UNIT-2

2.1 DATA PREPROCESSING

2.1.1 *Preprocessing*

Real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources. Low-quality data will lead to low-quality mining results, so we prefer a preprocessing concepts.

Data Preprocessing Techniques

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

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

* Data transformations, such as normalization, may be applied.

* Data reduction can reduce the data size by aggregating, eliminating redundant features, or clustering, for instance. These techniques are not mutually exclusive; they may work together.

2.1.2 Need for preprocessing

Incomplete, noisy and inconsistent data are common place properties of large real world databases and data warehouses.

Incomplete data can occur for a number of reasons:

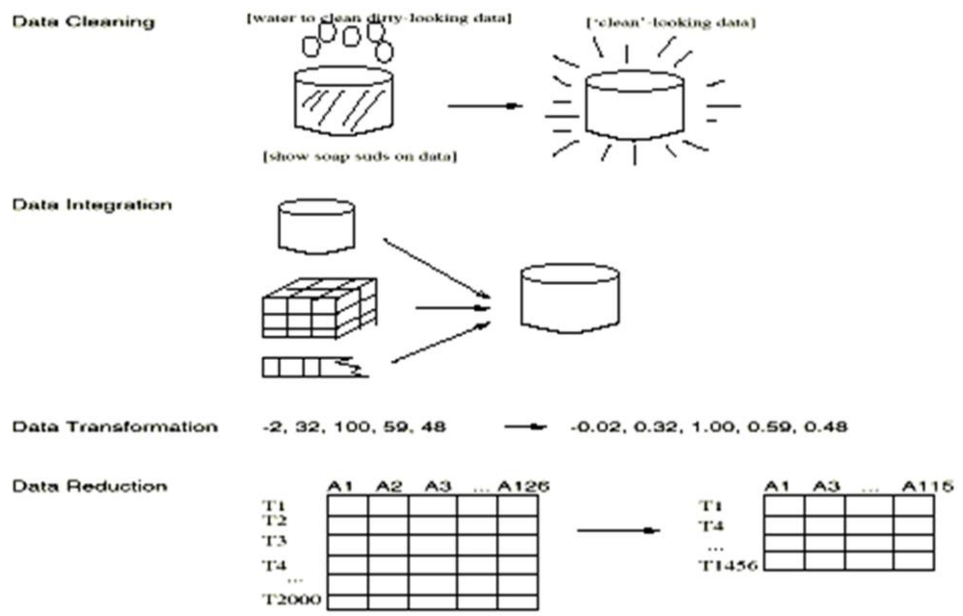
- Attributes of interest may not always be available
- Relevant data may not be recorded due to misunderstanding, or because of equipment malfunctions.
- Data that were inconsistent with other recorded data may have been deleted.
- Missing data, particularly for tuples with missing values for some attributes, may need to be inferred.
- The data collection instruments used may be faulty.
- There may have been human or computer errors occurring at data entry.
- Errors in data transmission can also occur.
- There may be technology limitations, such as limited buffer size for coordinating synchronized data transfer and consumption.

- Data cleaning routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies.
- Data integration is the process of integrating multiple databases cubes or files. Yet some attributes representing a given may have different names in different databases, causing inconsistencies and redundancies.
- Data transformation is a kind of operations, such as normalization and aggregation, are additional data preprocessing procedures that would contribute toward the success of the mining process.
- Data reduction obtains a reduced representation of data set that is much smaller in volume, yet produces the same(or almost the same) analytical results. There are a number of strategies for data reduction. These include

  1. Data aggregation (e.g., building a data cube)

  2. Attribute subset selection (e.g., removing irrelevant attributes through correlation analysis)

Forms of Data Preprocessing



3. Dimensionality reduction (e.g., using encoding schemes such as minimum length encoding or wavelets)

4. Numerosity reduction (e.g., ”replacing ” the data by alternating, Smaller representation such as clusters or parametric models).

5. Generalization (e.g., data is reduce with use of concept hierarchy)

* Data discretization is a form of data reduction which is very useful for automatic generation of concept hierarchies from numerical data.

2.2 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.

2.2.1 Missing Values

Many tuples have no recorded value for several attributes, such as customer income .so we can fill the missing values for this attributes.

The following methods are useful for performing missing values over several attributes:

1. Ignore the tuple: This is usually done when the class label 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 the missing values per attribute varies considerably.
2. Fill in the missing values manually: 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 value by the same constant, such as a label like “unknown” or -∞.
4. Use the attribute mean to fill in the missing value: For example, suppose that the average income of customers is $56,000. Use this value to replace the missing value for income.

5. Use the attribute mean for all samples belonging to the same class as the given tuple: If classifying customers according to credit risk, replace the missing value with the average income value for customers in the same credit risk category as that of give tuple.
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 the sets decision tree is constructed to predict the missing value for income.

2.2.2 *Noisy Data*

Noise is a random error or variance in a measured variable. Noise is removed using data smoothing techniques.

Binning: Binning methods smooth a sorted data value by consulting its “neighborhood,” that is the value 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.

Sorted data for price (in dollars): 3,7,14,19,23,24,31,33,38.

Example 1: Partition into (equal-frequency) bins:

Bin 1: 3,7,14
Bin 2: 19,23,24
Bin 3: 31,33,38

In the above method the data for price are first sorted and then partitioned into equal-frequency bins of size 3.

Smoothing by bin means:

Bin 1: 8,8,8
Bin 2: 22,22,22
Bin 3: 34,34,34

In smoothing by bin means method, each value in a bin is replaced by the mean value of the bin. For example, the mean of the values 3,7&14 in bin 1 is 8[(3+7+14)/3].

Smoothing by bin boundaries:

Bin 1: 3,3,14
Bin 2: 19,24,24
Bin 3: 31,31,38

In smoothing by bin boundaries, the maximum & minimum values in give bin or identify as the bin boundaries. Each bin value is then replaced by the closest boundary value. In general, the large the width, the greater the effect of the smoothing. Alternatively, bins may be equal-width, where the interval range of values in each bin is constant

Example 2: Remove the noise in the following data using smoothing techniques:

8, 4,9,21,25,24,29,26,28,15

Sorted data for price (in dollars):

4,8,9,15,21,21,24,25,26,28,29,34

- Partition into equal-frequency (equi-depth) bins:

  Bin 1: 4, 8,9,15
  Bin 2: 21,21,24,25
  Bin 3: 26,28,29,34

- Smoothing by bin means:

  Bin 1: 9,9,9,9
  Bin 2: 23,23,23,23

Bin 3: 29,29,29,29

- Smoothing by bin boundaries:

Bin 1: 4, 4,4,15
Bin 2: 21,21,25,25
Bin3: 26,26,26,34

Regression: Data can be smoothed by fitting the data to function, such as with regression. Linear regression involves finding the “best” line to fit two attributes (or variables), so that one attribute can be used to predict the other me. 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.

Clustering: Outliers may be detected by clustering, where similar values are organized into groups, or “clusters.” Intuitively, values that fall outside of the set of clusters may be considered outliers.

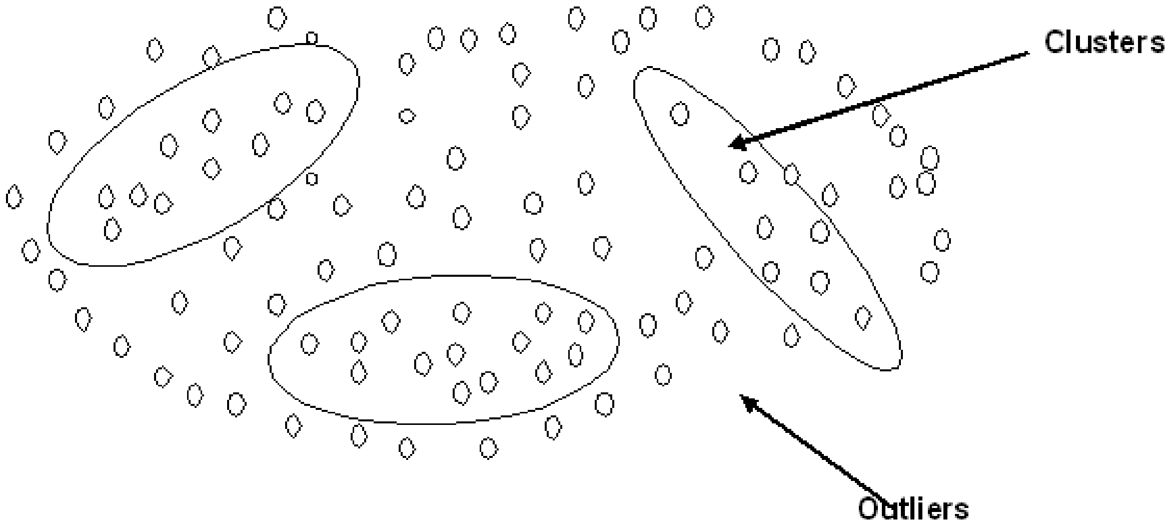


Figure 3.2 Outliers detected by clustering analysis

2.2.3 *Inconsistent Data*

Inconsistencies exist in the data stored in the transaction. Inconsistencies occur due to occur during data entry, functional dependencies between attributes and missing values. The inconsistencies can be detected and corrected either by manually or by knowledge engineering tools.

Data cleaning as a *process*

- Discrepancy detection
- Data transformations

1. *Discrepancy detection*

The first step in data cleaning is discrepancy detection. It considers the knowledge of meta data and examines the following rules for detecting the discrepancy.

Unique rules- each value of the given attribute must be different from all other values for that attribute.

Consecutive rules – Implies no missing values between the lowest and highest values for the attribute and that all values must also be unique .

Null rules - specifies the use of blanks, question marks, special characters, or other strings that may indicates the null condition

Discrepancy detection Tools:

- Data scrubbing tools - use simple domain knowledge (e.g., knowledge of postal addresses, and spell-checking) to detect errors and make corrections in the data
- Data auditing tools – analyzes the data to discover rules and relationship, and detecting data that violate such conditions.

2. *Data transformations*:

This is the second step in data cleaning as a process. After detecting discrepancies, we need to define and apply (a series of) transformations to correct them.
Data Transformations Tools:

- Data migration tools – allows simple transformation to be specified, such to replaced the string “gender” by “sex”.
- ETL (Extraction/Transformation/Loading) tools – allows users to specific transforms through a graphical user interface(GUI)

2.2.4 Disadvantages of Data Cleaning Process

* Nested discrepancies
* Lack of interactivity.
* Increased interactivity.

2.3 DATA INTEGRATION

Data mining often requires data integration - the merging of data from stores into a coherent data store, as in data warehousing. These sources may include multiple data bases, data cubes, or flat files.

Issues in Data Integration

- Schema integration & object matching.
- Redundancy.
- Detection & Resolution of data value conflict

Schema Integration & Object Matching

Schema integration & object matching can be tricky because same entity can be represented in different forms in different tables. This is referred to as the entity identification problem. Metadata can be used to help avoid errors in schema integration. The meta data may also be used to help transform the data (e.g., where data codes for may type in one data base may be “H”&”S”,&1&2in another).

Redundancy:

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

Give two attribute, such analysis can measure how strongly one attribute implies the other, based on available data. For numerical attributes, we can evaluate the correlation between two attributes, A and B, by computing the correlation coefficient

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

where

- n is the number of tuples
- $\overline{\boldsymbol{A}}$ mean value of A
- $\overline{\boldsymbol{B}}$ mean value of B

- $\boldsymbol{\sigma_A}$ Standard deviation of A
- $\boldsymbol{\sigma_B}$ Standard deviation of B

$$\overline{A} = \frac{\sum A}{n};\ \sigma_A = \sqrt{\frac{\sum(A-\overline{A})^2}{n-1}}$$

If

r $_{A\ ,B}$ >0 then A and B are positively correlated
r $_{A,B}$<0 then A and B are negatively correlated
r $_{A,B}$=0 then on correlation between A and B.

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 value from different sources may differ. This may be due to difference in representation, scaling, or encoding. For instance, a weight attribute may be stored in metric units in one system and British imperial units in another. For a hotel chain, the *price* of rooms in different cities may involve not only different currencies but also different services (such as free breakfast) and taxes. An attribute in one system may be recorded at a lower level of abstraction than the “same” attribute in another.

The semantic heterogeneity and structure of data pose great challenges in data integration.

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

2.4. DATA TRANSFROMATION

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

Data transformation can involve the following:

- Smoothing.
- Aggregation
- Generalization of data
- Normalization
- Attribute construction

Smoothing

Smoothing, this works to remove noise from the data. Such techniques include binning, regression, and clustering.

Aggregation

Aggregation, where summary or aggregation operations are applied to the data at multiple granularities.

Generalization of data,

Generalization of the data, where low-level or “primitive” (raw) data are replaced by high-level concepts through the use of concepts hierarchies. For example, categorical attributes, like *street* can be generalized to high-level concepts, like *city* or *country.*

Normalization

Normalization, where the attribute data are scaled so as to fall within a small specified range, such as –1.0 to 1.0, or 0.0 to 1.0

Data Normalization Methods

1. *min-max normalization*

2. *z-score normalization*
3. *normalization by decimal scaling*

Min-max normalization performs a linear transformation on the original data. Suppose that $\boldsymbol{min_A}$ and *max*A are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, *V*, of A to $\boldsymbol{V'}$ in the range [*new_min*A,- *new_max*A] by computing

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

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: Suppose that the minimum and maximum values for the attribute income are $1,000 and $15,000 respectively. Map income to the range [0.0,1.0]. By min-max normalization a value $12,000 for income is transformed to

$$= \frac{\mathbf{12{,}000-1{,}000}}{\mathbf{15{,}000-1{,}000}}\ \mathbf{(1.0 - 0) + 0}$$

= 0.785

z-score normalization (or *zero-mean normalization*). In this method, 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^l by computing

$$\boldsymbol{V' = \frac{V - \bar{A}}{\sigma_A}}$$

Where Ā and ⌐$_A$ are the mean and standard deviation, respectively, of attribute *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: Suppose that the mean and standard deviation of the value for the attribute income are $52,000 & $14,000, respectively. With z-score normalization, a value of $72,000 for income is transformed to

$$\mathbf{\frac{72{,}000 - 52{,}000}{14{,}000} = 1.42}$$

Normalization by decimal scaling normalizes by moving the decimal point of value of attribute *A*. The number of decimal points mood depends on the maximum absolute value of *A*. A value *V* of A is normalized to V^l by computing

$$\boldsymbol{V' = \frac{V}{10^j}}$$

Where *j* is the smallest integer such that Max (l$\boldsymbol{V'}$l) < 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, we therefore divide each value by 1000 (i.e., *j*=3) so that -986 normalizes -0.986 and 917 normalizes to 0.917.

Attribute Construction:

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

Example. Given one-dimensional data set *X* = {-5,023.0,17.6,9.23,1.11}, normalize the data set using

(a) Min-max normalization on interval [0,1],
(b) Min-max normalization on interval [-1,1],
(c) Standard deviation normalization.

- Min-max normalization [0,1]

$$V' = \frac{V - min_A}{max_A - \ min_A}\ (newmax_A \text{ – } newmin_A) + new_{min_f}$$

$$-5.0 \colon V' = \frac{-5.0 - (-5.0)}{23.0 - \ (-5.0)}\ \big(1 - (0)\big) + (0) = 0$$

$$23.0 \colon V' = \frac{23.0 - (-5.0)}{23.0 - \ (-5.0)}\ \big(1 - (0)\big) + (0) = 1$$

$$17.6 \colon V' = \frac{17.6 - (-5.0)}{23.0 - \ (-5.0)}\ \big(1 - (0)\big) + (0) = 0.807$$

$$7.23 \colon V' = \frac{7.23 - (-5.0)}{23.0 - \ (-5.0)}\ (1 - 0) + (0) = 0.436$$

$$1.11 \colon V' = \frac{1.11 - (-5.0)}{23.0 - \ (-5.0)}\ (1 - 0) + (0) = 0.563$$

b) Min-Max Normalization on interval [-1,1]

$$V' = \frac{V - min_A}{max_A - \ min_A}\ (newmax_A \text{ – } newmin_A)\ + new\_min_A$$

$$-5.0 \colon V' = \frac{-5.0 - (-5.0)}{23.0 - \ (-5.0)}\ \big(1 \text{ – } (-1)\big) + (-1) = 0$$

$$23.0 \colon V' = \frac{23.0 - (-5.0)}{23.0 - \ (-5.0)}\ (2) - (1) = 1$$

$$17.6 \colon V' = \frac{17.6 - (-5.0)}{23.0 - \ (-5.0)}\ (2) - (1) = 0.6143$$

$$7.23 \colon V' = \frac{7.23 - (-5.0)}{23.0 - \ (-5.0)}\ (2) - (1) = -0.1264$$

$$1.11 \colon V' = \frac{1.11 - (-5.0)}{23.0 - \ (-5.0)}\ (2) - (1) = -0.5635$$

C) Standard Deviation Normalization

Variance, $\boldsymbol{\sigma^2} = \frac{1}{N}\left[\sum x_i^2 - \frac{1}{N}(\sum x_i)^2\right]$

$\sum \boldsymbol{x_i}$=48.44 $(\sum x_i)^2$=2346.4

$\sum x_i^2$=917.26

$\boldsymbol{\sigma}^2$=89.6

Standard deviation, $\boldsymbol{\sigma}$=√89.6=9.46

Mean=$\frac{\sum x_i}{N}$=9.68

Standard deviation normalization, $V' = \frac{(V(i) - Mean(V))}{sd(V)}$

$$-5.0 \colon V' = \frac{(-5.0 - 9.68)}{9.46} = -1.5$$

$$\mathbf{23.0}: \boldsymbol{V}' = \frac{\mathbf{(23.0 - 9.68)}}{\mathbf{9.46}} = \mathbf{1.408}$$
$$\mathbf{17.6}: \boldsymbol{V}' = \frac{\mathbf{(17.6 - 9.68)}}{\mathbf{9.46}} = \mathbf{0.837}$$
$$\mathbf{7.23}: \boldsymbol{V}' = \frac{\mathbf{(7.23 - 9.68)}}{\mathbf{9.46}} = \mathbf{-0.258}$$
$$\mathbf{1.11}: \boldsymbol{V}' = \frac{\mathbf{(1.11 - 9.68)}}{\mathbf{9.46}} = \mathbf{-0.9}$$

2.5 DATA REDUCTION

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

Data Reduction Strategies

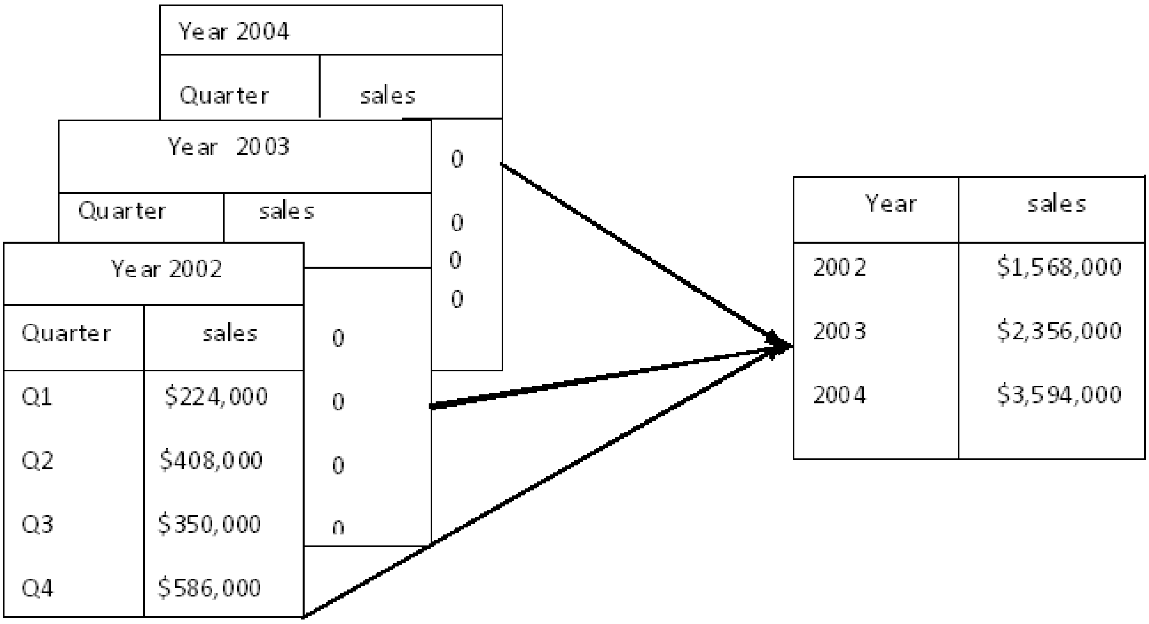
1. Data cube aggregation, where aggregation operations are applied to the data in the construction of a data cube.
2. Attribute subset selection, where irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.
3. Dimensionality reduction, where encoding mechanisms are used to reduced the data set size.
4. Numerosity reduction, where the data are replaced or estimated by alternative data smaller data representation.
5. Discretization and concept hierarchy generation, where raw data values for attributes are replaced by ranges or higher conceptual levels

2.5.1 Data cube Aggregation

Data cube aggregation, where aggregation operations are applied to the data for construction of a data cube. Data cubes store multidimensional aggregated information. Each cell holds an aggregate data value, corresponding to the data point in multidimensional space. Concept hierarchies may exist for each attribute, allowing the analysis of data at multiple level of abstraction. Data cubes provide fast access to pre computed summarized data, thereby benefiting on-line analytical processing as well as data mining.

The cube can be created in three ways:

- Based cuboid- The cube created at the lowest level of abstraction is referred to as base cuboid.
- Lattice of cuboids- Data cubes created for varying levels of abstraction are often referred to as cuboids.
- Apex cuboid - A cube at highest level of abstraction is the apex cuboid



Year 2002	
Quarter	sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

Year	sales
2002	$1,568,000
2003	$2,356,000
2004	$3,594,000

Fig .2.3(a) on the left, the sales are shown per quarter. On the right ,the data are aggregated to provide the annual sales.

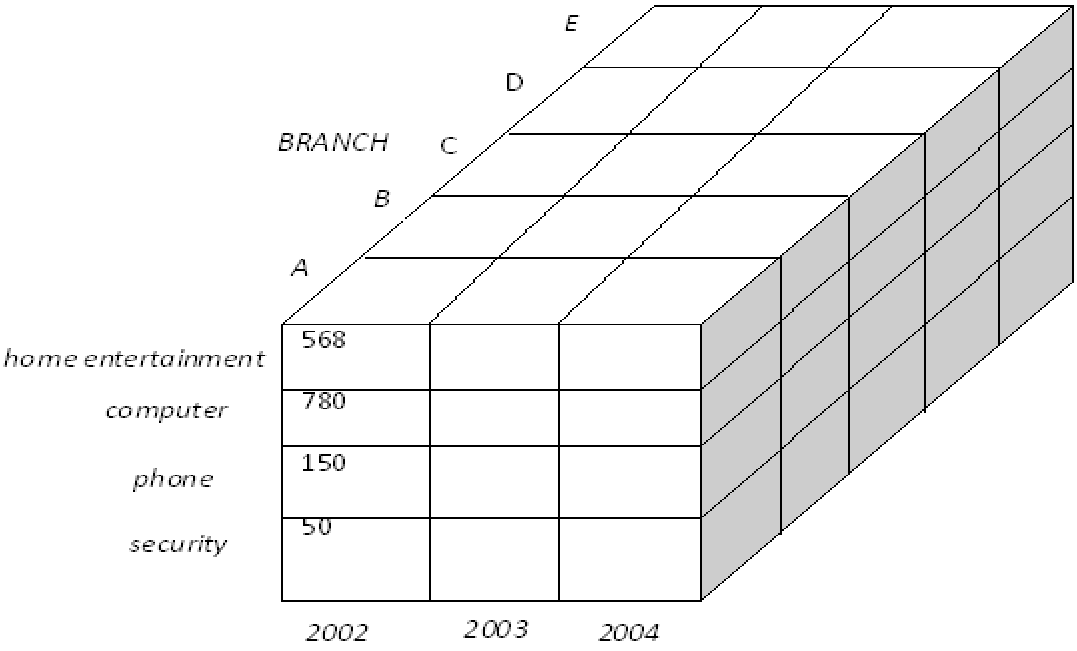


Fig 2.3(b) A data cube for sales

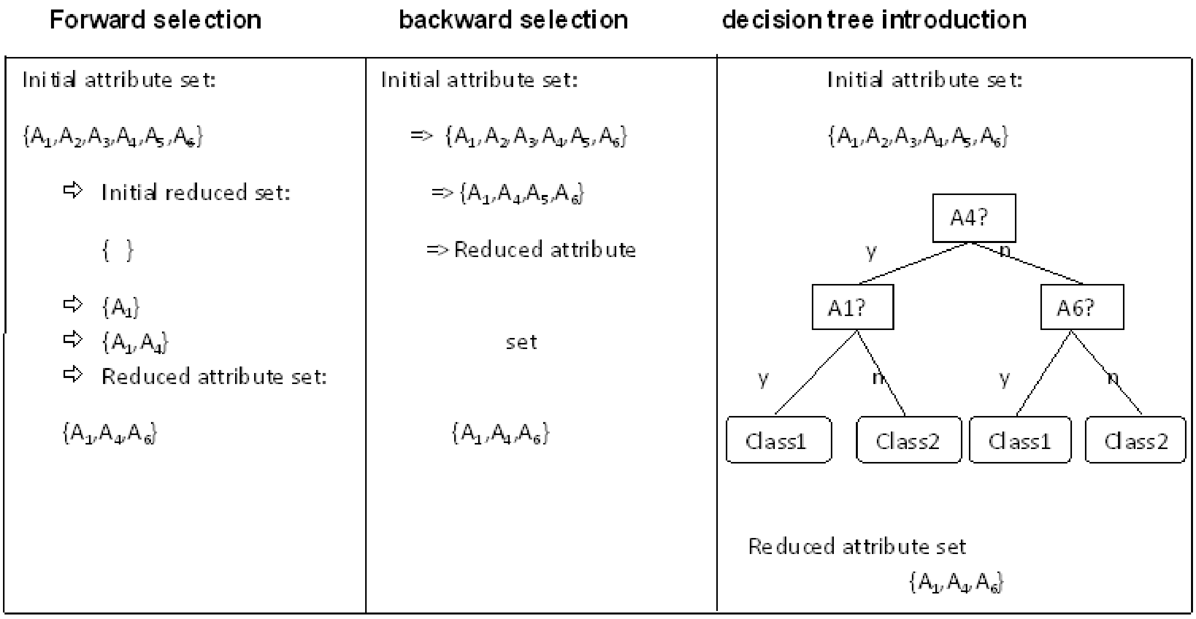
Each higher level of a abstraction further reduces the resulting data size. When replying to data mining requests, the smallest available cuboid relevant to the give task should be used.

2.5.2 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. It reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

To find out a 'good' subset from 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 to attribute space, they always make what looks to be the best choice at that time. Their strategy to make a locally optimal choice in the hope that this will lead to a globally optimal solution. Many other attributes evaluation measure can be used, such as the information gain measure used in building decision trees for classification.



Techniques for heuristic methods of attribute sub set selection

- ⇨ Stepwise forward selection
- ⇨ Stepwise backward elimination
- ⇨ Combination of forward selection and backward elimination
- ⇨ Decision tree induction

1. *Stepwise forward selection*: The procedure starts with an empty set of attributes as the reduced set. The best of 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 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 induction constructs a flowchart like structure where each internal node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each leaf node denotes a class prediction. At each node, the algorithm choices the"best" attribute to partition the data into individual classes. 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 from the reduced subset of attributes. Threshold measure is used as stopping criteria.

2.5.3 Dimensionality reduction

In dimensionality reduction, data encoding or transformations are applied so as to obtained reduced or "compressed" representation of the oriental data.

Dimension Reduction Types

- Lossless - If the original data can be *reconstructed* from the compressed data without any loss of information
- Lossy - If the original data can be reconstructed from the compressed data with loss of information, then the data reduction is called lossy.

Effective methods in lossy dimensional reduction

- Wavelet transforms
- Principal components analysis.

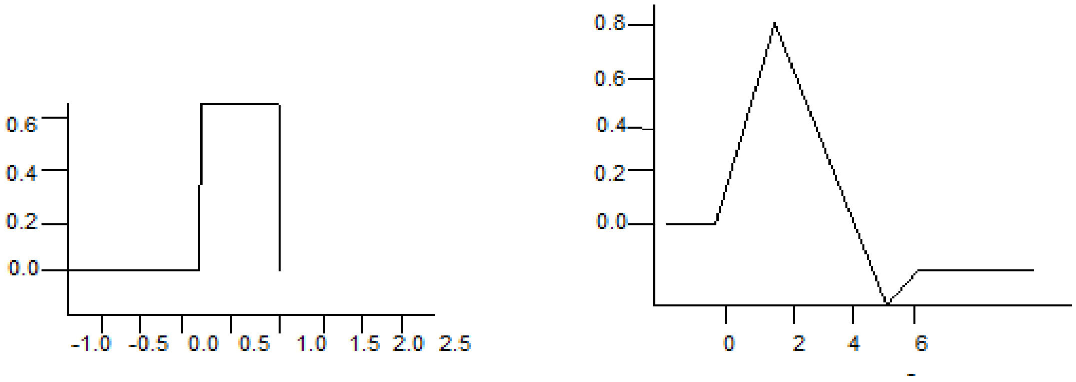
Wavelet Transforms

The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector, transforms it to a numerically different vector, of wavelet coefficients. The two vectors are of the same length. When applying this technique to data reduction, we consider each tuple as an n-dimensional data vector, that is, $X=(x_1,x_2,\ldots\ldots\ldots\ldots,x_n)$, depicting n measurements made on the tuple from n database attributes .

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 can take advantage of data sparsity are computationally very fast if performed in wavelet space. Given a set of coefficients, an approximation of the original data can be constructed by applying the inverse of the DWT used.

The numbers next to a wave let name is the number of vanishing moment of the wavelet this is a set of mathematical relationships that the coefficient must satisfy and is related to number of coefficients.

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



(a)haar2

(b) Daubechies4

Figure 3.5 Examples of wavelet families

Hierarchical pyramid algorithm for discrete wavelet transformation:

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 data.
3. The two functions are applied to pairs of data points in X, that is, to all pairs of measurements (X_{2i} , X_{2i+1}). This results in two sets of data of length *L/2*. In general, these represent a smoothed or low-frequency version of the input data and high frequency content of it, respectively.
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 length 2.

Principal Components Analysis

Suppose that the data to be reduced, which Karhunen-Loeve, K-L, method consists of tuples or data vectors describe by n attributes or dimensions. Principal components analysis, or 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. 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.

The basic procedure is as follows:

- The input data are normalized.
- 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.
- The principal components are sorted in order of decreasing significance or strength.

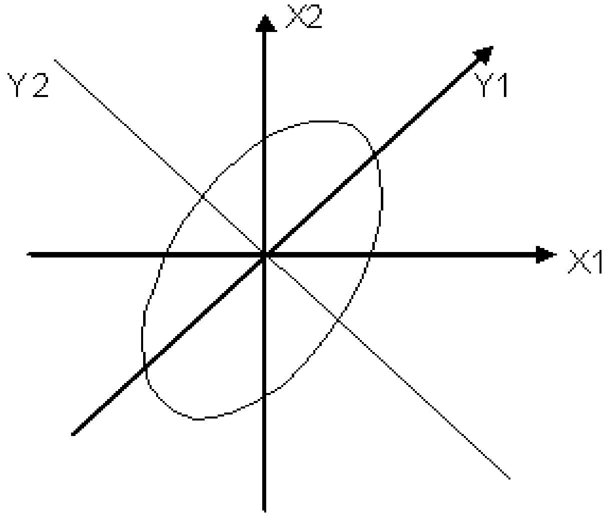


Figure 2.6 principal components analysis.

In the above figure, Y_1 and Y_2, for the given set of data originally mapped to the axes X_1 and X_2. This information helps identify groups or patterns within the data. 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.

- The size of the data can be reduced by eliminating the weaker components.

Advantage of PCA

- PCA is computationally inexpensive
- Multidimensional data of more than two dimensions can be handled by reducing the problem to two dimensions.
- Principal components may be used as inputs to multiple regression and cluster analysis.

2.5.4 Numerosity Reduction

Numerosity reduction is used to reduced the data volume by choosing alternative, smaller forms of the data representation

Techniques for Numerosity reduction:

- Parametric - In this model only the data parameters need to be stored, instead of the actual data. (e.g.,) Log-linear models, Regression
- Nonparametric – This method stores reduced representations of data include histograms, clustering, and sampling

Parametric model

1. Regression

- Linear regression

  In linear regression, the data are model 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=αX+β
  Where the variance of Y is assumed to be constant. The coefficients, α and β (called regression coefficients), specify the slope of the line and the Y- intercept, respectively.

- Multiple- linear regression

  Multiple linear regression is an extension of (simple) linear regression, allowing a response variable Y, to be modeled as a linear function of two or more predictor variables.

2. Log-Linear Models

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.

Nonparametric Model

1. Histograms

   A histogram for an attribute A partitions the data distribution of A into disjoint subsets, or
   buckets. If each bucket represents only a single attribute-value/frequency pair, the buckets are
   Called singleton buckets.
   Ex: The following data are bast of prices of commonly sold items at All Electronics. The numbers have been sorted: 1,1,5,5,5,5,5,8,8,10,10,10,10,12,14,14,14,15,15,15,15,15,18,18,18,18,18,18, 18,18,20,20,20,20,20,20,21,21,21,21,21,25,25,25,25,25,28,28,30,30,30

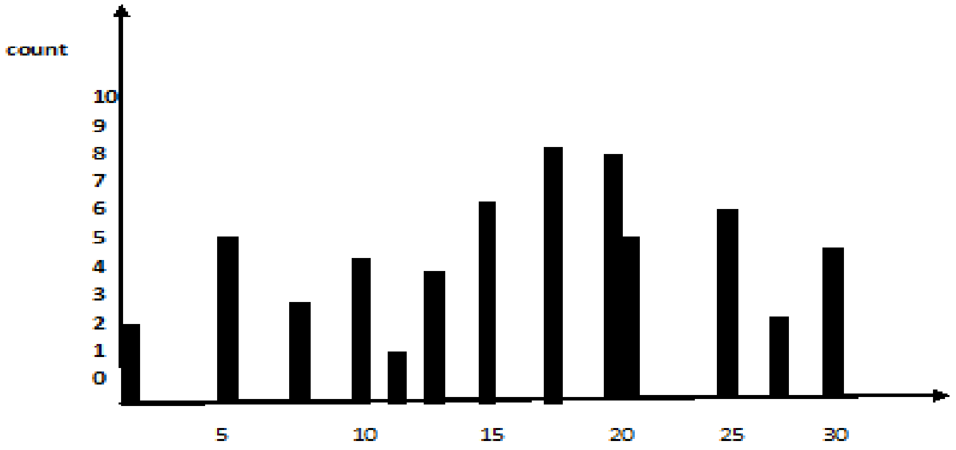


Figure 3.7 A Histogram for price using Singleton Buckets

There are several partitioning rules including the following:
Equal-width: The width of each bucket range is uniform

- (Equal-frequency (or equi-depth): the frequency of each bucket is constant

V-Optimal: The V-Optional 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.
MaxDiff: It is the difference between each pair of adjacent values. A bucket boundary is established between each pair for pairs having the β-1 largest differences, where β is the user-specified.

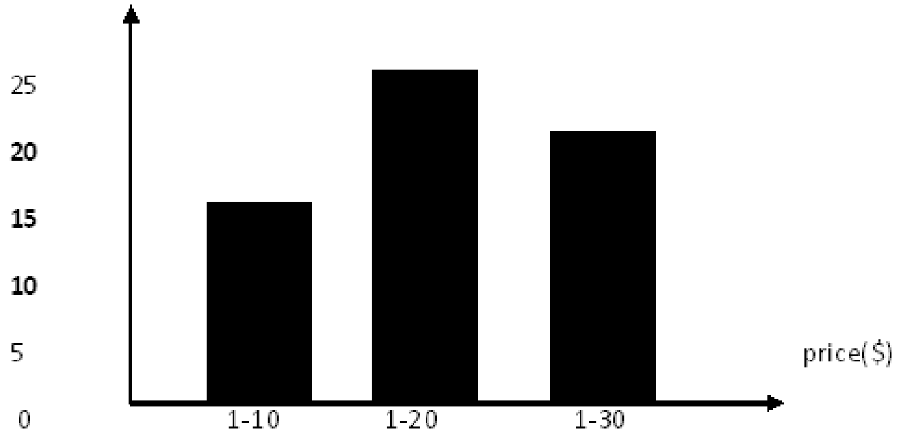


Figure.2.8 An equal-width histogram for price, where values are aggregated so ***that each bucket has a uniform width of $10.***

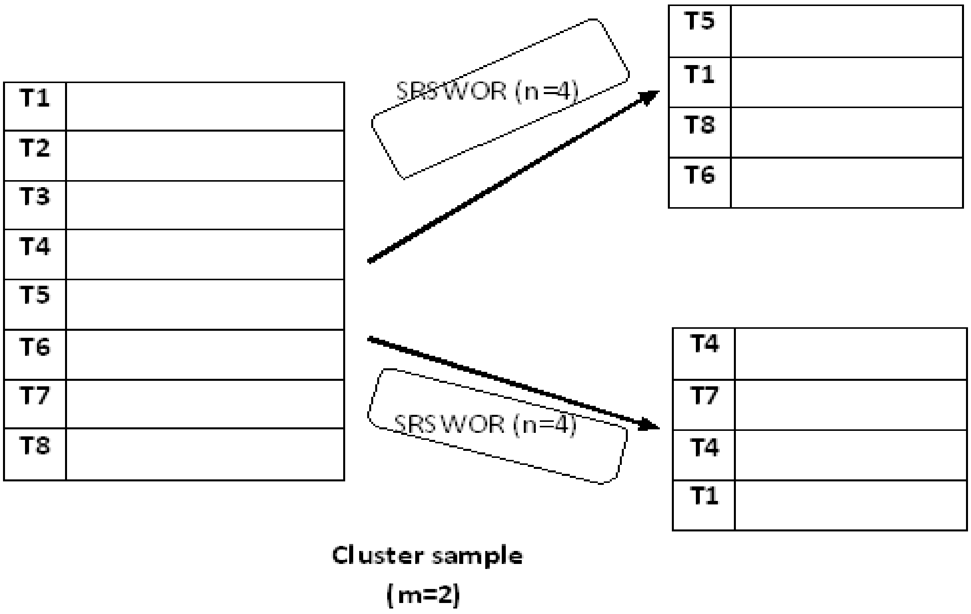
Clustering

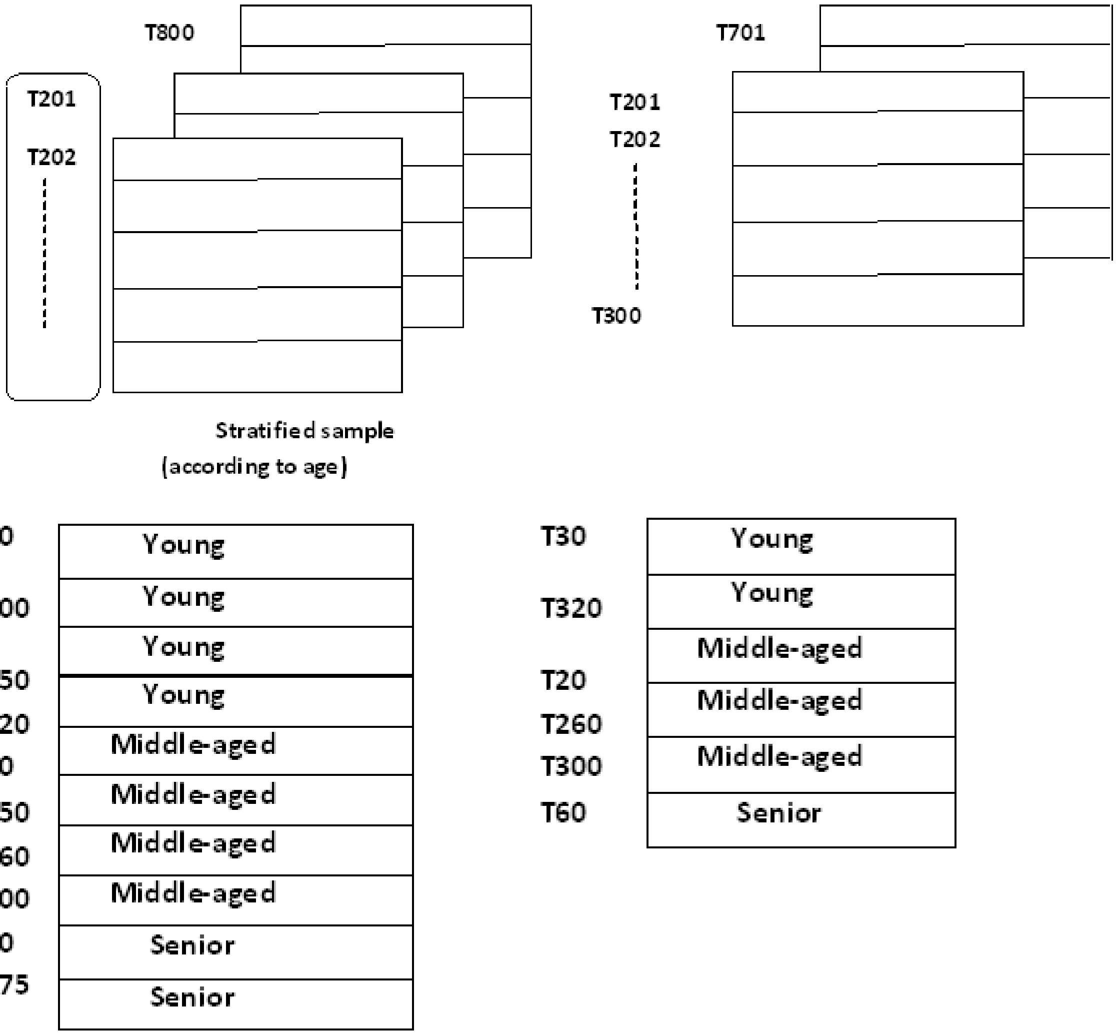
Clustering technique consider data tuples as objects. They partition the objects into groups or clusters, so that objects within a cluster are similar to one another and dissimilar to objects in other clusters. Similarity is defined in terms of how close the objects are in space, based on a distance 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.

In data reduction, the cluster representations of the data are used to replace the actual data. It is much more effective for data that can be organized into distinct clusters then for smeared data.

2. 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 sample (or subset) of the data. Suppose that a large data set D, contains N tuples, then the possible samples are:





T30	Young
T200	Young
T250	Young
T320	Young
T90	Middle-aged
T150	Middle-aged
T260	Middle-aged
T300	Middle-aged
T60	Senior
T275	Senior

T30	Young
T320	Young
T20	Middle-aged
T260	Middle-aged
T300	Middle-aged
T60	Senior

Figure 2.9. Sampling can be used for data reduction.

Simple Random sample without Replacement (SRS WOR) of size *n*: This is created by drawing '*n*' of the 'N' tuples from D (*n*<N), where the probability of drawing any tuple in D is 1/N, i.e., 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 .i.e., 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 m clusters can be obtained, where *m* < M.

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.

Advantages of sampling

- The cost of obtaining a sample is proportional to the size of the sample S, as opposed to the data set size N.
- Sampling complexity is potentially sub linear to the size of the data.
- It is possible to determine a sufficient sample size for estimating a given function within a specified degree of error.

- This sample size, s, may be extremely small in comparison to N.
- Sampling is a natural choice for the progressive refinement of a reduced data set.
- Data set can be further refined by simply increasing the sample size.

2.6 DATA DISCRETIZATION AND CONCEPT HIERARCHY GENERATION

2.6.1 DATA DISCRETIZATION

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

Features of Data Discretization

- Leads to a concise
- Easy-to-use
- Knowledge-level representation of mining results.

Categories of Data Discretization

- Supervised discretization- Uses class information.
- Unsupervised discretization or splitting – Does not uses class information.
- Top-down discretization or splitting – Here, 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.
- Bottom-up discretization or merging – Here, the process starts by considering all of the continuous values to form intervals, and then recursively applies this process to the resulting intervals.

2.6.2 Concept Hierarchy

A concept hierarchy for a given numerical attribute defines a discretization of attribute. Concept hierarchies can be used to reduced the data by collecting and replacing low-level concepts (such as numerical values for the attribute age) with higher-level concepts (such as youth, middle-aged, or senior).

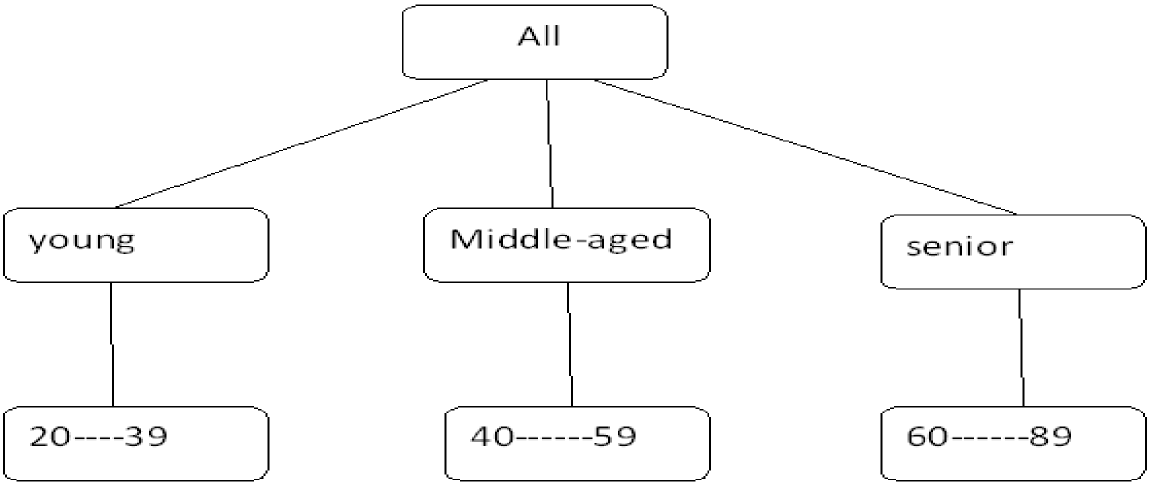


Fig:2.10 A concept hierarchy for the attribute age

2.6.3 Discretization and concept Hierarchy Generation for Numerical data

Concept hierarchies for numerical attributes can be constructed automatically based on data discretization.

Methods for handling numerical data over concept hierarchy

- Binning
- Histogram analysis
- Entropy-based discretization
- χ^2-merging
- Cluster analysis
- Discretization by intuitive partitioning

I. Binning

Binning is a top-down splitting technique based on a specified number of bins. These methods are also used as discretization methods for numerosity reduction and concepts hierarchy generation. These techniques can be applied recursively to the resulting partitions in order to generate concepts hierarchies. Binning does not use class information and is therefore an unsupervised discretization technique. It is sensitive to the user-specified number of bins, as well as the presence of outliers.

II. Histogram Analysis

Like binning, histogram analysis is an unsupervised discretization technique because it does not use class information. Histograms partition the value for an attribute A, into disjoint ranges called *buckets.* The histograms analysis algorithm can be applied recursively to each partition in order to automatically generate a multilevel concepts hierarchy, with the procedure terminating once a prespecified number of concepts levels has been reached.

III. Entropy-Based Discretization

Entropy is one of the most commonly used discretization measures. Entropy-based discretization is a supervised, top-down splitting technique. It explores class distribution information in its calculation and determination of split-points. To discretize a numerical attribute, A, the method selects the value of A that has the minimum entropy as a split-point, and recursively partitions the resulting intervals to arrive at a hierarchical discretization. Such discretization forms a concept hierarchy for A. Let D consist of data tuples defined by a set of attributes and a class-label attribute.

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 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. Suppose we want to classify the tuples in D by partitioning on attribute A and some split-point. Ideally, we would like this partitioning to result in an exact classification of the tuples. For example, if we had two classes, we would hope that all of the tuples of, say, class C1 will fall into one partition and all of the tuples of class C2 will fall into the other partition.
3. The process of determining a split-point is recursively applied to each partition obtained until some stopping criterion is met, such as when the minimum information

requirement on all candidate split-points is less than a small threshold, e, or when the number of intervals is greater than a threshold, max interval.

IV. Interval merging by χ^2 Analysis

Chi Merge, which employs a button-up approach by finding the best neighboring intervals and then merging these to form larger intervals, recursively. The method is supervised in that it uses class information.

Chi Merge 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 precedes recursively until a predefined stopping criterion is met.

V. 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 can be used to generate a concept hierarchy for A by following either a top down splitting strategy or a bottom-up merging strategy, where each cluster forms a node of the concept hierarchy. In the top down splitting strategy, each initial cluster or partition may be further decomposed into several sub clusters, forming a lower level of the hierarchy. In the bottom-up merging strategy, clusters are formed by repeatedly grouping neighboring clusters in order to form higher-level concepts.

VI. Discretization by intuitive partitioning

Although the above discretization methods are useful in the generation of numerical hierarchies, many users would like to see numerical ranges partitioned into relatively uniform, easy-to-read intervals that appear intuitive or natural.

The 3-4-5 rule can be used to segment numerical data into relatively uniform, natural seeming intervals. In general, the rule partitions a given range of data into 3, 4 or 5 relatively equal-width intervals, recursively and level by level, based on the value range at the most significant digit.

The rule is as follows:

- If an interval covers 3,6,7, or 9 distinct values at the most significant digit, then partition the range into 3 intervals(3 equal-width intervals for 3,6, and 9; and 3 intervals in the grouping of 2-3-2 for 7)
- If it covers 2, 4, or 8 distinct values at the most significant digit, then partition the range into 4 equal-width intervals.
- If it covers 1, 5, or 10 distinct values at the most significant digit, then partition the range into 5 equal-width intervals.

The rule can be recursively applied to each interval, creating a concept hierarchy for the given numerical attribute. Real-world data often contain extremely large positive and/or negative outlier values, which could distort any top-down discretization method, based on minimum and maximum data values.

2.6.4 concept Hierarchy Generation for categorical data

Categorical data are discrete data. Categorical attributes have a finite (but possibly large) Number of distinct values, with no ordering among the values. Methods of generating a concept hierarchy for categorical data.

- Specification of a partial ordering of attributes explicitly at the schema level by user or experts: Concept hierarchies for categorical or dimensions typically involve a group of attribute. A user or expert can easily define a concept hierarchy by specifying a partial or total ordering of the attributes at the schema level.
- Specification of a portion of a hierarchy by explicit data grouping: This is essentially the manual definition of a portion of a concept hierarchy. In a large database, it is unrealistic to define an entire concept hierarchy by explicit value enumeration. However, it is realistic to specify explicit groupings for a small portion of intermediate-level data.
- 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 they try to automatically generate the attribute so as to construct a meaningful concept hierarchy