UNIT-2

Getting to Know Your Data: Data objects and Attribute Types, Basic statistical descriptions of data, Measuring Data Similarity and Dissimilarity.
Data Preprocessing: An overview, Data Cleaning, Data integration, Data Reduction, Data Transformation and Discretization.

Getting to Know Your Data

Data objects and Attribute Types:
Data sets are made up of data objects.
A data object represents an entity— **in a sales database**, the objects may be customers, store items, and sales;
in a medical database, the objects may be patients;
in a university database, the objects may be students, professors, and courses.
Data objects are described by attributes. Data objects can also be referred to as samples, examples, instances, data points, or objects. If the data objects are stored in a database, they are data tuples. That is, the rows of a database correspond to the data objects, and the columns correspond to the attributes.

What Is an Attribute? :
An attribute is a data field, representing a characteristic or feature of a data object. Attributes describing a customer object can include, for example, customer ID, name, and address. Observed values for a given attribute are known as observations. A set of attributes used to describe a given object is called an attribute vector (or feature vector). The distribution of data involving one attribute (or variable) is called univariate. A bivariate distribution involves two attributes, and so on.

Types of Attributes:

- Nominal Attributes
- Binary Attributes
- Ordinal Attributes
- Numeric Attributes
- Discrete versus Continuous Attributes

Nominal attributes
Nominal means "relating to names." The values of a nominal attribute are symbols or names of things. Each value represents some kind of category, code, or state, and so nominal attributes are also referred to as categorical. The values do not have any meaningful order.

Examples:

- Hair color (blonde, gray, brown, black, etc.)
- Relationship status (married, cohabiting, single, etc.)
- Preferred mode of public transportation (bus, train, tram, etc.)
- Blood type (O negative, O positive, A negative, and so on)

Because nominal attribute values do not have any meaningful order about them and are not quantitative, it makes no sense to find the mean (average) value or median (middle) value for such an attribute, given a set of objects. One thing that is of interest, however, is the attribute's most commonly occurring value. This value, known as the mode, is one of the measures of central tendency.

Binary attributes: A binary attribute is a nominal attribute with only two categories or states: 0 or 1, where 0 means that the attribute is absent, and 1 means that it is present. Binary attributes are referred to as Boolean if the two states correspond to *true* and *false*.

For example, suppose the patient undergoes a medical test that has two possible outcomes. The attribute *medical test* is binary, where a value of 1 means the result of the test for the patient is positive, while 0 means the result is negative.

A binary attribute is symmetric if both of its states are equally valuable and carry the same weight. So, there is no preference on which outcome should be coded as 0 or 1. One such example could be the attribute gender having the states male and female.

A binary attribute is asymmetric if the outcomes of the states are not equally important, such as the positive and negative outcomes of a medical test for some severe disease. By convention, we code the most important outcome, which is usually the rarest one, by 1 (positive) and the other by 0 (negative).

Ordinal Attribute

An ordinal attribute is an attribute with possible values that have a meaningful order or ranking among them, but quantitative measure between successive values is not known.

For Example Suppose that drink size corresponds to the size of drinks available at a fast-food restaurant. This nominal attribute has three possible values: small, medium, and large. The values have a meaningful sequence (which corresponds to increasing drink size); however, we cannot tell from the values how much bigger, say, a medium is than a large.

Numeric attributes

A numeric attribute is quantitative, and it is presented as a measure of quantity, represented in integer or real values. Numeric attributes can be *interval-scaled* or *ratio-scaled*.

Interval-scaled attributes are measured on a scale of equal-size units. This type of data represents quantitative data with equal intervals between consecutive values. Interval data has no absolute zero point, and therefore, ratios cannot be computed.

Examples of interval data include temperature, IQ scores, and time. Interval data is used in data mining for clustering and prediction tasks. Because interval-scaled attributes are numeric, we can compute their mean value, in addition to the median and mode measures of central tendency.

A **ratio-scaled** attribute is a numeric attribute with an inherent zero-point.

This type of data is similar to interval data, but with an absolute zero point. In ratio data, it is possible to compute ratios of two values, and this makes it possible to make meaningful comparisons.

Examples of ratio data include height, weight, and income. Ratio data is used in data mining for prediction and association rule mining tasks.

Discrete versus Continuous Attributes:

Discrete Attribute

A variable or attribute is discrete if it can take a finite or a countably infinite set of values.

A discrete attribute has a finite or countably infinite set of values, which may or may not be represented as integers. The attributes hair color, smoker, medical test, and drink size each have a finite number of values, and so are discrete.

Basic Statistical Descriptions of Data:

For data preprocessing to be successful, it is essential to have an overall picture of your data. Basic statistical descriptions can be used to identify properties of the data and highlight which data values should be treated as noise or outliers.

This section discusses three areas of basic statistical descriptions, those are:

- **Measuring the Central Tendency**: Mean, Median, and Mode
- **Measuring the Dispersion of Data**: Range, Quar tiles, Variance, Standard Deviation, and Interquartile Range (IQR)
- **Graphic Displays of Basic Statistical Descriptions of Data**

Measuring the Central Tendency: Mean, Median, and Mode:

The central tendency is stated as the statistical measure that represents the single value of the entire distribution or a dataset. It aims to provide an accurate description of the entire data in the distribution.

Suppose that we have some attribute X, like salary, which has been recorded for a set of objects.

Let x1,x2,...,xN be the set of N observed values or observations for X. Here, these values may also be referred to as the data set (for X). If we were to plot the observations for salary, where would most of the values fall? This gives us an idea of the central tendency of the data. Measures of central tendency include the mean, median, mode, and midrange.

Mean:

Mean is the average of all data values which you work with. Mean is used to find the average value around which your data values range.

The most common and effective numeric measure of the "center" of a set of data is the (arithmetic) mean. Let x1,x2,...,xN be a set of N values or observations, such as for some numeric attribute X, like salary.
The mean of this set of values is

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + \cdots + x_N}{N}.$$

This corresponds to the built-in aggregate function, average (avg() in SQL), provided in relational database systems.
Example : Mean. Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110. Using above equation we have

$$\bar{x} = \frac{30 + 36 + 47 + 50 + 52 + 52 + 56 + 60 + 63 + 70 + 70 + 110}{12}$$
$$= \frac{696}{12} = 58.$$

Thus, the mean salary is $58,000.
weighted arithmetic mean:
Sometimes, each value xi in a set may be associated with a weight wi for i = 1,...,N. The weights reflect the significance, importance, or occurrence frequency attached to their respective values. In this case, we can compute

$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_N x_N}{w_1 + w_2 + \cdots + w_N}.$$

This is called the weighted arithmetic mean or the weighted average. Although the mean is the single most useful quantity for describing a data set, it is not always the best way of measuring the center of the data. – A major problem with the mean is its sensitivity to extreme (outlier) values. – Even a small number of extreme values can corrupt the mean.
• To offset the effect caused by a small number of extreme values, we can instead use the **trimmed mean**,
• Trimmed mean can be obtained after chopping off values at the high and low extremes.
For example, we can sort the values observed for salary and remove the top and bottom 2% before computing the mean. We should avoid trimming too large a portion (such as 20%) at both ends, as this can result in the loss of valuable information.
Median: Another measure of the center of data is the median.
• Suppose that a given data set of N distinct values is sorted in numerical order.
– If N is odd, the median is the middle value of the ordered set;
– If N is even, the median is the average of the middle two values.
In probability and statistics, the median generally applies to numeric data; however, we may extend the concept to ordinal data.
Example: Median. Let's find the median of the data. Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.
The data are already sorted in increasing order. There is an even number of observations (i.e., 12); therefore, the median is not unique. It can be any value within the two middlemost values of 52 and 56 (that is, within the sixth and seventh values in the list).
By convention, we assign the average of the two middlemost values
as the median; that is, 52+56⁄2 = 108 ⁄2 = 54.
Thus, the median is $54,000.
Suppose that we had only the first 11 values in the list. Given an odd number of values, the median is the middlemost value. This is the sixth value in this list, which has a value of $52,000.
We can approximate the median of the entire data set (e.g., the median salary) by interpolation using the formula

$$median = L_1 + \left(\frac{N/2 - (\sum freq)_l}{freq_{median}} \right) width,$$

Where
L1 is the lower boundary of the median interval, N is the number of values in the entire data set,
$(\sum freq)_l$ is the sum of the frequencies of all of the intervals that are lower than the median interval
$freq_{median}$ is the frequency of the median interval, and width is the width of the median interval.

Mode: The mode is another measure of central tendency. The mode for a set of data is the value that occurs most frequently in the set. Therefore, it can be determined for qualitative and quantitative attributes.

– It is possible for the greatest 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 unimodal, bimodal, and trimodal. In general, a data set with two or more modes is multimodal. At the other extreme, if each data value occurs only once, then there is no mode

Example: Mode.

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.

The two modes are $52,000 and $70,000.

Based on the properties of the data, the measures of central tendency are selected.If you have a symmetrical distribution of continuous data, all the three measures of central tendency hold good. But most of the times, the analyst uses the mean because it involves all the values in the distribution or dataset.

If you have skewed distribution, the best measure of finding the central tendency is the median.

If you have the original data, then both the median and mode are the best choice of measuring the central tendency.If you have categorical data, the mode is the best choice to find the central tendency.

Midrange: the number that is exactly halfway between the minimum and maximum numbers in a set of data. To work out the midrange, you must find the sum of both the smallest and largest, and divide it by 2.

The midrange can also be used to assess the central tendency of a numeric data set. It is the average of the largest and smallest values in the set. This measure is easy to compute using the SQL aggregate functions, max() and min().

The midrange of the data is 30,000+110,000⁄2 = $70,000.

In a unimodal frequency curve with perfect symmetric data distribution, the mean, median, and mode are all at the same center value, as shown in Figure 2.1(a).

Data in most real applications are not symmetric. They may instead be either positively skewed, where the mode occurs at a value that is smaller than the median (Figure 2.1b), or negatively skewed, where the mode occurs at a value greater than the median (Figure 2.1c).

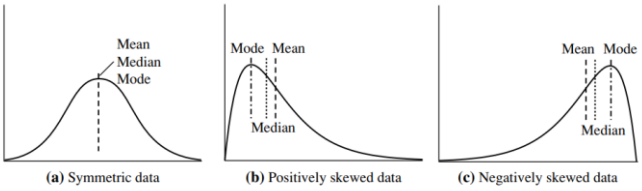


Figure 2.1 Mean, median, and mode of symmetric versus positively and negatively skewed data.

Measuring the Dispersion of Data: Range, Quartiles, Variance, Standard Deviation, and Interquartile Range:

Although measure of central tendency provides us with information on Mean, Media, and Mode, additional information about the data can be supplemented by measures of Dispersion. The main purpose of measures of dispersion is to get as much as possible a true picture of the data. Dispersion indicates the spread or variability of the data in a distribution. In other words, dispersion is the amount of spread of data about the centre of the distribution.

Range:

The range is the easiest dispersion of data or measure of variability. The range can measure by subtracting the lowest value from the massive Number. The wide range indicates high variability, and the small range specifies

low variability in the distribution. To calculate a range, prepare all the values in ascending order, then subtract the lowest value from the highest value.

Range = Highest_value – Lowest_value

Let x1,x2,...,xN be a set of observations for some numeric attribute, X. The range of the set is the difference between the largest (max()) and smallest (min()) values.

Quantiles:

Suppose that the data for attribute X are sorted in increasing numeric order. Imagine that we can pick certain data points so as to split the data distribution into equal-size consecutive sets, as in Figure 2.2. These data points are called quantiles. Quantiles are points taken at regular intervals of a data distribution, dividing it into essentially equalsize consecutive sets.

The kth -q-quantile for a given data distribution is the value x such that at most k/q of the data values are less than x and at most (q − k)/q of the data values are more than x, where k is an integer such that 0 < k < q. There are q − 1 q-quantiles.

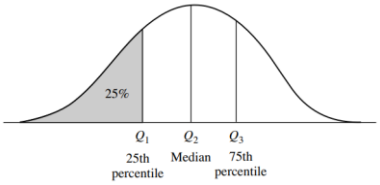


Figure 2.2 A plot of the data distribution for some attribute X

The 2-quantile is the data point dividing the lower and upper halves of the data distribution. It corresponds to the median.

The 4-quantiles are the three data points that split the data distribution into four equal parts; each part represents one-fourth of the data distribution. They are more commonly referred to as quartiles. The 100-quantiles are more commonly referred to as percentiles; they divide the data distribution into 100 equal-sized consecutive sets.

The median, quartiles, and percentiles are the most widely used forms of quantiles.

Interquartile range (IQR):

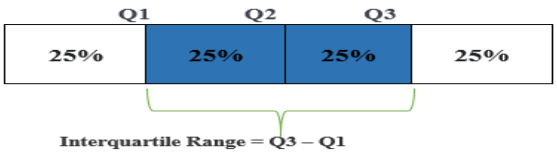
The quartiles give an indication of a distribution's center, spread, and shape.

The first quartile, denoted by Q1, is the 25th percentile. It cuts off the lowest 25% of the data.

The third quartile, denoted by Q3, is the 75th percentile—it cuts off the lowest 75% (or highest 25%) of the data.

The second quartile is the 50th percentile. As the median, it gives the center of the data 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



Example: Interquartile range. The quartiles are the three values that split the sorted data set into four equal parts.

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110.The data contain 12 observations.

Thus, the quartiles for this data are the third, sixth, and ninth values, respectively, in the sorted list.

Therefore, Q1 = $47,000 and Q3 is $63,000.

Thus, the interquartile range is IQR = 63 − 47 = $16,000.

(Note that the sixth value is a median, $52,000, although this data set has two medians since the number of data values is even.).

Variance and Standard Deviation:

Variance is one of the important measures of dispersion, Variance measure the variability of the data around its mean or average. In other words, variance indicates how the data is deviated or dispersed from its mean or

average. High variance means there is more variability or we can say that the data deviates more from its mean whereas low variance means there is less variability. If the variance is zero that means all the values in the data are identical. Variance can never be negative. It is denoted by (sigma square).

Formula for population variance:

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}$$

where **N** is the population size and the **X** are data points and **μ** is the population mean.

Formula for sample variance:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

where **n** is the sample size and **X** are the data points and x̄ (*X-bar*) is the sample mean.

The term depicted by this symbol: σ^2.

Variance and standard deviation are measures of data dispersion. They indicate how spread out a data distribution is. A low standard deviation means that the data observations tend to be very close to the mean, while a high standard deviation indicates that the data are spread out over a large range of values

The variance of N observations, x1,x2,...,xN , for a numeric attribute X is

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 = \left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right) - \bar{x}^2,$$

Let's understand variance with an example

Suppose I am traveling from Indore to Bhopal by car, my car speed data is 0,30,60,50,80,100 the average speed of the car is 53.33. Now we calculate the variance of car speed data, we get the variance 1055.55(by population formula). As we see variance is too far from its average which indicates our variance is too high which means my car speed is fluctuating a lot. So as a conclusion we say that the driver driving a car roughly that means he is not a good driver because the car speed data varying a lot.

Standard Deviation

Standard deviation is an important measure of dispersion and frequently used in statistics. Standard deviation is simply the square root of variance. It indicates how far away the dispersion of the dataset from its mean. It is denoted by (sigma). Simply standard deviation helps us to find the spread of the data about its mean or average. A low Standard deviation indicates that the data are less spread from their average where a high standard deviation indicates the data are more spread out from its average.

The square root of the variance is the standard deviation (SD or σ), which helps determine the consistency of an investment's returns over a period of time.

The formula of standard deviation for population:

$$\sigma = \sqrt{\frac{\Sigma(X-\mu)^2}{n}}$$

where **n** is the sample size and **X** are the data points and μ is the sample mean.

The formula of standard deviation for sample:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

where **n** is the sample size and **X** are the data points and x̄ (*X-bar*) is the sample mean.

where $\bar{x}$ is the mean value of the observation. The standard deviation, σ, of the observations is the square root of the variance, σ^2.

Example :Variance and standard deviation.

Suppose we have the following values for salary (in thousands of dollars), shown in increasing order: 30, 36, 47, 50, 52, 52, 56, 60, 63, 70, 70, 110. we found $\bar{x}$ = $58,000 using Eq. (2.1) for the mean.

To determine the variance and standard deviation of the data from that example, we set N = 12 and use above Eq. to obtain

$$\sigma^2 = \frac{1}{12}(30^2 + 36^2 + 47^2 \ldots + 110^2) - 58^2$$
$$\approx 379.17$$
$$\sigma \approx \sqrt{379.17} \approx 19.47.$$

The basic properties of the standard deviation, σ, as a measure of spread are as follows:
σ measures spread about the mean and should be considered only when the mean is chosen as the measure of center.
σ = 0 only when there is no spread, that is, when all observations have the same value. Otherwise, σ > 0

Five-Number Summary, Boxplots, and Outliers:

A common rule of thumb for identifying suspected outliers is to single out values falling at least 1.5 × IQR above the third quartile or below the first quartile. Because Q1, the median, and Q3 together 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 lowest and highest data values as well. This is known as the five-number summary.

Five number summary is a part of descriptive statistics and consists of five values and all these values will help us to describe the data.

- The minimum value (the lowest value)
- 25th Percentile or Q1
- 50th Percentile or Q2 or Median
- 75th Percentile or Q3
- Maximum Value (the highest value)

Example: How to calculate Five Number Summary

Let's understand this with the help of an example. Suppose we have some data such as: 11,23,32,26,16,19,30,14,16,10

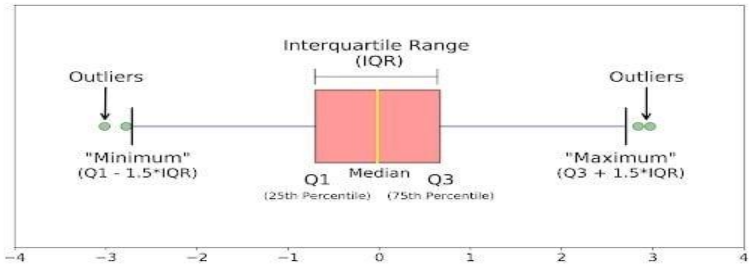
Here, in the above set of data points our Five Number Summary are as follows:

- First of all , we will arrange the data points in ascending order and then calculate the summary : 10,11,14,16,16,19,23,26,30,32
- Minimum value: 10
- 25th Percentile: 14
  Calculation of 25th Percentile: (25/100)*(n+1) = (25/100)*(11) = 2.75 i.e 3rd value of the data
- 50th Percentile : 17.5
  Calculation of 50th Percentile : (16+19)/2 = 17.5
- 75th Percentile : 26
  Calculation of 75th Percentile : (75/100)*(n+1) = (75/100)*(11) = 8.25 i.e 8th value of the data
- Maximum value: 32

Boxplots:

A boxplot is a standardized way of displaying the distribution of data based on its five-number summary ("minimum", first quartile [Q1], median, third quartile [Q3] and "maximum"). Boxplots can tell you about your outliers and their values, if your data is symmetrical, how tightly your data is grouped and if and how your data is skewed..

Boxplots are a popular way of visualizing a distribution. A boxplot incorporates the five-number summary as follows:



A boxplot is a graph that gives a visual indication of how a data set's mean, median, mode, minimum, maximum and outlier values are spread out and compare to each other.

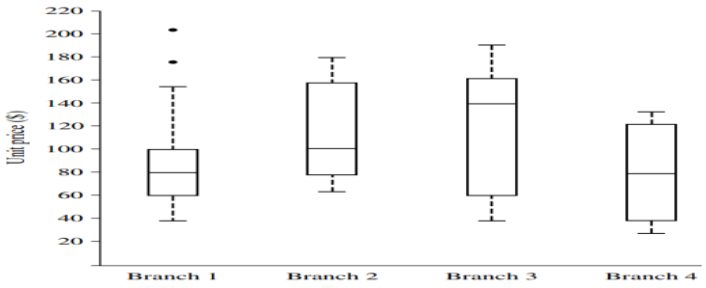
Minimum: The minimum value in the given dataset

- First Quartile (Q1): The first quartile is the median of the lower half of the data set.
- Median: The median is the middle value of the dataset, which divides the given dataset into two equal parts. The median is considered as the second quartile.
- Third Quartile (Q3): The third quartile is the median of the upper half of the data.
- Maximum: The maximum value in the given dataset.
- Apart from these five terms, the other terms used in the box plot are:
- Interquartile Range (IQR): The difference between the third quartile and first quartile is known as the interquartile range. (i.e.) IQR = Q3-Q1
- Outlier: The data that falls on the far left or right side of the ordered data is tested to be the outliers. Generally, the outliers fall more than the specified distance from the first and third quartile.
- (i.e.) Outliers are greater than Q3+(1.5 . IQR) or less than Q1-(1.5 . IQR).

As for whiskers of the boxplot, the left whisker shows the minimum data value and its variability in comparison to the IQR. The right whisker shows the maximum data value and its variability in comparison to the IQR. Whiskers also help present outlier values in comparison to the rest of the data, as outliers sit on the outside of whisker lines.

Example: Boxplot. Figure 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, Q1 is $60, and Q3 is $100.

Notice that two outlying observations for this branch were plotted individually, as their values of 175 and 202 are more than 1.5 times the IQR here of 40



Graphic Displays of Basic Statistical Descriptions of Data:

we study graphic displays of basic statistical descriptions.

These include

- Quantile Plots,
- Quantile–Quantile Plots,
- Histograms, And Scatter Plots.

Such graphs are helpful for the visual inspection of data, which is useful for data preprocessing. The first three of these show univariate distributions (i.e., data for one attribute), while scatter plots show bivariate distributions (i.e., involving two attributes).

Quantile Plot:

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 for the given attribute (allowing the user to assess both the overall behavior and unusual occurrences).

Note that the 0.25 percentile corresponds to quartile Q1, the 0.50 percentile is the median, and the 0.75 percentile is Q3. Let

$$f_i = \frac{i - 0.5}{N}.$$

On a quantile plot, xi is graphed against fi . 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 Q1, median, Q3, and other fi values at a glance.

Example: Quantile plot. Figure 2.4 shows a quantile plot for the unit price data of Table 2.1.

Table 2.1 A Set of Unit Price Data for Items Sold at a Branch of *AllElectronics*

Unit price ($)	*Count of items sold*
40	275
43	300
47	250
—	—
74	360
75	515
78	540
—	—
115	320
117	270
120	350

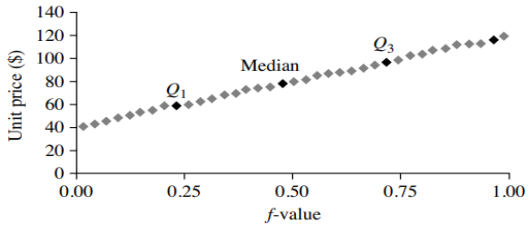


Figure 2.4 A quantile plot for the unit price data of Table 2.1.

Quantile–Quantile Plot:

Q-Q(quantile-quantile) plots play a very vital role to graphically analyze and compare two probability distributions by plotting their quantiles against each other. If the two distributions which we are comparing are exactly equal then the points on the Q-Q plot will perfectly lie on a straight line y = x.

Suppose that we have two sets of observations for the attribute or variable unit price, taken from two different branch locations. Let x1,...,xN be the data from the first branch, and y1,..., yM 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 yi against xi , where yi and xi are both (i − 0.5)/N 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, yi 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.

Example: Quantile–quantile plot. Figure 2.5 shows a quantile–quantile plot for unit price data of items sold at two branches of AllElectronics during a given time period. Each point corresponds to the same quantile for each data set and shows the unit price of items sold at branch 1 versus branch 2 for that quantile.

We see, for example, that at Q1, the unit price of items sold at branch 1 was slightly less than that at branch 2. In other words, 25% of items sold at branch 1 were less than or equal to $60, while 25% of items sold at branch 2 were less than or equal to $64.

At the 50th percentile (marked by the median, which is also Q2), we see that 50% of items sold at branch 1 were less than $78, while 50% of items at branch 2 were less than $85.

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.

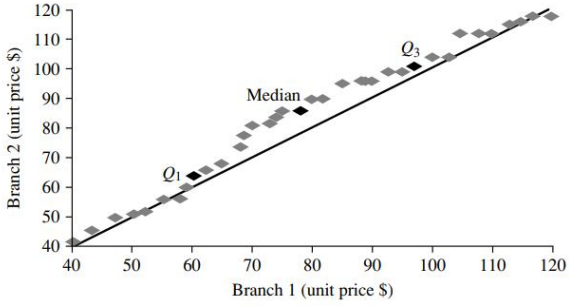


Figure 2.5 A q-q plot for unit price data from two AllElectronics branches.

Histograms:

Histograms (or frequency histograms) are at least a century old and are widely used. “Histos” means pole or mast, and “gram” means chart, so a histogram is a chart of poles. Plotting histograms is a graphical method for summarizing the distribution of a given attribute, X.

If X is nominal, such as automobile model or item type, then a pole or vertical bar is drawn for each known value of X. The height of the bar indicates the frequency (i.e., count) of that X value. The resulting graph is more commonly known as a bar chart.

If X is numeric, the term histogram is preferred. The range of values for X is partitioned into disjoint consecutive subranges.

The subranges, referred to as buckets or bins, are disjoint subsets of the data distribution for X. The range of a bucket is known as the width. Typically, the buckets are of equal width.

For example, a price attribute with a value range of $1 to $200 (rounded up to the nearest dollar) can be partitioned into subranges 1 to 20, 21 to 40, 41 to 60, and so on. For each subrange, a bar is drawn with a height that represents the total count of items observed within the subrange.

Although histograms are widely used, they may not be as effective as the quantile plot, q-q plot, and boxplot methods in comparing groups of univariate observations.

Example:

Create a bar chart representing the preference for sports of a group of 23 people.

Football – 12

Baseball – 10

Basketball – 8

Hockey – 3

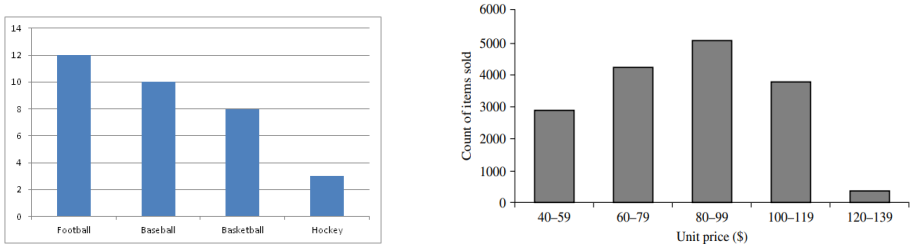


Figure 2.6 A histogram for the Table 2.1 data set.

Scatter Plots and Data Correlation:

The most useful graph for displaying the relationship between two quantitative variables is a scatterplot.

A scatterplot shows the relationship between two quantitative variables measured for the same individuals. The values of one variable appear on the horizontal axis, and the values of the other variable appear on the vertical axis. Each individual in the data appears as a point on the graph.

The scatter plot is a useful method for providing a first look at bivariate data to see clusters of points and outliers, or to explore the possibility of correlation relationships. Two attributes, X, and Y, are correlated if one attribute implies the other. Correlations can be positive, negative, or null (uncorrelated).

Figure shows a scatter plot for the set of data in Table 2.1.

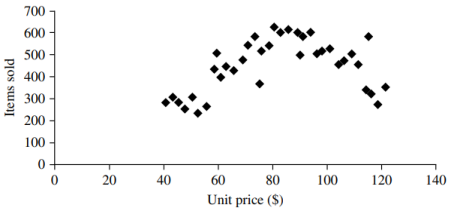


Figure shows a scatter plot for the set of data in Table 2.1.

If the plotted points pattern slopes. from lower left to upper right, this means that the values of X increase as the values of Y increase, suggesting a positive correlation (Figure 2.8a). If the pattern of plotted points slopes from upper left to lower right, the values of X increase as the values of Y decrease, suggesting a negative correlation (Figure 2.8b).

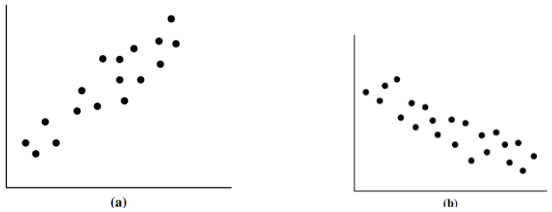


Figure 2.8 Scatter plots can be used to find (a) positive or (b) negative correlations between attributes

A line of best fit can be drawn to study the correlation between the variables. Figure 2.9 shows three cases for which there is no correlation relationship between the two attributes in each of the given data sets.

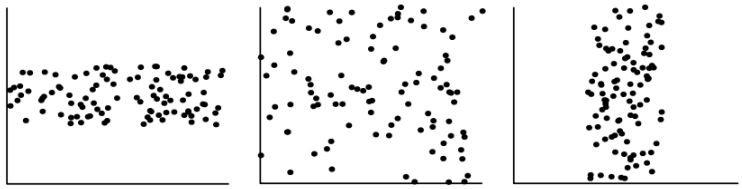


Figure 2.9 Three cases where there is no observed correlation between the two plotted attributes in each of the data sets.

Measuring Data Similarity and Dissimilarity:

In specific data-mining applications such as clustering, it is essential to find how similar or dissimilar objects are to each other.

Proximity measures:

Data mining is the process of finding interesting patterns in large quantities of data. While implementing clustering algorithms, it is important to be able to quantify the proximity of objects to one another. Proximity measures are mainly mathematical techniques that calculate the similarity/dissimilarity of data points. Usually, proximity is measured in terms of similarity or dissimilarity i.e., how alike objects are to one another.

Real-Life Example Use-case : Predicting COVID-19 patients on the basis of their symptoms

With the rise of COVID-19 cases, many people are not being able to seek proper medical advice due to the shortage of both human and infrastructure resources. As a result, we as engineers can contribute our bit to solve this problem by providing a basic diagnosis to help in identifying the people suffering from COVID-19. To help us we can make use of Machine Learning algorithms to ease out this task, among which clustering algorithms come in handy to use.

For this, we make two clusters based on the symptoms of the patients who are COVID-19 positive or negative and then predict whether a new incoming patient is suffering from COVID-19 or not by measuring the similarity/dissimilarity of the observed symptoms (features) with that of the infected person's symptoms.

Data Matrix versus Dissimilarity Matrix:

The objects are x1 = (x11,x12,...,x1p), x2 = (x21,x22,...,x2p), and so on, where xij is the value for object xi of the jth attribute. For brevity, we hereafter refer to object xi as object i. The objects may be tuples in a relational database, and are also referred to as data samples or feature vectors.

Main memory-based clustering and nearest-neighbor algorithms typically operate on either of the following **Two data structures**:

Data matrix (or object-by-attribute structure): This structure stores the n data objects in the form of a relational table, or n-by-p matrix (n objects ×p attributes)

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

Each row corresponds to an object. As part of our notation, we may use f to index through the p attributes.

Dissimilarity matrix (or object-by-object structure): This structure stores a collection of proximities that are available for all pairs of n objects.

It is often represented by an n-by-n table:

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

where d(i, j) is the measured dissimilarity or "difference" between objects i and j.

Measures of similarity can often be expressed as a function of measures of dissimilarity.

For example, for nominal data

$$\mathbf{sim(i, j) = 1 - d(i, j)}$$

where sim(i, j) is the similarity between objects i and j. Throughout the rest of this chapter, we will also comment on measures of similarity.

A data matrix is made up of two entities or "things," namely rows (for objects) and columns (for attributes). Therefore, the data matrix is often called a two-mode matrix. The dissimilarity matrix contains one kind of entity (dissimilarities) and so is called a one-mode matrix. Many clustering and nearest-neighbor algorithms operate on a dissimilarity matrix. Data in the form of a data matrix can be transformed into a dissimilarity matrix before applying such algorithms

Proximity Measures for Nominal Attributes:

A nominal attribute can take on two or more states For example, map color is a nominal attribute that may have, say, five states: red, yellow, green, pink, and blue.

Let the number of states of a nominal attribute be M. The states can be denoted by letters, symbols, or a set of integers, such as 1, 2,..., M. Notice that such integers are used just for data handling and do not represent any specific ordering.

"How is dissimilarity computed between objects described by nominal attributes?"

The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

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

where m is the number of matches (i.e., the number of attributes for which i and j are in the same state), and p is the total number of attributes describing the objects.

Weights can be assigned to increase the effect of m or to assign greater weight to the matches in attributes having a larger number of states

Example 2.17 Dissimilarity between nominal attributes.

Suppose that we have the sample data of Table 2.2, except that only the object-identifier and the attribute test-1 are available, where test-1 is nominal. (We will use test-2 and test-3 in later examples.)

Object Identifier	*test-1 (nominal)*	*test-2 (ordinal)*	*test-3 (numeric)*
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

Table 2.2 A Sample Data Table Containing Attributes of Mixed Type

Let's compute the dissimilarity matrix

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ d(4,1) & d(4,2) & d(4,3) & 0 \end{bmatrix}.$$

Since here we have one nominal attribute, test-1, we set p = 1 in Eq. (2.11) so that d(i, j) evaluates to 0 if objects i and j match, and 1 if the objects differ. Thus, we get

$$\begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

From this, we see that all objects are dissimilar except objects 1 and 4 (i.e., d(4,1) = 0)

Alternatively, similarity can be computed as

$$sim(i, j) = 1 - d(i, j) = \frac{m}{p}.$$

Proximity Measures for Binary Attributes:

dissimilarity and similarity measures for objects described by either symmetric or asymmetric binary attributes.

Recall that a binary attribute has only one of two states: 0 and 1, where 0 means that the attribute is absent, and 1 means that it is present .Given the attribute smoker describing a patient, for instance, 1 indicates that the patient smokes, while 0 indicates that the patient does not.

"So, how can we compute the dissimilarity between two binary attributes?"

One approach involves computing a dissimilarity matrix from the given binary data.

- If all binary attributes are thought of as having the same weight, we have the 2 × 2 contingency table of Table 2.3, where q is the number of attributes that equal 1 for both objects i and j,
- r is the number of attributes that equal 1 for object i but equal 0
- for object j, s is the number of attributes that equal 0
- for object i but equal 1 for object j, and t is the number of attributes that equal 0 for both objects i and j.

The total number of attributes is p, where p = q + r + s + t.

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

Table 2.3 Contingency Table for Binary Attributes

For symmetric binary attributes, each state is equally valuable. Dissimilarity that is based on symmetric binary attributes is called symmetric binary dissimilarity. If objects i and j are described by symmetric binary attributes, **then the dissimilarity between i and j is**

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

For asymmetric binary attributes, the two states are not equally important, such as the positive (1) and negative (0) outcomes of a disease test. Given two asymmetric binary attributes, the agreement of two 1s (a positive match) is then considered more significant than that of two 0s (a negative match). Therefore, such binary attributes are often considered "monary" (having one state).

The dissimilarity based on these attributes is called asymmetric binary dissimilarity, where the number of negative matches, t, is considered unimportant and is thus ignored in the following computation:

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

Complementarily, we can measure the difference between two binary attributes based on the notion of similarity instead of dissimilarity.

For example, the asymmetric binary similarity between the objects i and j can be computed as

$$sim(i, j) = \frac{q}{q+r+s} = 1 - d(i, j).$$

The coefficient sim(i, j) is called the **Jaccard coefficient.**

When both symmetric and asymmetric binary attributes occur in the same data set, the mixed attributes approach
can be applied.

Example Dissimilarity between binary attributes. Suppose that a patient record table (Table 2.4) contains the attributes name, gender, fever, cough, test-1, test-2, test-3, and test-4, where name is an object identifier, gender is a symmetric attribute, and the remaining attributes are asymmetric binary.

name	*gender*	*fever*	*cough*	*test-1*	*test-2*	*test-3*	*test-4*
Jack	M	Y	N	P	N	N	N
Jim	M	Y	Y	N	N	N	N
Mary	F	Y	N	P	N	P	N
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2.4 Relational Table where Patients are described by Binary Attributes

For asymmetric attribute values, let the values Y (yes) and P (positive) be set to 1, and the value N (no or negative) be set to 0. Suppose that the distance between objects (patients) is computed based only on the asymmetric attributes. According to Eq. the distance between each pair of the three patients—Jack, Mary, and Jim—is

$$d(Jack, Jim) = \frac{1+1}{1+1+1} = 0.67, \qquad d(Jack, Mary) = \frac{0+1}{2+0+1} = 0.33, \qquad d(Jim, Mary) = \frac{1+2}{1+1+2} = 0.75.$$

These measurements suggest that Jim and Mary are unlikely to have a similar disease because they have the highest dissimilarity value among the three pairs. Of the three patients, Jack and Mary are the most likely to have a similar disease.

Dissimilarity of Numeric Data: Minkowski Distance:

Distance measures that are commonly used for computing the dissimilarity of objects described by numeric attributes. These measures include the Euclidean, Manhattan, and Minkowski distances.

Euclidean distances:

The most popular distance measure is Euclidean distance (i.e., straight line or "as the crow flies"). Let i = (xi1, xi2,..., xip) and j = (xj1, xj2,..., xjp) be two objects described by p numeric attributes.

The Euclidean distance between objects i and j is defined as

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}.$$

Manhattan (or city block) distance: Another well-known measure is the Manhattan (or city block) distance, named so because it is the distance in blocks between any two points in a city (such as 2 blocks down and 3 blocks over for a total of 5 blocks).

It is defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|.$$

Both the Euclidean and the Manhattan distance satisfy the following mathematical properties:

- **Non-negativity**: d(i, j) ≥ 0: Distance is a non-negative number.
- **Identity of indiscernibles**: d(i, i) = 0: The distance of an object to itself is 0
- **Symmetry: d(i, j) = d(j, i)**: Distance is a symmetric function.
- **Triangle inequality**: d(i, j) ≤ d(i, k) + d(k, j): Going directly from object i to object j in space is no more than making a detour over any other object k. A measure that satisfies these conditions is known as metric.

Minkowski distance:

Minkowski distance is a generalization of the Euclidean and Manhattan distances. It is defined as

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

where h is a real number such that h ≥ 1.

It represents the Manhattan distance when h = 1 (i.e., L1 norm) and Euclidean distance when h = 2 (i.e., L2 norm).

Supremum distance:

The supremum distance (also referred to as Lmax, L∞ norm and as the Chebyshev distance) is a generalization of the Minkowski distance for h → ∞.

To compute it, we find the attribute f that gives the maximum difference in values between the two objects.

This difference is the supremum distance, defined more formally as:

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

The L∞ norm is also known as the uniform norm.

Example : Euclidean distance and Manhattan distance. Let x1 = (1, 2) and x2 = (3, 5) represent two objects as shown in Figure 2.23.

The Euclidean distance between the two is √ 2 2 + 3 2 = 3.61.

The Manhattan distance between the two is 2 + 3 = 5.

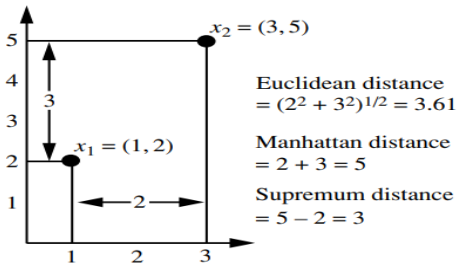


Figure 2.23 Euclidean, Manhattan, and supremum distances between two objects

Example : Supremum distance. Let's use the same two objects, x1 = (1, 2) and x2 = (3, 5), as in Figure 2.23.

The second attribute gives the greatest difference between values for the objects, which is 5 − 2 = 3.

This is the supremum distance between both objects. If each attribute is assigned a weight according to its perceived importance, the **weighted Euclidean distance can be computed as**

$$d(i,j) = \sqrt{w_1|x_{i1}-x_{j1}|^2 + w_2|x_{i2}-x_{j2}|^2 + \cdots + w_m|x_{ip}-x_{jp}|^2}.$$

Proximity Measures for Ordinal Attributes:

The values of an ordinal attribute have a meaningful order or ranking about them, yet the magnitude between successive values is unknown

These categories are organized into ranks. That is, the range of a numeric attribute can be mapped to an ordinal attribute f having Mf states.

For example, the range of the interval-scaled attribute temperature (in Celsius) can be organized into the following states: −30 to −10, −10 to 10, 10 to 30, representing the categories cold temperature, moderate temperature, and warm temperature, respectively.

Let M represent the number of possible states that an ordinal attribute can have. These ordered states define the ranking 1,..., Mf .

"How are ordinal attributes handled?"

Suppose that f is an attribute from a set of ordinal attributes describing n objects.

The dissimilarity computation with respect to f involves the following steps:

- The value of f for the ith object is xif , and f has Mf ordered states, representing the ranking 1,..., Mf . Replace each xif by its corresponding rank, rif ∈ {1,..., Mf }.
- Since each ordinal attribute can have a different number of states, it is often necessary to map the range of each attribute onto [0.0, 1.0] so that each attribute has equal weight. We perform such data normalization by replacing the rank rif of the ith object in the f th attribute by

$$z_{if} = \frac{r_{if}-1}{M_f-1}.$$

- Dissimilarity can then be computed using any of the distance measures ,for numeric attributes, using zif to represent the f value for the ith object.

Example: Dissimilarity between ordinal attributes. Suppose that we have the sample data except that this time only the object-identifier and the continuous ordinal attribute, test-2, are available.
There are three states for test-2: fair, good, and excellent, that is, Mf = 3.
For step 1, if we replace each value for test-2 by its rank, the four objects are assigned the ranks 3, 1, 2, and 3, respectively.
Step 2 normalizes the ranking by mapping rank 1 to 0.0, rank 2 to 0.5, and rank 3 to 1.0. For step 3,

Table 2.2 A Sample Data Table Containing Attributes of Mixed Type

Object Identifier	*test-1 (nominal)*	*test-2 (ordinal)*	*test-3 (numeric)*
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

we can use, say, the Euclidean distance which results in the following **dissimilarity matrix:**

$$\begin{bmatrix} 0 & & & \\ 1.0 & 0 & & \\ 0.5 & 0.5 & 0 & \\ 0 & 1.0 & 0.5 & 0 \end{bmatrix}.$$

Therefore, objects 1 and 2 are the most dissimilar, as are objects 2 and 4 (i.e., d(2,1) = 1.0 and d(4,2) = 1.0). since objects 1 and 4 are both excellent. Object 2 is fair, which is at the opposite end of the range of values for test-2.
Similarity values for ordinal attributes can be interpreted from dissimilarity as sim(i,j) = 1 − d(i,j).
Dissimilarity for Attributes of Mixed Types:"So, how can we compute the dissimilarity between objects of mixed attribute types?" One approach is to group each type of attribute together, performing separate data mining (e.g., clustering) analysis for each type.
A more preferable approach is to process all attribute types together, performing a single analysis. One such technique combines the different attributes into a single dissimilarity matrix, bringing all of the meaningful attributes onto a common scale of the interval [0.0, 1.0].
Suppose that the data set contains p attributes of mixed type. **The dissimilarity d(i, j) between objects i and j is defined as**

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}},$$

where the indicator $\delta^{(f)}_{ij}$ = 0 if either
(1) xif or xjf is missing (i.e., there is no measurement of attribute f for object i or object j), or
(2) xif = xjf = 0 and attribute f is asymmetric binary; otherwise, $\delta^{(f)}_{ij}$ = 1. The contribution of attribute f to the **dissimilarity between i and j (i.e., $d^{(f)}_{ij}$) is computed dependent on its type:**

- If f is numeric: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$, where h runs over all nonmissing objects for attribute f.
- If f is nominal or binary: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$; otherwise, $d_{ij}^{(f)} = 1$.
- If f is ordinal: compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$, and treat z_{if} as numeric.

Cosine Similarity:
A document can be represented by thousands of attributes, each recording the frequency of a particular word (such as a keyword) or phrase in the document. Thus, each document is an object represented by what is called a term-frequency vector.
For example, in following Table, we see that Document1 contains five instances of the word team, while hockey occurs three times. The word coach is absent from the entire document, as indicated by a count value of 0. Such data can be highly asymmetric.

Table:

Document Vector or Term-Frequency Vector

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

Applications using such structures include information retrieval, text document clustering, biological taxonomy, and gene feature mapping

Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words. Let x and y be two vectors for comparison.

Using the cosine measure as a similarity function, we have

$$sim(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}||||\boldsymbol{y}||},$$

Where ||x|| is the Euclidean norm of vector x = (x1, x2,..., xp), defined as $\sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$. Conceptually, it is the length of the vector. Similarly, ||y|| is the Euclidean norm of vector y. The measure computes the cosine of the angle between vectors x and y. A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match. The closer the cosine value to 1, the smaller the angle and the greater the match between vectors.

Example : Cosine similarity between two term-frequency vectors. Suppose that x and y are the first two term-frequency vectors in above Table. That is, x = (5,0,3,0,2,0,0,2,0,0) and y = (3,0,2,0,1,1,0,1,0,1). How similar are x and y? Using Eq. to compute the cosine similarity between the two vectors, we get:

$$\boldsymbol{x}^t \cdot \boldsymbol{y} = 5 \times 3 + 0 \times 0 + 3 \times 2 + 0 \times 0 + 2 \times 1 + 0 \times 1 + 0 \times 0 + 2 \times 1 + 0 \times 0 + 0 \times 1 = 25$$

$$||\boldsymbol{x}|| = \sqrt{5^2 + 0^2 + 3^2 + 0^2 + 2^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = 6.48$$

$$||\boldsymbol{y}|| = \sqrt{3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2} = 4.12$$

sim(x, y) = 0.94

When attributes are binary-valued, the cosine similarity function can be interpreted in terms of shared features or attributes. Suppose an object x possesses the ith attribute if xi = 1. Then x t · y is the number of attributes possessed (i.e., shared) by both x and y, and |x||y| is the geometric mean of the number of attributes possessed by x and the number possessed by y. Thus, sim(x, y) is a measure of relative possession of common attributes.

A simple variation of cosine similarity for the preceding scenario is

$$sim(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\boldsymbol{x} \cdot \boldsymbol{x} + \boldsymbol{y} \cdot \boldsymbol{y} - \boldsymbol{x} \cdot \boldsymbol{y}},$$

which is the ratio of the number of attributes shared by x and y to the number of attributes possessed by x or y. This function, known as the **Tanimoto coefficient or Tanimoto distance**, is frequently used in information retrieval and biology taxonomy.

Data Preprocessing

Data Preprocessing: An overview, Data Cleaning, Data integration, Data Reduction, Data Transformation and Discretization.

Data Preprocessing: An Overview

This section presents an overview of data preprocessing. illustrates the many elements defining data quality. This provides the incentive behind data preprocessing. outlines the major tasks in data preprocessing

Data Quality: Why Preprocess the Data?

Preprocessing of data is mainly to check the data quality.

The quality can be checked by the following:

- **Accuracy**: To check whether the data entered is correct or not.
- **Completeness**: To check whether the data is available or not recorded.
- **Consistency**: To check whether the same data is kept in all the places that do or do not match.
- **Timeliness:** The data should be updated correctly.

- **Believability:** The data should be trustable.
- **Interpretability:** The understandability of the data.

Major Tasks in Data Preprocessing:

Data Cleaning: This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, and duplicates. Various techniques can be used for data cleaning, such as imputation, removal, and transformation.

Data Integration: This involves combining data from multiple sources to create a unified dataset. Data integration can be challenging as it requires handling data with different formats, structures, and semantics. Techniques such as record linkage and data fusion can be used for data integration.

Data Transformation: This involves converting the data into a suitable format for analysis. Common techniques used in data transformation include normalization, standardization, and discretization. Normalization is used to scale the data to a common range, while standardization is used to transform the data to have zero mean and unit variance. Discretization is used to convert continuous data into discrete categories

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.

Major steps involved in data preprocessing, namely, data cleaning, data integration, data reduction, and data transformation.

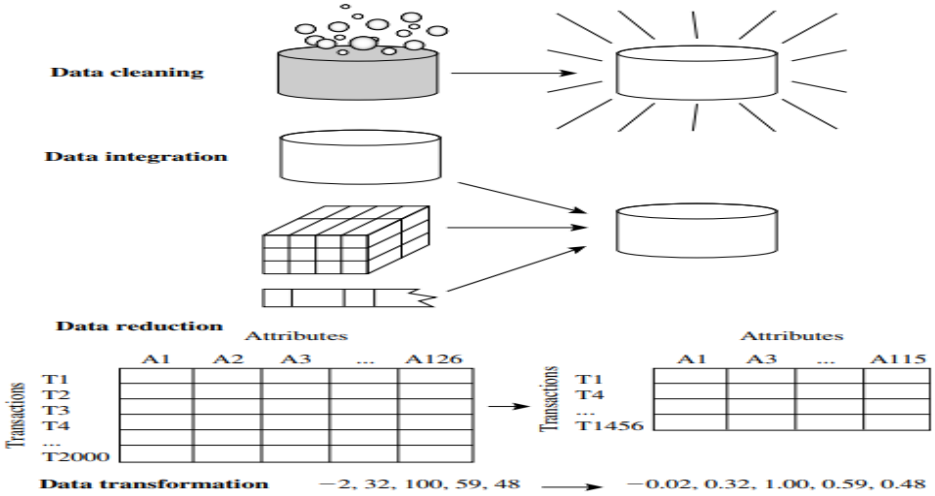


Figure 3.1 Forms of data preprocessing.

Data reduction strategies include dimensionality reduction and numerosity reduction.

- **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).

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

Data Discretization: This involves dividing continuous data into discrete categories or intervals. Discretization is often used in data mining and machine learning algorithms that require categorical data. Discretization can be achieved through techniques such as equal width binning, equal frequency binning, and clustering.

Data Normalization: This involves scaling the data to a common range, such as between 0 and 1 or -1 and 1. Normalization is often used to handle data with different units and scales. Common normalization techniques include min-max normalization, z-score normalization, and decimal scaling.

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.

- Ways of handling missing values.
- Explains data smoothing techniques.
- Approaches to data cleaning as a process.

Missing Values:

Imagine that you need to analyze All Electronics sales and customer data. You note that many tuples have no recorded value for several attributes such as customer income. How can you go about filling in the missing values for this attribute?

Let's look at the following methods:

1. **Ignore the tuple**: This is usually done when the class label is missing (assuming the mining task involves classification). 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. By ignoring the tuple, we do not make use of the remaining attributes' values in the tuple.

2**. Fill in the missing value manually**: In general, this approach is time consuming and may not be feasible given a large data set with many missing values.

3. **Use 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."

4. **Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value**:,

For normal (symmetric) data distributions, the mean can be used, while skewed data distribution should employ the median

For example, suppose that the data distribution regarding the income of All Electronics customers is symmetric and that the mean income is $56,000. Use this value to replace the missing value for income.

5. **Use the attribute mean or median for all samples belonging to the same class as the given tuple**: For example, if classifying customers according to credit risk, we may replace the missing value with the mean income value for customers in the same credit risk category as that of the given tuple. If the data distribution for a given class is skewed, the median value is a better choice.

6. **Use the most probable value to fill in the missing value**: This may be determined with regression, inference-based tools using a 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: "What is noise?" Noise is a random error or variance in a measured variable.

Let's look at the following data smoothing techniques:

- **Binning**
- **Regression**
- **Outlier Analysis**

Binning: Binning methods smooth a sorted data value by consulting its "neighborhood," that is, the 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 example, the data for price are first sorted and then partitioned into equal-frequency bins of size 3 (i.e., each bin contains three values).

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

Partition into (equal-frequency) bins:
Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:
Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:
Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Binning methods for data smoothing

- **In smoothing by bin means**, each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced by the value 9.
- Similarly, **smoothing by bin medians** can be employed, in which each bin value is replaced by the bin median.
- In **smoothing by bin boundaries**, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

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.

Outlier analysis: Outliers may be detected by clustering, for example, where similar values are organized into groups, or "clusters." values that fall outside of the set of clusters may be considered outliers.

Data Cleaning as a Process:

Missing values, noise, and inconsistencies contribute to inaccurate data. 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. Discrepancies may also arise from inconsistent data representations and inconsistent use of codes.

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

A unique rule says that each value of the given attribute must be different from all other values for that attribute.

A consecutive rule says that there can be no missing values between the lowest and highest values for the attribute, and that all values must also be unique (e.g., as in check numbers).

A null rule specifies the use of blanks, question marks, special characters, or other strings that may indicate the null condition.

There are a number of different commercial tools that can aid in the discrepancy detection step.

Data scrubbing tools use simple domain knowledge. Data auditing tools find discrepancies by analyzing the data to discover rules and relationships, and detecting data that violate such conditions.

Data Integration:

Data Integration is one of the major tasks of data preprocessing. Integrating multiple databases or data files into a single store of identical data is known as 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.

There are a number of issues to consider during data integration:

- Entity Identification Problem
- Redundancy and Correlation Analysis:
    1. χ 2 Correlation Test for Nominal Data
    2. Correlation Coefficient for Numeric Data
    3. Covariance of Numeric Data
- Tuple Duplication
- Data Value Conflict Detection and Resolution

Entity Identification Problem:

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.

Equivalent real-world entities from multiple data sources matched up are referred to this problem. Entity Identification Problem occurs during the data integration. During the integration of data from multiple resources, some data resources match each other and they will become redundant if they are integrated

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?

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.

Redundancies can be detected by:

1. χ 2 Correlation Test for Nominal Data
2. Correlation Coefficient for Numeric Data
3. Covariance of Numeric Data

1. χ 2 Correlation Test for Nominal Data:

For nominal data, a correlation relationship between two attributes, A and B, can be discovered by a χ 2 (chi-square) test.

Suppose A has c distinct values, namely a1,a2,...ac .

B has r distinct values, namely b1,b2,...br .

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. Let (Ai ,Bj) denote the joint event that attribute A takes on value ai and attribute B takes on value bj , that is, where (A = ai ,B = bj).

Each and every possible (Ai ,Bj) joint event has its own cell (or slot) in the table.

The χ 2 value (also known as the Pearson χ 2 statistic) is computed as

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

where oij is the observed frequency (i.e., actual count) of the joint event (Ai ,Bj) and eij is the expected frequency of (Ai ,Bj), which can be computed as

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

where n is the number of data tuples, count(A = ai) is the number of tuples having value ai for A, and count(B = bj) is the number of tuples having value bj for B.

The χ 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) × (c − 1) degrees of freedom. If the hypothesis can be rejected, then we say that A and B are statistically correlated.

Example: Correlation analysis of nominal attributes using χ 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 shown in

Table 3.1, where the numbers in parentheses are the expected frequencies. The expected frequencies are calculated based on the data distribution for both attributes using Eq. .

Table 3.1 Example 2.1's 2 × 2 Contingency Table Data

	male	*female*	*Total*
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

Note: Are *gender* and *preferred_reading* correlated?

we can verify the expected frequencies for each cell. For example, the expected frequency for the cell (male, fiction) is 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.

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

Using Eq. for χ 2 computation, we get

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

For this 2 × 2 table, the degrees of freedom are (2 − 1)(2 − 1) = 1.

For 1 degree of freedom, the χ 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 χ 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.

2.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, named after its inventer, Karl Pearson). 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, ai and bi 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, σA and σB are the respective standard deviations of A and and $\Sigma(a_i b_i)$is the sum of the AB cross-product (i.e., for each tuple, the value for A is multiplied by the value for B in that tuple).

Note that −1 ≤ rA,B ≤ +1.

- **If r A,B is greater than 0,** then A and B are positively correlated, meaning that the values of A increase as the values of B increase.
  The higher the value, the stronger the correlation (i.e., the more each attribute implies the other). Hence, a higher value may indicate that A (or B) may be removed as a redundancy.
- **If the resulting value is equal to 0**, then A and B are independent and there is no correlation between them.
- **If the resulting value is less than 0**, then A and B are negatively correlated, where the values of one attribute increase as the values of the other attribute decrease.
  This means that each attribute discourages the other. Scatter plots can also be used to view correlations between attributes.

3 .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 {(a1,b1),...,(an,bn)}.

The mean values of A and B, respectively, are also known as the expected values on A and B, that is,

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n} \quad \textbf{and} \quad E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}.$$

The covariance between A and B is defined as

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

If we compare correlation coefficient and covariance, we see that

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

where σA and σB are the standard deviations of A and B, respectively. It can also be shown that

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}.$$

For two attributes A and B that tend to change together,

- if A is larger than $\bar{A}$ (the expected value of A), then B is likely to be larger than $\bar{B}$ (the expected value of B). Therefore, the covariance between A and B is positive.
- if one of the attributes tends to be above its expected value when the other attribute is below its expected value, then the covariance of A and B is negative.
- If A and B are independent (i.e., they do not have correlation)

Example:

Covariance analysis of numeric attributes. Consider Table 3.2, which presents a simplified example of stock prices observed at five time points for All Electronics and High Tech, a high-tech company. If the stocks are affected by the same industry trends, will their prices rise or fall together?

Table 3.2

Stock Prices for *AllElectronics* and *HighTech*

Time point	AllElectronics	HighTech
t1	6	20
t2	5	10
t3	4	14
t4	3	5
t5	2	5

$$E(AllElectronics) = \frac{6+5+4+3+2}{5} = \frac{20}{5} = \$4 \quad \textbf{and} \quad E(HighTech) = \frac{20+10+14+5+5}{5} = \frac{54}{5} = \$10.80.$$

Thus, using Eq. we compute

$$Cov(AllElectroncis, HighTech) = \frac{6\times 20 + 5\times 10 + 4\times 14 + 3\times 5 + 2\times 5}{5} - 4\times 10.80$$

$$= 50.2 - 43.2 = 7.$$

Therefore, given the positive covariance we can say that stock prices for both companies rise together

Tuple Duplication: Duplicate tuples also increase the size of the database and make it to be complex. Duplicate tuples and attributes cause inconsistencies in attributes and inconsistencies in the database or data sets. Duplicate tuples generally occur due to inaccurate data entry or updating the files of similar data occurrences.

In some situations Duplicate tuples cause the serious issue, for example, A purchase order database contains attributes such as purchaser's name and addresses if another purchaser has the same name and due to technical issues if these two purchasers have the same addresses then it becomes difficult to find the particular customer who has ordered the product. We can handle the duplicate tuples by removing them from the data set during the data cleaning process in data mining. Removing the duplicate tuples is the only way to handle the redundancy caused due to it.

Data Value Conflict Detection and Resolution:

Data conflict means the data merged from the different sources do not match. Like the attribute values may differ in different data sets. The difference maybe because they are represented differently in the different data sets. For suppose the price of a hotel room may be represented in different currencies in different cities. This kind of issues is detected and resolved during data integration.

Data Reduction:

Imagine that you have selected data from the All Electronics data warehouse for analysis. The data set will likely be huge! Complex data analysis and mining on huge amounts of data can take a long time, making such analysis impractical or infeasible.

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

Overview of Data Reduction Strategies

Individual Techniques:

- Wavelet Transforms
- Wavelet Transforms
- Attribute Subset Selection
- Regression and Log-Linear Models: Parametric Data Reduction
- Histograms
  Clustering
- Sampling
- Data Cube Aggregation

Data reduction strategies include

- Dimensionality Reduction,
- Numerosity Reduction, And
- Data Compression

Curse of dimensionality

◦ When dimensionality increases, data becomes increasingly sparse
◦ Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
◦ The possible combinations of subspaces will grow exponentially.

Dimensionality reduction

◦ Avoid the curse of dimensionality
◦ Help eliminate irrelevant features and reduce noise
◦ Reduce time and space required in data mining
◦ Allow easier visualization.

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.

Numerosity reduction:

Numerosity reduction techniques replace the original data volume by alternative, smaller forms of data representation.

These techniques may be parametric or non-parametric.

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.

Data compression:

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.

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

Individual Techniques:

Wavelet Transforms:

The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector X, transforms it to a numerically different vector, X, of wavelet coefficients. The two vectors are of the same length. **"How can this technique be useful for data reduction if the wavelet transformed data are of the same length as the original data?"**

The usefulness lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients.

- For example, all wavelet coefficients larger than some user-specified threshold can be retained. All other coefficients are set to 0.
- The resulting data representation is therefore very sparse, so that operations that can take advantage of data sparsity are computationally very fast if performed in wavelet space.
- The technique also works to remove noise without smoothing out the main features of the data, making it effective for datacleaning as well. .

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.

Figure 3.4 shows some wavelet families. Popular wavelet transforms include the Haar2, Daubechies-4, and Daubechies-6.

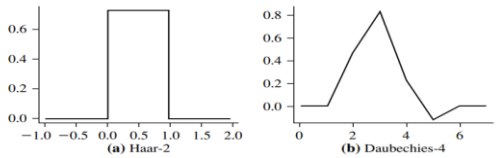


Figure 3.4 Examples of wavelet families

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 ≥ 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 (x2i ,x2i+1). 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 highfrequency 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 Component Analysis:

Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of the dataset. PCA identifies the most important features in the dataset and removes the redundant ones.

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 ≤ n. The original data are thus projected onto a much smaller space, resulting in dimensionality reduction.

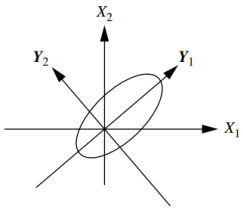
The basic procedure is as follows:

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. That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on.

For example, the following Figure shows the first two principal components, Y1 and Y2, for the given set of data originally mapped to the axes X1 and X2. This information helps identify groups or patterns within the data.



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 may be used as inputs to multiple regression and cluster analysis. In comparison with wavelet transforms, PCA tends to be better at handling sparse data, whereas wavelet transforms are more suitable for data of high dimensionality.

Attribute Subset Selection:

- Attribute subset selection 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.

"How can we find a 'good' subset of the original attributes?"

- For n attributes, there are 2 n possible subsets.
- An exhaustive search for the optimal subset of attributes can be prohibitively expensive, especially as n and the number of data classes increase. Therefore, heuristic methods that explore a reduced search space are commonly used for attribute subset selection.
- These methods are typically greedy in that, while searching through attribute space, they always make what looks to be the best choice at the time.

Basic heuristic methods of attribute subset selection include the techniques that follow, some of which are illustrated in Figure 3.6.

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: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ A_4? Y N A_1? A_6? Y N Y N Class 1 Class 2 Class 1 Class 2 $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$

Figure 3.6 Greedy (heuristic) methods for attribute subset selection

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 flowchartlike structure where each internal (nonleaf) 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.

Regression and Log-Linear Models: Parametric Data Reduction:

Regression and log-linear models can be used to approximate the given data.

- **In (simple) 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 modeled as a linear function of two or more predictor variables.

Log-linear models approximate discrete multidimensional probability distributions. Given a set of tuples in n dimensions (e.g., described by n attributes), we can consider each tuple as a point in an n-dimensional space.

- Log-linear models can be used to estimate the probability of each point in a multidimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations.
- This allows a higher-dimensional data space to be constructed from lower-dimensional spaces. Log-linear models are therefore also useful for dimensionality reduction (since the lower-dimensional points together typically occupy less space than the original data points) and data smoothing (since aggregate estimates in the lower-dimensional space are less subject to sampling variations than the estimates in the higher-dimensional space).

Histograms:

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: Histograms.

The following data are a list of All Electronics 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, 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

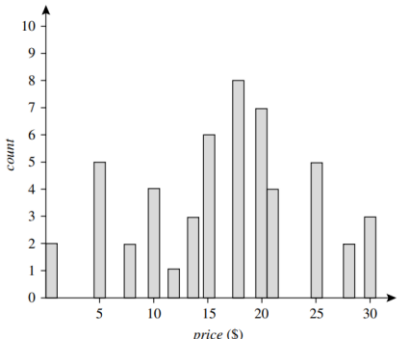


Figure 3.7 A histogram for price using singleton buckets—each bucket represents one price–value/ frequency pair.

"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)

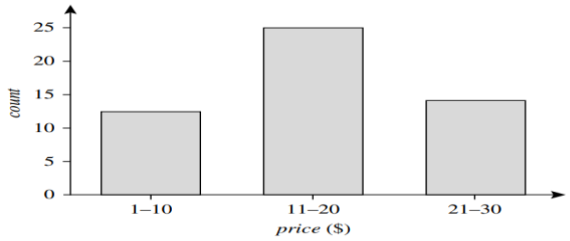


Figure 3.8 An equal-width histogram for price, where values are aggregated so that each bucket has a uniform width of $10.

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.

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).

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.

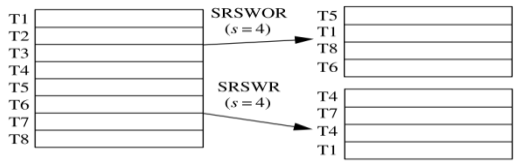
Sampling:

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.

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. **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.



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. Other clustering criteria conveying rich semantics can also be explored. **For example**, in a spatial database, we may choose to define clusters geographically based on how closely different areas are located.

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.

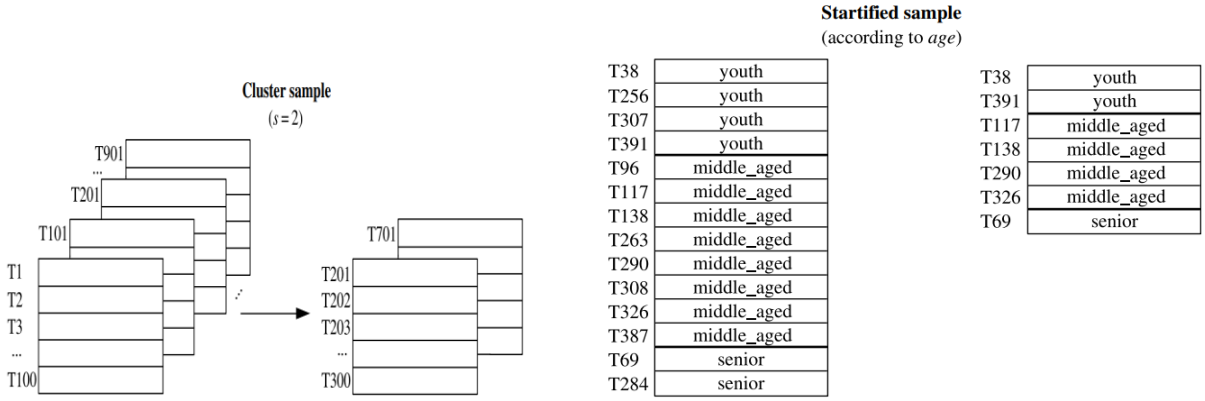


Figure 3.9 Sampling can be used for data reduction.

An advantage of sampling for data reduction is that the cost of obtaining a sample is proportional to the size of the sample, s, as opposed to N, the data set size.

Data Cube Aggregation:

Imagine that you have collected the data for your analysis. These data consist of the All Electronics 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.

Figure 3.10 Sales data for a given branch of All Electronics 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 cubes store multidimensional aggregated information

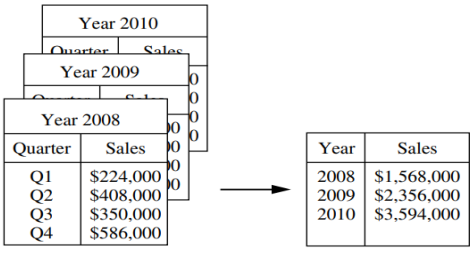


Figure 3.10

For example, Figure 3.11 shows a data cube for multidimensional analysis of sales data with respect to annual sales per item type for each All Electronics branch. Each cell holds an aggregate data value, corresponding to the data point in multidimensional space. (For readability, only some cell values are shown.)

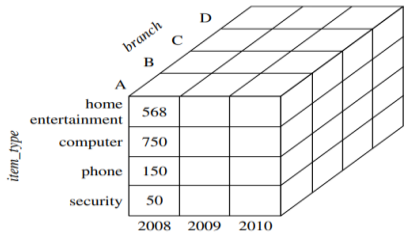


Figure 3.11 A data cube for sales at AllElectronics

Concept hierarchies may exist for each attribute, allowing the analysis of data at multiple abstraction levels.

For example, a hierarchy for branch could allow branches to be grouped into regions, based on their address.

Data cubes provide fast access to pre computed, summarized data, thereby benefiting online analytical processing as well as data mining.

The cube created at the lowest abstraction level is referred to as the **base cuboid**. The base cuboid should correspond to an individual entity of interest such as sales or customer.

Data cubes created for varying levels of abstraction are often referred to as **cuboids**, so that a data cube may instead refer to a lattice of cuboids

A cube at the highest level of abstraction is the **apex cuboid**.

For the sales data in Figure 3.11, the apex cuboid would give one total—the total sales for all three years, for all item types, and for all branches.

Data Transformation and Data Discretization:

Data transformation routines convert the data into appropriate forms for mining.

For example, in normalization, attribute data are scaled so as to fall within a small range such as 0.0 to 1.0.

Other examples are data discretization and concept hierarchy generation.

Data discretization transforms numeric data by mapping values to interval or concept labels. Such methods can be used to automatically generate concept hierarchies for the data, which allows for mining at multiple levels of granularity.

Discretization techniques include binning, histogram analysis, cluster analysis, decision tree analysis, and correlation analysis.

For nominal data, concept hierarchies may be generated based on schema definitions as well as the number of distinct values per attribute.

In this preprocessing step, the data are transformed or consolidated so that the resulting mining process may be more efficient, and the patterns found may be easier to understand.

Data Transformation Strategies Overview

Methods of Data Transformation:

- Discretization by Binning
- Discretization by Histogram Analysis
- Discretization by Cluster, Decision Tree, and Correlation Analyses
- Concept Hierarchy Generation for Nominal Data

In data transformation, the data are transformed or consolidated into forms appropriate for mining.

Strategies for data transformation include the following:

- **Smoothing**, which works to remove noise from the data. Techniques include binning, regression, and clustering.
- **Attribute construction** (or feature construction), where new attributes are constructed and added from the given set of attributes to help the mining process.
- **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.
- **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.

- **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.
- **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.

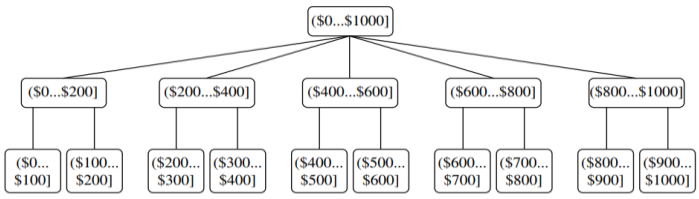


Figure 3.12 A concept hierarchy for the attribute price, where an interval ($X ...$Y] denotes the range from $X (exclusive) to $Y (inclusive).

Data Transformation by Normalization:

Transforming the data to fall within a smaller or common range such as [−1,1] or [0.0, 1.0].

Normalizing the data attempts to give all attributes an equal weight. Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering.

There are many methods for data normalization:

- min-max normalization,
- z-score normalization, and
- normalization by decimal scaling. For our discussion,

let A be a numeric attribute with n observed values, v1, v2,..., vn.

Min-max normalization performs a linear transformation on the original data. Suppose that minA and maxA are the minimum and maximum values of an attribute, A.

Min-max normalization maps a value, vi , of A to vi' in the range [new $\min_A$,new $\max_A$] by computing v

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A.$$

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: Min-max normalization.

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

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

In 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, vi , of A is normalized to vi' by computing

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

Where $\bar{A}$ = 1 n= 1/ n= (v1+v2+v3-------vn)

and σA is computed as the square root of the variance of A. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers that dominate the min-max normalization.

Example: z-score normalization. Suppose that the mean and standard deviation of the values for the attribute income are $54,000 and $16,000, respectively. With z-score normalization, a value of $73,600 for income is transformed to

$$\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225.$$

A variation of this z-score normalization replaces the standard deviation of Eq. by the mean absolute deviation of A. The mean absolute deviation of A, denoted sA, is

$$s_A = \frac{1}{n}(|v_1 - \bar{A}| + |v_2 - \bar{A}| + \cdots + |v_n - \bar{A}|).$$

Thus, z-score normalization using the mean absolute deviation is

$$v_i' = \frac{v_i - \bar{A}}{s_A}.$$

The mean absolute deviation, sA, is more robust to outliers than the standard deviation, σA. When computing the mean absolute deviation, the deviations from the mean (i.e., |xi − ¯x|) are not squared; hence, the effect of outliers is somewhat reduced.

Normalization by decimal scaling normalizes by moving the decimal point of values of attribute A.

The number of decimal points moved depends on the maximum absolute value of A. A value, vi , of A is normalized to v 0 i by computing

$$v_i' = \frac{v_i}{10^j},$$

where j is the smallest integer such that $max(|v_i'|) < 1.$

Example: 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.

Data Discretization:

Discretization techniques can be categorized based on whether it uses class information, as:

– Supervised discretization the discretization process uses class information

– Unsupervised discretization the discretization process does not use class information

Discretization techniques can be categorized based on which direction it proceeds, as:

– Top-down If the process starts by first finding one or a few points (called split points or cut points) to split the entire attribute range, and then repeats this recursively on the resulting intervals Data Discretization and Concept Hierarchy Generation

– Bottom-up starts by considering all of the continuous values as potential split- points, removes some by merging neighborhood values to form intervals, and then recursively applies this process to the resulting intervals.

There are different techniques of discretization:

1. **Discretization by binning:** It is unsupervised method of partitioning the data based on equal partitions , either by equal width or by equal frequency
2. **Discretization by Cluster:** clustering can be applied to discretize numeric attributes. It partitions the values into different clusters or groups by following top down or bottom up strategy
3. **Discretization By decision tree:** it employs top down splitting strategy. It is a supervised technique that uses class information.
4. **Discretization By correlation analysis:** Chi Merge employs a bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively
5. **Discretization by histogram:** Histogram analysis is unsupervised learning because it doesn't use any class information like binning. There are various partition rules used to define histograms.

Discretization by Binning:

The sorted values are distributed into a number of buckets, or bins, and then replacing each bin value by the bin mean or median.

Binning is:
– a top-down splitting technique based on a specified number of bins. Data Discretization and Concept Hierarchy Generation
– an unsupervised discretization technique, because it does not use class information.

Binning methods:
– Equal-width (distance) partitioning
– Equal-depth (frequency) partitioning

Equal-width (distance) partitioning
– Divides the range into N intervals of equal size: uniform grid
– if A and B are the lowest and highest values of the attribute, the width of intervals will be: W = (B –A)/N. – The most straightforward, but outliers may dominate Data Discretization and Concept Hierarchy Generation presentation
– Skewed data is not handled well

Example:
- Sorted data for price (in dollars):
  - 4, 8, 15, 21, 21, 24, 25, 28, 34
- W = (B –A)/N = (34 – 4) / 3 = 10
  - Bin 1: 4-14, Bin2: 15-24, Bin 3: 25-34
- Equal-width (distance) partitioning:
  - Bin 1: 4, 8
  - Bin 2: 15, 21, 21, 24
  - Bin 3: 25, 28, 34

Equal-depth (frequency) partitioning
– Divides the range into N intervals, each containing approximately same number of samples
– Good data scaling
– Managing categorical attributes can be tricky

Example:
- Sorted data for price (in dollars):
  - 4, 8, 15, 21, 21, 24, 25, 28, 34
- Equal-depth (frequency) partitioning:
  - Bin 1: 4, 8, 15
  - Bin 2: 21, 21, 24
  - Bin 3: 25, 28, 34

Discretization by Histogram Analysis:
Histogram analysis is an unsupervised discretization technique because it does not use class information.

- A histogram partitions the values of an attribute, A, into disjoint ranges, called buckets or bins.
- If each bucket represents only a single attribute–value/frequency pair, the buckets are called singleton buckets. Singleton buckets are useful for storing high-frequency outliers.
- Histograms are effective at approximating sparse data, 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.
- **There are two types of histograms**: Equal-width(or distance) and Equal-frequency(or equal-depth).
- In an **equal-width histogram**, the width of each bucket range is uniform. It divides the range into N intervals of equal size. If A and B are the lowest and highest values of the attribute, the width of intervals will be: W = (B –A)/N and the interval boundaries are: A+w, A+2w, …, A+(k-1)w.

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). It divides the range

into N intervals, each containing approximately same number of samples. It's good for data scaling but managing categorical attributes can be tricky.

Discretization by Cluster, Decision Tree, and Correlation Analyses:

Cluster 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 based on similarity, and store cluster representation (e.g., centroid and diameter) only. It partitions the data set into clusters.
- There are many choices of clustering definitions and clustering algorithms. Eg: K-Means and K-Medoid algorithm.
- **Properties of clusters**: (i) All the data points in a cluster should be similar to each other. (ii) The data points from different clusters should be as different as possible.

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

Correlation Analysis:

- It is a supervised discretization method, i.e it uses class information.
- It uses bottom-up merge, i.e it finds the best neighboring intervals (those having similar distributions of classes, i.e., low $\chi 2$ values) to merge.
- It's also known as Chi Merge algorithm.
- It is performed recursively by finding best neighboring intervals that have a similar distribution of classes and merge them, until a predefined stopping condition.
- **Steps include**: (i) 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. (ii) Adjacent intervals with the least Chi-2 values are merged because low Chi-2 values for a pair indicate similar class distributions. (iii) This merging process proceeds recursively until a predefined stopping criterion is met.

Concept Hierarchy Generation for Nominal Data:

We now look at data transformation 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.

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, as follows:

1.Specification of a partial ordering of attributes explicitly at the schema level by users or experts: Concept hierarchies for nominal attributes or dimensions 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:

Example: a relational database or a dimension location of a data warehouse may contain the following group of attributes: street, city, province or state, and country.

A user or expert can easily define a concept hierarchy by Data Discretization and Concept Hierarchy Generation specifying ordering of the attributes at the schema level.

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.

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 Data Discretization and Concept Hierarchy Generation hierarchy.

Example: Suppose a user selects a set of location-oriented attributes, street, country, province_or_state, and city, from the All Electronics database, but does not specify the hierarchical ordering among the attributes.

Example: Concept hierarchy generation based on the number of distinct values per attribute.

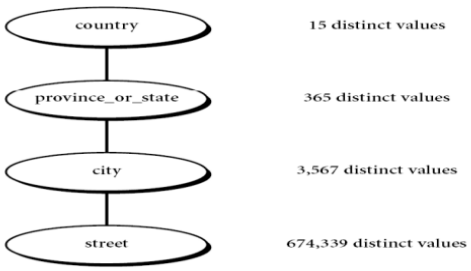
Suppose a user selects a set of location-oriented attributes—street, country, province or state, and city—from the All Electronics database, but does not specify the hierarchical ordering among the attributes.

A concept hierarchy for location can be generated automatically, as illustrated in following figure:

- First, sort the attributes in ascending order based on the number of distinct values in each attribute. This results in the following (where 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.

Concept Hierarchy Generation for Categorical Data

- Automatic generation of a schema concept hierarchy based on the number of distinct attribute values.
- The attribute with the most distinct values is placed at the lowest level of the hierarchy
- Exceptions, e.g., weekday, month, quarter, year



4.Specification of only a partial set of attributes: Sometimes a user can be careless when defining 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.

Example: Concept hierarchy generation using pre specified semantic connections.

Suppose that a data mining expert (serving as an administrator) has pinned together the five attributes number, street, city, province or state, and country, because they are closely linked semantically regarding the notion of location.

UNITWISE QUESTIONS

1. What are the methods for data discretization,
2. With the help of diagram explain major tasks in data pre-processing and Discuss issues to consider during data integration
3. Describe the process of data cleaning
4. What is the curse of dimensionality? how to reduce it?
5. Data processing is necessary before data mining process, justify your answer.
6. Illustrate binning methods for data smoothing.
7. In real word data, tuples with missing values for some attributes are a common occurrence. Describe various methods for handling this problem.
8. Discuss issues to consider during data integration
9. Describe the process of data reduction
10. Explain data discretization and transformation with examples