



DATA PREPROCESSING



Introduction

Real-world DBs are highly noisy, missing and inconsistent data
Preprocessing techniques are used to Improve the quality, efficiency and mining results

- **Data Cleaning:** remove noise and correct the inconsistencies in data
- **Data Integration:** merges data from multiple sources into coherent data store such as a data warehouse
- **Data Reduction:** reduce data size by performing aggregations, eliminating redundant features and clustering
- **Data Transformations:** 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 measures

Why Preprocess the Data?

- **Data Quality:** Data have quality if they satisfy the requirements of the intended use.
- Many factors comprising data quality
 - Accuracy
 - Completeness
 - Consistency
 - Timeliness
 - Believability
 - Interpretability

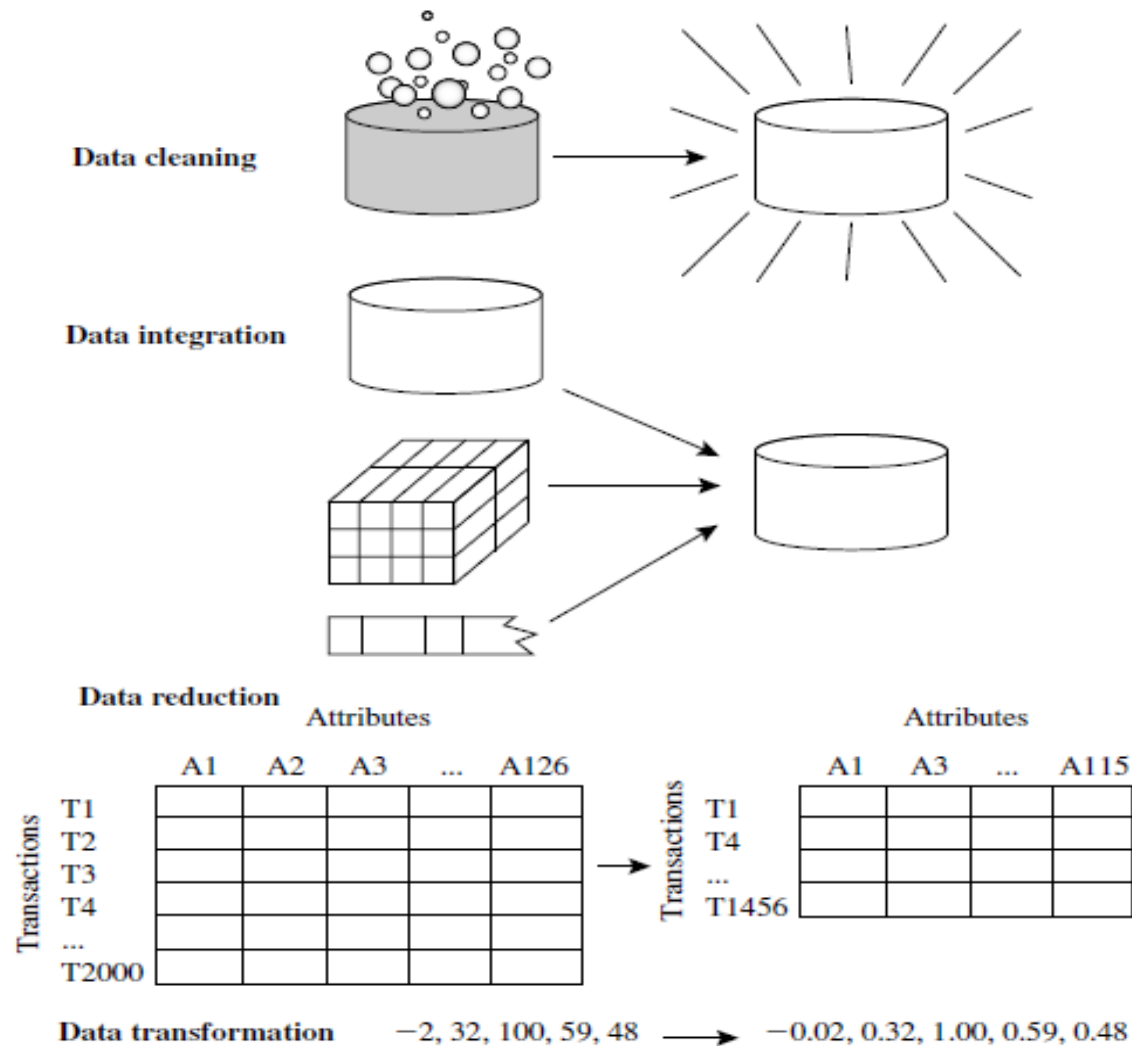
Cont.,

- Several attributes for various tuples have no record values, that is called missing data that reduces the quality by reporting errors, unusual values and inconsistencies
- The data you wish to analyze by the DM techniques are
 - Incomplete (lacking attribute values or containing aggregate data)
 - Inaccurate or noisy (having incorrect attr values that deviate from the expected)
 - Inconsistent (contains discrepancies in the dept codes used to categorize items)

Major Tasks in Data Preprocessing

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

Data Preprocessing



Data Cleaning

- Data cleaning routines attempt to
 - Fill in missing values
 - smooth out noise while identifying outliers
 - correct inconsistencies in the data
 - Resolve redundancy caused by data integration

Missing Values

1. Ignore the tuple,
2. Fill in the missing value manually- time consuming may not be feasible for large datasets
3. Use a global constant to fill in the missing value- like *unknown* or ∞
4. Use a measure of central tendency- mean for symmetric data and median for skewed data
5. Use the attribute mean or median for all samples belonging to the same class as the given tuple
6. Use the most probable value to fill in the missing value
 - Determined with regression, inference based tools using a Bayesian formalism or decision tree induction

Noisy Data

- Noise is a random error or variance in a measured variable
 - Boxplots and scatter plots are used to identifying the outliers which may represent noise
 - Ex: attribute “*price*” , we have to smooth out the data to remove noise

Smoothing techniques

- **Binning:** smooth the sorted data value by consulting its neighbourhood i.e. the values around it
- The sorted values are distributed into a no.of buckets or bins and the perform local smoothing

- Smooth by Bin mean
- Smooth by bin medians
- Smooth by bin boundaries(min and max values are as bin boundaries)

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

- **Regression:** is a technique that conforms data values to a function
- **Linear Regression** involves finding the **best** line to fit two attributes 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 data are fit to multidimensional surface

- **Outlier analysis:** detected by clustering
- **Ex:** similar values are organized into groups or clusters, values outside of the set of clusters are considered as

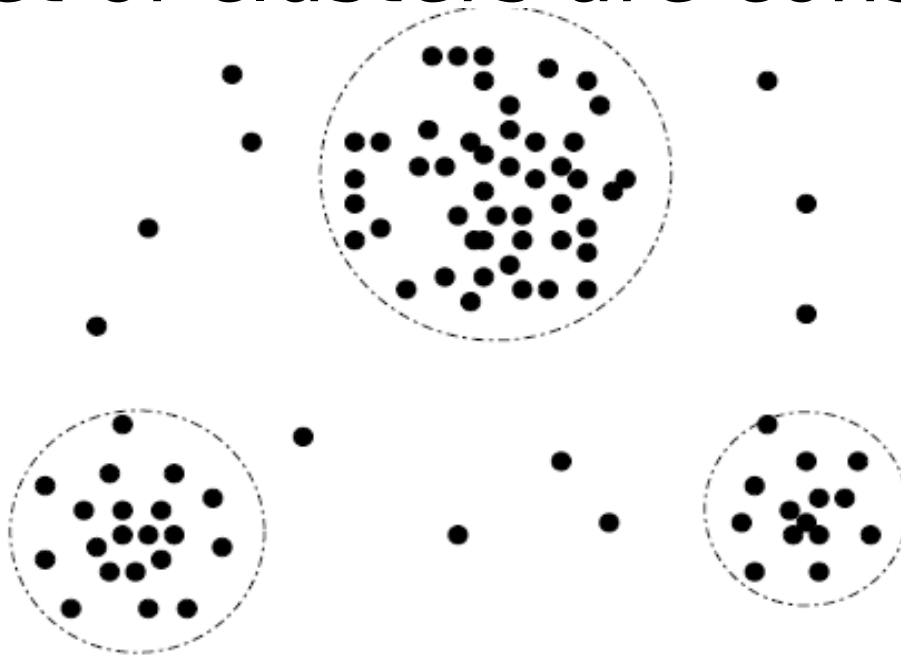


Figure 3.3 A 2-D customer data plot with respect to customer locations in a city, showing three data clusters. Outliers may be detected as values that fall outside of the cluster sets.

Data Integration

- The semantic **heterogeneity and structure of the data** pose great challenges in the data integration
 - Entity identification problem – How can we match schema and objects from different sources?
 - Correlation tests on numeric data and nominal data – Specifies the correlation between objects

Entity Identification Problem

- No. of issues are considered during data integration
 - Schema integration and object matching can be tricky
- The equivalent of real world entities from multiple data sources is known as **entity identity problem**
 - Ex: Different representations and Different scales like cust_id and customer_id how they are referred
 - Metadata for each attribute (name, meaning, datatype, range of values permitted for the attribute, null rules for handling blanks, zeros)
 - Such metadata may also be used to help avoid errors in schema integration

Redundancy and Correlation Analysis

- An attribute may be redundant if it can be derived from another attribute.
- Careful integration of data from multiple sources may help/avoid redundancies
- Some redundancies can be detected by correlation analysis
- Given two attributes, correlation analysis can measure how strongly one attribute implies the other, based on the available data
 - For Nominal data, we use Chi-Square test
 - For Numeric attribute, we use Correlation Coefficient and Covariance

χ^2 Correlation data for Nominal data

- Correlation relationship between two attributes, A and B, can be discovered by Chi-Square test
- Suppose A has c distinct values, namely $a_1, a_2, \dots, a_c$ B has r distinct values, namely $b_1, b_2, \dots, b_r$
- The data tuples described by contingency table, with the **c** values of A making up the **columns**, **r** values of B making up the **rows**.
- Chi-Square test is computed as :

- $$Chi - Square = \sum_{i=1}^c \sum_{j=1}^r \frac{(O_{ij} - e_{ij})^2}{e_{ij}}$$
 - O_{ij} is Observed frequency(actual count) of the joint event (A_i, B_j)
 - e_{ij} is Expected frequency

- $e_{ij} = \frac{\text{count}(A=a_i) \times \text{count}(B=b_j)}{n}$
 - n is the total number of data tuples
- Chi-Square statistic tests the hypothesis that A and B are independent, i.e., there is no correlation b/w them
- The test is based on the significant level, with $(r-1) \times (c-1)$ degree of freedom
- In terms of a p-value and a chosen significance level (α), the test can be interpreted as follows:
 - **If p-value $\leq \alpha$:** significant result, reject null hypothesis (H_0), dependent.
 - **If p-value $> \alpha$:** not significant result, fail to reject null hypothesis (H_0), independent.

Correlation Coefficient for Numeric Data

- Also known as Pearson's product moment coefficient

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

- n is the number of tuples
- a_i and b_i are respective values of A and B in tuple i
- $\bar{A}$ and $\bar{B}$ are respective mean values of A and B
- σA and σB are respective standard deviation of A and B
- $\sum(a_i, b_i)$ is the sum of the AB cross-product
- Note:** $-1 \leq r_{A,B} \leq +1$
- If $r_{A,B}$ is greater than 0, then A and B are Positively Correlated (values of A increase as the values of B increase)
- If the resulting value = 0, then A and B are independent, and there is no correlation between them
- If the resulting value is < 0 , then A and B are Negatively correlated (values of A increase as the values of B decrease)

Covariance of Numeric Data

- In P&S, correlation and covariance are similar measures for assessing how much two attributes change together
- Consider two numeric attr X and Y, and a set of n observations

$$COV(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1}$$

- If **COV(A,B)** is greater than 0, then A and B are Positively Correlated (values of A increase as the values of B increase)
- If the resulting value = 0, then A and B are independent, and there is no correlation between them
- If the resulting value is < 0, then A and B are Negatively correlated (values of A increase as the values of B decrease)

DATA TRANSFORMATION

- **Smoothing:** To remove noise from the data (Techniques include binning, regression and clustering)
- **Attribute construction (or feature construction):** New attributes are constructed from the given set of attributes to help the mining process.
- **Aggregation:** Aggregation operations are applied to the data. Ex- the daily sales data may be aggregated so as to compute monthly and annual total amounts.
- **Normalization:** 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.
- **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*). The labels, in turn, can be recursively organized into higher-level concepts, resulting in a *concept hierarchy* for the numeric attribute.
- **Concept hierarchy generation for nominal data:** 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

Data Transformation by Normalization

- Changing the measurement unit of attribute values/units is called normalization
- Transforming the data to fall within a smaller range such as $[-1,1]$ or $[0.0,1.0]$
- Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as clustering
- Data Normalization methods
 - Min-max normalization
 - Z-score normalization
 - Normalization by Decimal Scaling

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 preserves the relationships among the original data values.
- Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to

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

Z(Zero) - Score Normalization

- The values for an attribute A , are normalized based on mean and standard deviation of A .
- A value, v_i , of A *normalized* to v'_i by computing

$$v' = \frac{v - \mu_A}{\sigma_A}$$

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 *normalized* to v'_i by computing

$$v' = \frac{v}{10^j}$$

- Where j is the smallest integer, $\text{Max}(|v|)$

Data Reduction

- To obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (almost 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
 - Dimensionality reduction, e.g., reduces the no.of random variables or attributes
 - Wavelet transforms
 - Principal Components Analysis (PCA)
 - Attribute subset selection
 - Numerosity reduction, Replace the original data volume by alternative or smaller forms
 - Parametric methods: Regression and Log-Linear Models
 - Non - parametric methods: Histograms, clustering, sampling
 - Data compression:
 - Lossy

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 such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.
- It reduces the number of attributes appearing in the discovered patterns, helping to make 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.

Greedy(Heuristic) methods for Attribute Subset Selection

- **Stepwise forward selection:**
 - starts with an empty set of attributes as the reduced set.
 - Best of the 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.
- **Stepwise backward elimination:**
 - Starts with the full set of attributes.
 - At each step, it removes the worst attribute remaining in the set.
- **Combination of forward selection and backward elimination:**
 - The stepwise forward selection and backward elimination methods can be combined
 - At each step, the procedure selects the best attribute and removes the worst from among the remaining attributes.
- **Decision tree induction:**
 - Decision tree algorithms (e.g., ID3, C4.5, and CART) were originally intended for classification.
 - Decision tree induction constructs a flow chart like structure where each internal (nonleaf) node denotes a test on an attribute, each branch corresponds to an outcome of the test, and each external (leaf) node denotes a class prediction.
 - At each node, the algorithm chooses the “best” attribute to partition the data into individual classes.
- When decision tree induction is used for attribute subset selection, 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 form the reduced subset of attributes.

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: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ <pre> graph TD A4["A4?"] -- Y --> A1["A1?"] A4 -- N --> A6["A6?"] A1 -- Y --> C1_1("Class 1") A1 -- N --> C2_1("Class 2") A6 -- Y --> C1_2("Class 1") A6 -- N --> C2_2("Class 2") </pre> $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$

Greedy (heuristic) methods for attribute subset selection.

Thank You