

HIT391

MACHINE LEARNING: ADVANCEMENTS AND APPLICATIONS



CHARLES
DARWIN
UNIVERSITY
AUSTRALIA

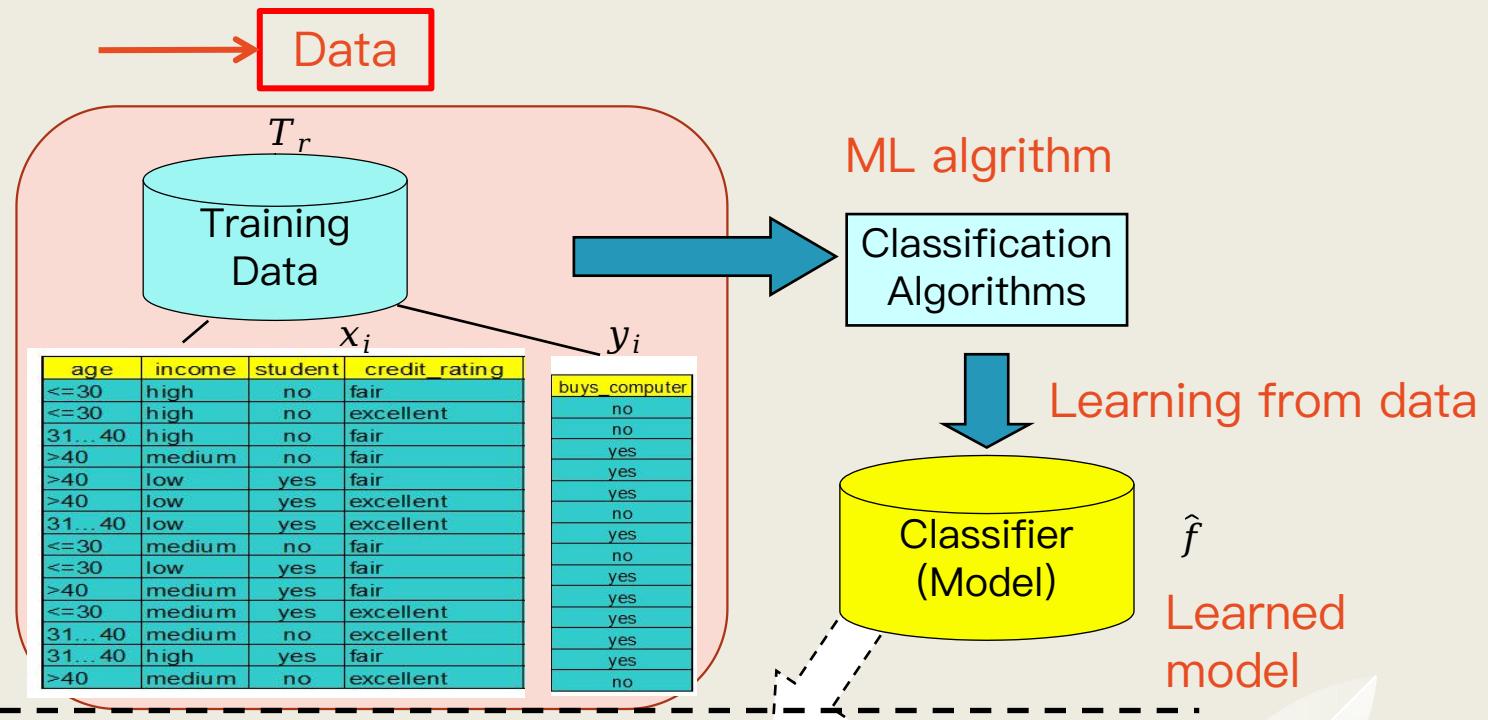
Week 3: Data Analysis and Data Pre-processing

■ Learning Outcomes

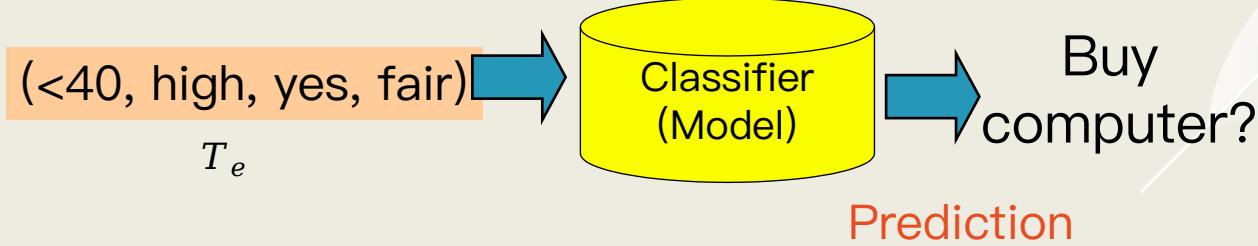
- Data analysis**
- Data preprocessing**

An example of Supervised Learning

Step 1: Training



Step2: Testing



Outline

- An Overview
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization
- Summary

An Overview



Data Quality: Why Preprocess the Data?

- Measures for **data quality**: a multidimensional view
 - **Accuracy**: correct or wrong, accurate or not, e.g.,
Synthetic data
 - **Completeness**: not recorded, unavailable, ...e.g., missing
data.
 - **Consistency**: some modified but some not, dangling, ...
 - **Timeliness**: timely update? e.g., collected 10 years ago
 - **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. e.g., all data need to be normalized a range [0,1]
 - Concept hierarchy generation

Data Cleaning



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”
 - 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, e.g., recommender system

	Country	Age	Salary	Purchased	Gender	City
0	Australia	44.0	72000.0	No	M	Sydney
1	New Zealand	27.0	48000.0	Yes	M	Auckland
2	Germany	30.0	52000.0	No	M	Bonn
3	New Zealand	38.0	61000.0	No	F	Auckland
4	Germany	40.0	NaN	Yes	F	Berlin
5	Australia	35.0	58000.0	Yes	NaN	NaN
6	New Zealand	NaN	52000.0	No	NaN	Wellington
7	Australia	48.0	79000.0	Yes	M	Sydney

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: smarter

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 Integration



Data integration

- **Data integration:**
 - Combines data from multiple sources into a coherent store
- **Schema integration:** e.g., A.cust-id , 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

- **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 (Numeric Data)

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

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

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 $\Sigma(a_i b_i)$ 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.
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated

Covariance (Numeric Data)

- 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.

Co-Variance: An Example

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

- It can be simplified in computation as

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

- 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?
 - $E(A) = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4$
 - $E(B) = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6$
 - $Cov(A,B) = (2 \times 5 + 3 \times 8 + 5 \times 10 + 4 \times 11 + 6 \times 14) /5 - 4 \times 9.6 = 4$
- Thus, A and B rise together since $Cov(A, B) > 0$.

Data Reduction



Data Reduction Methods

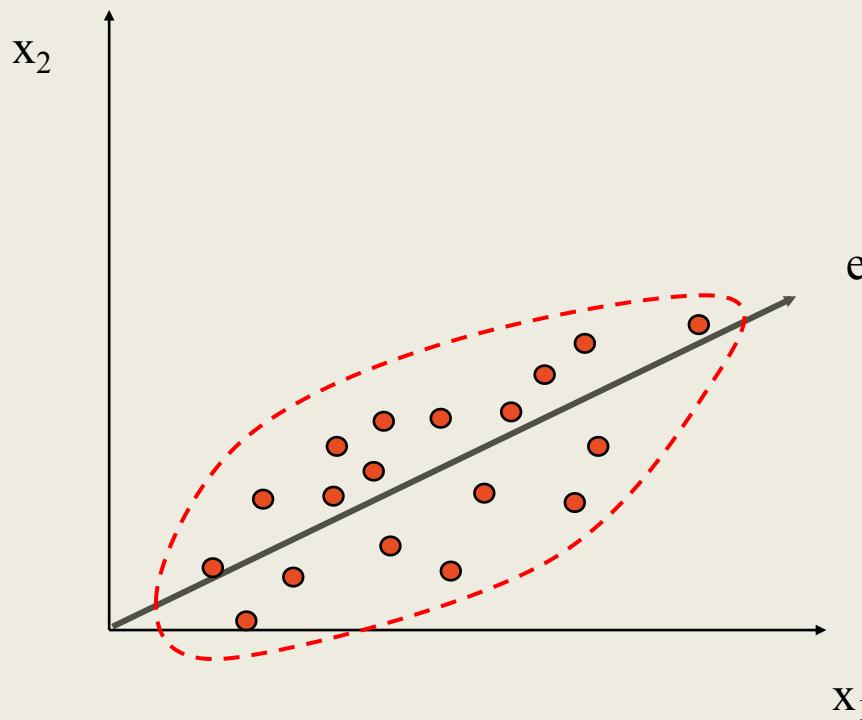
- **Data reduction:** Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very **long time** to run on the complete data set.
- Data reduction strategies
 - **Dimensionality reduction**, e.g., remove unimportant attributes
 - Principal Components Analysis (PCA)
 - Feature subset selection, feature creation
 - **Data reduction**
 - Regression and Log-Linear Models
 - Histograms, clustering, sampling
 - Data cube aggregation
 - **Data compression**

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

Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in **dimensionality reduction**. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



Steps of PCA

- Given N data vectors from n -dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
 - Normalize input data: Each attribute falls within the same range
 - Compute k orthonormal (unit) vectors, i.e., *principal components*
 - Each input data (vector) is a linear combination of the k principal component vectors
 - The principal components are sorted in order of decreasing “significance” or strength
 - Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data)
- Works for numeric data only

Attribute Subset Selection

- Another way to reduce dimensionality of data
- 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

Data Reduction

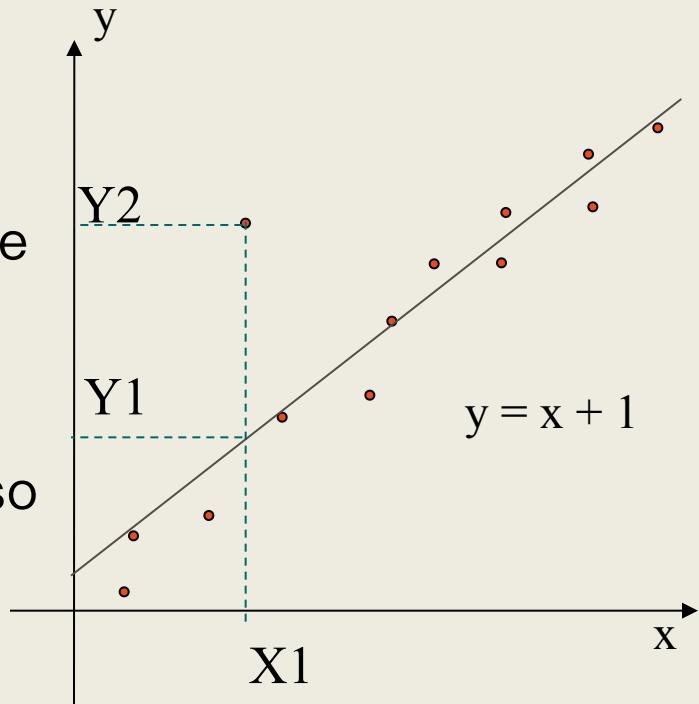
- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods** (e.g., regression)
 - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
 - Ex.: Log-linear models—obtain value 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, ...

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

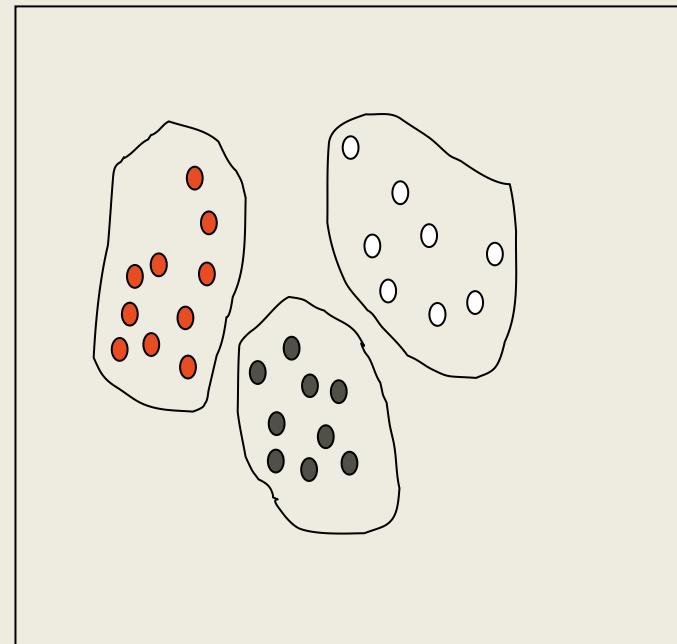
- Regression analysis: A collective name for techniques for the modeling and analysis of numerical data consisting of values of a **dependent variable** (also called **response variable** or *measurement*) and of one or more *independent variables* (aka. **explanatory variables** or **predictors**)
- 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

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

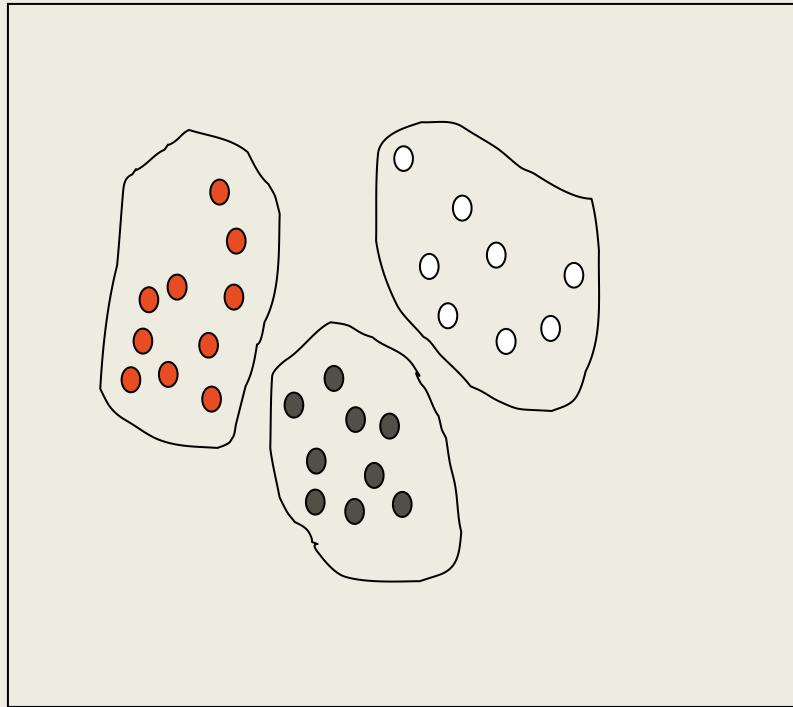
- Sampling: obtaining a small sample s to represent the whole data set N
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a **representative** subset of the data
 - Simple random sampling may have very poor performance in the presence of skew
 - Develop adaptive sampling methods, e.g., stratified sampling:
- Note: Sampling may not reduce database I/Os (page at a time)

Types of Sampling

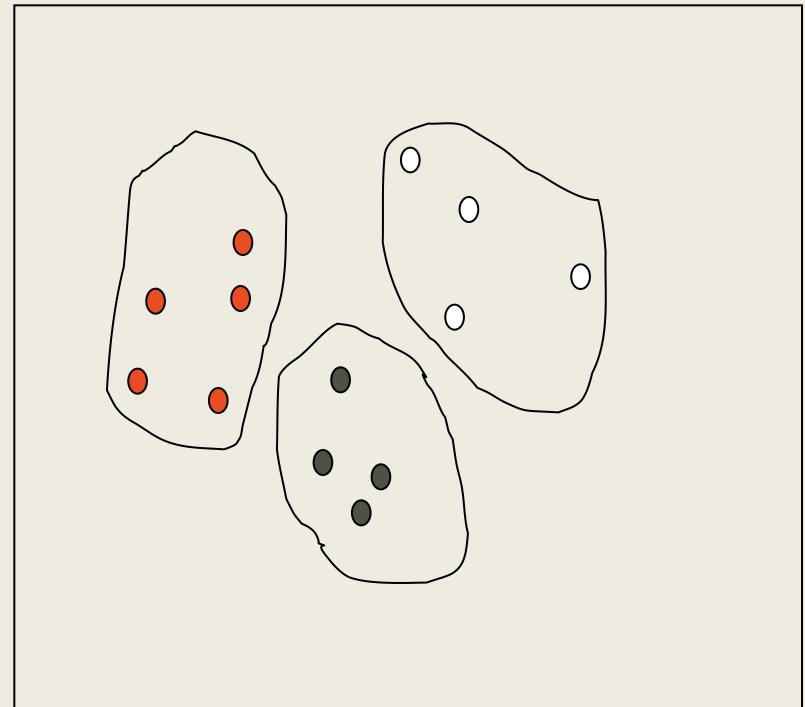
- **Simple random sampling**
 - There is an equal probability of selecting any particular item
- **Sampling without replacement**
 - Once an object is selected, it is removed from the population
- **Sampling with replacement**
 - A selected object is not removed from the population
- **Stratified sampling:**
 - 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

Stratified Sampling

Raw Data



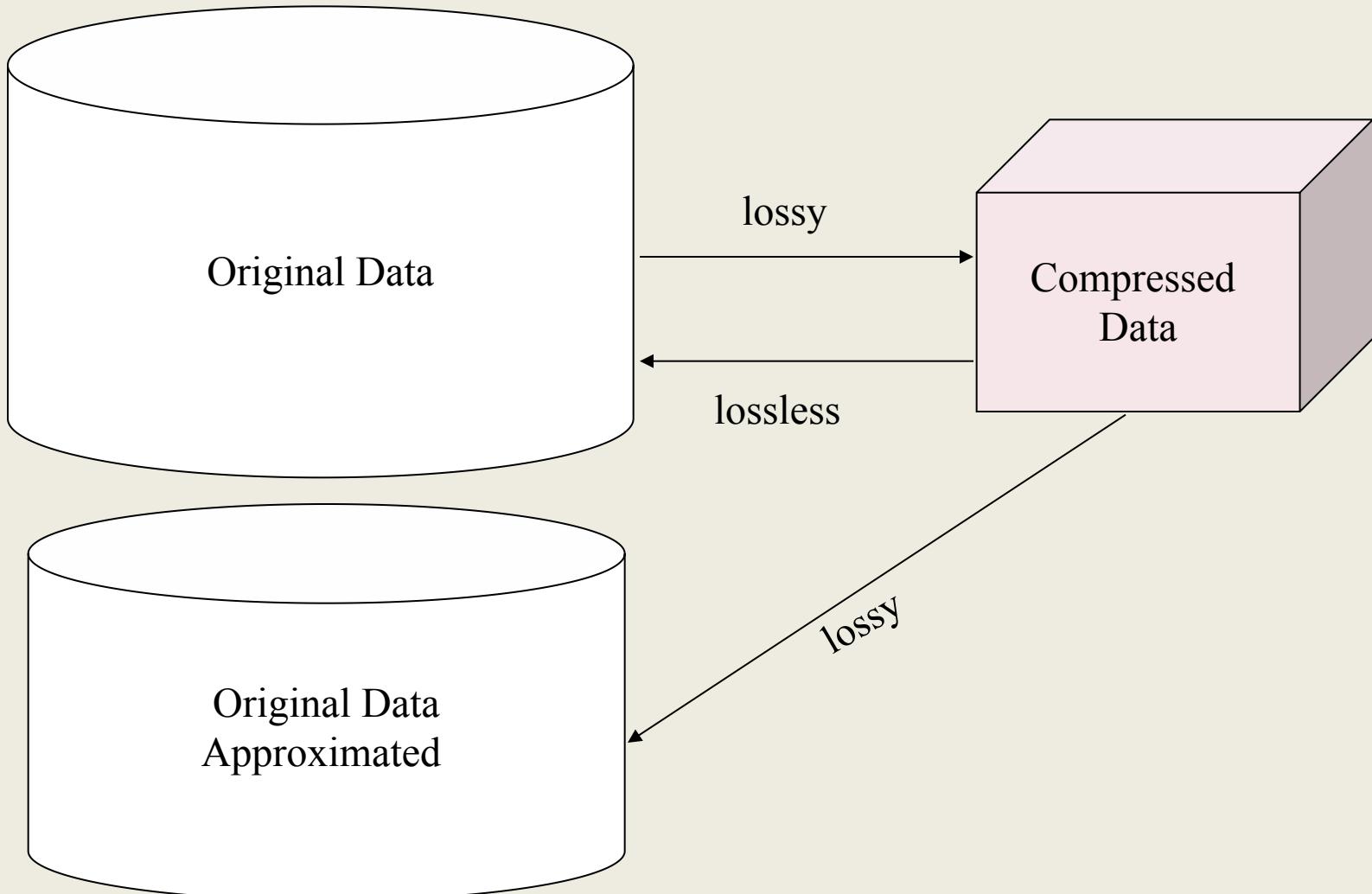
Cluster/Stratified Sample



Data Compression

- String compression
 - There are extensive theories and well-tuned algorithms
 - Typically lossless, but only limited manipulation is possible without expansion
- Audio/video compression
 - Typically lossy compression, with progressive refinement
 - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
 - Typically short and vary slowly with time
- Dimensionality and numerosity reduction may also be considered as forms of data compression

Data Compression



Data Transformation



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
- Methods
 - Smoothing: Remove noise from data
 - Attribute/feature construction
 - New attributes constructed from the given ones
 - Aggregation: Summarization
 - Normalization: Scaled to fall within a smaller, specified range
 - min-max normalization
 - z-score normalization
 - normalization by decimal scaling
 - Discretization: one-hot encoding, Concept hierarchy climbing

Normalization

- **Min-max normalization:** to $[new_min_A, new_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

- 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**

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

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
 - Clustering analysis (unsupervised, top-down split or bottom-up merge)
 - Decision-tree analysis (supervised, top-down split)
 - Correlation analysis (unsupervised, bottom-up merge)

Discretization by Classification & Correlation Analysis

- 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. For numeric data, use discretization methods shown.

Summary

- **Data quality:** accuracy, completeness, consistency, timeliness, believability, interpretability
- **Data cleaning:** e.g. missing/noisy values, outliers
- **Data integration** from multiple sources:
 - Entity identification problem
 - Remove redundancies
 - Detect inconsistencies
- **Data reduction**
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression
- **Data transformation and data discretization**
 - Normalization
 - Concept hierarchy generation

References



- Han, Jiawei, Jian Pei, and Hanghang Tong. Data mining: concepts and techniques. Morgan kaufmann, 2022.
- Chapter 3 - Concepts and Techniques, Jiawei Han, Micheline Kamber, and Jian Pei, University of Illinois at Urbana-Champaign & Simon Fraser University ©2011 Han, Kamber & Pei.