DATA MINING AND BIG DATA ANALYTICS (18MCA52C)

UNIT II: Data Pre-processing

FACULTY

Dr. K. ARTHI MCA, M.Phil., Ph.D.,

Assistant Professor,

Postgraduate Department of Computer Applications,

Government Arts College (Autonomous),

Coimbatore-641018.

UNIT II

Data Pre-processing

- There are several data preprocessing techniques.
- Data cleaning can be applied to remove noise and correct inconsistencies in data.
- Data integration merges data from multiple sources into a coherent data store such as a data warehouse.
- Data reduction can reduce data size by aggregating, eliminating redundant features, or clustering.
- Data transformations (e.g., normalization) may be applied, where data are scaled to fall within a smaller range like 0.0 to 1.0.
- This can improve the accuracy and efficiency of mining algorithms involving distance measurements.
- These techniques are not mutually exclusive; they may work together.
- For example, data cleaning can involve transformations to correct wrong data, such as by transforming all entries for a date field to a common format.

Data Quality: Why Preprocess the Data?

- Data have quality if they satisfy the requirements of the intended use.
- There are many factors comprising data quality, including **accuracy, completeness, consistency, timeliness, believability, and interpretability.**
- **D**ata to be analyzed by data mining techniques are:

  incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data);

  inaccurate or noisy (containing errors, or values that deviate from the expected); and

  inconsistent (e.g., containing discrepancies in the department codes used to categorize items).

- Three of the elements defining data quality: accuracy, completeness, and consistency.
- Inaccurate, incomplete, and inconsistent data are common-place properties of large real-world databases and data warehouses.
- There are many possible reasons for inaccurate data (i.e., having incorrect attribute values).
- The data collection instruments used may be faulty. There may have been human or computer errors occurring at data entry.
- Users may purposely submit incorrect data values for mandatory fields when they do not wish to submit personal information (e.g., by choosing the default value “January 1” displayed for birthday). This is known as **disguised missing data.**
- Errors in data transmission can also occur.
- There may be technology limitations such as limited buffer size for coordinating synchronized data transfer and consumption.
- Incorrect data may also result from inconsistencies in naming conventions or data codes, or inconsistent formats for input fields (e.g., date).
- Duplicate tuples also require data cleaning.
- Incomplete data can occur for a number of reasons.
- Attributes of interest may not always be available, such as customer information for sales transaction data.
- Other data may not be included simply because they were not considered important at the time of entry.
- Relevant data may not be recorded due to a misunderstanding or because of equipment malfunctions.
- Data that were inconsistent with other recorded data may have been deleted.
- Furthermore, the recording of the data history or modifications may have been overlooked.
- Missing data, particularly for tuples with missing values for some attributes, may need to be inferred.
- Data quality depends on the intended use of the data.

- Two different users may have very different assessments of the quality of a given database.

- For example, a marketing analyst may need to access the database mentioned before for a list of customer addresses.

- Some of the addresses are outdated or incorrect, yet overall, 80% of the addresses are accurate.

- The marketing analyst considers this to be a large customer database for target marketing purposes and is pleased with the database's accuracy, although, as sales manager, you found the data inaccurate.

- **Timeliness** also affects data quality.

- The fact that the month-end data are not updated in a timely fashion has a negative impact on the data quality.

- Two other factors affecting data quality are **believability and interpretability**.

- Believability reflects how much the data are trusted by users, while interpretability reflects how easy the data are understood.

- Suppose that a database, at one point, had several errors, all of which have since been corrected. The past errors, however, had caused many problems for sales department users, and so they no longer trust the data.

- Even though the database is now accurate, complete, consistent, and timely,sales department users may regard it as of low quality due to poor believability and interpretability.

Major Tasks in Data Preprocessing

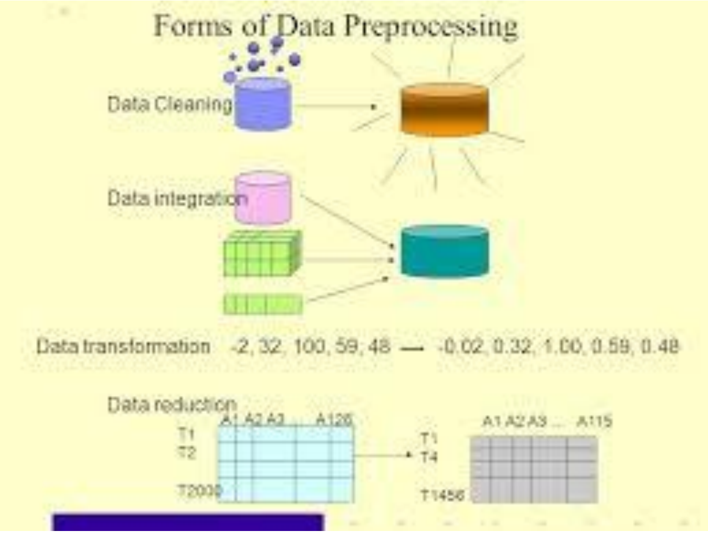
Major steps involved in data preprocessing, namely, data cleaning, data integration, data reduction, and data transformation.

- Data cleaning routines work to "clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies.

- This would involve integrating multiple databases, data cubes, or files (i.e., data integration).

- Data reduction obtains a reduced representation of the data set that is much smaller in volume, yet produces the same (or almost the same) analytical results.

- Data reduction strategies include **dimensionality reduction and numerosity reduction**.
- In dimensionality reduction, data encoding schemes are applied so as to obtain a reduced or “compressed” representation of the original data.
- Examples include data compression techniques (e.g., wavelet transforms and principal components analysis), attribute subset selection (e.g., removing irrelevant attributes), and attribute construction (e.g., where a small set of more useful attributes is derived from the original set).
- In numerosity reduction, the data are replaced by alternative, smaller representations using parametric models (e.g., regression or log-linear models) or nonparametric models (e.g., histograms, clusters, sampling, or data aggregation).
- **Discretization and concept hierarchy generation** can also be useful, where raw data values for attributes are replaced by ranges or higher conceptual levels. For example, raw values for age may be replaced by higher-level concepts, such as youth, adult, or senior.
- Discretization and concept hierarchy generation are powerful tools for data mining in that they allow data mining at multiple abstraction levels. Normalization, data discretization, and concept hierarchy generation are forms of data transformation.



Forms of Data Preprocessing

Data Cleaning

Data integration

Data transformation -2, 32, 100, 59, 48 — -0.02, 0.32, 1.00, 0.59, 0.48

Data reduction

Forms of data preprocessing

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

1. Ignore the tuple:

- This is usually done when the class label is missing (assuming the mining task involves classification).
- This method is not very effective, unless the tuple contains several attributes with missing values.
- It is especially poor when the percentage of missing values per attribute varies considerably.
- By ignoring the tuple, we do not make use of the remaining attributes' values in the tuple.
- Such data could have been useful to the task at hand.

2. Fill in the missing value manually:

In general, this approach is time consuming and may not be feasible given a large data set with many missing values.

3. Use a global constant to fill in the missing value:

- Replace all missing attribute values by the same constant such as a label like "Unknown" or 1.
- If missing values are replaced by, say, "Unknown," then the mining program may mistakenly think that they form an interesting concept, since they all have a value in common—that of "Unknown." Hence, although this method is simple, it is not foolproof.

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

- measures of central tendency, which indicate the "middle" value of a data distribution.

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

5. Use the attribute mean or median for all samples belonging to the same class as the given tuple:

- For example, if classifying customers according to credit risk, we may replace the missing value with the mean income value for customers in the same credit risk category as that of the given tuple.
- If the data distribution for a given class is skewed, the median value is a better choice.

6. Use the most probable value to fill in the missing value:

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

- Methods 3 through 6 bias the data—the filled-in value may not be correct.
- Method 6, however, is a popular strategy. In comparison to the other methods, it uses the most information from the present data to predict missing values.
- By considering the other attributes' values in its estimation of the missing value for income, there is a greater chance that the relationships between income and the other attributes are preserved.

Noisy Data

"What is noise?" Noise is a random error or variance in a measured variable. In we saw how some basic statistical description techniques (e.g., boxplots and scatter plots), and methods of data visualization can be used to identify outliers, which may represent noise.

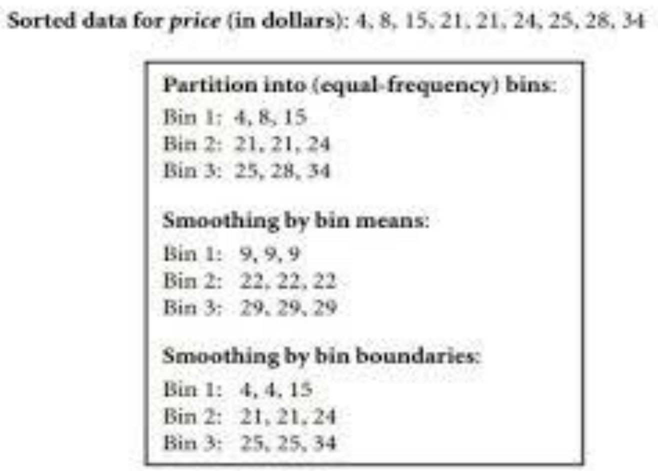
Data smoothing techniques

Binning:

Binning methods smooth a sorted data value by consulting its "neighbor-hood," that is, the values around it. The sorted values are distributed into a number of "buckets," or bins.

Because binning methods consult the neighborhood of values, they perform local smoothing.

Figure 3.2 illustrates some binning techniques.



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

Partition into (equal-frequency) bins:

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

Smoothing by bin means:

Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:

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

In this example, the data for price are first sorted and then partitioned into equal-frequency bins of size 3 (i.e., each bin contains three values). In smoothing by bin means, each value in a bin is replaced by the mean value of the bin. For example, the mean of the

values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced

by the value 9. Similarly, smoothing by bin medians can be employed, in which each bin value is replaced by the bin median.

In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

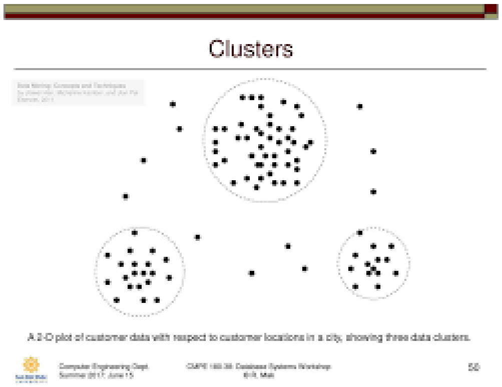
In general, the larger the width, the greater the effect of the smoothing. Alternatively, bins may be equal width, where the interval range of values in each bin is constant. Binning is also used as a discretization technique.

Regression:

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

Outlier analysis:

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



Clusters

Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, and Jian Pei Elsevier, 2011

A 2-D plot of customer data with respect to customer locations in a city, showing three data clusters.

Computer Engineering Dept. Summer 2017: June 15

CMPE 180-38 Database Systems Workshop © R. Mak

50

3.2.3 Data Cleaning as a Process

- Missing values, noise, and inconsistencies contribute to inaccurate data.
- So far, we have looked at techniques for handling missing data and for smoothing data.
- “But data cleaning is a big job. What about data cleaning as a process? How exactly does one proceed in tackling this task? Are there any tools out there to help?”
- The first step in data cleaning as a process is discrepancy detection.
- Discrepancies can be caused by several factors, including poorly designed data entry forms that have many optional fields, human error in data entry, deliberate errors (e.g., respondents not wanting to divulge information about themselves), and data decay (e.g., outdated addresses).
- Discrepancies may also arise from inconsistent data representations and inconsistent use of codes.
- Other sources of discrepancies include errors in instrumentation devices that record data and system errors.
- Errors can also occur when the data are (inadequately) used for purposes other than originally intended.

- There may also be inconsistencies due to data integration (e.g., where a given attribute can have different names in different databases).

"So, how can we proceed with discrepancy detection?"

- As a starting point, use any knowledge you may already have regarding properties of the data. Such knowledge or "data about data" is referred to as metadata.

  As a data analyst, you should be on the lookout for the inconsistent use of codes and

  any inconsistent data representations (e.g., "2010/12/25" and "25/12/2010" for date)

- **Field overloading** is another error source that typically results when developers squeeze new attribute definitions into unused (bit) portions of already defined attributes (e.g.,

  an unused bit of an attribute that has a value range that uses only, say, 31 out of

  32 bits).

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

- A unique rule says that each value of the given attribute must be different from

all other values for that attribute.

- A consecutive rule says that there can be no missing values between the lowest and highest values for the attribute, and that all values must also be unique (e.g., as in check numbers).
- A null rule specifies the use of blanks, question marks, special characters, or other strings that may indicate the null condition (e.g., where a value for a given attribute is not available), and how such values should be handled.

Reasons for missing values may include

(1) the person originally asked to provide a value for the attribute refuses and/or finds

that the information requested is not applicable (e.g., a license number attribute left

blank by nondrivers);

(2) the data entry person does not know the correct value; or

(3) the value is to be provided by a later step of the process.

- The null rule should specify how to record the null condition, for example, such as to store zero for numeric attributes, a blank for character attributes, or any other conventions that may be in use (e.g., entries like “don’t know” or “?” should be transformed to blank).

- There are a number of different commercial tools that can aid in the discrepancy

  detection step**.**

    - **Data scrubbing tools** use simple domain knowledge (e.g., knowledge of postal addresses and spell-checking) to detect errors and make corrections in the data. These tools rely on parsing and fuzzy matching techniques when cleaning data from multiple sources.

- **Data auditing tools** find discrepancies by analyzing the data to discover rules and relationships, and detecting data that violate such conditions. They are variants of data mining tools. For example, they may employ statistical analysis to find correlations, or clustering to identify outliers.

- Commercial tools can assist in the data transformation step.

- **Data migration tools** allow simple transformations to be specified such as to replace the string “gender” by “sex.”

- **ETL (extraction/transformation/loading) t**ools allow users to specify transforms

  through a graphical user interface (GUI).

Data Integration

- Data mining often requires data integration—the merging of data from multiple data

  stores. Careful integration can help reduce and avoid redundancies and inconsistencies in the resulting data set. This can help improve the accuracy and speed of the subsequent

  data mining process.

  3.3.1 Entity Identification Problem

- It is likely that your data analysis task will involve data integration, which combines data

  from multiple sources into a coherent data store, as in data warehousing.

- These sources may include multiple databases, data cubes, or flat files.

- There are a number of issues to consider during data integration.
- Schema integration and object matching can be tricky.
- How can equivalent real-world entities from multiple data sources be matched up?
- This is referred to as the entity identification problem.
- For example, how can the data analyst or the computer be sure that customer id in one database and cust number in another refer to the same attribute? Examples of metadata for each attribute include the name, meaning, data type, and range of values permitted for the attribute, and null rules for handling blank, zero, or null values.
- Such metadata can be used to help avoid errors in schema integration.
- The metadata may also be used to help transform the data (e.g., where data codes for pay type in one database may be "H" and "S" but 1 and 2 in another).
- Hence, this step also relates to data cleaning

3.3.2 Redundancy and Correlation Analysis

- Redundancy is another important issue in data integration.
- An attribute (such as annual revenue, for instance) may be redundant if it can be "derived" from another attribute or set of attributes.
- Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.
- Some redundancies can be detected by correlation analysis.
- Given two attributes, such analysis can measure how strongly one attribute implies the other, based on the available data.

- For nominal data(gender, hair colour), we use (chi-square) test. For numeric attributes, we can use the correlation coefficient and covariance, both of which access how one attribute's values vary from those of another.

  p.95 Correlation Test for Nominal Data

3.3.3 Tuple Duplication

- In addition to detecting redundancies between attributes, duplication should also be detected at the tuple level (e.g., where there are two or more identical tuples for a given unique data entry case).
- The use of denormalized tables (often done to improve performance by avoiding joins) is another source of data redundancy. Inconsistencies often arise between various duplicates, due to inaccurate data entry or updating some but not all data occurrences.
- For example, if a purchase order database contains attributes for the purchaser's name and address instead of a key to this information in a purchaser database, discrepancies can occur, such as the same purchaser's name appearing with different addresses within the purchase order database.

3.3.4 Data Value Conflict Detection and Resolution

- Data integration also involves the detection and resolution of data value conflicts. For example, for the same real-world entity, attribute values from different sources may differ. This may be due to differences in representation, scaling, or encoding.
- For instance, a weight attribute may be stored in metric units in one system and British imperial units in another.
- For a hotel chain, the price of rooms in different cities may involve not only different currencies but also different services (e.g., free breakfast) and taxes.

- When exchanging information between schools, for example, each school may have its own curriculum and grading scheme. One university may adopt a quarter system, offer three courses on database systems, and assign grades from A,C to F, whereas another may adopt a semester system, offer two courses on databases, and assign grades from 1 to 10.
- It is difficult to work out precise course-to-grade transformation rules between the two universities, making information exchange difficult.
- Attributes may also differ on the abstraction level, where an attribute in one sys-tem is recorded at, say, a lower abstraction level than the "same" attribute in another.

3.4 Data Reduction

- Complex data analysis and mining on huge amounts of data can take a long time, making such analysis impractical or infeasible.
- Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data.
- That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results.

3.4.1 Overview of Data Reduction Strategies

- Data reduction strategies include dimensionality reduction, numerosity reduction, and data compression.
- Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration.
- Dimensionality reduction methods include wavelet transforms and principal components analysis which transform or project the original data onto a smaller space.
- Attribute subset selection is a method of dimensionality reduction in which irrelevant,

weakly relevant, or redundant attributes or dimensions are detected and removed.

- Numerosity reduction techniques replace the original data volume by alternative,smaller forms of data representation.
- These techniques may be parametric or non-parametric. For parametric methods, a model is used to estimate the data, so that typically only the data parameters need to be stored, instead of the actual data. (Outliers may also be stored.)
- Regression and log-linear models are examples.
- Nonparametric methods for storing reduced representations of the data include histograms, clustering, sampling, and data cube aggregation.
- In data compression, transformations are applied so as to obtain a reduced or "compressed" representation of the original data.
- If the original data can be reconstructed from the compressed data without any information loss, the data reduction is called lossless.
- If, instead, we can reconstruct only an approximation of the original data, then the data reduction is called lossy.
- There are several lossless algorithms for string compression; however, they typically allow only limited data manipulation.
- Dimensionality reduction and numerosity reduction techniques can also be considered forms of data compression.
- There are many other ways of organizing methods of data reduction. The computational time spent on data reduction should not outweigh or "erase" the time saved by mining on a reduced data set size.

3.4.2 Wavelet Transforms

- The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector X, transforms it to a numerically different vector, X, of wavelet coefficients. The two vectors are of the same length.

“How can this technique be useful for data reduction if the wavelet transformed data are of the same length as the original data?”

- The usefulness lies in the fact that the wavelet transformed data can be truncated. A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients.
- For example, all wavelet coefficients larger than some user-specified threshold can be retained. All other coefficients are set to 0.
- The resulting data representation is therefore very sparse, so that operations that can take advantage of data sparsity are computationally very fast if performed in wavelet space.
- The technique also works to remove noise without smoothing out the main features of the data, making it effective for datacleaning as well.
- Given a set of coefficients, an approximation of the original data can be constructed by applying the inverse of the DWT used.
- 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 version will provide a more accurate approximation of the original data.
- Hence, for an equivalent approximation, the DWT requires less space than the DFT.
- Unlike the DFT, wavelets are quite localized in space, contributing to the conservation of local detail.
- There is only one DFT, yet there are several families of DWTs.
- Figure 3.4 shows some wavelet families. Popular wavelet transforms include the Haar-2, Daubechies-4, and Daubechies-6.

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

The method is as follows:

1.The length, L, of the input data vector must be an integer power of 2. This condition

can be met by padding the data vector with zeros as necessary (L n).

2. Each transform involves applying two functions. The first applies some data smoothing, such as a sum or weighted average. The second performs a weighted difference, which acts to bring out the detailed features of the data.

3. The two functions are applied to pairs of data points in X, that is, to all pairs of

measurements.x2i, x2i. This results in two data sets of length L=2. In general,

these represent a smoothed or low-frequency version of the input data and the high-frequency content of it, respectively.

4. The two functions are recursively applied to the data sets obtained in the previous

loop, until the resulting data sets obtained are of length 2.

5. Selected values from the data sets obtained in the previous iterations are designated

the wavelet coefficients of the transformed data.

- Equivalently, a matrix multiplication can be applied to the input data in order to

obtain the wavelet coefficients, where the matrix used depends on the given DWT.

- The matrix must be orthonormal, meaning that the columns are unit vectors and are mutu-ally orthogonal, so that the matrix inverse is just its transpose.

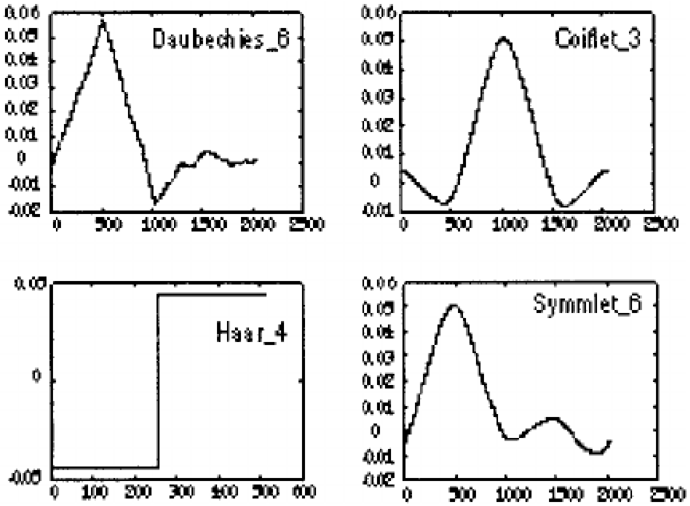
- this property allows the reconstruction of the data from the smooth and smooth-difference data sets.

- By factoring the matrix used into a product of a few sparse matrices, the resulting “fast DWT” algorithm has a complexity of O.n/ for an input vector of length n.

- Wavelet transforms can be applied to multidimensional data such as a data cube.

- This is done by first applying the transform to the first dimension, then to the second, and so on.

- The computational complexity involved is linear with respect to the number of cells in the cube.
- Wavelet transforms give good results on sparse or skewed data and on data with ordered attributes.
- Lossy compression by wavelets is reportedly better than JPEG compression, the current commercial standard.
- Wavelet transforms have many real-world applications, including the compression of fingerprint images, computer vision, analysis of time-series data, and data cleaning.



Examples of wavelet families. The number next to a wavelet name is the number of vanishing moments of the wavelet. This is a set of mathematical relationships that the coefficients must satisfy and is related to the number of coefficients.

3.4.3 Principal Components Analysis

- principal components analysis as a method of dimesionality reduction.
- Suppose that the data to be reduced consist of tuples or data vectors described

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

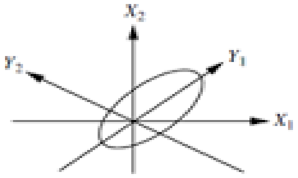
The basic procedure is as follows:

1. The input data are normalized, so that each attribute falls within the same range. This step helps ensure that attributes with large domains will not dominate attributes with smaller domains.

2. PCA computes k orthonormal vectors that provide a basis for the normalized input data. These are unit vectors that each point in a direction perpendicular to the others. These vectors are referred to as the principal components. The input data are a linear combination of the principal components.

3. The principal components are sorted in order of decreasing “significance” or strength. The principal components essentially serve as a new set of axes for the data, providing important information about variance. That is, the sorted axes are such that the first axis shows the most variance among the data, the second axis shows the next highest variance, and so on.

For example, Figure shows the first two princi-pal components, Y1 and Y2, for the given set of data originallymapped to the axes X1 and X2. This information helps identify groups or patterns within the data.



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

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

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

3.4.4 Attribute Subset Selection

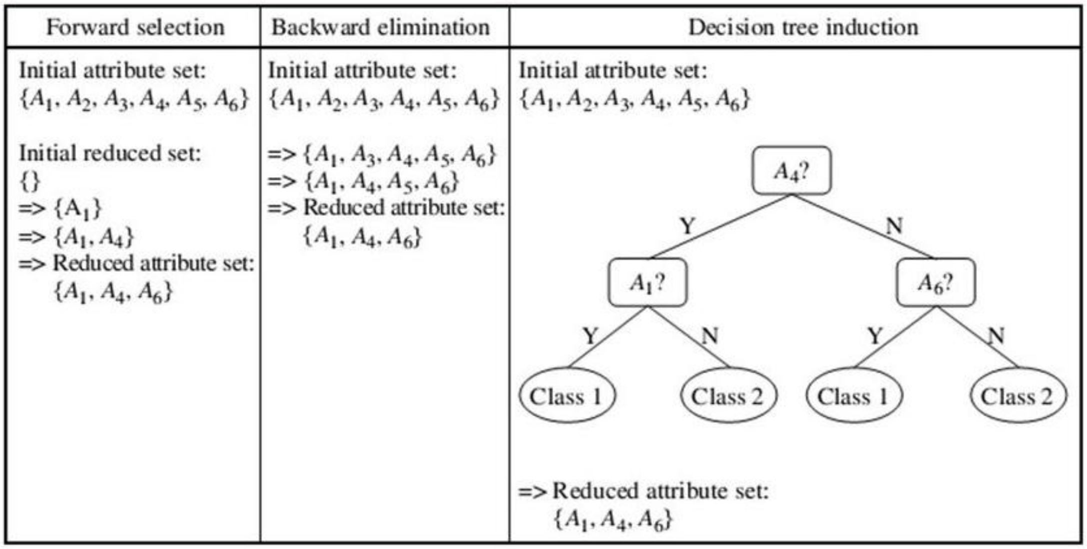
- Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task or redundant.
- Attribute subset selection reduces the data set size by removing irrelevant or redundant attributes (or dimensions).
- The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.
- Mining on a reduced set of attributes has an additional benefit: It reduces the number of attributes appearing in the discovered patterns, helping tomake the patterns easier to understand.

“How can we find a ‘good’ subset of the original attributes?”

- For n attributes, there are 2^n possible subsets.
- An exhaustive search for the optimal subset of attributes can be prohibitively expensive, especially as n and the number of data classes increase.
- Therefore, heuristic methods that explore a reduced search space are commonly used for attribute subset selection.

- These methods are typically greedy in that, while searching through attribute space, they always make what looks to be the best choice at the time.
- Their strategy is to make a locally optimal choice in the hope that this will lead to a globally optimal solution.
- Such greedy methods are effective in practice and may come close to estimating an optimal solution.
- The “best” (and “worst”) attributes are typically determined using tests of statistical significance, which assume that the attributes are independent of one another.
- Many other attribute evaluation measures can be used such as the information gain measure used in building decision trees for classification.
- Basic heuristic methods of attribute subset selection include the techniques that

  follow, some of which are illustrated in Figure



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

A_4?
Y
N
A_1?
A_6?
Y
N
Y
N
Class 1
Class 2
Class 1
Class 2

Greedy (heuristic) methods for attribute subset selection.

THANK YOU

This content is taken from the text books and reference books prescribed in the syllabus.