Unit 3
Data Preprocessing

Prepared By

Arjun Singh Saud, Asst. Prof. CDCSIT

Data Preprocessing Concept

- Raw data collected from the environment can be dirty. A data is said to be dirty if it is *incomplete, noisy, or inconsistent*.
- **Incomplete**: Data that lacks attribute values or lacks certain attributes of interest or contains only aggregate data is called incomplete data. For example, occupation=" "
- **Noisy**: Data that contains errors or outliers is called noisy data. For example, Salary="-10"
- **Inconsistent**: Data that contains discrepancy among data is called inconsistent data. For example, Birth date "03/07/1997" Age="42".

Data Preprocessing Concept

- Quality of data affects quality of data mining results. In order to improve quality of data and consequently the quality of data mining result, raw data need to be preprocessed.
- Data preprocessing is one of the critical step of data mining. Data preprocessing methods can be divided into following categories:
  - Data Cleaning
  - Data Integration
  - Data Transformation
  - Data Reduction
  - Data Discretization

Data Cleaning

- Real world data tend to be incomplete, noisy and inconsistent. data cleaning routines attempts to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

Ways for Handling Missing Values

- **Ignore the Tuple**: This is usually done when class label is missing. This method is not very effective, unless tuple contains several attributes with missing values.
- **Fill in the missing value manually**: this approach is time consuming and may not be feasible given a large data set with missing values.

Data Cleaning

- **Use a global constant to fill in the missing value**: Replace all missing attribute values by the same constant, such as label like "unknown". if missing value are replaced by ,say unknown then the mining program may mistakenly think that they form an interesting concept ,since they all have a value common – that of "unknown".
- **Use the attribute mean to fill in the missing value**: For example, suppose that the average income of customers is $28,000.Use this value to replace the missing value for income.

Data Cleaning

- **Use the attribute mean for all samples belonging to the same class as the given tuple**: For example, if classifying customers according to credit risk, replace the missing value with the average income value for customers in the same credit risk category as that of the given tuple.
- **Use the most probable value to fill in the missing value**: This may be determined with regression, Bayesian formalism, or decision tree induction. For example, using the other customer attributes in your data set, you may construct a decision tree to predict the missing values for income.

Data Cleaning

Ways of Smoothing Out Noisy Data

- **Binning Methods**: Theses methods smooth sorted data values by consulting values around it. Sorted values are distributed into a number of buckets or bins and then data can be smoothed by using bin mean, bin median or bin boundaries. Consider the example given below:
- Sorted data for price: 4, 8, 15, 21, 21, 24, 25, 28, 34

  Partitioning into Bins:
  - Bin1: 4, 8, 15
  - Bin2: 21, 21, 24
  - Bin3: 25, 28, 34

Data Cleaning

Smoothing by Bin Means

- Each value in a bin is replaced by the mean value of the bin.

  Example: Smoothing of Example Data
  - Bin1: 9, 9, 9
  - Bin2: 22, 22, 22
  - Bin3: 29, 29, 29

Data Cleaning

Smoothing by Bin Median

- Each bin value is replaced by the bin median.

  Example: Smoothing of Example Data
  - Bin1: 8, 8, 8
  - Bin2: 21, 21, 21
  - Bin3: 28, 28, 28

Data Cleaning

Smoothing by Bin Boundaries

- The minimum and maximum values in a given bin are identified as the *bin boundaries*. Each bin value is then replaced by the closest boundary value.
- Example: Smoothing of Example Data
  - Bin1: 4, 4, 15
  - Bin2: 21, 21, 24
  - Bin3: 25, 25, 34

Data Cleaning

Ways of Smoothing Out Noisy Data

- **Outlier Analysis:** Clustering is used to divide given data into different groups or clusters on the basis of similarity measures.
- Clustering can be also be used for outlier detection. Data can be smoothed by detecting and removing outliers.

Data Cleaning

Ways of Smoothing Out Noisy Data

- **Regression:** Data can be smoothed by using regression. We use regression to find a function that best fits through the given data and then this function can be used for smoothing data. We can either use linear or non-linear regression.
- Linear regression involves finding the "best" line to fit two attributes (or variables) so that one attribute can be used to predict the other. Multiple linear regression is an extension of linear regression.

Data Integration

- Data integration is the process of combining data from multiple sources into a coherent data store. These sources may include multiple databases, data cubes, or flat files.
- Two major issues to consider during data integration are: Schema Integration and Handling data redundancy.
- 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 is sure that customer_id in one database, and cust_number in another refer to the same entity?

Data Integration

- Databases and data warehouses typically have metadata which 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 attributes.
- Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.
- Redundant data may be able to be detected by correlation analysis.

Data Transformation

- Data transformation is the process of transforming data into the form that is appropriate for mining. Data transformation can involve the following:
- **Smoothing**: It is used to remove the noise from data. Such techniques include binning, clustering, and regression.
- **Aggregation**: Here 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.

Data Transformation

- **Discretization**: Values for numeric attributes, like age, may be mapped to higher level concepts, like young, middle aged, and senior.
- **Normalization**: Here the attribute data are scaled so as to fall within a small specified range, such as -1 to +1, or 0 to 1.
- **Attribute construction**: Here new attributes are constructed and added from the given set of attributes to help the mining process.

Data Transformation

- **Concept Hierarchy Generation for Nominal Data:** Here low level 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.

Data Transformation

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.
- In general, expressing an attribute in smaller units will lead to a larger range for that attribute, and thus tend to give such an attribute greater effect or "weight."

Data Transformation

Data Transformation by Normalization

- 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, 1].
- Normalizing the data attempts to give all attributes an equal weight. There are many methods for data normalization. The major normalization methods are: *min-max normalization and z-score normalization*.

Data Transformation

- **Min-max Normalization:** It performs a linear transformation on the original data. Suppose that *min* and *max* are the minimum and maximum values of an attribute, *A*. Min-max normalization maps a value, *v* , of *A* to *nv* in the range [*new_min*,*new_max*] using following formula.

$$nv = \frac{v - \min}{\max - \min}(new\_\max - new\_\min) + new\_\min$$

Data Transformation

- **Z-score Normalization:** In z-score normalization (or zero-mean normalization), the values for an attribute, A, are normalized based on the mean and standard deviation of A. The value, v , of A is normalized to nv as below. It is also called standard normalization.

$$nv = \frac{v - \mu}{\sigma} \qquad where\ \mu\ is\ mean$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(v_i - \mu)^2}{N}},$$

where, μ is mean and n is number of data points

Data Transformation

Example

- Normalize the following data using Min-Max, and Z-score Normalization.

Salary	Age
45000	42
32000	26
58000	48
37000	32

Data Transformation

Solution

Min-Max Normalization

Salary	Age
0.50	0.73
0.00	0.00
1.00	1.00
0.19	0.27

Normalization of Salary

Salary=45000

$$nsal = \frac{45000 - 32000}{58000 - 32000}(1-0) + 0 = 0.5$$

Normalization of Age

Age=42

$$nage = \frac{42 - 26}{48 - 26}(1-0) + 0 = 0.73$$

Data Transformation

Solution

Z-Score Normalization

Salary	Age
0.20	0.59
-1.12	-1.29
1.53	1.29
-0.61	-0.59

Calculate mean and standard deviation of salary

$$\mu = 43000$$

$$\sigma = 9823.44$$

Now, transform salary=45000

$$nsal = \frac{45000 - 43000}{9823.44} = 0.20$$

Data Reduction

- 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. That is, mining on the reduced data set should be more efficient yet produce the same (or almost the same) analytical results.
- Data reduction strategies include *dimensionality reduction*, *numerosity reduction*, and *data compression*.

Data Reduction

Dimensionality Reduction

- **Dimensionality reduction** is the process of reducing the number of random variables or attributes under consideration.
- Dimensionality reduction methods include *wavelet* transformation and principal components analysis, which 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.

Data Reduction

Dimensionality Reduction

- The discrete wavelet transform (DWT) is a linear signal processing technique that, when applied to a data vector *X*, transforms it to a numerically different vector, *X'*, of wavelet coefficients. The two vectors are of the same length.
- A compressed approximation of the data can be retained by storing only a small fraction of the strongest of the wavelet coefficients. For example, all wavelet coefficients larger than some user-specified threshold can be retained.

Data Reduction

Dimensionality Reduction

- Principal Component analysis (PCA) converts K-dimensional data vector into c-dimensional vector, where c <= k is the number of orthogonal vectors that can be best used to represent data. These c orthogonal data vectors are called principal components of the original dataset.
- Each data vector is a represented as linear combination of the c principal component vectors.
- PCA Works for numeric data only and is useful when the number of dimensions is large.

Data Reduction

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.) Regression and log-linear models are examples.
- *Nonparametric methods* for storing reduced representations of the data include *histograms*, *clustering*, *sampling*, and *data cube aggregation*.

Data Reduction

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**.

Data Discretization and Concept Hierarchy Generation

- 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).
- Data Discretization and Concept Hierarchy Generation can be performed using binning, histogram analysis or decision tree induction approaches.

Data Discretization and Concept Hierarchy Generation

Discretization and Concept Hierarchy Generation by Binning

- Binning is a top-down splitting technique based on a specified number of bins. Binning methods for data smoothing can also be used as discretization methods for data reduction and concept hierarchy generation.
- For example, attribute values can be discretized by applying binning, and then replacing each bin value by the bin mean or median.
- These techniques can be applied recursively to the resulting partitions to generate concept hierarchies.

Data Discretization and Concept Hierarchy Generation

Discretization and Concept Hierarchy Generation by Histogram Analysis

- Histograms use binning to approximate data distributions and are a popular form of data reduction. A histogram for an attribute, *A*, partitions the data distribution of *A* into disjoint subsets, referred to as *buckets* or *bins*.
- The histogram analysis algorithm can be applied recursively to each partition in order to automatically generate a multilevel concept hierarchy, with the procedure terminating once a pre-specified number of concept levels has been reached.

Data Discretization and Concept Hierarchy Generation

Discretization and Concept Hierarchy Generation by Clustering

- A clustering algorithm can be applied to discretize a numeric attribute, *A*, by partitioning the values of *A* into clusters or groups.
- Clustering takes the distribution of *A* into consideration, as well as the closeness of data points, and therefore is able to produce high-quality discretization results.
- Clustering can be used to generate a concept hierarchy for *A* by following either a top-down splitting strategy or a bottom-up merging strategy, where each cluster forms a node of the concept hierarchy.

Data Mining Primitives

- Data mining task can be specified in the form of a data mining query, which is input to the data mining system. A data mining query is defined in terms of data mining primitives.
- These primitives allow the user to interactively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles. The data mining primitives specify the following.
- **The set of task-relevant data to be mined**: This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest.

Data Mining Primitives

- **The kind of knowledge to be mined**: This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.
- **The background knowledge to be used in the discovery process**: This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction.

Data Mining Primitives

- **The interestingness measures and thresholds for pattern evaluation**: They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns.
- **The expected representation for visualizing the discovered patterns**: This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.

Data Mining Primitives

- Designing a comprehensive data mining language is challenging because data mining covers a wide spectrum of tasks. The design of an effective data mining query language requires a deep understanding of the power, limitation, and underlying mechanisms of the various kinds of data mining tasks.
- DMQL (Data Mining Query Language) is based on the above primitives. The language adopts an SQL-like syntax, so that it can easily be integrated with the relational query language, SQL. Let's look at how it can be used to specify a data mining task.

Data Mining Primitives

Example

- Suppose, as a marketing manager of and Electronics store, you would like to classify customers based on their buying patterns. You are especially interested in those customers whose salary is no less than $40,000, and who have bought more than $1,000 worth of items, each of which is priced at no less than $100. In particular, you are interested in the customer's age, income, the types of items purchased, the purchase location, and where the items were made. You would like to view the resulting classification in the form of rules. This data mining query is expressed in DMQL as follows.

Data Mining Primitives

Example

use database ElectronicsDB
use hierarchy location-hierarchy for T.branch, age-hierarchy for C.age
mine classification as PromisingCustomers
in relevance to C.age, C.income, I.type, I.place-made, T.branch
from customer C, item I, transaction T
where I.itemID = T.itemID and C.custID = T.custID and C.income ≥40,000 and I.price ≥ 100
group by T.custID