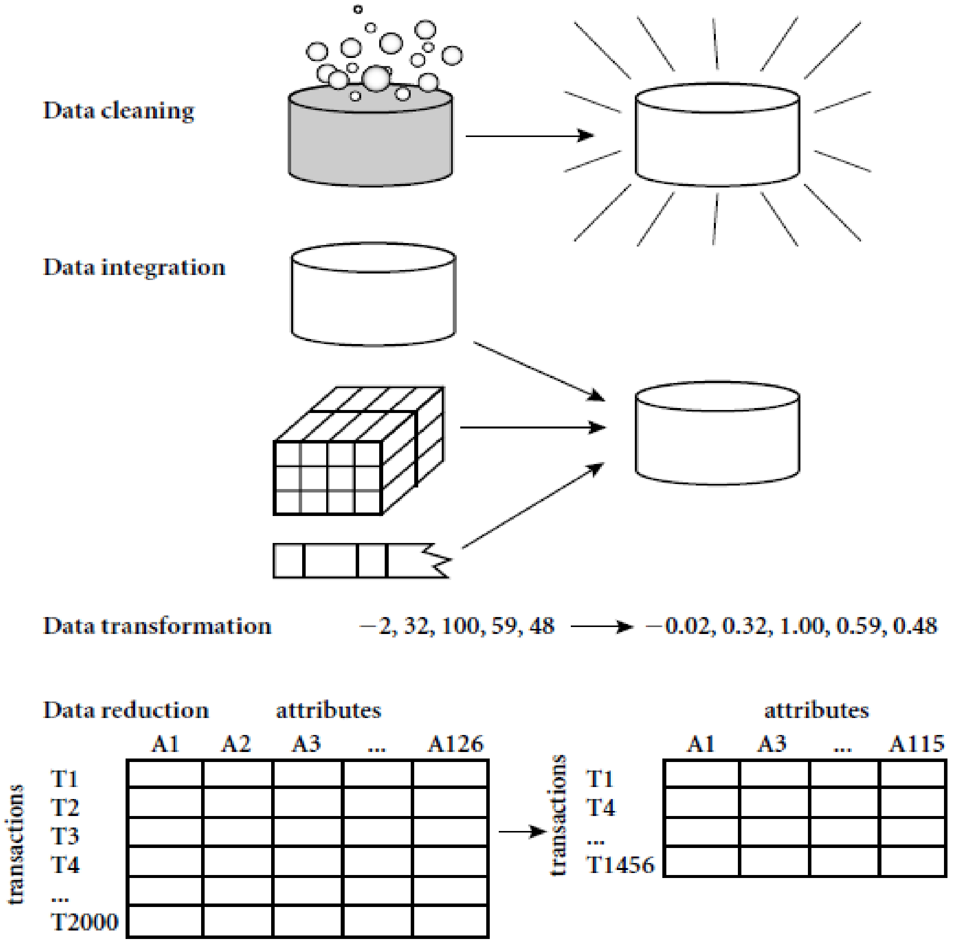
UNIT –II:

Data Pre-processing: Why Pre-process the Data? Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation. (Han & Kamber)

<u>Why Pre-process the Data:-</u> Today's real-world databases are highly susceptible to noise, and consists of missing, and inconsistent data due to their huge size. Data preprocessing is done to improve the quality of the data. Preprocessed data improve the efficiency and ease of the mining process. There are a number of data preprocessing techniques. They are

1. Data cleaning can be applied to remove noise and correct inconsistencies in the data.
2. Data integration merges data from multiple sources into a single data store, such as a data warehouse or a data cube.
3. Data transformations, such as normalization, may be applied. Normalization may improve the accuracy and efficiency of mining algorithms.
4. Data reduction can reduce the data size by aggregating, eliminating redundant features, or clustering.



Forms of data preprocessing.

Mining Descriptive Statistical Measures in Large Databases:- For many data mining tasks, users would like to learn more data characteristics regarding both central tendency and data dispersion. Measures of central tendency include mean, median, mode, and midrange, while measures of data dispersion include quartiles, outliers, variance, and other statistical measures. These descriptive statistics are of great help in understanding the distribution of the data.

1. Measuring the central tendency:- The most common and most effective numerical measure of the “center" of a set of data is the (arithmetic) mean. Let x_1, x_2, …., x_n be a set of n values or observations. The mean of this set of values is

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

This corresponds to the built-in aggregate function, average (avg() in SQL), provided in relational database systems. In most data cubes, sum and count are saved in precomputation. Thus, the derivation of average is straightforward, using the formula

average = sum/count.

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

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}.$$

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

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

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

Based on the categorization of measures, the median is neither a distributive measure nor an algebraic measure Although it is not easy to compute the exact median value in a large database, an approximate median can be computed efficiently. For example, for grouped data, the median, obtained by interpolation, is given by

$$median = L_1 + \left(\frac{n/2 + (\sum f)_l}{f_{median}}\right)c.$$

where L1 is the lower class boundary of (i.e., lowest value for) the class containing the median, n is the number of values in the data, (Pf)l is the sum of the frequencies of all of the classes that are lower than the median class, and fmedian is the frequency of the median class, and c is the size of the median class interval.

Another measure of central tendency is the mode. The mode for a set of data is the value that occurs most frequently in the set. It is possible for the 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. If a data set has more than three modes, it is multimodal. At the other extreme, if each data value occurs only once, then there is no mode. For unimodal frequency curves that are moderately skewed (asymmetrical), we have the following empirical relation

$$\text{mode} = 3\text{median}-2\text{mean}$$

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

Measuring the dispersion of data

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

The kth percentile of a set of data in numerical order is the value x having the property that k percent of the data entries lies at or below x. Values at or below the median M (discussed in the previous subsection) correspond to the 50-th percentile.

→The most commonly used percentiles other than the median are quartiles. The first quartile, denoted by Q1, is the 25-th percentile; and the third quartile, denoted by Q3, is the 75-th percentile.

The quartiles together with the median give some indication of the center, spread, and shape of a distribution.

The distance between the first and third quartiles is a simple measure of spread that gives the range covered by the middle half of the data. This distance is called the interquartile range (IQR), and is denoted as

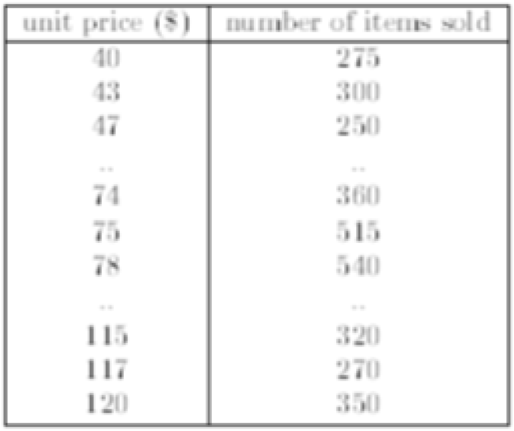
$$IQR = Q3 - Q1$$

We should be aware that no single numerical measure of spread, such as IQR, is very useful for describing skewed distributions. The spreads of two sides of a skewed distribution are unequal. Therefore, it is more informative to also provide the two quartiles Q1 and Q3, along with the median, M.

One 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, M, and Q3 contain no information about the endpoints (e.g., tails) of the data, a fuller summary of the shape of a distribution can be obtained by providing the highest and lowest data values as well. This is known as the **five-number summary**. The five-number summary of a distribution consists of the median M, the quartiles Q1 and Q3, and the smallest and largest individual observations, written in the order Minimum; Q1; M; Q3; Maximum:

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

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



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

A set of data.

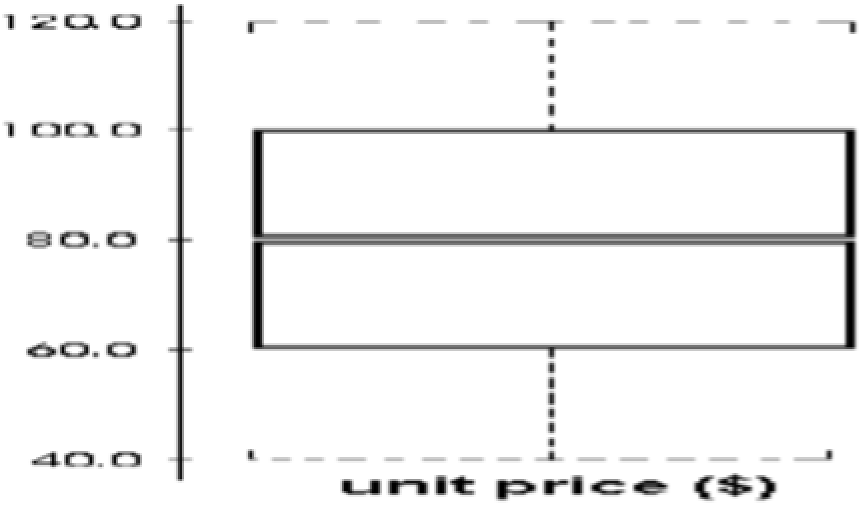


Fig. A boxplot for the above dataset

Graph displays of basic statistical class descriptions

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

A **histogram,** or frequency histogram, is a univariate graphical method. It denotes the frequencies of the classes present in a given set of data. A histogram consists of a set of rectangles where the area of each rectangle is proportional to the relative frequency of the class it represents.

Figure 5.5 shows a histogram for the data set of Table 5.11, where classes are de_ned by equi-width ranges representing $10 increments. Histograms are at least a century old, and are a widely used univariate graphical method. However, they may not be as effective as the quantile plot, Q-Q plot and boxplot methods for comparing groups of univariate observations.

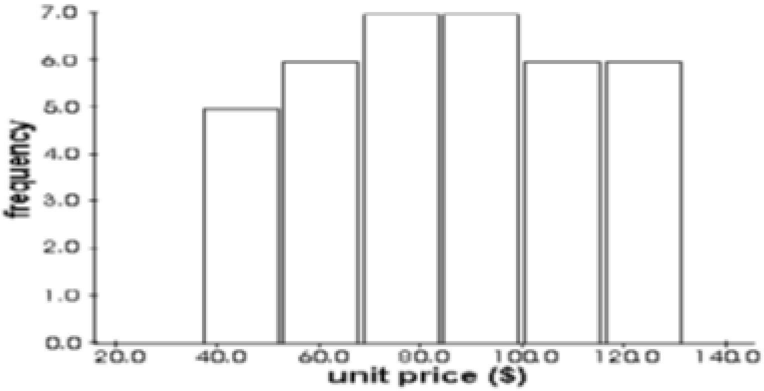
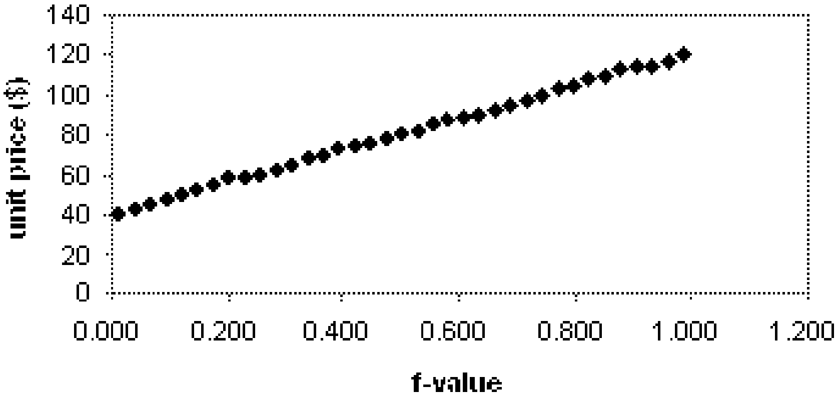


Fig: A histogram for the above dataset

A **quantile plot** is a simple and effective way to have a first look at data distribution. First, it displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences). Second, it plots quantile information. The mechanism used in this step is slightly different from the percentile computation.



A **Q-Q plot,** or quantile-quantile plot, is a powerful visualization method for comparing the distributions of two or more sets of univariate observations. When distributions are compared, the goal is to understand how the distributions differ from one data set to the next.

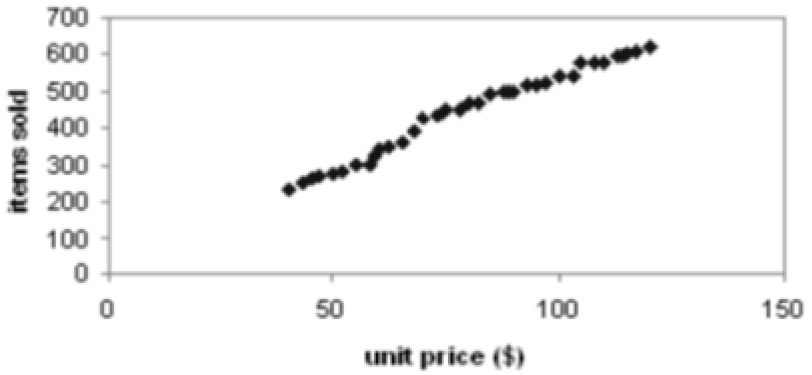


Fig: A Q-Q plot for the above dataset

A **scatter plot** is one of the most effective graphical methods for determining if there appears to be a relation-ship, pattern, or trend between two quantitative variables. To construct a scatter plot, each pair of values is treated as a pair of coordinates in an algebraic sense, and plotted as points in the plane. The scatter plot is a useful exploratory method for providing a first look at bivariate data to see how they are distributed throughout the plane, for example, and to see clusters of points, outliers, and so forth. Figure 5.8 shows a scatter plot for the set of data in Table

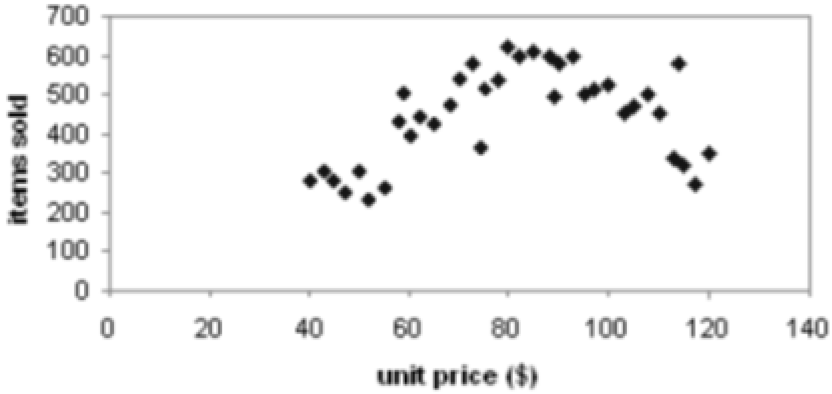


Fig: A Scatter Plot for the above dataset

A **loess curve** is another important exploratory graphic aid which adds a smooth curve to a scatter plot in order to provide better perception of the pattern of dependence. The word loess is short for local regression. Figure 5.9 shows a loess curve for the set of data in above Table.

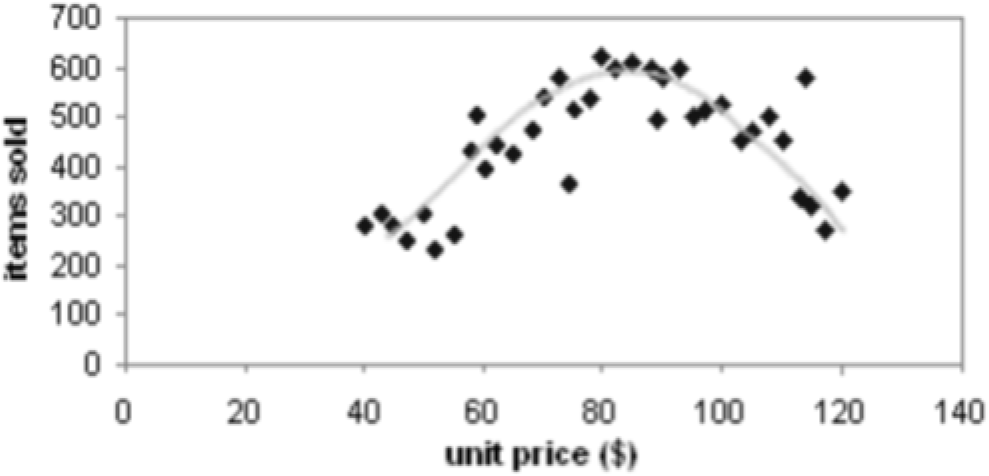


Fig: a loess curve for the above dataset

Data Cleaning:- Real-world data tend to be incomplete, noisy, and inconsistent. Data cleaning routines attempt to fill in missing values, smooth out noisy data, and correct inconsistencies in the data.

Missing values:- Filling the missing values for any attribute can be done by using the following methods.

1. Ignore the tuple:- This is usually done when the class label is missing. This method is not very effective, unless the tuple contains several attributes with missing values.
2. Fill in the missing value manually:-This approach is time-consuming and may not be feasible for 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 - ∞.
4. Use the attribute mean to fill in the missing value:- Replace all missing attribute values by the mean of the attribute.
5. Use the attribute mean for all samples belonging to the same class as the given tuple:- Replace all missing attribute values by the mean of same class.
6. Use the most probable value to fill in the missing value:- Most probable value may be calculated by using regression or decision tree induction.

Noisy data:- A data is said to be noisy if its attribute values are invalid and incorrect. Noise is a random error or variance in a measured variable. “Smooth” out the data to remove noise. Some of the data smoothing techniques that are commonly used are.

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

Commonly used binning methods are

a) *Smoothing by bin means:*- In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.

b) *Smoothing by bin median:-* In smoothing by bin medians, each bin value is replaced by the bin median.

c) *Smoothing by bin means:-* 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.

Example:- Smooth out the following prices 21, 8, 28, 4, 34, 21, 15, 25, 24.

Data for price are first sorted and then partitioned into equidepth bins of depth 3.

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

Partition into (equi-width) 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 median:

Bin 1: 8, 8, 8
Bin 2: 21, 21, 21
Bin 3: 28, 28, 28

Smoothing by bin boundaries:

Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Figure 3.2: Binning methods for data smoothing.

2. Clustering:- Outliers may be detected by clustering. Similar values are organized into groups clusters. Values which fall outside all clusters may be considered outliers.

3. Combined computer and human inspection:- Outliers may be identified through a combination of computer and human inspection. This is much faster than having to manually search through the entire database. The garbage patterns can then be removed from the database.

4. Regression:- Data can be smoothed by using regression. Linear regression involves finding the best line to fit two variables, so that one variable can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two variables are involved to predict the unknown value.

5. Inconsistent data:- Data inconsistencies may be corrected manually using external references. For example, errors made at data entry may be corrected by performing a paper trace. Knowledge engineering tools may also be used to detect the violation of known data constraints.

Data integration:- Data integration combines data from multiple sources into a single data store, such as large database or data warehouse. Major Issues that are to be considered during data integration are

Entity identification problem:- Sometimes customer_id in one database, and cust_number in another refer to the same entity. Data analyst or computer decide whether they both refer to the same entity by examining the metadata of the datawarehose. Metadata is data about the data. Such metadata can be used to avoid errors in schema integration.

Redundancy:- Redundancy is another important issue. An attribute may be redundant if it can be “derived" from another table, such as annual revenue. Some redundancies can be detected by correlation analysis. The correlation between attributes A and B can be measured by

$$r_{A,B} = \frac{\sum (A - A)(B - B)}{(n - 1)\, \sigma_A \, \sigma_B}$$

If the correlation factor(r) is greater than 1, then A and B are positively correlated. The higher the value, the more each attribute implies the other. Hence, a high value may indicate that A (or B) may be removed as a redundancy. If the resulting value is equal to 1, then A and B are independent and there is no correlation between them. If the resulting value is less than 1, then A and B are negatively correlated.

Detection and resolution of data value conflicts:- A third important issue in data integration is the detection and resolution of data value conflicts. For example, for the same real world entity, attribute values from different sources may differ. This may be due to differences in metric units used in the system. The price of different hotels may involve different currencies.

Careful integration of the data from multiple sources can help reduce and avoid redundancies and inconsistencies in the resulting data set. This can help improve the accuracy and speed of the subsequent mining process.

Data transformation:- In data transformation, the data are transformed or consolidated into forms appropriate for mining. Data transformation can involve the following:

1. Normalization, where the attribute data are scaled so as to fall within a small specified range, such as -1.0 to 1.0, or 0 to 1.0.
2. Smoothing, which works to remove the noise from data. Such techniques include binning, clustering, and regression.
3. Aggregation, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.
4. Generalization of the data, where low level or 'primitive' (raw) data are replaced by higher level concepts through the use of concept hierarchies. For

example, categorical attributes, like street, can be generalized to higher level concepts, like city or county. Similarly, values for numeric attributes, like age, may be mapped to higher level concepts, like young, middle-aged, and senior.

5. An attribute is normalized by scaling its values so that they fall within a small specified range, such as 0 to 1.0. There are three main methods for data normalization. They are
    - min-max normalization,
    - z-score normalization, and
    - normalization by decimal scaling.

Min-max normalization performs a linear transformation on the original data. Suppose that min_A and max_A are the minimum and maximum values of an attribute A. Min-max normalization maps a value v of A to v^1

$$v^1 = \frac{v - min_A}{max_A - min_A} (new\_max_A – new\_min_A) + new\_min_A$$

Example:- Suppose that the maximum and minimum values for the attribute income are \$98,000 and \$12,000,respectively. Map the income to the range [0; 1]. By min-max normalization, a value of \$73,600 for income is transformed to

$$v^1 = \frac{73{,}600 - 12{,}000}{98{,}000 – 12{,}000} (1.0 - 0) + 0 \quad = 0.716$$

In z-score normalization (or zero-mean normalization), the values for an attribute A are normalized based on the mean and standard deviation of A. A value v of A is normalized to v^1 by computing

$$v^1 = \frac{v – A}{\sigma_A}$$

where A stands for mean of A and σ_A for standard deviation. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers which dominate the min-max normalization.

Example :- 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

$$v^1 = \frac{73{,}600 – 54{,}000}{16{,}000} = 1.225$$

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 v of A is normalized to v^1 by computing

$$v^1 = \frac{v}{10^j}$$

where j is the smallest integer such that $Max(|v^1|) < 1$.

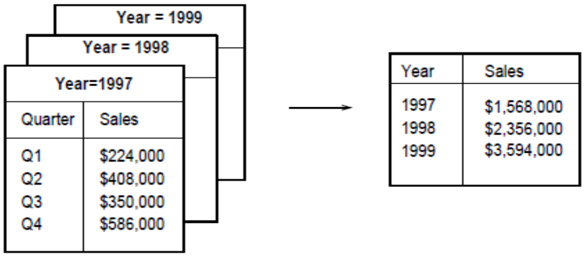
Example:- Suppose that the recorded values of A range from -986 to 917. The maximum absolute value of A is 986. To normalize by decimal scaling, divide each value by 1,000 (i.e., j = 3) so that -986 normalizes to -0.986

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

1. Data cube aggregation 2. Dimension reduction 3. Data compression
4. Numerosity reduction 5. Discretization and concept hierarchy generation

1. Data cube aggregation:- In Data cube aggregation, aggregation operations are applied to the data in the construction of a data cube.

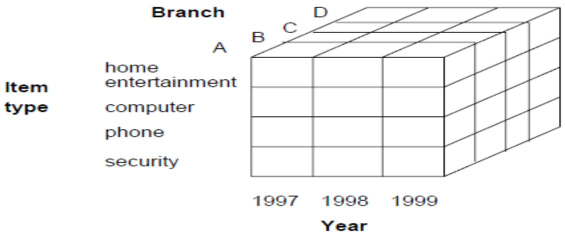
Suppose AllElectronics have their data as sales per quarter for the years 1997 to 1999 as shown in fig(a) . But the management are interested in the annual sales (total per year), rather than the total per quarter. Thus the data can be aggregated so that the resulting data summarize the total sales per year instead of per quarter. This aggregation is illustrated in Figure b. The resulting data set is smaller in volume, without loss of information necessary for the analysis task.


Year = 1999
Year = 1998

Year=1997	
Quarter	Sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

Year	Sales
1997	$1,568,000
1998	$2,356,000
1999	$3,594,000

Fig(a)



Fig(b)

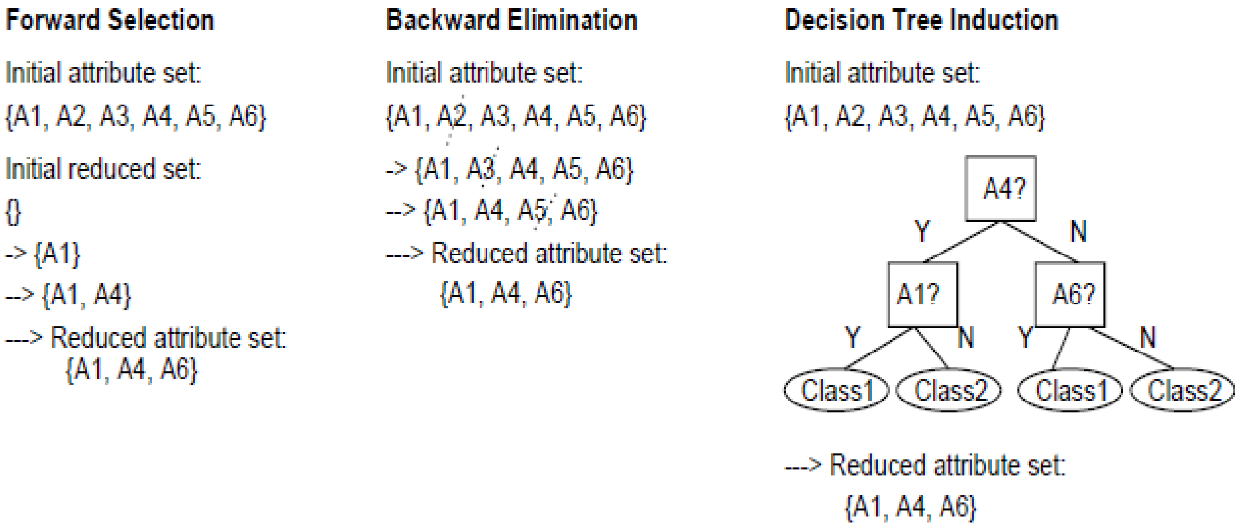
Datacubes store multidimensional aggregated information. Data cube consists of many cells and each cell holds an aggregate data value at multiple levels of abstraction. Data cubes provide fast access to precomputed, summarized data, thereby benefiting on-line analytical processing as well as data mining.

The cube created at the lowest level of abstraction is referred to as the base cuboid. A cube for the highest level of abstraction is the apex cuboid. For the sales data represented in the cube , the apex cuboid would give one total i.e the total sales for all three years, for all item types, and for all branches. Data cubes created for varying levels of abstraction are sometimes referred to as cuboids, so that a data cube may instead refer to a lattice of cuboids. Each higher level of abstraction further reduces the resulting data size.

2. Dimension reduction:- In Dimension reduction, irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.

Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task, or redundant. In analyzing customer music interest attributes such as the customer's telephone number are likely to be irrelevant and attributes such as age or music taste become relevant attributes .

The 'best' (and 'worst') attributes are typically selected using greedy methods. Some of the methods of attribute subset selection are



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

2. Step-wise 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 forward selection and backward elimination:- The step-wise forward selection and backward elimination methods can be combined, where at each step one selects the best attribute and removes the worst from among the remaining attributes.

4. Decision tree induction:- Decision tree induction constructs a flow-chart-like structure where each internal (non-leaf) node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each external (leaf) node denotes a class prediction. At each node, the algorithm chooses the “best” attribute to partition the data into individual classes. When decision tree induction is used for attribute subset selection, a tree is constructed from the given data. All attributes that do not appear in the tree are assumed to be irrelevant. The set of attributes appearing in the tree form the reduced subset of attributes.

3. Data compression:- In Data compression encoding mechanisms are used to reduced or “compressed" representation of the original data. If the original data can be reconstructed from the compressed data without any loss of information, the data compression technique used is called lossless. If, instead, we can reconstruct only an approximation of the original data, then the data compression technique is called lossy. Two popular and effective methods of lossy data compression are

1. Wavelet transforms and 2. Principal components analysis.

1. Wavelet transforms:- The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector D, transforms it to a numerically different vector, D0, of wavelet coefficients. This technique be useful for data reduction if the wavelet transformed data are of the same length as the original data.

The DWT is closely related to the discrete Fourier transform (DFT), a signal processing technique involving sines and cosines. In general the DWT achieves better lossy compression. That is, if the same number of coefficients are retained for a DWT and a DFT of a given data vector, the DWT version will provide a more accurate approximation of the original data.

Popular wavelet transforms include the Daubechies-4 and the Daubechies-6 transforms. Wavelet transforms can be applied to multidimensional data, such as a data cube. Wavelet transforms give good results on sparse or skewed data, and data with ordered attributes.

There is only one DFT, yet there are several DWTs. The general algorithm for a discrete wavelet transform is
as follows.

1. The length, L, of the input data vector must be an integer power of two. This condition can be met by padding the data vector with zeros, as necessary.
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.
3. The two functions are applied to pairs of the input data, resulting in two sets of data of length L=2. In general, these respectively represent a smoothed version of the input data, and the high-frequency content of it.
4. The two functions are recursively applied to the sets of data obtained in the previous loop, until the resulting data sets obtained are of desired length.
5. A selection of values from the data sets obtained in the above iterations are designated the wavelet coefficients of the transformed data.

2. Principal components analysis:- Principal components analysis is a method of data compression. PCA can be used as a form of dimensionality reduction. However, unlike attribute subset selection, which reduces the attribute set size by retaining a subset of the initial set of attributes, PCA “combines" the essence of attributes by creating an alternative, smaller set of variables. The initial data can then be projected onto this smaller set.

PCA can be applied to ordered and unordered attributes, and can handle sparse data and skewed data. Multidimensional data of more than two dimensions can be handled by reducing the problem to two dimensions.

3. Numerosity reduction

Can we reduce the data volume by choosing alternative, `smaller' forms of data representation?" Techniques of numerosity reduction can indeed be applied for this purpose. 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 be stored, instead of the actual data. (Outliers may also be stored). Log-linear models, which estimate discrete multidimensional probability distributions, are an example. Non-parametric methods for storing reduced representations of the data include histograms, clustering, and sampling.

Let's have a look at each of the numerosity reduction techniques mentioned above.

4.Regression and log-linear models

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

In linear regression, the data are modeled to to 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 = mX+c

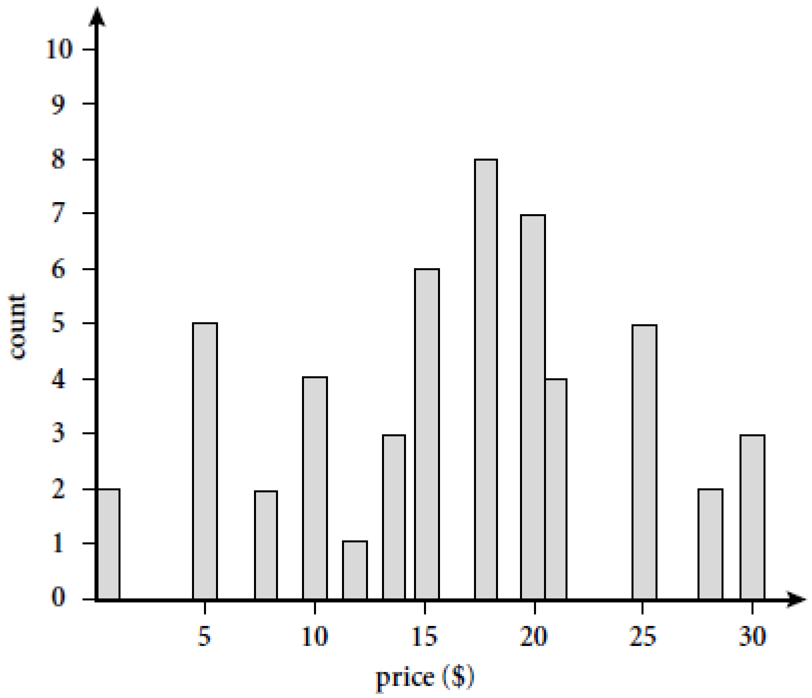
Multiple linear regression is an extension of linear regression allowing a response variable Y to be modeled as a linear function of a multidimensional feature vector. Log-linear models approximate discrete multidimensional probability distributions. The method can be used to

estimate the probability of each cell in a base cuboid for a set of discretized attributes, based on the smaller cuboids making up the data cube lattice. This allows higher order data cubes to be constructed from lower order ones.

5.Histograms

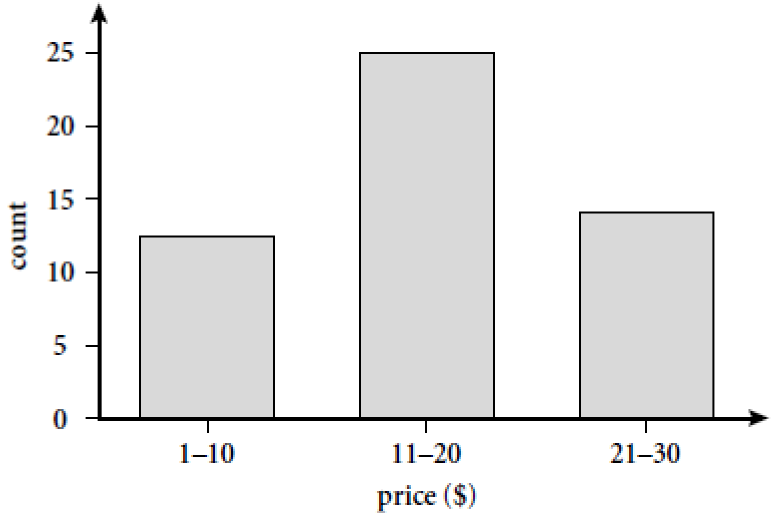
Histograms use binning to approximate data distributions and are a popular form of data reduction. A histogram

for an attribute A partitions the data distribution of A into disjoint subsets, or buckets. The buckets are displayed on a horizontal axis, while the height (and area) of a bucket typically represents the average frequency of the values represented by the bucket. 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.



ⁱ A histogram for *price* using singleton buckets—each bucket represents one price-value/ frequency pair.

Example The following data are a list of prices of commonly sold items at AllElectronics (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.



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

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

1. Equi-width: In an equi-width histogram, the width of each bucket range is constant (such as the width of $10 for the buckets in Figure 3.8).

2. Equi-depth (or equi-height): In an equi-depth histogram, the buckets are created so that, roughly, the frequency of each bucket is constant (that is, each bucket contains roughly the same number of contiguous data samples).

3. V-Optimal: If we consider all of the possible histograms for a given number of buckets, the V-optimal histogram is the one with the least variance. Histogram variance is a weighted sum of the original values that each bucket represents, where bucket weight is equal to the number of values in the bucket.

4. MaxDiff: In a MaxDiff histogram, we consider the difference between each pair of adjacent values. A bucket boundary is established between each pair for pairs having the 1 largest differences, where _ is user-specified.

V-Optimal and MaxDiff_ histograms tend to be the most accurate and practical. Histograms are highly effective

at approximating both sparse and dense data, as well as highly skewed, and uniform data.

6.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

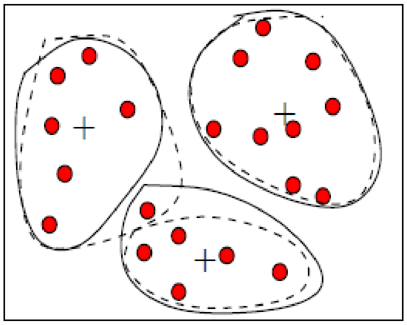


Figure 3.9: A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters. Each cluster centroid is marked with a "+".

7.Sampling:

Sampling can be used as a data reduction technique since it allows a large data set to be represented by a much

smaller random sample (or subset) of the data. Suppose that a large data set, D, contains N tuples. Let's have a look at some possible samples for D.

1. **Simple random sample without replacement (SRSWOR) of size n**: This is created by drawing n of the

N tuples from D (n < N), where the probably of drawing any tuple in D is 1=N, i.e., all tuples are equally likely.

2. **Simple random sample with replacement (SRSWR) of size n**: This is similar to SRSWOR, except that each time a tuple is drawn from D, it is recorded and then replaced. That is, after a tuple is drawn, it is placed back in D so that it may be drawn again.

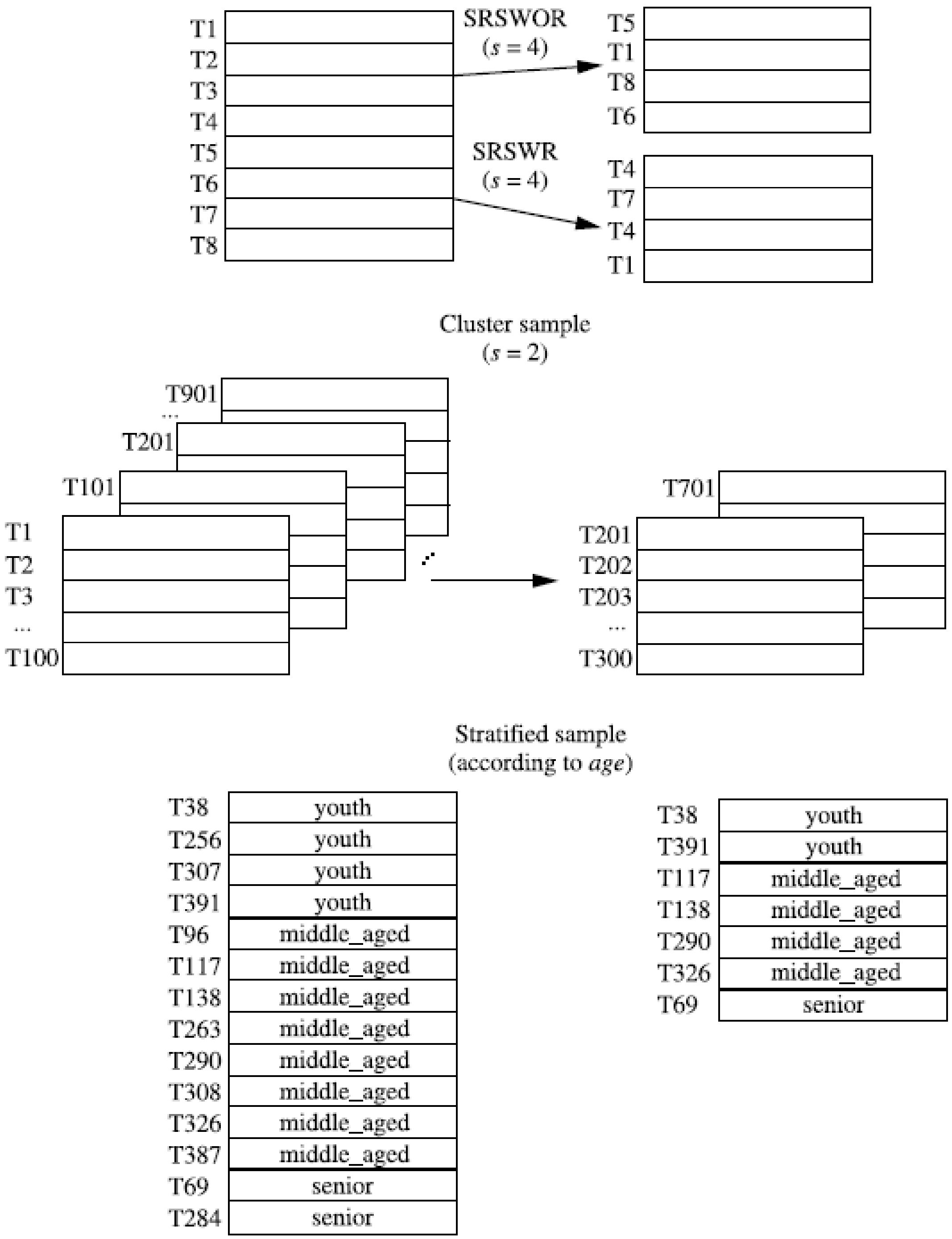
3**. Cluster sample**: If the tuples in D are grouped into M mutually disjoint “clusters", then a SRS of m clusters can be obtained, where m < M. For example, tuples in a database are usually retrieved a page at a time, so

that each page can be considered a cluster. A reduced data representation can be obtained by applying, say, SRSWOR to the pages, resulting in a cluster sample of the tuples.

4. **Stratified sample**: If D is divided into mutually disjoint parts called “strata", a stratified sample of D is generated by obtaining a SRS at each stratum. This helps to ensure a representative sample, especially when

the data are skewed. For example, a stratified sample may be obtained from customer data, where 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.



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

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

Sampling can be used for data reduction.

Data Discretization and Concept Hierarchy Generation:-

Data discretization techniques can be used to reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values.

Discretization and Concept Hierarchy Generation for Numerical Data

i) Binning

Binning is a top-down splitting technique based on a specified number of bins. For example, attribute values can be discretized by applying equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median, as in smoothing by bin means or smoothing by bin medians, respectively. These techniques can be applied recursively to the resulting partitions in order to generate concept hierarchies.

ii) Histogram Analysis

Histograms partition the values for an attribute, A, into disjoint ranges called buckets.

In an equal-width histogram, for example, the values are partitioned into equal-sized partitions or ranges.

With an equal frequency histogram, the values are partitioned so that, ideally, each partition contains the same number of data tuples.

iii) Entropy-Based Discretization

Entropy is one of the most commonly used discretization measures.

Let D consist of data tuples defined by a set of attributes and a class-label attribute.

The class-label attribute provides the class information per tuple. The basic method for entropy-based discretization of an attribute A within the set is as follows:

1. Each value of A can be considered as a potential interval boundary or split-point (denoted split point) to partition the range of A. That is, a split-point for A can partition the tuples in D into two subsets satisfying the conditions A <= split point and A > split point, respectively, thereby creating a binary discretization.

2. Entropy-based discretization, as mentioned above, uses information regarding the class label of tuples.

iv)Interval Merging by χ^2 Analysis

ChiMerge is a χ^2 based discretization method. ChiMerge proceeds as follows. Initially, each distinct value of a numerical attribute A is considered to be one interval. c2 tests are performed for every pair of adjacent intervals. Adjacent intervals with the least c2 values are merged together, because low c2 values for a pair indicate similar class distributions. This merging process proceeds recursively until a predefined stopping criterion is met.

iv) *Cluster Analysis*

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

Data Discretization and Concept Hierarchy Generation for categorical data:

There are several methods for the generation of concept hierarchies for categorical data. Some of them are

Specification of a partial ordering of attributes explicitly at the schema level by users or experts

A user or expert can easily define a concept hierarchy by 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

Specification of a portion of a hierarchy by explicit data grouping

we can easily specify explicit groupings for a small portion of intermediate-level data. For example, after specifying that province and country form a hierarchy at the schema level, a user could define some intermediate levels manually, such as **{**Urbana, Champaign, Chicago} < Illinois

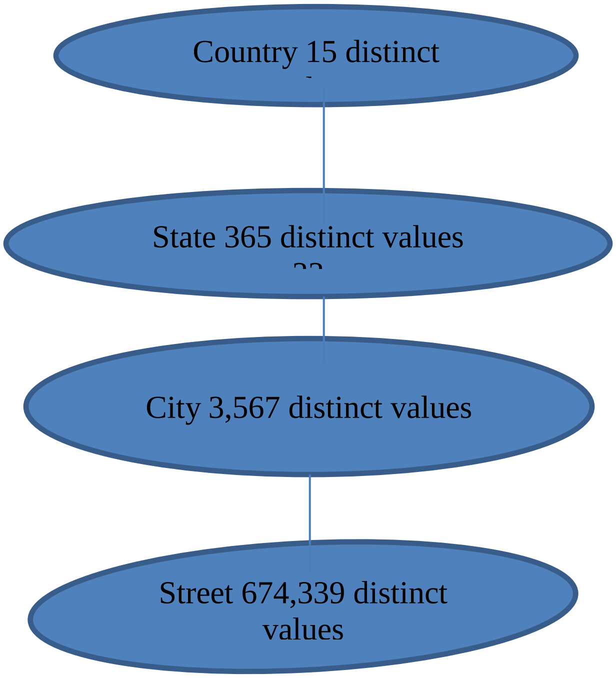
Specification of a set of attributes, but not of their partial ordering

A user may specify a set of attributes forming a concept hierarchy, but omit to explicitly state their partial ordering. The system can then try to automatically generate the attribute ordering so as to construct a meaningful concept hierarchy.

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


Country 15 distinct
State 365 distinct values
City 3,567 distinct values
Street 674,339 distinct values