Chapter 2: Data Preprocessing

CSE 435:Data Mining

Md. Atikuzzaman
Lecturer
Department of Computer Science & Engineering
Green University of Bangladesh
atik@cse.green.edu.bd

Table of Contents

Why Preprocess Data? The Reality of Raw Data

Real-world data is often messy. Analyzing poor-quality data leads to poor-quality results. This is known as the **"Garbage In, Garbage Out" (GIGO)** principle.

Data Quality Dimensions

Data has high quality if it meets the requirements of its intended use. Key dimensions include:

- **Accuracy:** Are the values correct?
- **Completeness:** Is the data all there?
- **Consistency:** Is the data consistent across different sources?
- **Timeliness:** Is the data up-to-date?
- **Believability:** How much do we trust the data?
- **Interpretability:** Is the data understandable?

Major Tasks in Data Preprocessing

Data preprocessing is a set of techniques used to convert raw data into a clean, consistent, and usable format.

- **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 reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - **incomplete:** lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., `Occupation=""` (missing data)
  - **noisy:** containing noise, errors, or outliers
    - e.g., `Salary="-10"` (an error)
  - **inconsistent:** containing discrepancies in codes or names, e.g.,
    - `Age="42"`, `Birthday="03/07/2010"`
    - Was rating "1, 2, 3", now rating "A, B, C"
    - discrepancy between duplicate records
  - **Intentional** (e.g., *disguised missing data*)
    - Jan. 1 as everyone's birthday?

Incomplete (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?

It's common for data to have missing values. How do we handle them?

- **Ignore the Tuple:** Effective only if the dataset is large and the number of missing tuples is small.
- **Fill in the Value Manually:** Practical only for very small datasets.
- **Use a Global Constant:** Replace missing values with a constant like "Unknown" or $-\infty$. Can be misleading.
- **Use a Measure of Central Tendency:**
  - For numerical data, replace with the attribute **mean** or **median**.
  - For categorical data, replace with the attribute **mode**.
- **Use the Most Probable Value:** Use techniques like regression or a decision tree to predict and fill in the missing value.

Data Cleaning - Noisy Data

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

How to Handle Noisy Data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

Data Cleaning - Noisy Data(Binning)

Binning

This method smooths sorted data values by consulting their "neighborhood."

1. **Sort** the data.
2. **Partition** into equal-frequency bins.
3. **Smooth** using bin means, medians, or boundaries.

Example: Binning

Consider sorted prices: 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into 3 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} (mean is 9)
- Bin 2: {22, 22, 22} (mean is 22)
- Bin 3: {29, 29, 29} (mean is 29)

Data Integration

Data integration involves merging data from multiple sources. Key challenges include:

- **Entity Identification:** Matching records for the same real-world entity.
- **Redundancy and Correlation:** Detecting if an attribute can be derived from another.

For Nominal Data: Chi-Square (χ^2) Test

This test assesses if two attributes are independent.

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

Where o_{ij} is the observed frequency and e_{ij} is the expected frequency. A large χ^2 value indicates the attributes are related.

Use Cases of the Chi-Square (χ^2) Test

The χ^2 test is fundamental for understanding associations between nominal (categorical) attributes.

Feature Selection

- **Goal:** To determine if a categorical feature is relevant for predicting a categorical target variable.
- **Example:** In a model to predict customer churn (`Churn: Yes/No`), we can use a χ^2 test to see if there is a statistically significant association between 'Churn' and other categorical features like 'Contract Type' (Monthly, Yearly) or 'Payment Method' (Bank Transfer, Credit Card).
- **Outcome:** Features showing a strong association (a high χ^2 value) are likely good predictors and should be kept for model training.

Use Cases of the Chi-Square (χ^2) Test

Test of Independence in EDA

- **Goal:** To discover relationships between different attributes during exploratory data analysis.
- **Example:** A retail company operating in Bangladesh could test if a customer's 'Preferred Product Category' (e.g., Electronics, Apparel, Groceries) is independent of their 'Location' (e.g., Dhaka, Chittagong, Sylhet).
- **Outcome:** Finding that the variables are *not* independent can reveal key business insights, such as specific product categories being unusually popular in certain regions.

Chi-Square (χ^2) Test: An Example

Are Gender and Ice Cream Preference independent?

Observed Counts (o_{ij}):

	Chocolate	Vanilla	**Total**
Male	40	20	**60**
Female	30	70	**100**
Total	**70**	**90**	**160**

Expected Counts (e_{ij}): $e_{ij} = \frac{\text{row total} \times \text{column total}}{\text{grand total}}$

- $e_{11} = (60 \times 70)/160 = 26.25$
- $e_{12} = (60 \times 90)/160 = 33.75$
- $e_{21} = (100 \times 70)/160 = 43.75$
- $e_{22} = (100 \times 90)/160 = 56.25$

Chi-Square (χ^2) Test: An Example

$$\chi^2 = \frac{(40-26.25)^2}{26.25} + \frac{(20-33.75)^2}{33.75} + \frac{(30-43.75)^2}{43.75} + \frac{(70-56.25)^2}{56.25}$$

$$\chi^2 = 7.25 + 5.63 + 4.35 + 3.39 = \mathbf{20.62}$$

With degrees of freedom

$$df = (r-1) \times (c-1)$$

$(2-1)(2-1) = 1$, the critical value at $\alpha = 0.05$ is 3.84. Since $20.62 > 3.84$, we reject the null hypothesis and conclude that Gender and Ice Cream Preference are dependent.

df	$\alpha = 0.10$	$\alpha = \mathbf{0.05}$	$\alpha = 0.01$
1	2.71	**3.84**	6.63
2	4.61	5.99	9.21
3	6.25	7.81	11.34

Data Integration

For Numeric Data: Correlation Coefficient (Pearson's)

Measures the linear relationship between two attributes, A and B.

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

A value near $+1$ indicates a strong positive correlation, -1 a strong negative correlation, and 0 no linear correlation.

Use Cases of Pearson's Correlation Coefficient

Pearson's correlation coefficient (r) is essential for understanding the linear interplay between continuous attributes.

Detecting Multicollinearity

- **Goal:** To identify if two or more predictor variables in a regression model are highly correlated.
- **Example:** In a model to predict house prices, features like 'Square Footage' and 'Number of Rooms' are likely to have a very high positive correlation ($r \approx +0.9$).
- **Outcome:** High multicollinearity can make regression models unstable. By identifying it, we can decide to remove one of the highly correlated features, thereby improving the model's reliability. This is a form of data reduction.

Use Cases of Pearson's Correlation Coefficient

Feature Selection

- **Goal:** To find which numeric features have the strongest linear relationship with a numeric target variable.
- **Example:** An agricultural analyst studying crop yields in Rupganj wants to predict 'Paddy Yield (kg/acre)'. They can calculate the correlation between the yield and features like 'Annual Rainfall (mm)', 'Fertilizer Used (kg)', and 'Average Temperature'.
- **Outcome:** Features with a high absolute correlation value (e.g., $|r| > 0.6$) are often the most powerful predictors for linear models.

Correlation Coefficient: An Example

Are Study Hours and Test Scores correlated? Let's analyze data for 5 students.

- Study Hours (A): $\{2, 4, 5, 6, 8\}$
- Test Score (B): $\{60, 70, 75, 80, 95\}$

1. Calculate Means:

- $\bar{A} = (2 + 4 + 5 + 6 + 8)/5 = 5$
- $\bar{B} = (60 + 70 + 75 + 80 + 95)/5 = 76$

2. Calculate Standard Deviations (σ):

- $\sigma_A = \sqrt{\frac{(2-5)^2 + \ldots + (8-5)^2}{5-1}} = 2.236$
- $\sigma_B = \sqrt{\frac{(60-76)^2 + \ldots + (95-76)^2}{5-1}} = 13.454$

Correlation Coefficient: An Example

3. Calculate Covariance Term:

- $\sum(a_i - \bar{A})(b_i - \bar{B}) = (-3)(-16) + (-1)(-6) + (0)(-1) + (1)(4) + (3)(19) = 48 + 6 + 0 + 4 + 57 = 115$

4. Calculate Pearson's Coefficient (r):

$$r_{A,B} = \frac{115}{(5-1) \times 2.236 \times 13.454} = \frac{115}{120.22} \approx \mathbf{0.957}$$

The result of 0.957 indicates a very strong positive linear correlation between study hours and test scores.

Task 3: Data Reduction

The goal is to get a reduced representation of the data that is much smaller but produces nearly the same analytical results.

Strategies

- **Dimensionality Reduction:** Reduces the number of attributes.
  - **Principal Component Analysis (PCA):** Finds new orthogonal axes (principal components) that capture maximum variance.
  - **Wavelet Transforms:** Decomposes a signal into significant frequency components.
- **Numerosity Reduction:** Replaces data with a smaller representation.
  - **Parametric:** Store model parameters instead of data (e.g., Regression).
  - **Non-parametric:** Use methods like histograms, clustering, or sampling.

Dimensionality Reduction: Principal Component Analysis (PCA)

PCA is a technique to reduce the dimensionality of a dataset while preserving as much 'variability' (i.e., information) as possible.

The Goal of PCA

Transform a set of correlated variables into a smaller set of uncorrelated variables called **principal components**. The first principal component accounts for the most variance, the second for the next most, and so on.

The PCA Algorithm in Brief

1. **Standardize:** Scale all features to have zero mean and unit variance.
2. **Covariance Matrix:** Compute the covariance matrix (Σ)
3. **Eigendecomposition:** Calculate eigenvectors and eigenvalues
4. **Select Components:** Choose the top k eigenvectors.
5. **Transform:** Project the original data onto the selected components to get the new, reduced feature space.

Applications of Principal Component Analysis (PCA)

PCA is a versatile technique used across many fields for data exploration, visualization, and preprocessing.

- **Data Visualization:** Visually identify patterns in high-dimensional data by reducing it to 2D or 3D. For instance, a bank in Dhaka can plot complex customer data with many features (income, age, etc.) onto a simple scatter plot to discover distinct market segments.
- **Improve Model Performance:** Combat the "curse of dimensionality" by reducing the number of input features. This speeds up model training and helps prevent overfitting. A garment factory in Rupganj, for example, could transform 50 correlated sensor readings into a few principal components to build a faster and more robust machine failure prediction model.
- **Image Compression and Noise Reduction:** Save storage space and clean up data by isolating the most important information (the signal) from the noise. This is highly effective for compressing satellite imagery of agricultural lands in the Dhaka Division, retaining key features while discarding redundant data.

Task 4: Data Transformation

This involves changing the data into a format that is more suitable for mining.

Common Transformation Strategies

- **Smoothing:** Removing noise from data.
- **Aggregation:** Summarizing data (e.g., aggregating daily sales to monthly totals).
- **Attribute Construction:** Creating new, more useful attributes from existing ones (e.g., creating ‘area‘ from ‘height‘ and ‘width‘).
- **Normalization:** Scaling attribute values to fall within a specified range.

Data Discretization

Discretization transforms continuous attributes into nominal ones by dividing their range into a finite number of intervals.

Discretization Methods

- **Binning:** Can be equal-width or equal-frequency.
- **Clustering Analysis:** Groups values into a predefined number of clusters (intervals).
- **Concept Hierarchy Generation:** For categorical data, this involves grouping low-level concepts into higher-level ones.
  - Example: Rupganj $\subset$ Dhaka Division $\subset$ Bangladesh

Summary

Preprocessing is a critical, foundational step in the data mining process.

- **Data Cleaning** fixes imperfections like missing values and noise.
- **Data Integration** unifies data from different sources.
- **Data Reduction** makes massive datasets manageable.
- **Data Transformation & Discretization** standardizes and prepares data for mining algorithms.

Investing time in preprocessing ensures that your mining results are **valid, accurate, and meaningful**.

References

[1] Jiawei Han, Micheline Kamber, & Jian Pei, *Data Mining: Concepts and Techniques*, 4th Edition, Morgan Kaufmann, 2012.

[2] David J. Hand, Heikki Mannila, & Padhraic Smyth, *Principles of Data Mining*, First Edition, A Bradford Book, 2001.

[3] Richard O. Duda, Peter E. Hart, & David G. Stork, *Pattern Classification*, 2nd Edition, Wiley, 2001.