent. **Data cleaning (or data cleansing) routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.** In this section, you will study basic methods for data cleaning.

#### **❖ Missing Values**

##### **(a) Ignoring the tuple:**

This is usually done when the class label is missing (assuming the mining task involves classification or description). This method is not very effective unless the tuple contains several attributes with missing values. It is especially poor when the percentage of missing values per attribute varies considerably.

##### **(b) Manually filling in the missing value:**

In general, this approach is time-consuming and may not be a reasonable task for large data sets with many missing values, especially when the value to be filled in is not easily determined.

##### **(c) Using a global constant to fill in the missing value:**

Replace all missing attribute values by the same constant, such as a label like “Unknown,” or  $-\infty$ . If missing values are replaced by, say, “Unknown,” then the mining program may mistakenly think that they form an interesting concept, since they all have

a value in common that of "Unknown." Hence, although this method is simple, it is not recommended.

**(d) Using the attribute mean for quantitative (numeric) values or attribute mode for categorical (nominal) values, for all samples belonging to the same class as the given tuple:**

For example, if classifying customers according to credit risk, replace the missing value with the average income value for customers in the same credit risk category as that of the given tuple.

**(e) Using the most probable value to fill in the missing value:**

This may be determined with regression, inference-based tools using Bayesian formalism, or decision tree induction. For example, using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for income.

❖ **Noisy data:**

Noise is a random error or variance in a measured variable. Data smoothing tech is used for removing such noisy data.

Several Data smoothing techniques:

**1 Binning methods:**

Binning methods smooth a sorted data value by consulting the neighborhood", or values around it. The sorted values are distributed into a number of 'buckets', or bins. Because binning methods consult the neighborhood of values, they perform local smoothing.

In this technique,

1. The data for first sorted
2. Then the sorted list partitioned into equi-depth of bins.
3. Then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.

a. **Smoothing by bin means:** Each value in the bin is replaced by the mean value of the bin.

b. **Smoothing by bin medians:** Each value in the bin is replaced by the bin median.

c. **Smoothing by boundaries:** The min and max values of a bin are identified as the bin boundaries. Each bin value is replaced by the closest boundary value.

Example:

Sorted data for price (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

<b>Partition into (equal-frequency) bins:</b>	
Bin 1:	4, 8, 15
Bin 2:	21, 21, 24
Bin 3:	25, 28, 34
<b>Smoothing by bin means:</b>	
Bin 1:	9, 9, 9
Bin 2:	22, 22, 22
Bin 3:	29, 29, 29
<b>Smoothing by bin boundaries:</b>	
Bin 1:	4, 4, 15
Bin 2:	21, 21, 24
Bin 3:	25, 25, 34

---

Binning methods for data smoothing.

**2. Clustering :** Outliers may be detected by clustering, for example, where similar values are organized into groups, or “clusters.” Intuitively, values that fall outside of the set of clusters may be considered outliers. [ Refer Figure ]

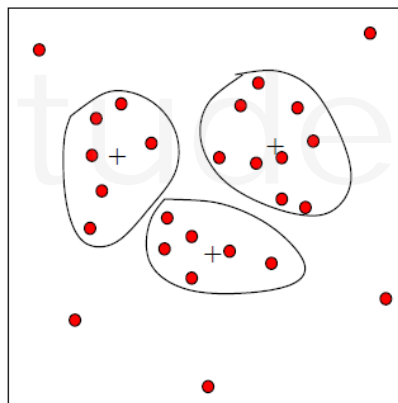


Figure 3.3: Outliers may be detected by clustering analysis.

**3. Combined computer and human inspection:** Outliers may be identified through a combination of **computer and human inspection**.

In one application, for example, an information-theoretic measure was used to help identify outlier patterns in a handwritten character database for classification. The measure's value reflected the “**surprise**” content of the predicted character label with respect to the known label. Outlier patterns may be informative (e.g., identifying useful data exceptions, such as different versions of the characters “0” or “7”), or “**garbage**” (e.g., mislabeled characters). Patterns whose surprise content is above a threshold are output to a list. A human can then sort through the patterns in the list to identify the actual garbage ones.

This is much faster than having **to manually search through the entire database**. The garbage patterns can then be removed from the (training) database. The garbage patterns can be excluded from use in subsequent data mining.

#### 4. Regression:

Data smoothing can also be done by regression, a technique that conforms data values to a function. **Linear regression** involves finding the “best” line to fit two attributes (or variables) so that one attribute can be used to predict the other. **Multiple linear regression** is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.

**Regression** :smooth by fitting the data into regression functions.

Linear regression involves finding the best of line to fit two variables, so that one variable can be used to predict the other.

Multiple Linear regression is an extension of linear regression, where more than two variables are involved and the data are fit to a multidimensional surface.

Using regression to find a mathematical equation to fit the data helps smooth out the noise.

#### Data Cleaning as a Process

The first step in data cleaning as a process is discrepancy detection. Discrepancies can be caused by several factors, including poorly designed data entry forms that have many optional fields, human error in data entry, deliberate errors (e.g., respondents not wanting to divulge information about themselves), and data decay (e.g., outdated addresses). Discrepancies may also arise from inconsistent data representations and the inconsistent use of codes.

“So, how can we proceed with discrepancy detection?” As a starting point, use any knowledge you may already have regarding properties of the data. Such knowledge or “data about data” is referred to as metadata.

The data should also be examined regarding unique rules, consecutive rules, and null rules.

**Field overloading:** is a kind of source of errors that typically occurs when developers compress new attribute definitions into unused portions of already defined attributes.

**Unique rule** is a rule says that each value of the given attribute must be different from all other values of that attribute

**Consecutive rule** is a rule says that there can be no missing values between the lowest and highest values of the attribute and that all values must also be unique.

**Null rule** specifies the use of blanks, question marks, special characters or other strings that may indicate the null condition and how such values should be handled.

There are a number of different commercial tools that can aid in the step of discrepancy detection. Data scrubbing tools use simple domain knowledge (e.g., knowledge of postal addresses, and spell-checking) to detect errors and make corrections in the data. These tools rely on parsing and fuzzy matching techniques when cleaning data from multiple sources. Data auditing tools find discrepancies by analyzing the data to discover rules and relationships, and detecting data that violate such conditions. They are variants of data mining tools.

## **Data Integration**

Data mining often requires data integration—the merging of data from multiple data stores. Careful integration can help reduce and avoid redundancies and inconsistencies in the resulting data set. This can help improve the accuracy and speed of the subsequent data mining process.

### **1. Entity Identification Problem**

There are a number of issues to consider during data integration. Schema integration and object matching can be tricky. How can equivalent real-world entities from multiple data sources be matched up? This is referred to as the **entity identification problem**.

**For example**, how can the data analyst or the computer be sure that `customer_id` in one database and `cust_number` in another refer to the same attribute? Examples of metadata for each attribute include the name, meaning, data type, and range of values permitted for the attribute, and null rules for handling blank, zero, or null values. Such metadata can be used to help avoid errors in schema integration.

### **2. Redundancy and Correlation Analysis**

Redundancy is another important issue in data integration. An attribute (such as annual revenue, for instance) may be redundant if it can be “derived” from another attribute or set of attributes. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set. Some redundancies can be detected by correlation analysis.

Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data. For nominal data, we use the  $\chi^2$  (chi-square) test. For numeric attributes, we can use the **correlation coefficient and covariance**, both of which access how one attribute’s values vary from those of another.

### **$\chi^2$ Correlation Test for Nominal Data**

For nominal data, a correlation relationship between two attributes, A and B, can be discovered by a  $\chi^2$  (chi-square) test. Suppose A has  $c$  distinct values, namely  $a_1, a_2, \dots, a_c$ . B has  $r$  distinct values, namely  $b_1, b_2, \dots, b_r$ . The data tuples described by A and B can be shown as a contingency table, with the  $c$  values of A making up the columns and the  $r$  values of B making up the rows. The  $\chi^2$  value (also known as the Pearson  $\chi^2$  statistic) is computed as,



$$\chi^2 = \sum_{i=1}^c \sum_{j=1}^r \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

where  $o_{ij}$  is the observed frequency (i.e., actual count) of the joint event and  $e_{ij}$  is the expected frequency which can be computed as,

$$e_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{n},$$

where  $n$  is the number of data tuples.

The  $\chi^2$  statistic tests the hypothesis that  $A$  and  $B$  are independent, that is, there is no correlation between them. The test is based on a significance level, with  $(r - 1) \times (c - 1)$  degrees of freedom. We illustrate the use of this statistic in Example 3.1. If the hypothesis can be rejected, then we say that  $A$  and  $B$  are statistically correlated.

Example: Correlation analysis of nominal attributes using  $\chi^2$ . Suppose that a group of 1500 people was surveyed. The gender of each person was noted. Each person was polled as to whether his or her preferred type of reading material was fiction or nonfiction. Thus, we have two attributes, gender and preferred reading. The observed frequency (or count) of each possible joint event is summarized in the contingency table, where the numbers in parentheses are the expected frequencies, the expected frequency for the cell (male, fiction) is,

$$e_{11} = \frac{\text{count}(\text{male}) \times \text{count}(\text{fiction})}{n} = \frac{300 \times 450}{1500} = 90,$$

and so on. Notice that in any row, the sum of the expected frequencies must equal the total observed frequency for that row, and the sum of the expected frequencies in any column must also equal the total observed frequency for that column.

Example 2.1's  $2 \times 2$  Contingency Table Data

	<i>male</i>	<i>female</i>	<i>Total</i>
<i>fiction</i>	250 (90)	200 (360)	450
<i>non-fiction</i>	50 (210)	1000 (840)	1050
<i>Total</i>	300	1200	1500

Note: Are *gender* and *preferred\_reading* correlated?

Using Eq. (3.1) for  $\chi^2$  computation, we get

$$\begin{aligned} \chi^2 &= \frac{(250 - 90)^2}{90} + \frac{(50 - 210)^2}{210} + \frac{(200 - 360)^2}{360} + \frac{(1000 - 840)^2}{840} \\ &= 284.44 + 121.90 + 71.11 + 30.48 = 507.93. \end{aligned}$$

For this  $2 \times 2$  table, the degrees of freedom are  $(2 - 1)/(2 - 1) = 1$ . For 1 degree of freedom, the  $\chi^2$  value needed to reject the hypothesis at the 0.001 significance level is 10.828 (taken from the table of upper percentage points of the  $\chi^2$  distribution, typically available from any textbook on statistics). Since our computed value is above this, we



can reject the hypothesis that gender and preferred reading are independent and conclude that the two attributes are (strongly) correlated for the given group of people.

### Correlation Coefficient for Numeric Data

For numeric attributes, we can evaluate the correlation between two attributes, A and B, by computing the correlation coefficient (also known as Pearson's product moment coefficient). This is,

$$r_{A,B} = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^n (a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B},$$

where n is the number of tuples,  $a_i$  and  $b_i$  are the respective values of A and B in tuple i,  $\bar{A}$  and  $\bar{B}$  are the respective mean values of A and B,  $\sigma_A$  and  $\sigma_B$  are the respective standard deviations of A and B.

### Covariance of Numeric Data

In probability theory and statistics, correlation and covariance are two similar measures for assessing how much two attributes change together. Consider two numeric attributes A and B, and a set of n observations. The mean values of A and B, respectively are also known as the expected values on A and B, that is,

$$E(A) = \bar{A} = \frac{\sum_{i=1}^n a_i}{n}$$

and

$$E(B) = \bar{B} = \frac{\sum_{i=1}^n b_i}{n}.$$

The **covariance** between A and B is defined as

$$Cov(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n}. \quad (3.4)$$

If we compare Eq. (3.3) for  $r_{A,B}$  (correlation coefficient) with Eq. (3.4) for covariance, we see that

$$r_{A,B} = \frac{Cov(A, B)}{\sigma_A\sigma_B}, \quad (3.5)$$

where  $\sigma_A$  and  $\sigma_B$  are the standard deviations of A and B, respectively.

### 3. Tuple Duplication

In addition to detecting redundancies between attributes, duplication should also be detected at the tuple level (e.g., where there are two or more identical tuples for a given unique data entry case). The use of denormalized tables (often done to improve performance by avoiding joins) is another source of data redundancy. Inconsistencies often arise between various duplicates, due to inaccurate data entry or updating some but not all data occurrences.

**For example**, if a purchase order database contains attributes for the purchaser's name and address instead of a key to this information in a purchaser database, discrepancies can occur, such as the same purchaser's name appearing with different addresses within the purchase order database.

#### **4. Data Value Conflict Detection and Resolution**

Data integration also involves the detection and resolution of data value conflicts. For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding.

For instance, a weight attribute may be stored in metric units in one system and British imperial units in another. For a hotel chain, the price of rooms in different cities may involve not only different currencies but also different services (e.g., free breakfast) and taxes. When exchanging information between schools, for example, each school may have its own curriculum and grading scheme. One university may adopt a quarter system, offer three courses on database systems, and assign grades from A+ to F, whereas another may adopt a semester system, offer two courses on databases, and assign grades from 1 to 10. It is difficult to work out precise course-to-grade transformation rules between the two universities, making information exchange difficult.

### **Data Reduction**

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results.

Data reduction strategies include dimensionality reduction, numerosity reduction, and data compression.

**Numerosity reduction** techniques replace the original data volume by alternative, smaller forms of data representation. These techniques may be parametric or nonparametric. For parametric methods, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. (Outliers may also be stored.) Regression and log-linear models are examples. Nonparametric methods for storing reduced representations of the data include histograms, clustering, sampling, and data cube aggregation .

In **data compression**, transformations are applied so as to obtain a reduced or “compressed” representation of the original data. If the original data can be reconstructed from the compressed data without any information loss, the data reduction is called lossless. If, instead, we can reconstruct only an approximation of the original data, then the data reduction is called lossy. There are several lossless algorithms for string compression; however, they typically allow only limited data

manipulation. Dimensionality reduction and numerosity reduction techniques can also be considered forms of data compression.

**Dimensionality reduction** is the process of reducing the number of random variables or attributes under consideration. Dimensionality reduction methods include wavelet transforms and principal components analysis, which transform or project the original data onto a smaller space. Attribute subset selection is a method of dimensionality reduction in which irrelevant, weakly relevant, or redundant attributes or dimensions are detected and removed.

### Wavelet Transforms

The general procedure for applying a discrete wavelet transform uses a hierarchical pyramid algorithm that halves the data at each iteration, resulting in fast computational speed. The method is as follows:

1. The length,  $L$ , of the input data vector must be an integer power of 2. This condition can be met by padding the data vector with zeros as necessary ( $L \geq n$ ).
2. Each transform involves applying two functions. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference, which acts to bring out the detailed features of the data.
3. The two functions are applied to pairs of data points in  $X$ , that is, to all pairs of measurements. This results in two data sets of length  $L/2$ . In general, these represent a smoothed or low-frequency version of the input data and the high-frequency content of it, respectively.
4. The two functions are recursively applied to the data sets obtained in the previous loop, until the resulting data sets obtained are of length 2.
5. Selected values from the data sets obtained in the previous iterations are designated the wavelet coefficients of the transformed data.

### Principal Components Analysis

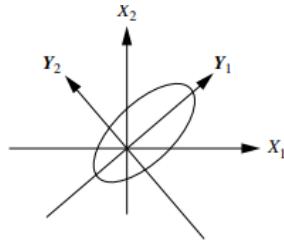
Suppose that the data to be reduced consist of tuples or data vectors described by  $n$  attributes or dimensions. Principal components analysis (PCA; also called the Karhunen-Loeve, or K-L, method) searches for  $k$   $n$ -dimensional orthogonal vectors that can best be used to represent the data, where  $k \leq n$ .

The basic procedure is as follows:

1. The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.
2. PCA computes  $k$  orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the principal components. The input data are a linear combination of the principal components.

3. The principal components are sorted in order of decreasing “significance” or strength. The principal components essentially serve as a new set of axes for the data providing important information about variance.

4. Because the components are sorted in decreasing order of “significance,” the data size can be reduced by eliminating the weaker components, that is, those with low variance. Using the strongest principal components, it should be possible to reconstruct a good approximation of the original data.



---

Principal components analysis.  $Y_1$  and  $Y_2$  are the first two principal components for the given data.

### Attribute Subset Selection

Attribute subset selection<sup>4</sup> reduces the data set size by removing irrelevant or redundant attributes (or dimensions). The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes. Mining on a reduced set of attributes has an additional benefit: It reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

The “best” (and “worst”) attributes are typically determined using tests of statistical significance, which assume that the attributes are independent of one another. Many other attribute evaluation measures can be used such as the information gain measure used in building decision trees for classification.<sup>5</sup> Basic heuristic methods of attribute subset selection include the techniques that follow,

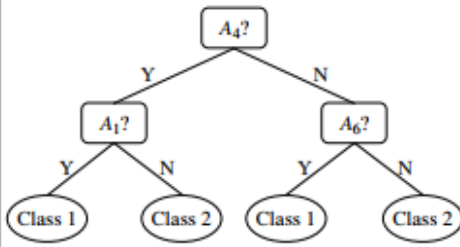
1. **Stepwise forward selection:** The procedure starts with an empty set of attributes as the reduced set. The best of the original attributes is determined and added to the reduced set. At each subsequent iteration or step, the best of the remaining original attributes is added to the set.

2. **Stepwise backward elimination:** The procedure starts with the full set of attributes. At each step, it removes the worst attribute remaining in the set.

3. **Combination of forward selection and backward elimination:** The stepwise forward selection and backward elimination methods can be combined so that, at each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.

4. **Decision tree induction:** Decision tree algorithms (e.g., ID3, C4.5, and CART) were originally intended for classification. Decision tree induction constructs a

flowchart like structure where each internal (non leaf) node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each external (leaf) node denotes a class prediction. At each node, the algorithm chooses the “best” attribute to partition the data into individual classes.

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$    $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$

When decision tree induction is used for attribute subset selection, a tree is constructed from the given data. All attributes that do not appear in the tree are assumed to be irrelevant. The set of attributes appearing in the tree form the reduced subset of attributes.

The stopping criteria for the methods may vary. The procedure may employ a threshold on the measure used to determine when to stop the attribute selection process.

## Numerosity reduction

Numerosity reduction techniques replace the original data volume by alternative, smaller forms of data representation. These techniques may be parametric or nonparametric.

For parametric methods, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. (Outliers may also be stored.) Regression and log-linear models are examples.

Nonparametric methods for storing reduced representations of the data include histograms, clustering, sampling.

### ▪ Histograms : Non - Parametric method for numerosity reduction

Histograms use binning to approximate data distributions and are a popular form of data reduction. A histogram for an attribute, A, partitions the data distribution of A into disjoint subsets, referred to as buckets or bins.

If each bucket represents only a single attribute–value/frequency pair, the buckets are called singleton buckets. Often, buckets instead represent continuous ranges for the given attribute.

### Example

The following data are a list of prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

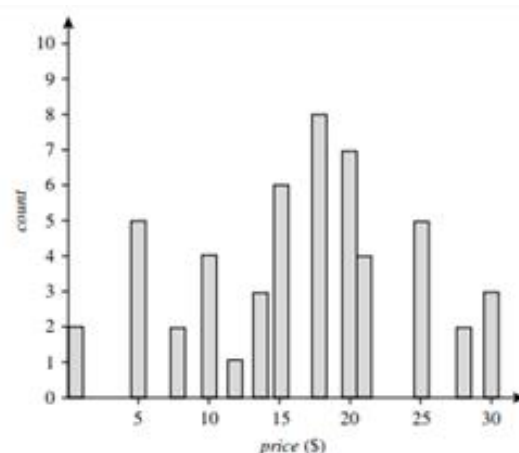
Figure 3.7 shows a histogram for the data using singleton buckets. To further reduce the data, it is common to have each bucket denote a continuous value range for the given attribute. In Figure 3.8, each bucket represents a different \$10 range for price.

“How are the buckets determined and the attribute values partitioned?” There are several partitioning rules, including the following:

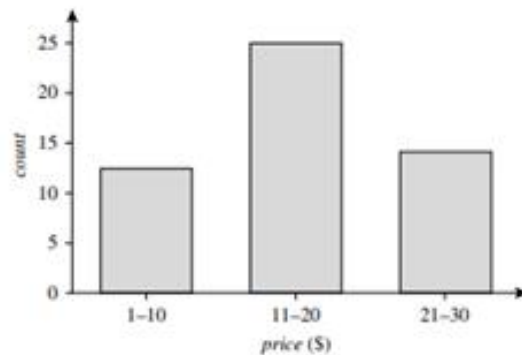
**Equal-width:** In an equal-width histogram, the width of each bucket range is uniform (e.g., the width of \$10 for the buckets in Figure 3.8).

**Equal-frequency** (or equal-depth): In an equal-frequency histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (i.e., each bucket contains roughly the same number of contiguous data samples).

Histograms are highly effective at approximating both sparse and dense data, as well as highly skewed and uniform data. The histograms described before for single attributes can be extended for multiple attributes. Multidimensional histograms can capture dependencies between attributes. These histograms have been found effective in approximating data with up to five attributes. More studies are needed regarding the effectiveness of multidimensional histograms for high dimensionalities. Singleton buckets are useful for storing high-frequency outliers.



**Figure 3.7** A histogram for *price* using singleton buckets—each bucket represents one price–value/frequency pair.

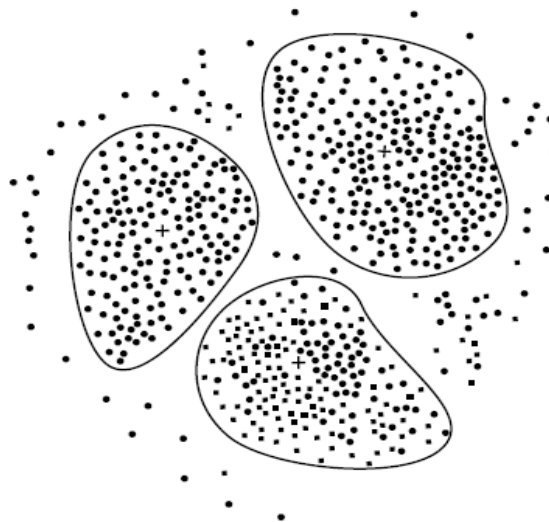


**Figure 3.8** An equal-width histogram for *price*, where values are aggregated so that each bucket has a uniform width of \$10.

## Clustering

Clustering techniques consider data tuples as objects. They partition the objects into groups, or clusters, so that objects within a cluster are “similar” to one another and “dissimilar” to objects in other clusters.

Similarity is commonly defined in terms of how “close” the objects are in space, based on a distance function. The “**quality**” of a cluster may be represented by its **diameter**, the maximum distance between any two objects in the cluster. **Centroid distance** is an alternative measure of cluster quality and is defined as the average distance of each cluster object from the cluster centroid (denoting the “average object,” or average point in space for the cluster).



A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster centroid is marked with a “+”, representing the average point in space for that cluster. Outliers may be detected as values that fall outside of the sets of clusters.

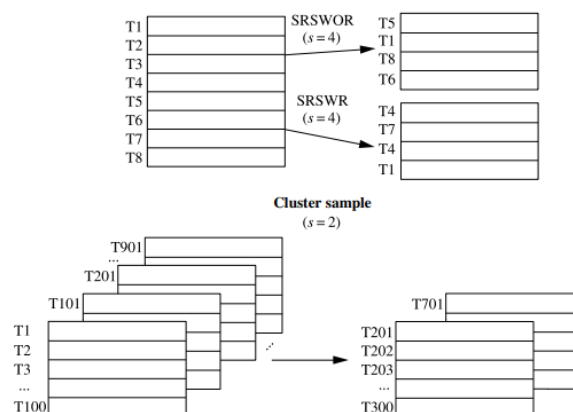


In data reduction, the cluster representations of the data are used to replace the actual data. The effectiveness of this technique depends on the data's nature. It is much more effective for data that can be organized into distinct clusters than for smeared data.

## ▪ Sampling

Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random data sample (or subset). Suppose that a large data set,  $D$ , contains  $N$  tuples. Let's look at the most common ways that we could sample  $D$  for data reduction, as illustrated in Figure 3.9.

1. **Simple random sample without replacement (SRSWOR)** of size  $s$ : This is created by drawing  $s$  of the  $N$  tuples from  $D$  ( $s < N$ ), where the probability of drawing any tuple in  $D$  is  $1/N$ , that is, all tuples are equally likely to be sampled.
2. **Simple random sample with replacement (SRSWR)** of size  $s$ : This is similar to SRSWOR, except that each time a tuple is drawn from  $D$ , it is recorded and then replaced. That is, after a tuple is drawn, it is placed back in  $D$  so that it may be drawn again.
3. **Cluster sample**: If the tuples in  $D$  are grouped into  $M$  mutually disjoint "clusters," then an SRS of  $s$  clusters can be obtained, where  $s < M$ . For example, tuples in a database are usually retrieved a page at a time, so that each page can be considered a cluster. A reduced data representation can be obtained by applying, say, SRSWOR to the pages, resulting in a cluster sample of the tuples.
4. **Stratified sample**: If  $D$  is divided into mutually disjoint parts called strata, a stratified sample of  $D$  is generated by obtaining an SRS at each stratum. This helps ensure a representative sample, especially when the data are skewed. For example, a stratified sample may be obtained from customer data, where a stratum is created for each customer age group. In this way, the age group having the smallest number of customers will be sure to be represented.



**Stratified sample**  
(according to *age*)

T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

- **Regression and Log-Linear Models: Parametric Data Reduction**

Regression and log-linear models can be used to approximate the given data. In linear regression, the data are modeled to fit a straight line.

For example, a random variable,  $Y$  (called a response variable), can be modeled as a linear function of another random variable,  $x$  (called a predictor variable), with the equation

$$y = wx + b,$$

where the variance of  $y$  is assumed to be constant. In the context of data mining,  $x$  and  $y$  are numeric database attributes. The coefficients,  $w$  and  $b$  (called regression coefficients), specify the slope of the line and the  $y$ -intercept, respectively. These coefficients can be solved for by the method of least squares, which minimizes the error between the actual line separating the data and the estimate of the line. Multiple linear regression is an extension of (simple) linear regression, which allows a response variable,  $y$ , to be modelled as a linear function of two or more predictor variables.

- **Data Cube Aggregation**

Imagine that you have collected the data for your analysis. These data consist of sales per quarter, for the years 2008 to 2010. You are, however, interested in the annual sales (total per year), rather than the total per quarter. Thus, the data can be aggregated so that the resulting data summarize the total sales per year instead of per quarter. The resulting data set is smaller in volume, without loss of information necessary for the analysis task.

Year 2010	
Quarter	Sales
Q1	\$224,000
Q2	\$408,000
Q3	\$350,000
Q4	\$586,000

Year 2009	
Quarter	Sales
Q1	\$224,000
Q2	\$408,000
Q3	\$350,000
Q4	\$586,000

Year 2008	
Quarter	Sales
Q1	\$224,000
Q2	\$408,000
Q3	\$350,000
Q4	\$586,000

Year	Sales
2008	\$1,568,000
2009	\$2,356,000
2010	\$3,594,000

Sales data for a given branch of *AllElectronics* for the years 2008 through 2010. On the *left*, the sales are shown per quarter. On the *right*, the data are aggregated to provide the annual sales.

## Data Transformation and Data Discretization

In data transformation, the data are transformed or consolidated into forms appropriate for mining. Strategies for data transformation include the following:

1. **Smoothing**, which works to remove noise from the data. Techniques include binning, regression, and clustering.

2. **Attribute construction** (or feature construction), where new attributes are constructed and added from the given set of attributes to help the mining process.

3. **Aggregation**, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for data analysis at multiple abstraction levels.

4. **Normalization**, where the attribute data are scaled so as to fall within a smaller range, such as -1.0 to 1.0, or 0.0 to 1.0.

5. **Discretization**, where the raw values of a numeric attribute (e.g., age) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior). The labels, in turn, can be recursively organized into higher level concepts, resulting in a concept hierarchy for the numeric attribute. Figure 3.12 shows a concept hierarchy for the attribute price. More than one concept hierarchy can be defined for the same attribute to accommodate the needs of various users.

6. **Concept hierarchy generation** for nominal data, where attributes such as street can be generalized to higher-level concepts, like city or country. Many hierarchies for nominal attributes are implicit within the database schema and can be automatically defined at the schema definition level.

## Data Transformation by Normalization

The measurement unit used can affect the data analysis. For example, changing measurement units from meters to inches for height, or from kilograms to pounds for weight, may lead to very different results. In general, expressing an attribute in smaller units will lead to a larger range for that attribute, and thus tend

to give such an attribute greater effect or “weight.” To help avoid dependence on the choice of measurement units, the data should be normalized or standardized. This involves transforming the data to fall within a smaller or common range such as [-1,1] or [0.0, 1.0].

There are many methods for data normalization. We study min-max normalization, z-score normalization, and normalization by decimal scaling. For our discussion, let A be a numeric attribute with n observed values,  $v_1, v_2, \dots, v_n$ .

**Min-max normalization** performs a linear transformation on the original data. Suppose that  $\min_A$  and  $\max_A$  are the minimum and maximum values of an attribute, A. Min-max normalization maps a value,  $v_i$ , of A to  $v'_i$  in the range [new  $\min_A$ , new  $\max_A$ ] by computing

$$v'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A.$$

Min-max normalization preserves the relationships among the original data values. It will encounter an “out-of-bounds” error if a future input case for normalization falls outside of the original data range for A.

**Example.** Suppose that the minimum and maximum values for the attribute income are 12,000 and 98,000, respectively. We would like to map income to the range [0.0,1.0]. By min-max normalization, a value of 73,600 for income is transformed to,

$$\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716.$$

**z-score normalization** (or zero-mean normalization), the values for an attribute, A, are normalized based on the mean (i.e., average) and standard deviation of A. A value,  $v_i$ , of A is normalized to  $v'_i$  by computing

$$v'_i = \frac{v_i - \bar{A}}{\sigma_A},$$

where  $\bar{A}$  and  $\sigma_A$  are the mean and standard deviation, respectively, of attribute A.

**Decimal scaling.** Suppose that the recorded values of A range from -986 to 917. The maximum absolute value of A is 986. To normalize by decimal scaling, we therefore divide each value by 1000 (i.e.,  $j=3$ ) so that -986 normalizes to -0.986 and 917 normalizes to 0.917.

Note that normalization can change the original data quite a bit, especially when using z-score normalization or decimal scaling. It is also necessary to save the normalization parameters (e.g., the mean and standard deviation if using z-score normalization) so that future data can be normalized in a uniform manner.

## **Discretization by Binning**

Binning is a top-down splitting technique based on a specified number of bins. Binning methods are also used as discretization methods for data reduction and concept hierarchy generation.

For example, attribute values can be discretized by applying equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median, as in smoothing by bin means or smoothing by bin medians, respectively. These techniques can be applied recursively to the resulting partitions to generate concept hierarchies.

Binning does not use class information and is therefore an unsupervised discretization technique. It is sensitive to the user-specified number of bins, as well as the presence of outliers.

## **Discretization by Histogram Analysis**

A histogram partitions the values of an attribute,  $A$ , into disjoint ranges called buckets or bins. In an equal-width histogram, for example, the values are partitioned into equal-size partitions or ranges. With an equal-frequency histogram, the values are partitioned so that, ideally, each partition contains the same number of data tuples. The histogram analysis algorithm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy, with the procedure terminating once a prespecified number of concept levels has been reached. A minimum interval size can also be used per level to control the recursive procedure. This specifies the minimum width of a partition, or the minimum number of values for each partition at each level..

## **Discretization by Cluster, Decision tree and correlation Analysis**

Cluster analysis is a popular data discretization method. A clustering algorithm can be applied to discretize a numeric attribute,  $A$ , by partitioning the values of  $A$  into clusters or groups. Clustering takes the distribution of  $A$  into consideration, as well as the closeness of data points, and therefore is able to produce high-quality discretization results.

Clustering can be used to generate a concept hierarchy for  $A$  by following either a top-down splitting strategy or a bottom-up merging strategy, where each cluster forms a node of the concept hierarchy. In the former, each initial cluster or partition may be further decomposed into several subclusters, forming a lower level of the hierarchy. Clusters are formed by repeatedly grouping neighboring clusters in order to form higher-level concepts. decision tree-based discretization uses class information, it is more likely that the interval boundaries (split-points) are defined to occur in places that may help improve classification accuracy

Measures of correlation can be used for discretization. ChiMerge is a  $\chi^2$ -based discretization method. ChiMerge proceeds as follows. Initially, each

distinct value of a numeric attribute  $A$  is considered to be one interval.  $\chi^2$  tests are performed for every pair of adjacent intervals. Adjacent intervals with the least  $\chi^2$  values are merged together, because low  $\chi^2$  values for a pair indicate similar class distributions. This merging process proceeds recursively until a predefined stopping criterion is met.

## Concept Hierarchy Generation for Nominal Data \*\*\*\*\*

Nominal attributes have a finite (but possibly large) number of distinct values, with no ordering among the values. Examples include geographic location, job category, and item type. Manual definition of concept hierarchies can be a tedious and time-consuming task for a user or a domain expert. Fortunately, many hierarchies are implicit within the database schema and can be automatically defined at the schema definition level. The concept hierarchies can be used to transform the data into multiple levels of granularity. For example, data mining patterns regarding sales may be found relating to specific regions or countries, in addition to individual branch locations.

Four methods for the generation of concept hierarchies for nominal data,

### 1. Specification of a partial ordering of attributes explicitly at the schema level by users or experts:

Concept hierarchies for nominal attributes or dimensions typically involve a group of attributes. A user or expert can easily define a concept hierarchy by specifying a partial or total ordering of the attributes at the schema level.

For example, suppose that a relational database contains the following group of attributes: street, city, province or state, and country. Similarly, a data warehouse location dimension may contain the same attributes.

A hierarchy can be defined by specifying the total ordering among these attributes at the schema level such as street < city < province or state < country.

### 2. Specification of a portion of a hierarchy by explicit data grouping:

In a large database, it is unrealistic to define an entire concept hierarchy by explicit value enumeration. On the contrary, we can easily specify explicit groupings for a small portion of intermediate-level data.

For example, after specifying that *province* and *country* form a hierarchy at the schema level, a user could define some intermediate levels manually, such as {Alberta, Saskatchewan, Manitoba}  $\subset$  {prairies\_Canada} and {British Columbia, prairies Canada}  $\subset$  Western\_Canada."

### 3. Specification of a set of attributes, but not of their partial ordering:

A user may specify a set of attributes forming a concept hierarchy, but omit to explicitly state their partial ordering. The system can then try to

automatically generate the attribute ordering so as to construct a meaningful concept hierarchy.

“Without knowledge of data semantics, how can a hierarchical ordering for an arbitrary set of nominal attributes be found?”

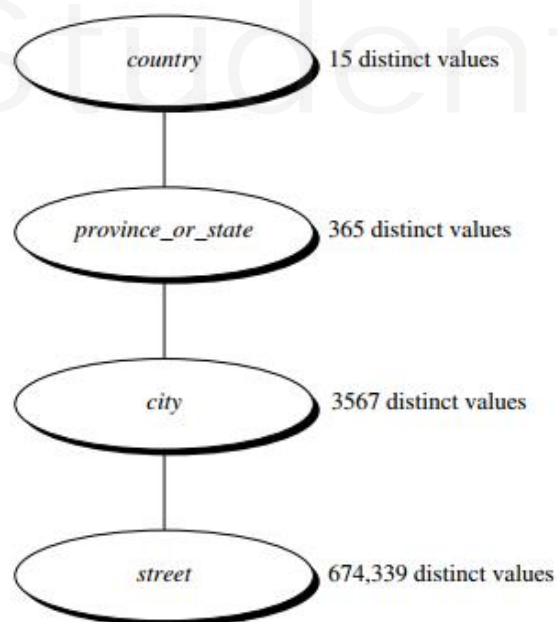
Consider the observation that since higher-level concepts generally cover several subordinate lower-level concepts, an attribute defining a high concept level (e.g., country) will usually contain a smaller number of distinct values than an attribute defining a lower concept level (e.g., street). Based on this observation, a concept hierarchy can be automatically generated based on the number of distinct values per attribute in the given attribute set.

The attribute with the most distinct values is placed at the lowest hierarchy level. The lower the number of distinct values an attribute has, the higher it is in the generated concept hierarchy. This heuristic rule works well in many cases. Some local-level swapping or adjustments may be applied by users or experts, when necessary, after examination of the generated hierarchy.

### **Example**

Suppose a user selects a set of location-oriented attributes—street, country, province or state, and city—from the AllElectronics database, but does not specify the hierarchical ordering among the attributes.

A concept hierarchy for location can be generated automatically, as illustrated in Figure.



First, sort the attributes in ascending order based on the number of distinct values in each attribute. (The number of distinct values per attribute is shown in parentheses): country (15), province or state (365), city (3567), and street (674,339).

Second, generate the hierarchy from the top down according to the sorted order, with the first attribute at the top level and the last attribute at the bottom level.



Finally, the user can examine the generated hierarchy, and when necessary, modify it to reflect desired semantic relationships among the attributes.

#### **4. Specification of only a partial set of attributes:**

Sometimes a user can be careless when defining a hierarchy, or have only a vague idea about what should be included in a hierarchy. Consequently, the user may have included only a small subset of the relevant attributes in the hierarchy specification.

For example, instead of including all of the hierarchically relevant attributes for location, the user may have specified only street and city. To handle such partially specified hierarchies, it is important to embed data semantics in the database schema so that attributes with tight semantic connections can be pinned together. In this way, the specification of one attribute may trigger a whole group of semantically tightly linked attributes to be “dragged in” to form a complete hierarchy. Users, however, should have the option to override this feature, as necessary.

\*\*\*\*\*

KTUStudents.in