

Information Processing Technology of Internet of Things

Chapter 1 Data and Data Preprocessing

Wu Liu

Beijing Key Lab of Intelligent Telecomm. Software and Multimedia
Beijing University of Posts and Telecommunications

1.2 Data Preprocessing



Outline

- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary



Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
 - Accuracy: correct or wrong, accurate or not
 - Completeness: not recorded, unavailable, ...
 - Consistency: some modified but some not, dangling, ...
 - Timeliness: timely update?
 - Believability: how trustable the data are correct?
 - Interpretability: how easily the data can be understood?

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

Outline

- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary



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?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
 - a global constant : e.g., “unknown”, a new class?!
 - the attribute mean
 - the attribute mean for all samples belonging to the same class
 - the most probable 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 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
 - 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 as a Process

- Data discrepancy detection
 - Use metadata (e.g., domain, range, dependency, distribution)
 - Check field overloading
 - Check uniqueness rule, consecutive rule and null rule
 - Use commercial tools
 - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
 - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)
 - Data migration and integration
 - Data migration tools: allow transformations to be specified
 - ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface
 - Integration of the two processes: discrepancy detection and data transformation
 - Iterative and interactive
-



Outline

- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
 - Data Cleaning
 - Data Integration 
 - Data Reduction
 - Data Transformation and Data Discretization
 - Summary
-



Data Integration

- **Data integration:**
 - Combines data from multiple sources into a coherent store
- Schema integration: e.g., $A.cust-id \equiv B.cust-\#$
 - Integrate metadata from different sources
- Entity identification problem:
 - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
 - For the same real world entity, attribute values from different sources are different
 - Possible reasons: different representations, different scales, e.g., metric vs. British units



Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
 - *Object identification*: The same attribute or object may have different names in different databases
 - *Derivable data*: One attribute may be a “derived” attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by *correlation analysis and covariance analysis*
- Careful integration of the data from multiple sources may help *reduce/avoid redundancies and inconsistencies* and improve mining speed and quality



Correlation Analysis (Nominal Data)

- **X² (chi-square) test**

$$\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}$$

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

$$e_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{n}$$

- The larger the X² value, the more likely the variables are related
- The cells that contribute the most to the X² value are those whose actual count is very different from the expected count
- Correlation does not imply causality
 - # of hospitals and # of car-theft in a city are correlated
 - Both are causally linked to the third variable: population



Chi-Square Calculation: An Example

	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- χ^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250 - 90)^2}{90} + \frac{(50 - 210)^2}{210} + \frac{(200 - 360)^2}{360} + \frac{(1000 - 840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group
-



Correlation Analysis (Numeric Data)

- Correlation coefficient (also called **Pearson's product moment coefficient**)

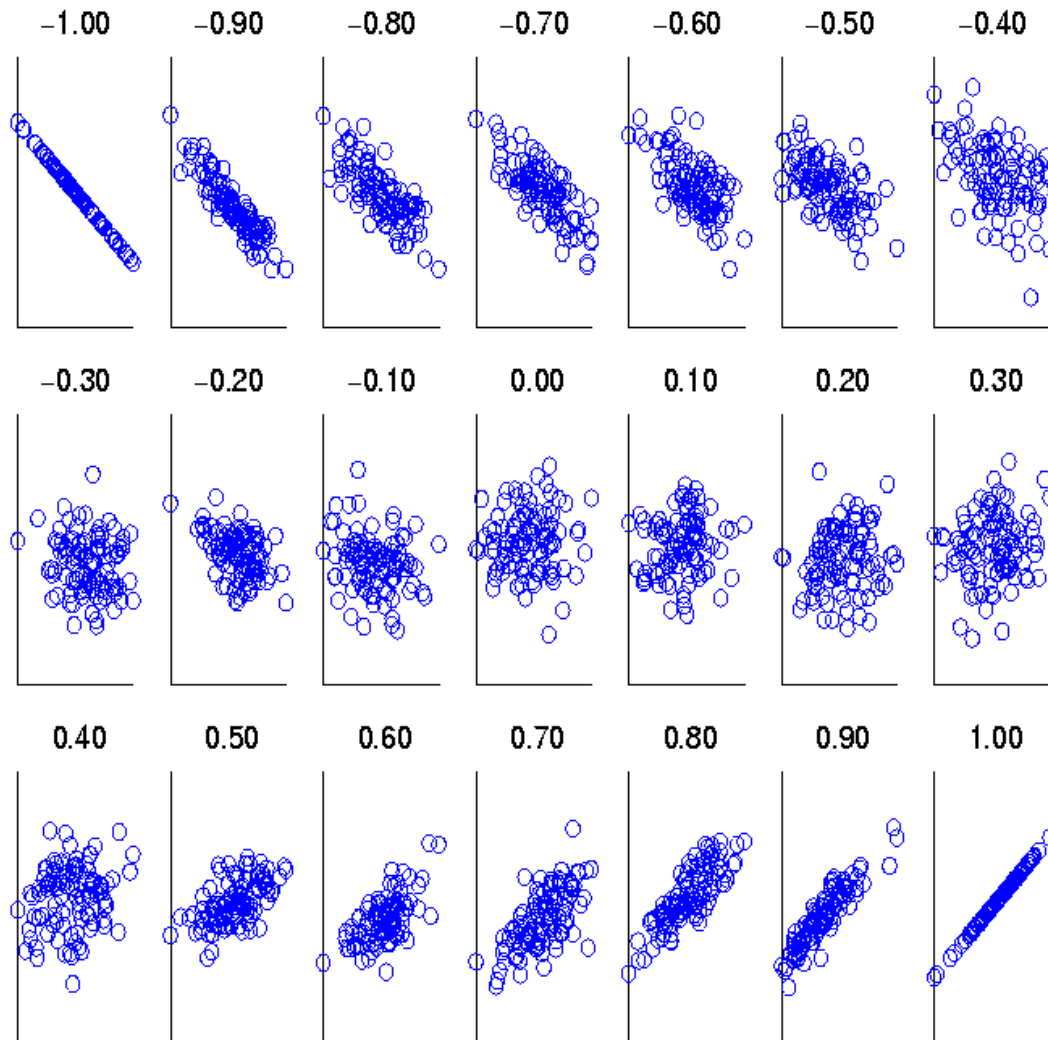
$$r_{A,B} = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^n (a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B}$$

where n is the number of tuples, \bar{A} and \bar{B} are the respective means of A and B , σ_A and σ_B are the respective standard deviation of A and B , and $\sum(a_i b_i)$ is the sum of the AB cross-product.

- If $r_{A,B} > 0$, A and B are positively correlated (A 's values increase as B 's). The higher, the stronger correlation. **A higher value may indicate that A (or B) may be removed as a redundancy.**
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated



Visually Evaluating Correlation



**Scatter plots
showing the
similarity from
-1 to 1.**

Covariance (Numeric Data)

$$E(A) = \bar{A} = \frac{\sum_{i=1}^n a_i}{n}$$

$$E(B) = \bar{B} = \frac{\sum_{i=1}^n b_i}{n}$$

- Covariance is similar to correlation

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

Correlation coefficient: $r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$

where n is the number of tuples, \bar{A} and \bar{B} are the respective mean or **expected values** of A and B , σ_A and σ_B are the respective standard deviation of A and B

- **Positive covariance:** If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values
- **Negative covariance:** If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value
- **Independence:** $Cov_{A,B} = 0$ but the converse is not true:
 - Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence



Co-Variance: An Example

$$\text{Cov}(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n}$$


- It can be simplified in computation as

$$\text{Cov}(A, B) = E(A \cdot B) - \bar{A}\bar{B}$$

- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?
 - $E(A) = (2 + 3 + 5 + 4 + 6) / 5 = 20/5 = 4$
 - $E(B) = (5 + 8 + 10 + 11 + 14) / 5 = 48/5 = 9.6$
 - $\text{Cov}(A, B) = (2 \times 5 + 3 \times 8 + 5 \times 10 + 4 \times 11 + 6 \times 14) / 5 - 4 \times 9.6 = 4$
- Thus, **A and B rise together since $\text{Cov}(A, B) > 0$.**



Outline

- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
 - Data Cleaning
 - Data Integration
 - Data Reduction 
 - Data Transformation and Data Discretization
 - Summary
-



Data Reduction Strategies

- **Data reduction:** Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the 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., remove unimportant attributes
 - Wavelet transforms
 - Principal Components Analysis (PCA)
 - Feature subset selection, feature creation
 - **Numerosity reduction** (some simply call it: Data Reduction)
 - Regression and Log-Linear Models
 - Histograms, clustering, sampling
 - Data cube aggregation
 - **Data compression**



Data Reduction: Dimensionality Reduction

■ **Curse of dimensionality**

- When dimensionality increases, **data becomes increasingly sparse**
- **Density and distance** between points, which is critical to clustering, outlier analysis, **becomes less meaningful**
- The possible combinations of subspaces will grow exponentially

■ **Dimensionality reduction**

- Avoid the curse of dimensionality
- Help eliminate irrelevant features and reduce noise
- Reduce time and space required in data mining
- Allow easier visualization

■ **Dimensionality reduction techniques**

- Wavelet transforms
- Principal Component Analysis
- Supervised and nonlinear techniques (e.g., feature selection)



Attribute Subset Selection

- Way to reduce dimensionality of data. **Attribute subset selection reduces the data set size by removing irrelevant or redundant attributes** (or dimensions).
- Redundant attributes
 - Duplicate much or all of the information contained in one or more other attributes
 - E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
 - Contain no information that is useful for the data mining task at hand
 - E.g., students' ID is often irrelevant to the task of predicting students' GPA



Heuristic Search in Attribute Selection

- There are 2^d possible attribute combinations of d attributes
- Typical heuristic attribute selection methods:
 - Best single attribute under the attribute independence assumption: choose by significance tests
 - Best step-wise feature selection (**Stepwise forward selection**):
 - The best single-attribute is picked first
 - Then next best attribute condition to the first, ...
 - Step-wise attribute elimination (**Stepwise backward elimination**):
 - Repeatedly eliminate the worst attribute
 - Best combined attribute selection and elimination (**Combination of forward selection and backward elimination**)

Forward selection	Backward elimination
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$
Initial reduced set: $\{\}$	$\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$
$\Rightarrow \{A_1\}$	$\Rightarrow \{A_1, A_4, A_5, A_6\}$
$\Rightarrow \{A_1, A_4\}$	\Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$
\Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$	



Attribute Creation (Feature Generation)

- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
 - Three general methodologies
 - Attribute extraction
 - Domain-specific
 - Mapping data to new space (see: data reduction)
 - E.g., Fourier transformation, wavelet transformation, manifold approaches (not covered)
 - Attribute construction
 - Combining features (see: discriminative frequent patterns in Chapter on “Advanced Classification”)
 - Data discretization
-



Data Reduction 2: Numerosity Reduction

- Reduce data volume by choosing alternative, *smaller forms of data representation*
- **Parametric methods** (e.g., regression)
 - Assume the data fits some model, estimate the model parameters, store only the parameters, and discard the data (except possible outliers)
 - Ex.: Log-linear models—obtain value at a point in m -D space as the product on appropriate marginal subspaces
- **Non-parametric methods**
 - Do not assume models
 - Major families: histograms, clustering, sampling, ...



Parametric Data Reduction: Regression and Log-Linear Models

■ Linear regression

- Data modeled to fit a straight line
- Often uses the **least-square method** to fit the line

■ Multiple regression

- Allows a response variable Y to be modeled as **a linear function of multidimensional feature vector**

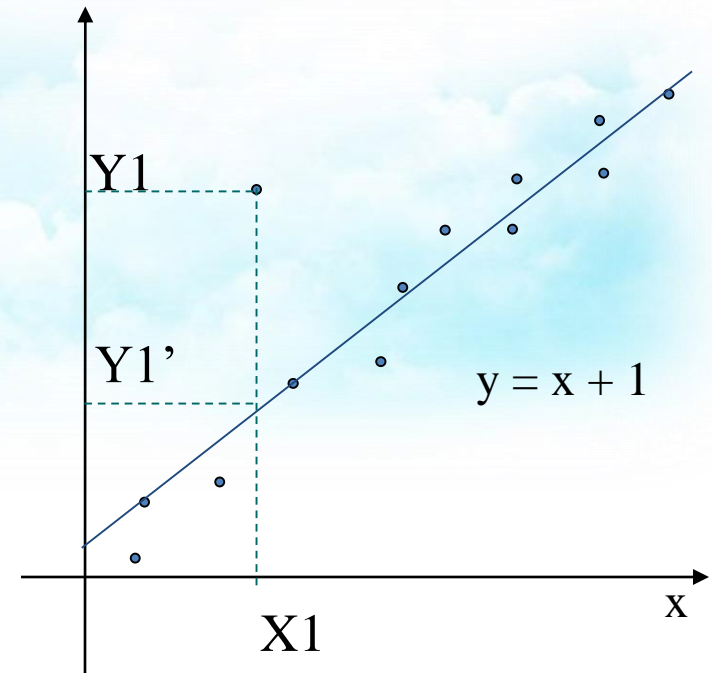
■ Log-linear model

- Approximates discrete multidimensional probability distributions



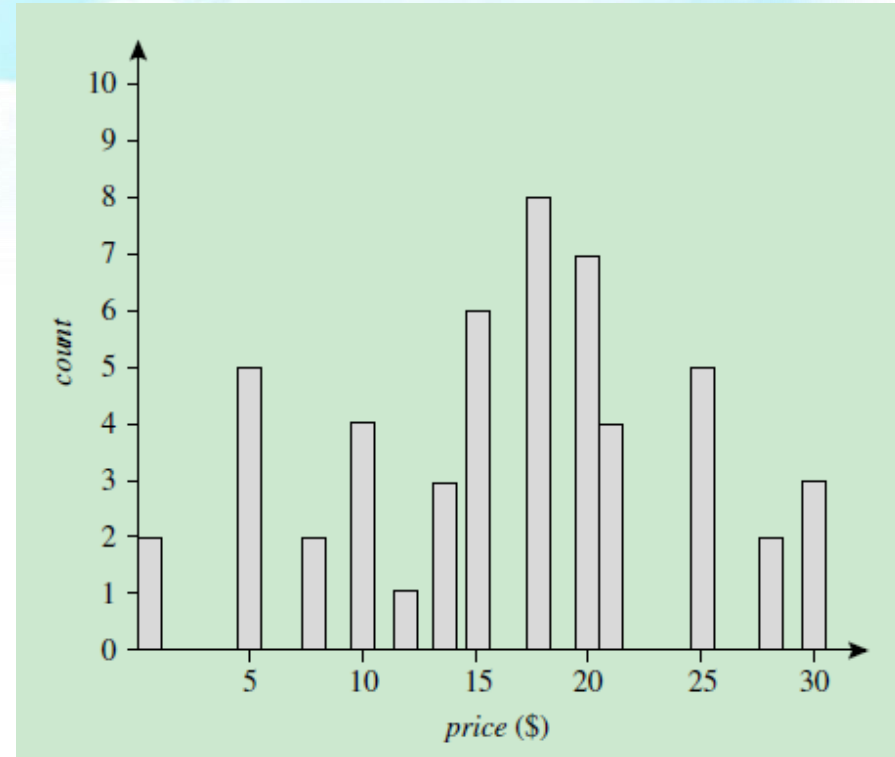
Regression Analysis

- Regression analysis: A collective name for techniques for the modeling and analysis of numerical data consisting of values of a **dependent variable** (also called **response variable** or *measurement*) and of one or more **independent variables** (aka. **explanatory variables** or **predictors**)
- The parameters are estimated so as to give a "best fit" of the data
- Most commonly the best fit is evaluated by using the **least squares method**, but other criteria have also been used
- Used for prediction (including forecasting of time-series data), inference, hypothesis testing, and modeling of causal relationships



Histogram Analysis

- Histograms use binning to approximate data distributions and are a popular form of data reduction.
- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
 - Equal-width: equal bucket range
 - Equal-frequency (or equal-depth): the frequency of each bucket is constant



A histogram for *price* using *singleton buckets*—each bucket represents one price–value/frequency pair.



Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms



Sampling

- Sampling: obtaining a small sample s to represent the whole data set N
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a **representative** subset of the data
 - Simple random sampling may have very poor performance in the presence of skew
 - Develop adaptive sampling methods, e.g., stratified sampling:

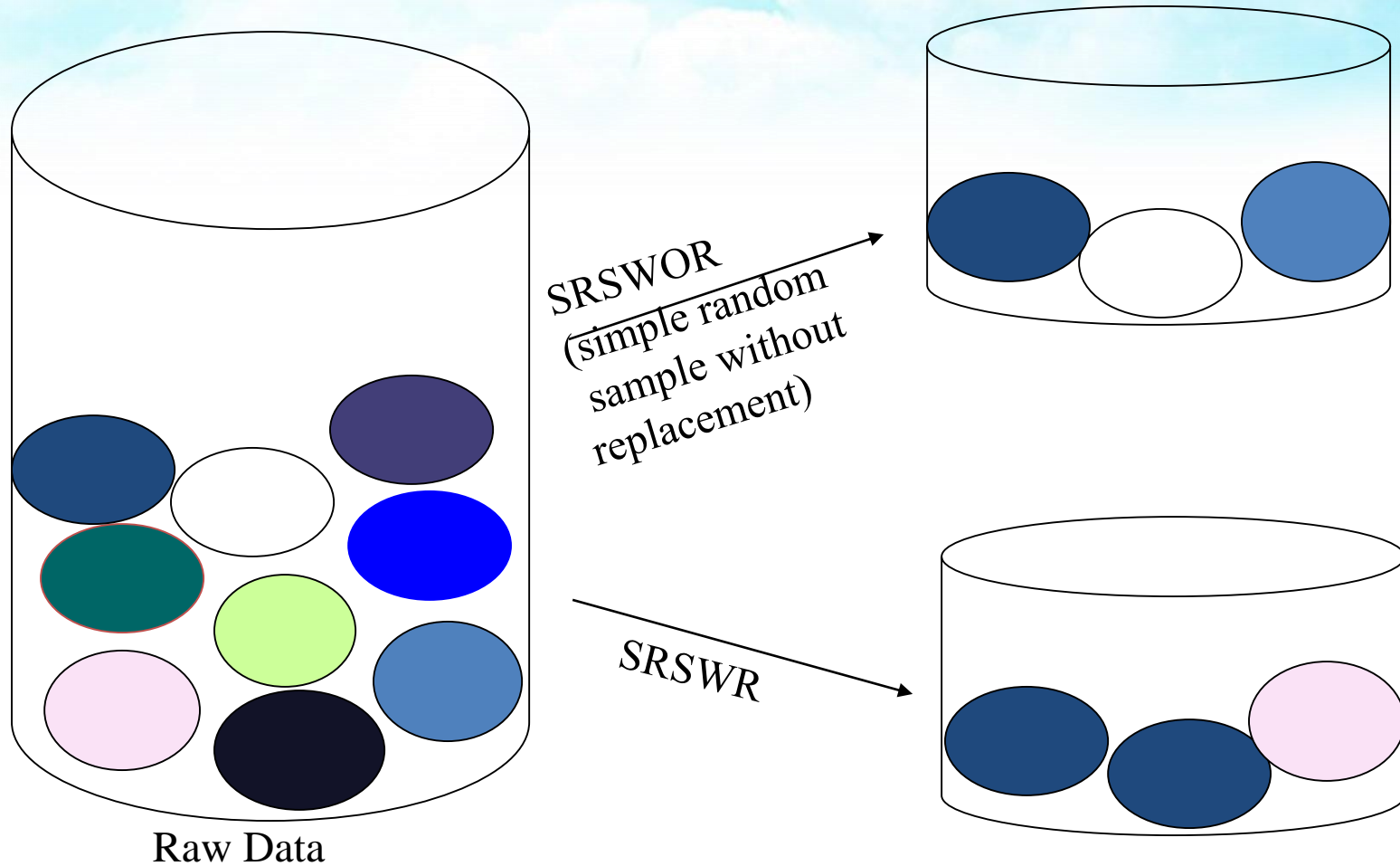


Types of Sampling

- **Simple random sampling**
 - There is an **equal probability** of selecting any particular item
- **Sampling without replacement**
 - Once an object is selected, it is removed from the population
- **Sampling with replacement**
 - A selected object is not removed from the population
- **Stratified sampling:** If D is divided into mutually disjoint parts called strata, a stratified sample of D is generated by obtaining a simple random sampling at each stratum.
 - Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)
 - Used in conjunction with skewed data

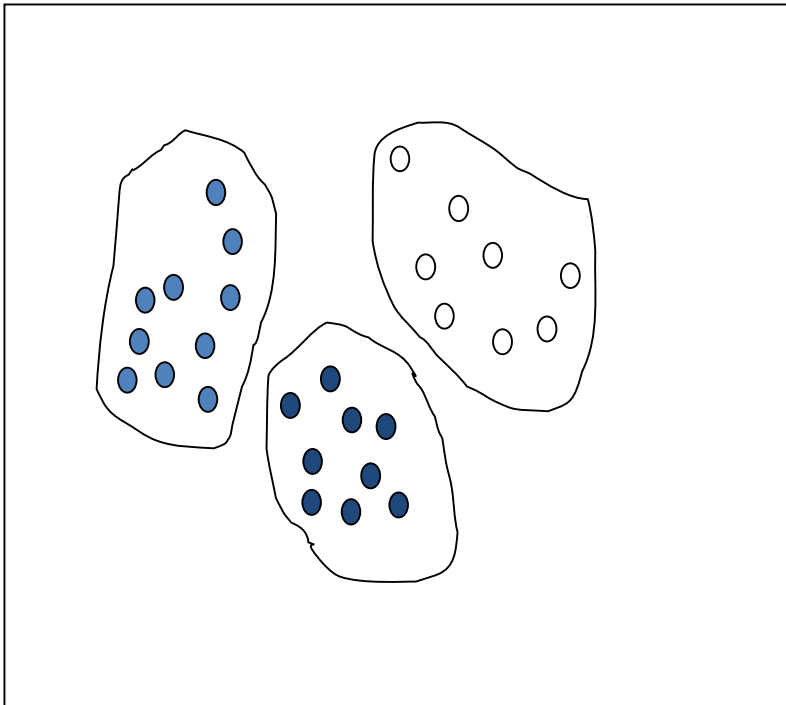


Sampling: With or without Replacement

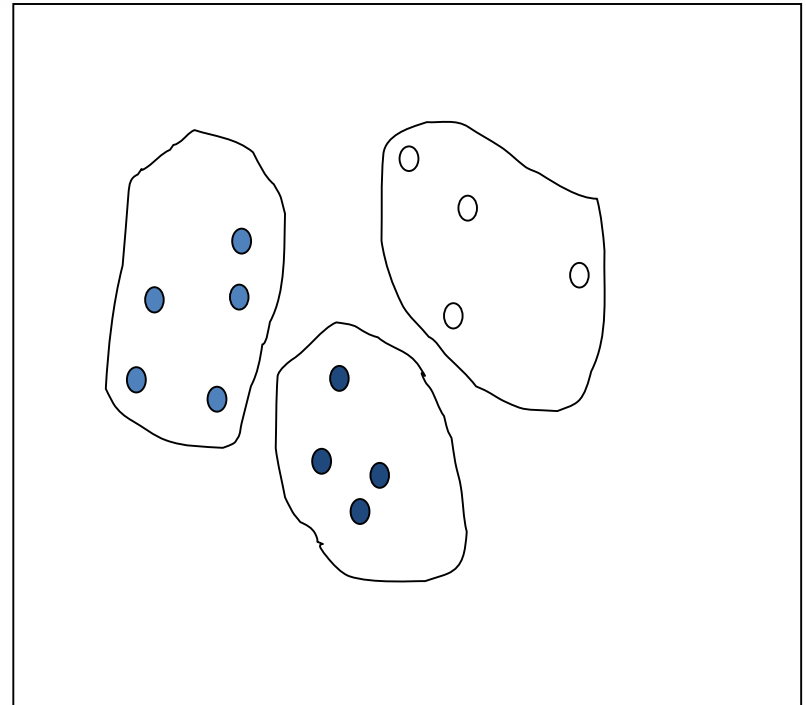


Sampling: Cluster or Stratified Sampling


Raw Data



Cluster/Stratified Sample



Outline

- Data Preprocessing: An Overview
 - Data Quality
 - Major Tasks in Data Preprocessing
 - Data Cleaning
 - Data Integration
 - Data Reduction
 - Data Transformation and Data Discretization 
 - Summary
-



Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values s.t. each old value can be identified with one of the new values.
- In data transformation, the data are transformed or consolidated into forms appropriate for mining.
- Methods
 - **Smoothing**: Remove noise from data
 - **Attribute/feature construction**
 - New attributes constructed from the given ones
 - **Aggregation**: Summarization, data cube construction
 - **Normalization**: Scaled to fall within a smaller, specified range
 - min-max normalization
 - z-score normalization
 - normalization by decimal scaling
 - **Discretization**: Concept hierarchy climbing



Normalization

- **Min-max normalization:** to $[\text{new_min}_A, \text{new_max}_A]$

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

- 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-score normalization** (μ : mean, σ : standard deviation):

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

- Ex. Let $\mu = 54,000$, $\sigma = 16,000$. Then $\frac{73,600 - 54,000}{16,000} = 1.225$

- **Normalization by decimal scaling**

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



Standardizing Numeric Data

- Z-score: $z = \frac{x - \mu}{\sigma}$
 - X: raw score to be standardized, μ : mean of the population, σ : standard deviation
 - the distance between the raw score and the population mean in units of the standard deviation
 - negative when the raw score is below the mean, “+” when above
- An alternative way: Calculate the mean absolute deviation
$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$
where
$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \dots + x_{nf}).$$
 - standardized measure (z-score):
$$z_{if} = \frac{x_{if} - m_f}{s_f}$$
- Using mean absolute deviation is more robust than using standard deviation



Discretization

- Three types of attributes
 - Nominal—values from an unordered set, e.g., color, profession
 - Ordinal—values from an ordered set, e.g., military or academic rank
 - Numeric—real numbers, e.g., integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
 - Interval labels can then be used to replace actual data values
 - Reduce data size by discretization
 - Supervised vs. unsupervised
 - Split (top-down) vs. merge (bottom-up)
 - Discretization can be performed recursively on an attribute
 - Prepare for further analysis, e.g., classification



Data Discretization Methods

- Typical methods: All the methods can be applied recursively
 - Binning
 - Top-down split, unsupervised
 - Histogram analysis
 - Top-down split, unsupervised
 - Clustering analysis (unsupervised, top-down split or bottom-up merge)
 - Decision-tree analysis (supervised, top-down split)
 - Correlation (e.g., χ^2) analysis (unsupervised, bottom-up merge)
-



Simple Discretization: Binning

- **Equal-width** (distance) partitioning
 - 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
 - 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 equal-frequency (**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



Discretization by Classification & Correlation Analysis

- Classification (e.g., decision tree analysis)
 - Supervised: Given class labels, e.g., cancerous vs. benign
 - Using *entropy* to determine split point (discretization point)
 - Top-down, recursive split
 - Details to be covered in Chapter “Classification”
 - Correlation analysis (e.g., Chi-merge: χ^2 -based discretization)
 - Supervised: use class information
 - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
 - Merge performed recursively, until a predefined stopping condition
-



Concept Hierarchy Generation

- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: **Recursively reduce the data by collecting and replacing low level concepts** (such as numeric values for *age*) **by higher level concepts** (such as *youth*, *adult*, or *senior*)
- Concept hierarchies can be explicitly specified by domain experts and/or data warehouse designers
- Concept hierarchy can be automatically formed for both numeric and nominal data



Concept Hierarchy Generation for Nominal Data

- Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
 - $street < city < state < country$
- Specification of a hierarchy for a set of values by explicit data grouping
 - $\{Urbana, Champaign, Chicago\} < Illinois$
- Specification of only a partial set of attributes
 - E.g., only $street < city$, not others
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
 - E.g., for a set of attributes: $\{street, city, state, country\}$



Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
 - The attribute with the most distinct values is placed at the lowest level of the hierarchy
 - Exceptions, e.g., weekday, month, quarter, year

