Data Mining: Data Preprocessing

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  - noisy: containing errors or outliers
  - inconsistent: containing discrepancies in codes or names
- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data

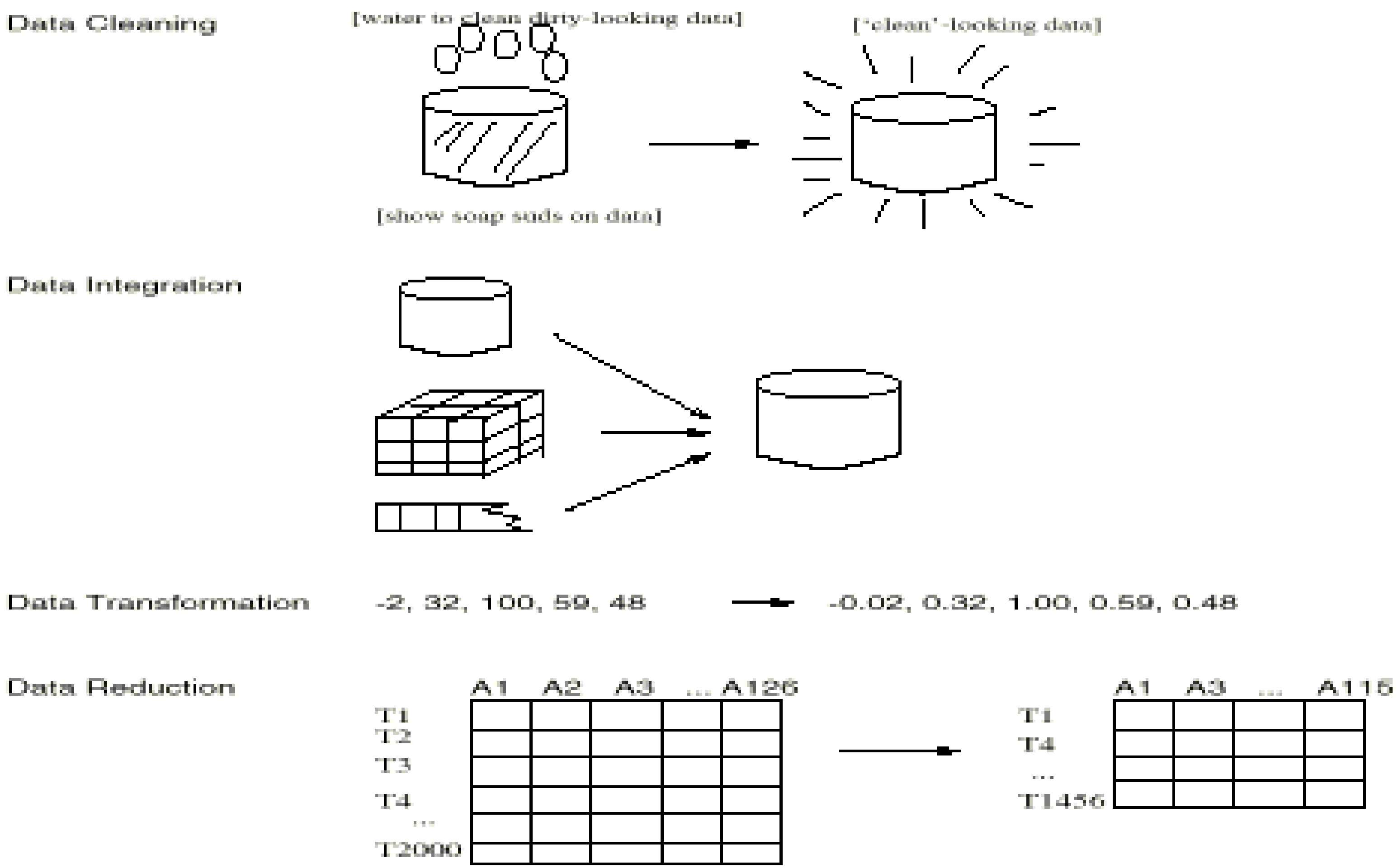
Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility

Major Tasks in Data Preprocessing

- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data

Forms of data preprocessing



Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Data Cleaning

- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data

Missing Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred.

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably)
- Fill in the missing value manually: tedious + infeasible?
- Use a global constant to fill in the missing value: e.g., "unknown", a new class?!
- Use the attribute mean to fill in the missing value
- Use the most probable value to fill in the missing value: inference-based such as Bayesian formula or decision tree

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which requires data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

How to Handle Noisy Data?

- Binning method:
  - first sort data and partition into (equi-depth) bins
  - then smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human
- Regression
  - smooth by fitting the data into regression functions

Simple Discretization Methods: Binning

- Equal-width (distance) partitioning:
  - It divides the range into N intervals of equal size: uniform grid
  - if A and B are the lowest and highest values of the attribute, the width of intervals will be: $W = (B-A)/N$.
  - The most straightforward
  - But outliers may dominate presentation
  - Skewed data is not handled well.
- Equal-depth (frequency) partitioning:
  - It divides the range into N intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky.

Binning Methods for Data Smoothing

* Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
* Partition into (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
  - Bin 3: 26, 26, 26, 34

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Data Integration

- Data integration:
  - combines data from multiple sources into a coherent store.
  - Careful integration can help reduce and avoid redundancies and inconsistencies in resulting data set.
  - This can help improve the accuracy and speed of the subsequent data mining process.

Data Integration

- 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?

Data Integration

- When **matching attributes** from one database to another during integration, special attention must be paid to the structure of the data.
- For example, in one system, a discount may be applied to the order, whereas in another system it is applied to each individual line item within the order.
- If this is not caught before integration, items in the target system may be improperly discounted.

Handling Redundant Data

- Redundant data occur often when integration of multiple databases
  - The same attribute may have different names in different databases Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Data Transformation

Strategies for data transformation are:

- **Smoothing:** remove noise from data
- **Attribute Construction**: new attributes are constructed
- **Aggregation:** summarization, data cube construction
- **Generalization:** concept hierarchy climbing
- **Normalization:** scaled to fall within a small, specified range like 0.0 to 1.0

  Discretization: raw values of a numeric attribute (e.g., age) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior).

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.
- To help avoid dependence on the choice of measurement units, the data should be **normalized or standardized**.
- This involves transforming the data to fall within a smaller or common range such as [−1,1] or [0.0, 1.0].

Data Transformation by Normalization

Methods for Normalization:

- min-max normalization

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

- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- normalization by decimal scaling

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max(| v' |)<1

Example

Example

Data Transformation by Normalization

Let A be the numeric attribute with n observed values v1, v2………..vn

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

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

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

Data Transformation by Normalization

In **z-score normalization** (or *zero-mean 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

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

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

Data Transformation by Normalization

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, v_i, of A is normalized to v_i' by computing

$$v_i' = \frac{v_i}{10^j},$$

where j is the smallest integer such that $max(|v_i'|) < 1$.

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 917 normalizes to 0.917. ■

Discretization

- The raw values of a numeric attribute (e.g., age) are replaced by interval labels (e.g., 0–10, 11–20, etc.) or conceptual labels (e.g., youth, adult, senior).
- Three types of attributes:
  - Nominal — values from an unordered set
  - Ordinal — values from an ordered set
  - Continuous — real numbers
- Discretization:
  - divide the range of a continuous attribute into intervals
  - Some classification algorithms only accept categorical attributes.
  - Reduce data size by discretization
  - Prepare for further analysis

Discretization

- Discretization techniques can be categorized based on how the discretization is performed, such as whether it uses **class information** or which direction it proceeds (i.e., **top-down vs. bottom-up**).
- If the discretization process uses class information, then we say it is **supervised discretization**.
- If the process starts by first finding one or a few points to split the entire attribute range, and then repeats this recursively on the resulting intervals, it is called **top-down discretization or splitting**.

Discretization

- This contrasts with **bottom-up discretization or merging**, which starts by considering all of the continuous values as potential split-points, removes some by merging neighborhood values to form intervals, and then recursively applies this process to the resulting intervals.

Data Reduction Strategies

- Warehouse may store terabytes of data: Complex data, so analysis/mining may take a very long time to run on the complete data set
- **Data reduction**
  - Obtains a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- **Data reduction strategies**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression

Dimensionality Reduction

- Feature selection (i.e., attribute subset selection):
  - Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration.
  - It 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

- Numerosity reduction techniques replace the original data volume by alternative, smaller forms of data representation.
  - These techniques may be **parametric or nonparametric**.
  - 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.)
  - **Nonparametric methods** for storing reduced representations of the data include histograms, clustering, sampling etc

Data Compression

- 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.
- Dimensionality reduction and numerosity reduction techniques can also be considered forms of data compression.

Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Discretization and Concept hierachy

- Discretization
  - reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values.
- Concept hierarchies
  - reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior).

Discretization for numeric data

- Binning
- Histogram analysis
- Clustering analysis