UNIT –II: Syllabus

Data Pre-processing: Data Preprocessing: An Overview,Data Cleaning,Data Integration,Data Reduction,Data Transformation and Data Discretization

UNIT-II

DATA PREPROCESSING

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.

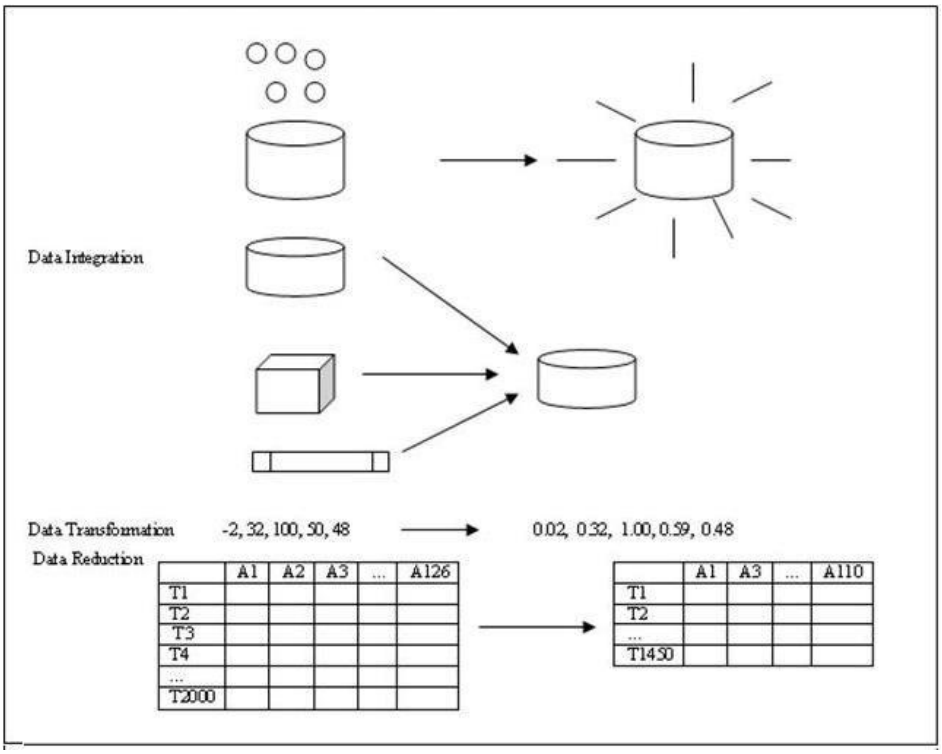
Data Preprocessing Techniques

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

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.



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.

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 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.

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 ofthe 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:

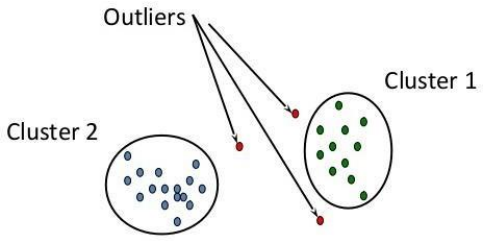
Bin 1: 9,9,9,9
Bin 2: 23,23,23,23
Bin 3: 29,29,29,29

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. Multiple linear regressions 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.



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

a) Discrepancy detection

b) Data transformations

a) Discrepancy detection

The first step in data cleaning is discrepancy detection. It considers the knowledge ofmeta 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.

b) 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)

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

a) Schema integration & object matching.
b) Redundancy.
c) Detection & Resolution of data value conflict

a) 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.

b) 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 and covariance analysis.

For Nominal data, we use the χ^2 (Chi-Square) test.

For Numeric attributes we can use the correlation coefficient and covariance.

χ^2 Correlation analysis for numerical data:

For nominal data, a correlation relationship between two attributes, A and B, can be discovered by a $\chi 2$ (Chi-Square) test. Suppose A has c distinct values, namely a_1, a_2, a_3, ……., a_c. B has r distinct values, namely b_1, b_2, b_3, …., b_r. The data tuples are described by table.

The $\chi 2$ value is computed as

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

Where o_{ij} is the observed frequency of the joint event (A_i,B_j) and e_{ij} is the expected frequency of (A_i,B_j), which can computed as

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

For Example,

	Male	Female	Total
Fiction	250	200	450
Non_Fiction	50	1000	1050
Total	300	1200	1500

$$e_{11} = \frac{count\ male\ X count\ (fiction)}{n} = \frac{300X450}{1500} = 90$$

$$e_{12} = \frac{count\ male\ X count\ (non_fiction)}{n} = \frac{300X1050}{1500} = 210$$

$$e_{21} = \frac{count\ female\ X count\ (fiction)}{n} = \frac{1200\ X450}{1500} = 360$$

$$e = \frac{count\ female\ X count\ (non_fiction)}{n} = \frac{1200\ X1050}{1500} = 840$$

22 n 1500

	Male	Female	Total
Fiction	250 (90)	200 (360)	450
Non_Fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

For χ2 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 X 2 table, the degrees of freedom are (2-1(2-1)=1. For 1 degree of freedom, the x2 value needed to reject the hypothesis at the 0.001 significance level is 10.828(from statistics table). Since our computed value is greater than this, we can conclude that two attributes are strongly correlated for the given group of people.

Correlation Coefficient for Numeric data:

For Numeric attributes, we can evaluate the correlation between two attributes, A and B, by computing the correlation coefficient. This is

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

For Covariance between A and B defined as

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

***c)* Detection and Resolution of Data Value Conflicts.**

A third important issue in data integration is the *detection and resolution of data valueconflicts*. For example, for the same real–world entity, attribute value from differentsources 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.

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.

4. Data Reduction:

Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results.

Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.

Data reduction strategies

4.1.Data cube aggregation

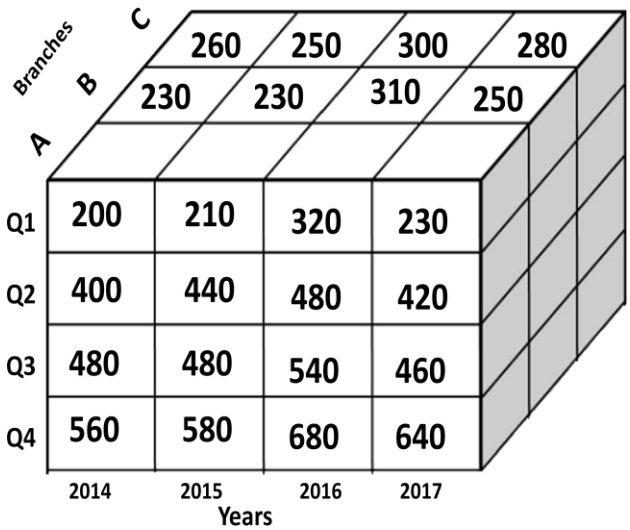
4.2.Attribute Subset Selection

4.3.Numerosity reduction — e.g., fit data into models

4.4.Dimensionality reduction - Data Compression

Data cube aggregation:

For example, the data consists of AllElectronics sales per quarter for the years 2014 to 2017.You are, however, interested in the annual sales, 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.



Year/Quarter	2014	2015	2016	2017
Quarter 1	200	210	320	230
Quarter 2	400	440	480	420
Quarter 3	480	480	540	460
Quarter 4	560	580	680	640

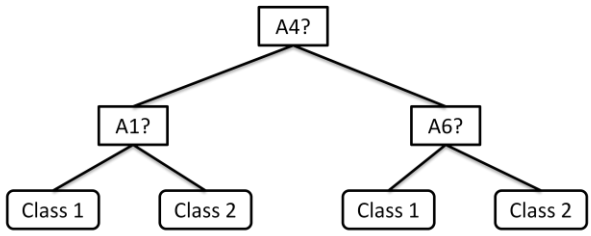
Year	Sales
2014	1640
2015	1710
2016	2020
2017	1750

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.

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.

Initial attribute set: {A1, A2, A3, A4, A5, A6} Initial Reduced Set: ➢ { } ➢ { A1 } ➢ { A1, A4 } ➢ { A1, A4, A6 } Reduced Attribute Set: { A1, A4, A6 }	Initial attribute set: {A1, A2, A3, A4, A5, A6} {A1, A2, A3, A4, A5, A6} {A1, A3, A4, A5, A6} {A1, A4, A5, A6} {A1, A4, A6} Reduced Attribute Set: { A1, A4, A6 }	Initial attribute set: {A1, A2, A3, A4, A5, A6}  Reduced Attribute Set: { A1, A4, A6 }

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.

Numerosity Reduction:

Numerosity reduction is used to reduce 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

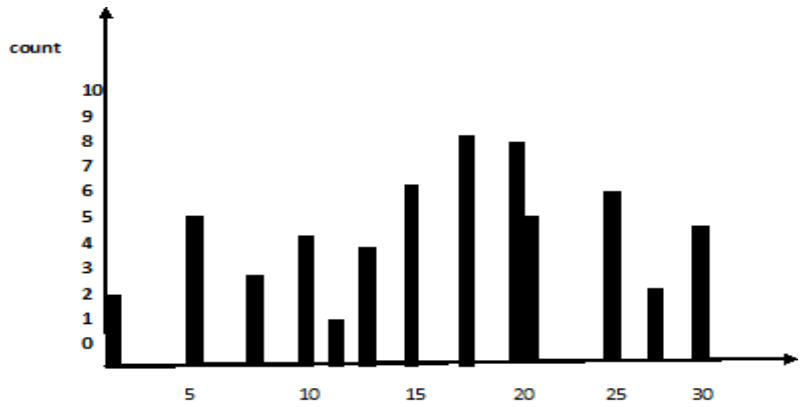


Figure 3.7 A Histogram for price using Singleton Buckets

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

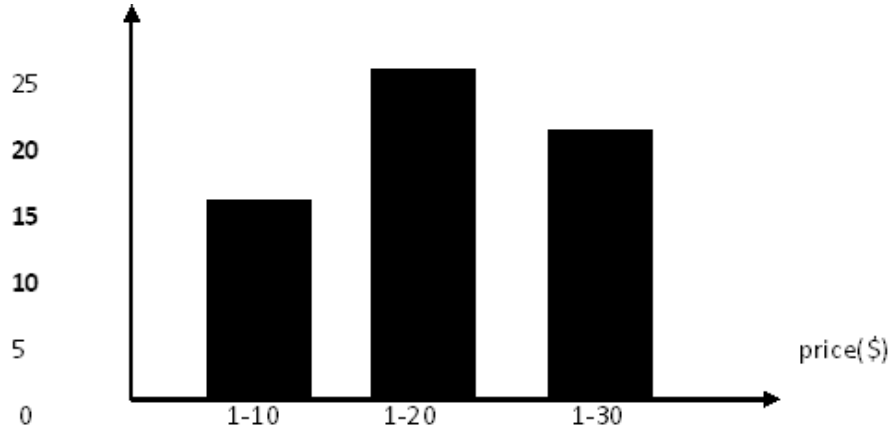


Figure.2.8 An equal-width histogram for price, where values are aggregated so ***that each bucket has a uniform width of $10.***

2. 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.

3. 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 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.

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

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

Figure 2.9. Sampling can be used for data reduction.

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

a) Wavelet transforms

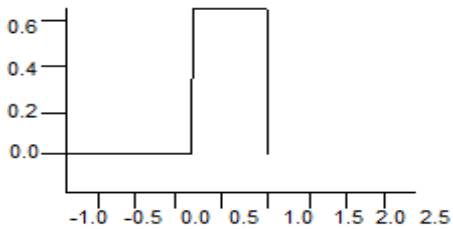
b) Principal components analysis.

a) Wavelet transforms:

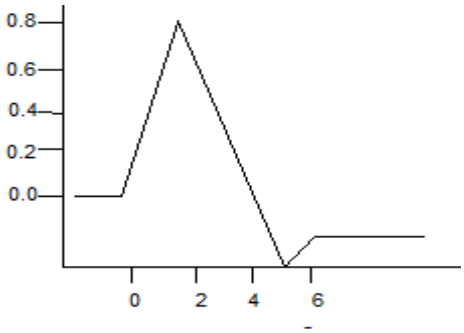
The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector, transforms it to a numerically different vector, of wavelet coefficients. The two vectors are of the same length. When applying this technique to data reduction, we consider each tuple as an n-dimensional data vector, that is, X=(x1,x2,……,xn), 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.

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.



(a)haar2



(b) Daubechies4

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 (X2i , X2i+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.

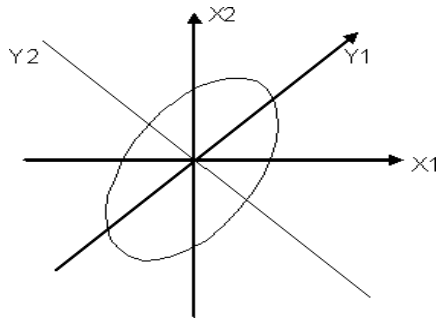
4. The two functions are recursively applied to the sets of data obtained in the previous loop, until the resulting data sets obtained are of length 2.

b) 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.



In the above figure, 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. 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

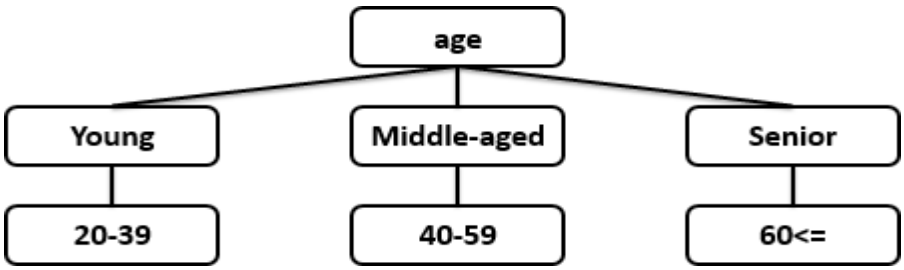
- PCA is computationally inexpensive
- Multidimensional data of more than two dimensions can be handled by reducing the problem to two dimensions.
- Principal components may be used as inputs to multiple regression and cluster analysis.

5. Data Transformation and Discretization

Data transformation is the process of converting data from one format or structure into another format or structure.

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

1. **Smoothing**, which works to remove noise from the data. Techniques include binning, regression, and clustering.
2. **Attribute construction** (or *feature construction*), where new attributes are constructed and added from the given set of attributes to help the mining process.
3. **Aggregation**, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for data analysis at multiple abstraction levels.
4. **Normalization**, where the attribute data are scaled so as to fall within a smaller range, such as -1.0 to 1.0, or 0.0 to 1.0.
5. **Discretization**, where the raw values of a numeric attribute (e.g., *age*) are replaced byinterval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., *youth, adult, senior*).The labels, in turn, can be recursively organized into higher-level concepts, resulting in a *concept hierarchy* for the numeric attribute. Figure 3.12 shows a concept hierarchy for the attribute *price*. More than one concept hierarchy can be defined for the same attribute to accommodate the needs of various users.
6. **Concept hierarchy generation for nominal data**, where attributes such as *street* can be generalized to higher-level concepts, like *city* or *country*. Many hierarchies for nominal attributes are implicit within the database schema and can be automatically defined at the schema definition level.



5.1 Data Transformation by Normalization:

The measurement unit used can affect the data analysis. For example, changing measurement units from meters to inches for *height*, or from kilograms to pounds for *weight*, may lead to very different results.

For distance-based methods, normalization helps prevent attributes with initially large ranges (e.g., *income*) from outweighing attributes with initially smaller ranges (e.g., binary attributes). It is also useful when given no prior knowledge of the data.

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

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

Min-max normalization preserves the relationships among the original data values. It will encounter

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new_max_A - new_min_A) + new_min_A.$$

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

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

b) Z-Score Normalization

The values for an attribute, *A*, are normalized based on the mean (i.e., average) and standard deviation of *A*. A value, v_i, of *A* is normalized to v_i' by computing

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

where*A* and σA are the mean and standard deviation, respectively, of attribute A.

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

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

c) Normalization by Decimal Scaling:

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 vi' 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 917normalizes to 0.917.

5.2. Data Discretization

a) Discretization by 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-frequencybinning, and then replacing each bin value by the bin mean or median, as in *smoothing by bin means* or *smoothing by bin medians*, respectively. These techniques can be applied recursively to the resulting partitions to generate concept hierarchies.

b) Discretization by Histogram Analysis:

Like binning, 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.

In an equal-width histogram, for example, the values are partitioned into equal-size partitions or ranges (e.g., for price, where each bucket has a width of $10).With an equal-frequency histogram, the values are partitioned so that, ideally, each partition contains the same number of data tuples.

c) Discretization by Cluster, Decision Tree, and Correlation Analyses

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.

Techniques to generate decision trees for classification can be applied to discretization. Such techniques employ a top-down splitting approach. Unlike the other methods mentioned so far, decision tree approaches to discretization are supervised, that is, they make use of class label information.

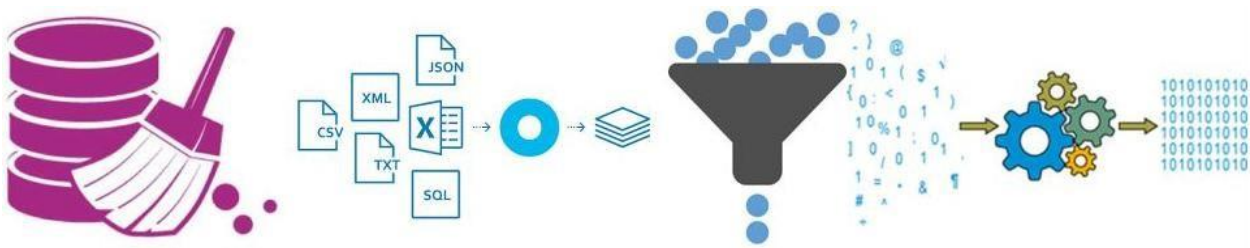
Concept Hierarchy Generation for Nominal Data

Nominal attributes have a finite (but possibly large) number of distinct values, with no ordering among the values. Examples include geographic location, job category, and item type.

Manual definition of concept hierarchies can be a tedious and time-consuming task for a user or a domain expert. Fortunately, many hierarchies are implicit within the database schema and can be automatically defined at the schema definition level. The concept hierarchies can be used to transform the data into multiple levels of granularity.

1. **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 a partial or total ordering of the attributes at the schema level.
2. **Specification of a portion of a hierarchy by explicit data grouping:** In a large database, it is unrealistic to define an entire concept hierarchy by explicit value enumeration. For example, after specifying that province and country form a hierarchy at the schema level, a user could define some intermediate levels manually.
3. **Specification of a set of attributes, but not of their partial ordering:** A user may specify a set of attributes forming a concept hierarchy, but omit to explicitly

State their partial ordering. The system can then try to automatically generate the attribute ordering so as to construct a meaningful concept hierarchy.

4. **Specification of only a partial set of attributes:** Sometimes a user can be careless when defining a hierarchy, or have only a vague idea about what should be included in a hierarchy. Consequently, the user may have included only a small subset of there irrelevant attributes in the hierarchy specification.

- ✓ **Data cleaning** routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.
- ✓ **Data integration** combines data from multiple sources to form a coherent data store. The resolution of semantic heterogeneity, metadata, correlation analysis ,tuple duplication detection, and data conflict detection contribute to smooth data integration.
- ✓ **Data reduction** techniques obtain a reduced representation of the data while minimizing the loss of information content. These include methods of *dimensionality reduction*, *numerosity reduction*, and *data compression*.
- ✓ **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.