Data cleaning

Data is real world is dirty and which means incompleteness, noisy and inconsistency. Data cleaning is the process of filling the missing values, smoothing out the noise and identifying outliers and correcting the inconsistency.

Missing values:

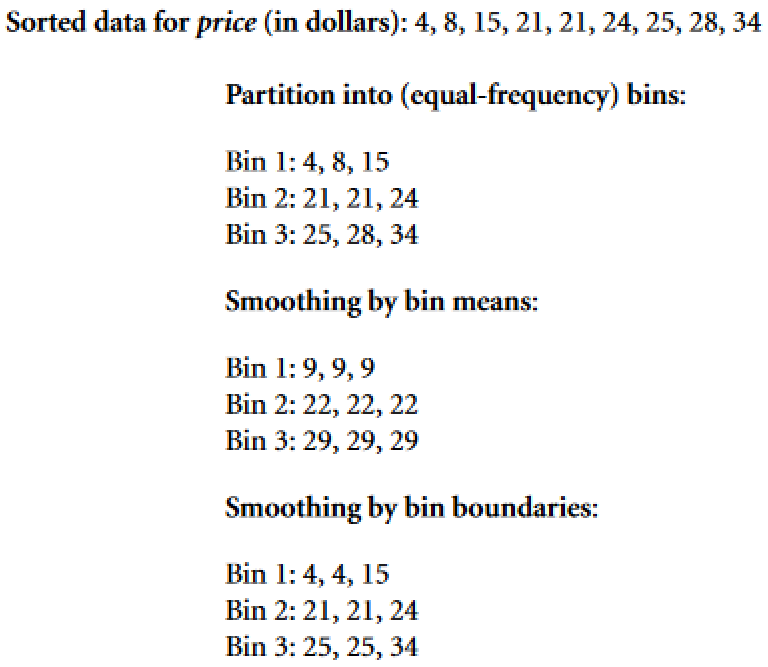
Some of the attribute for data object might not contain the recorded value due to unavailability or human error. Following measures can be applied to solve the missing value problem:

1. Ignore the tuple
2. Fill in the missing value manually
3. Use a global constant to fill in the missing value (“Unknown”, “TBD”, -∞)
4. Use global attribute mean
5. Use attribute mean for all the samples belonging to same class
6. Use the most probable value (regression)

Noisy Data:

Noise is a random error or variance in a measured variable. Here are the strategies to smooth out the noise:

1. Binning: Binning method smooth a sorted value by consulting its neighbor. The data objects are sorted and distributed in buckets called bins. We can then allot bin mean, bin median or bin boundaries to smooth out the attribute.



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

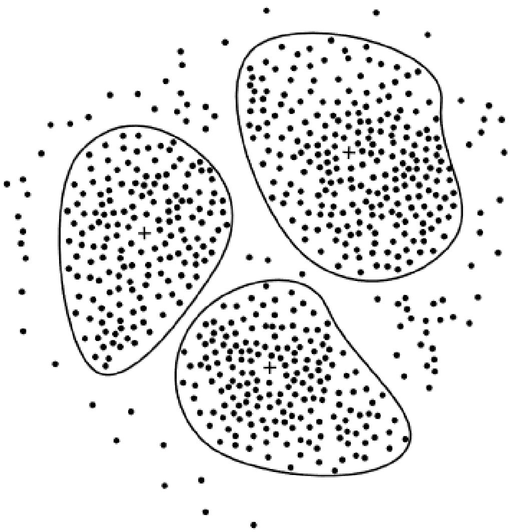
Smoothing by bin means:

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

2. 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 attributes (or variables).
3. 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



Data integration and transformation

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 multiple issues that needs to be considered during data integration. Some of them are:

- Schema integration
- Object Matching
- Redundancy (mostly seen in case of derived attributes)

Data Transformation:

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

- **Smoothing**, which works to remove noise from the data. Such techniques include binning, regression, and clustering.
- **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.
- **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 country. Similarly, values for numerical attributes, like age, may be mapped to higher-level concepts, like youth, middle-aged, and senior.
- **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.

Min-max normalization

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

$$v' = \frac{v - \mathrm{min}_a}{maxA - minA}(new\_max_{A} - new\_min_{B}) + new\_min_{A}$$

Z-score Normalization:

In z-score normalization, attribute A are normalized based on the mean and standard deviation of A. a value v of A is normalized to v' by computing:

$$v' = \frac{v - \mu_a}{\sigma_a}$$

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

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.

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 reduce the data set size.

4. **Numerosity reduction**

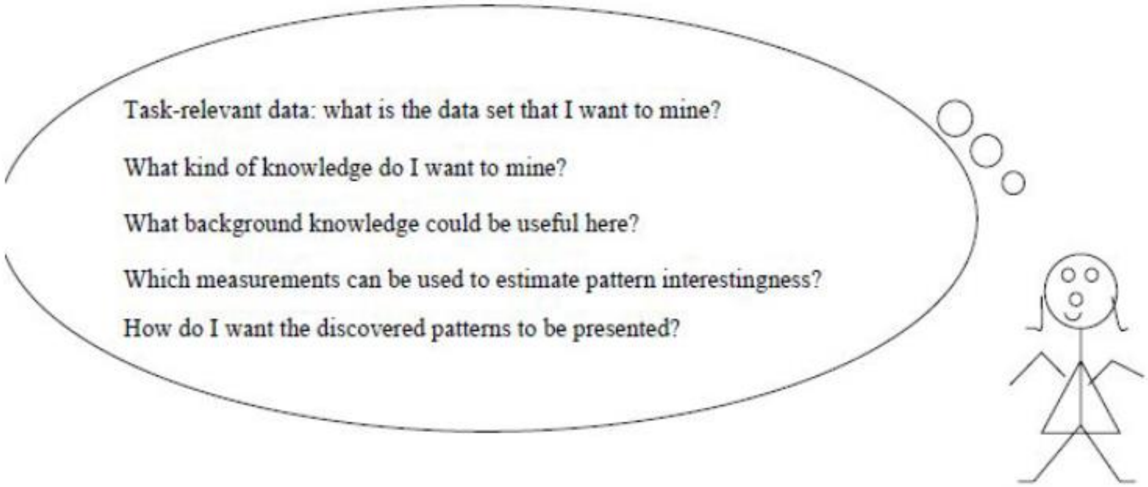
where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need store only the model parameters instead of the actual data) or nonparametric methods such as clustering, sampling, and the use of histograms.

5. **Discretization and concept hierarchy generation**

where raw data values for attributes are replaced by ranges or higher conceptual levels. Data discretization is a form of numerosity reduction that is very useful for the automatic generation of concept hierarchies. Discretization and concept hierarchy generation are powerful tools for data mining, in that they allow the mining of data at multiple levels of abstraction.

Data mining primitives

A data mining query is defined in terms of the following primitives



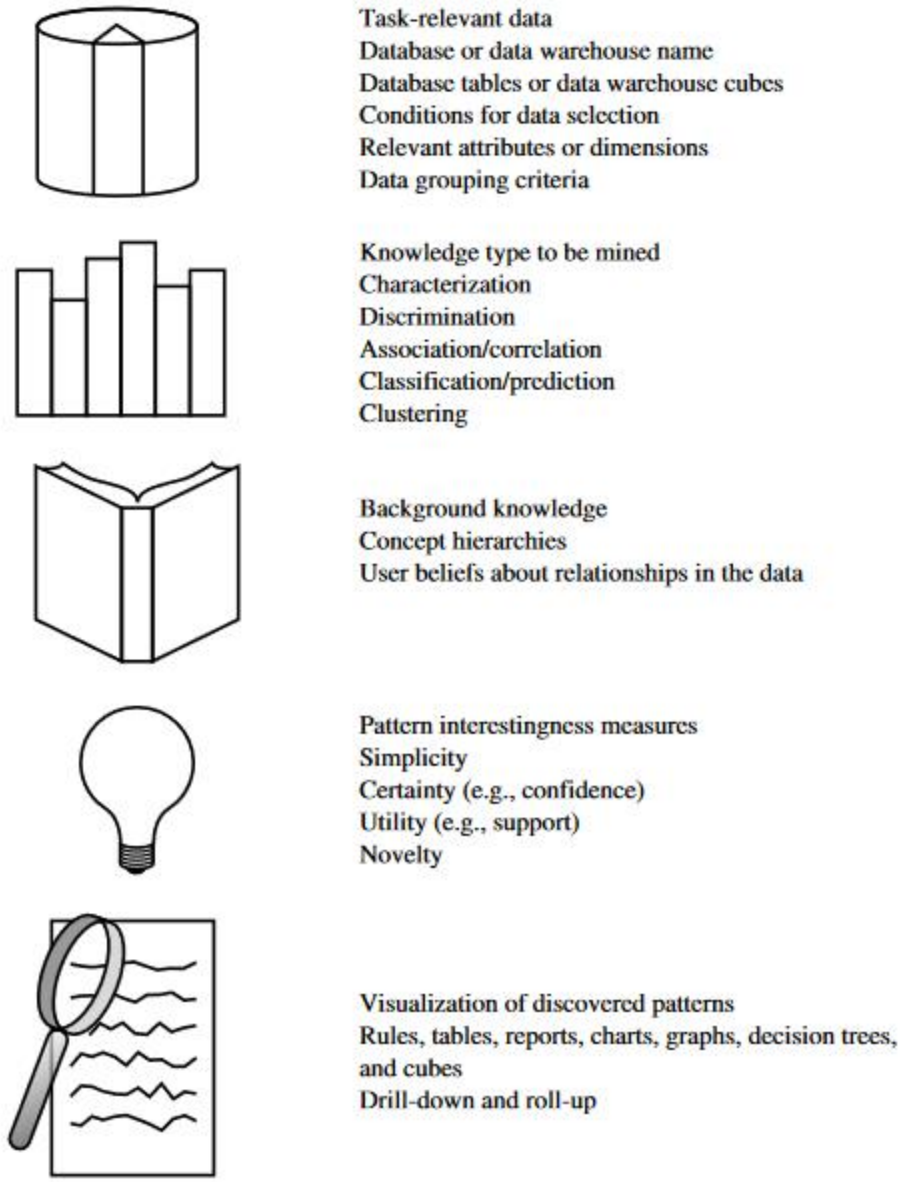
Task-relevant data: what is the data set that I want to mine?

What kind of knowledge do I want to mine?

What background knowledge could be useful here?

Which measurements can be used to estimate pattern interestingness?

How do I want the discovered patterns to be presented?



Task-relevant data
Database or data warehouse name
Database tables or data warehouse cubes
Conditions for data selection
Relevant attributes or dimensions
Data grouping criteria

Knowledge type to be mined
Characterization
Discrimination
Association/correlation
Classification/prediction
Clustering

Background knowledge
Concept hierarchies
User beliefs about relationships in the data

Pattern interestingness measures
Simplicity
Certainty (e.g., confidence)
Utility (e.g., support)
Novelty

Visualization of discovered patterns
Rules, tables, reports, charts, graphs, decision trees,
and cubes
Drill-down and roll-up

Task - Relevant Data

This is the database portion to be investigated

The kinds of knowledge to be mined

This specifies the data mining functions to be performed, such as characterization, discrimination, association, classification

Background Knowledge

Users can specify background knowledge, or knowledge about the domain to be mined. This knowledge is useful for guiding the knowledge discovery process, and for evaluating the patterns found. A popular form of background knowledge is concept hierarchies

Interestingness Measures

These functions are used to separate uninteresting patterns from knowledge. They may be used to guide the mining process, or after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include support (the percentage of task-relevant data tuples for which the rule pattern appears), and confidence (the strength of the implication of the rule)

Presentation and Visualization of Discovered Patterns

This refers to the form in which discovered patterns are to be displayed. Users can choose from different forms for knowledge presentation, such as rules, tables, charts, graphs, decision trees, and cubes