Data Preprocessing Techniques for Data Mining

Introduction

Data pre-processing is an often neglected but important step in the data mining process. The phrase "Garbage In, Garbage Out" is particularly applicable to data mining and machine learning. Data gathering methods are often loosely controlled, resulting in out-of-range values (e.g., Income: -100), impossible data combinations (e.g., Gender: Male, Pregnant: Yes), missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis.

If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data pre-processing includes cleaning, normalization, transformation, feature extraction and selection, etc. The product of data pre-processing is the final training set.

Data Pre-processing Methods

Raw data is highly susceptible to noise, missing values, and inconsistency. The quality of data affects the data mining results. In order to help improve the quality of the data and, consequently, of the mining results raw data is pre-processed so as to improve the efficiency and ease of the mining process. Data preprocessing is one of the most critical steps in a data mining process which deals with the preparation and transformation of the initial dataset. Data preprocessing methods are divided into following categories:

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

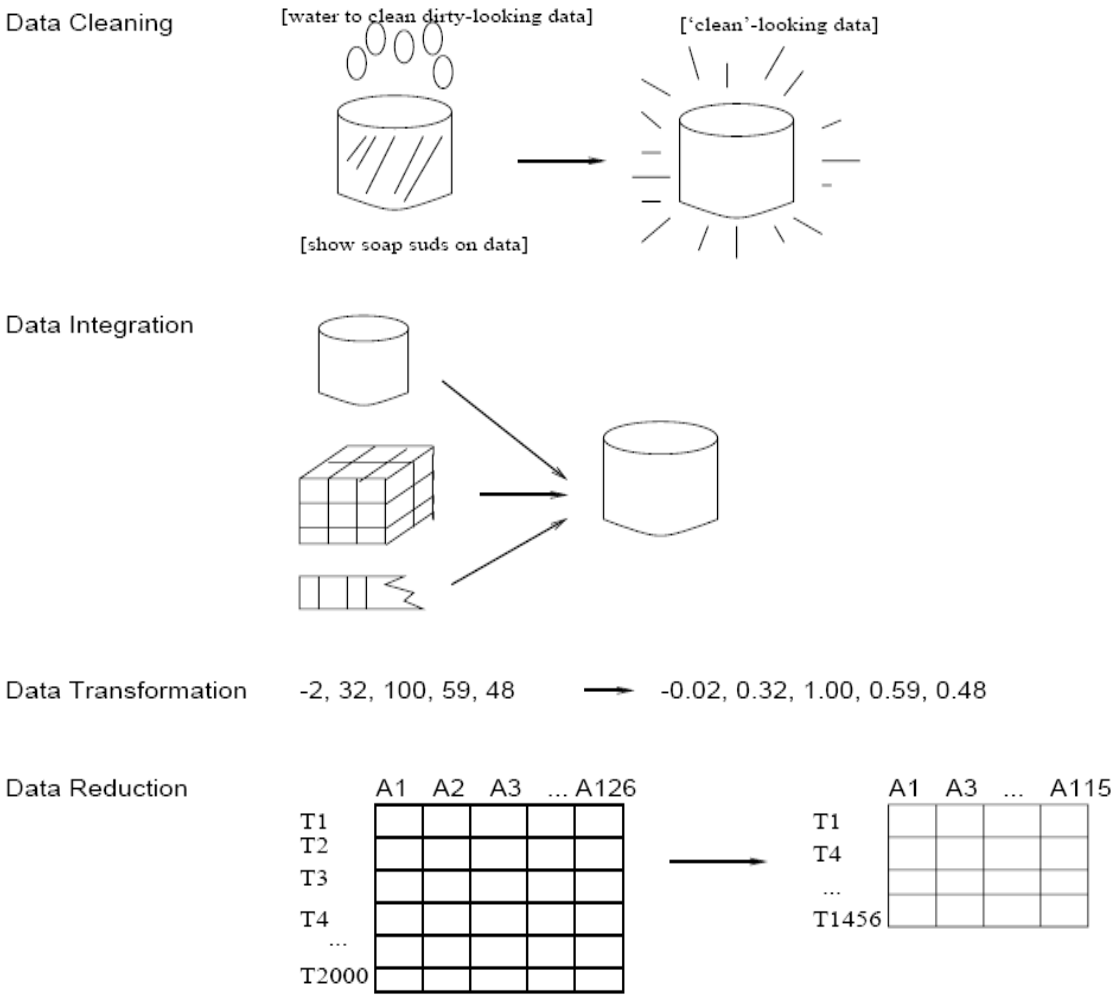

Figure 1: Forms of Data Preprocessing

Data Cleaning

Data that is to be analyze by data mining techniques can be incomplete (lacking attribute values or certain attributes of interest, or containing only aggregate data), noisy (containing errors, or outlier values which deviate from the expected), and inconsistent (e.g., containing discrepancies in the department codes used to categorize items).Incomplete, noisy, and inconsistent data are commonplace 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, such as customer information for sales transaction data. Other data may not be included simply because it was 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 history or modifications to the data may have been overlooked. Missing data, particularly for tuples with missing values for some attributes, may need to be inferred. Data can be noisy, having incorrect attribute values, owing to the following. 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. Incorrect data may also result

from inconsistencies in naming conventions or data codes used. Duplicate tuples also require data cleaning. Data cleaning routines work to “clean" the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. Dirty data can cause confusion for the mining procedure. Although most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust. Instead, they may concentrate on avoiding over fitting the data to the function being modelled. Therefore, a useful pre-processing step is to run your data through some data cleaning routines.

Missing Values: If it is noted that there are many tuples that have no recorded value for several attributes, then the missing values can be filled in for the attribute by various methods described below:

1. *Ignore the tuple*: This is usually done when the class label is missing (assuming the mining task involves classification or description). 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.
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 -∞. 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 recommended.
4. *Use the attribute mean to fill in the missing value*
5. *Use the attribute mean for all samples belonging to the same class as the given tuple.*
6. *Use the most probable value to fill in the missing value*: This may be determined with inference-based tools using a Bayesian formalism or decision tree induction.

Methods 3 to 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.

Noisy Data: Noise is a random error or variance in a measured variable. Given a numeric attribute such as, say, price, how can the data be “smoothed" to remove the noise? The following data smoothing techniques describes this.

1. **Binning methods**: Binning methods smooth a sorted data value by consulting the \neighborhood", or 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 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.
2. **Clustering:** Outliers may be detected by clustering, where similar values are organized into groups or \clusters".

3. **Combined computer and human inspection**: Outliers may be identified through a combination of computer and human inspection. In one application, for example, an information-theoretic measure was used to help identify outlier patterns in a handwritten character database for classification. The measure's value reflected the \surprise" content of the predicted character label with respect to the known label. Outlier patterns may be informative (e.g., identifying useful data exceptions, such as different versions of the characters \0" or \7"), or \garbage" (e.g., mislabeled characters). Patterns whose surprise content is above a threshold are output to a list. A human can then sort through the patterns in the list to identify the actual garbage ones.

   This is much faster than having to manually search through the entire database. The garbage patterns can then be removed from the (training) database.

4. **Regression:** Data can be smoothed by fitting the data to a function, such as with regression. Linear regression involves finding the \best" line to fit two variables, so that one variable can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two variables are involved and the data are fit to a multidimensional surface. Using regression to find a mathematical equation to fit the data helps smooth out the noise.

Inconsistent data: There may be inconsistencies in the data recorded for some transactions. Some data inconsistencies may be corrected manually using external references. For example, errors made at data entry may be corrected by performing a paper trace. This may be coupled with routines designed to help correct the inconsistent use of codes. Knowledge engineering tools may also be used to detect the violation of known data constraints. For example, known functional dependencies between attributes can be used to find values contradicting the functional constraints.

Data Integration

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 can be tricky. How can like 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 entity? Databases and data warehouses typically have metadata - that is, data about the data. Such metadata can be used to help avoid errors in schema integration. Redundancy is another important issue. An attribute may be redundant if it can be \derived" from another table, such as annual revenue. Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

Data Transformation

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

1. *Normalization*, where the attribute data are scaled so as to fall within a small specified range, such as -1.0 to 1.0, or 0 to 1.0.

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

3. *Aggregation*, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.

4. Generalization of the data, where low level or 'primitive' (raw) data are replaced by higher level concepts through the use of concept hierarchies. For example, categorical attributes, like street, can be generalized to higher level concepts, like city or county. Similarly, values for numeric attributes, like age, may be mapped to higher level concepts, like young, middle-aged, and senior.

Data Reduction

Complex data analysis and mining on huge amounts of data may take a very long time, making such analysis impractical or infeasible. Data reduction techniques have been helpful in analyzing reduced representation of the dataset without compromising the integrity of the original data and yet producing the quality knowledge. The concept of data reduction is commonly understood as either reducing the volume or reducing the dimensions (number of attributes). There are a number of methods that have facilitated in analyzing a reduced volume or dimension of data and yet yield useful knowledge. Certain partition based methods work on partition of data tuples.That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results

Strategies for data reduction include the following.

1. *Data cube aggregation*, where aggregation operations are applied to the data in the construction of a data cube.

2. *Dimension reduction*, where irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.

3. *Data compression*, where encoding mechanisms are used to reduce the data set size. The methods used for data compression are wavelet transform and Principal Component Analysis.

4. *Numerosity reduction*, where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need store only the model parameters instead of the actual data e.g. regression and log-linear models), or nonparametric methods such as clustering, sampling, and the use of histograms.

5.Discretization and concept hierarchy generation, where raw data values for attributes are replaced by ranges or higher conceptual levels. Concept hierarchies allow the mining of data at multiple levels of abstraction, and are a powerful tool for data mining.

Conclusion

Data preparation is an important issue for both data warehousing and data mining, as real-world data tends to be incomplete, noisy, and inconsistent. Data preparation includes data cleaning, data integration, data transformation, and data reduction. Data cleaning routines can be used to fill in missing values, smooth noisy data, identify outliers, and correct data inconsistencies. Data integration combines data from multiples sources to form a coherent data store. Metadata, correlation analysis, data conflict detection, and the resolution of semantic heterogeneity contribute towards smooth data integration. Data transformation routines conform the data into appropriate forms for mining. For example, attribute data may be normalized so as to fall between a small range, such as 0 to 1.0. Data reduction techniques such as data cube aggregation, dimension reduction, data compression, numerosity reduction, and discretization can be used to obtain a reduced representation of the data, while minimizing the loss of information content. Concept hierarchies organize the values of attributes or dimensions into gradual levels of abstraction. They are a form a discretization that is particularly useful in multilevel mining. Automatic generation of concept hierarchies for categoric data may be based on the number of distinct values of the attributes defining the hierarchy. For numeric data, techniques such as data segmentation by partition rules, histogram analysis, and clustering analysis can be used. Although several methods of data preparation have been developed, data preparation remains an active and important area
of research.