Data Mining:

Concepts and Techniques

(3rd ed.)

— Chapter 3 —

Jiawei Han, Micheline Kamber, and Jian Pei
University of Illinois at Urbana-Champaign &
Simon Fraser University

Chapter 3: Data Preprocessing

- 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
    - Human or computer error, limited buffer size etc
  - 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

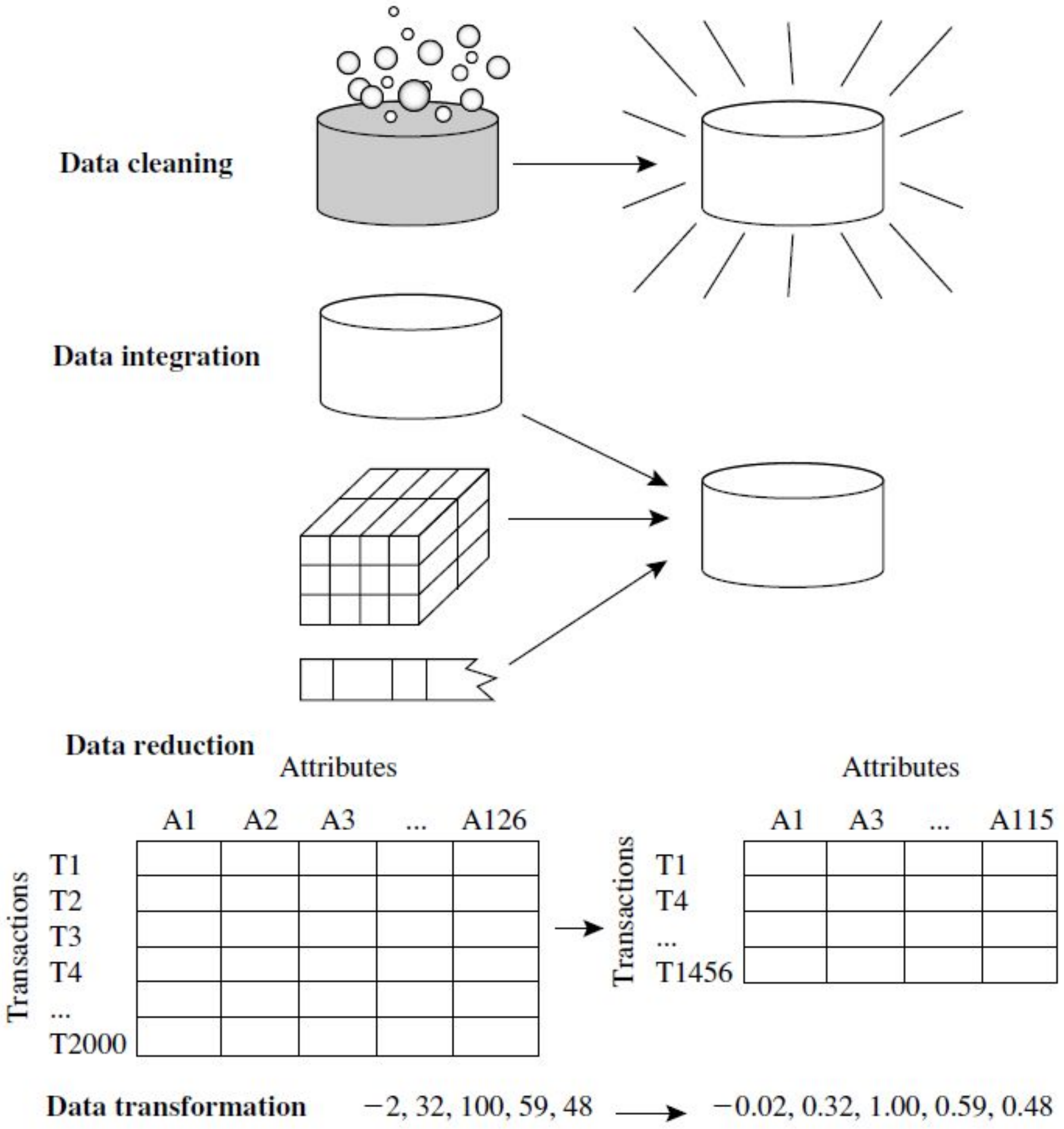
- **Data cleaning**
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies

- **Data integration**
  - Integration of multiple databases, data cubes, or files
    - Eg.: Customer_ID, C_ID
    - FIRST NAME, MIDDL NAME, LAST NAME

Major Tasks in Data Preprocessing

- **Data reduction**
  - Dimensionality reduction
    - To obtain reduce or compressed representation
  - Data compression
    - Wavelet transformation, PCA, Attribute subset selection, Attribute construction
  - Numerosity reduction
    - Replace data by alternative smaller representation
- **Data transformation and data discretization**
  - Normalization
    - [0,1]
  - Concept hierarchy generation
    - Age attribute replace by higher-level concepts such as youth, adult, senior

Forms of data preprocessing



Chapter 3: Data Preprocessing

- 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"
    - 'M', 'm', 'Male', or 'MALE'
    - Did we sell 'aples', 'apples', or 'APPLES' this month.
      - The best way to spot them can be via a frequency chart or making a distinct display of all values in the column.
    - 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?

1. Ignore the tuple:

- usually done when class label is missing (when doing classification)
- not effective when the % of missing values per attribute varies considerably

Developer	Experience	Salary
Java	1	20000
Python	4	🟥
Java	1.5	25000
Python	2	40000
Python	4	80000
🟥	2	🟥

How to Handle Missing Data?

2. Fill in the missing value manually:

- tedious + infeasible?

Student Pr.	Present
80	P
25	P
34	P
31	🟥
0	A
78	P

How to Handle Missing Data?

3. Fill in it automatically with

- a global constant : e.g., “unknown”, a new class?!
- the attribute mean (normal/symmetric distribution)
- median (asymmetric distribution)

Developer	Experience	Salary
Java	1	20000
Python	4	🟥
Java	1.5	25000
Python	2	40000
Python	4	80000
🟥	2	🟥

How to Handle Missing Data?

- Fill in it automatically with
  - the attribute mean for all samples belonging to the same class: smarter (For ex. Mean value of Python developer)
  - the most probable value: inference-based such as Bayesian formula or decision tree

Developer	Experience	Salary
Java	1	20000
Python	4	🟥
Java	1.5	25000
Python	2	40000
Python	4	80000
Java	2	🟥

Noisy Data

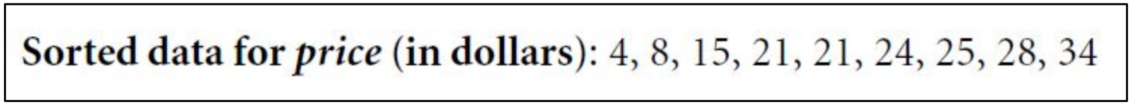
- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which require data cleaning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

Data Inconsistency

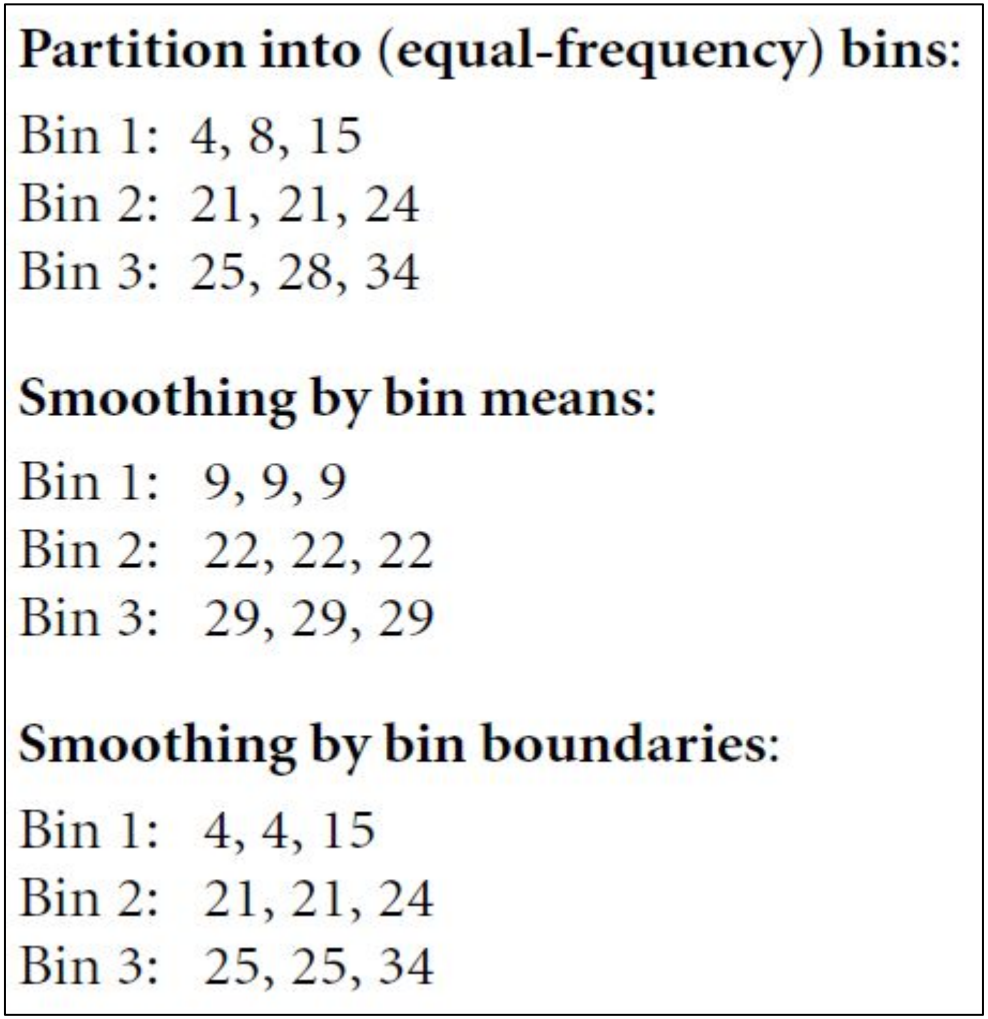
- Example of Data Inconsistency
  - 'M', 'm', 'Male', or 'MALE’,
  - Did we sell 'aples', 'apples', or 'APPLES' this month.
    - The best way to spot them can be via a frequency chart or making a distinct display of all values in the column.
- Some operations on inconsistent records will include but are not limited to:
  - Converting strings to lower or proper case
  - Removing white spaces
  - Renaming column names

How to Handle Noisy Data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.



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

How to Handle Noisy Data?

- Regression
  - smooth by fitting the data into regression functions
    - Linear Regression (best-fit line for two attributes)
    - Multiple linear regression (more than two attributes)
- 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

- *But data cleaning is a big job.*
- *What about data cleaning as a process?*
- *How exactly does one proceed in tackling this task?*
- *Are there any tools out there to help?"*
- *Two steps*
  - *Discrepancy detection*
  - *Data Transformation*

Data Cleaning as a Process

1. Data discrepancy detection

- Use metadata (e.g., domain, range, dependency, distribution)
- Check uniqueness rule
- Consecutive rule (e.g. cheque number)
- Null rule
- Use commercial tools
  - **Data scrubbing:** use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
    - For example, if the email-id stored is Vipul.gmail.com then correct it to Vipul@gmail.com
  - **Data auditing:** by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)
    - After performing data auditing, we discover
      - Missing values (column-wise)
      - Invalid mobile (because of 9 digits stored for mobile number)
      - Inconsistent Name ("Vadodara" and "Baroda" used for same city)

Data Cleaning as a Process

2. Data transformation

- Data migration and integration
  - **Data migration tools:** allow transformations to be specified
    - E.g replace string “gender” by “sex”
  - **ETL (Extraction/Transformation/Loading) tools:** allow users to specify transformations through a graphical user interface

- Integration of the two processes
  - Iterative and interactive

Chapter 3: Data Preprocessing

- 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

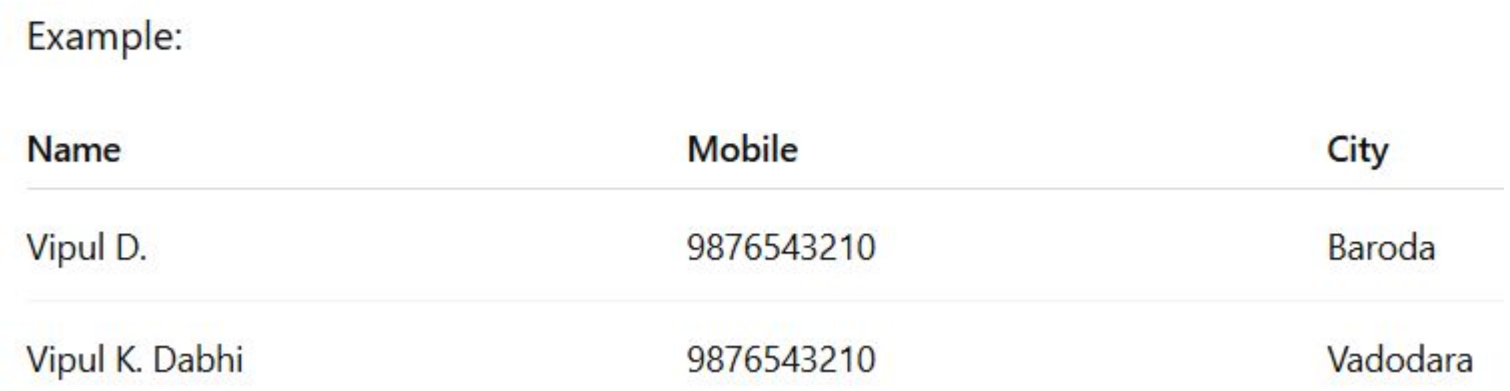
- **Data integration**:
  - Combines data from multiple sources into a coherent store

1. Entity identification problem
2. Redundancy and Correlation Analysis
3. Tuple duplication
4. Data value conflict detection and resolution

1. Entity identification problem:

- How can equivalent real-world entities from multiple data sources be matched up?
- Schema integration: e.g., A.cust-id ≡ B.cust-#
  - Integrate metadata from different sources
- Object matching:
  - Identify real world entities from multiple data sources
  - e.g., Bill Clinton = William Clinton
- attribute values from different sources are different so use of metadata
- Possible reasons: different representations, different scales,
  - E.g one system , a **discount** applied to the whole order amount, in another system **each individual line item is discounted.**

Entity Identification



Example:

Name	Mobile	City
Vipul D.	9876543210	Baroda
Vipul K. Dabhi	9876543210	Vadodara

- These two rows are clearly referring to the same person (entity) despite differences.
- **Entity Identification detects this redundancy**.
- Correlation analysis needs **clean, unique entities** to avoid bias.

2. Redundancy and Correlation Analysis

- 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

Object Identification

- Identify *distinct objects* or *distinct features/attributes* within the data.
- For example:
  - "price" and "cost" features (columns) may be duplicates
  - "experience_years" and "total_experience" may be the same feature
- If both are included:
  - correlation becomes artificially high
  - multicollinearity appears
  - ML models become unstable

Correlation Analysis (Numeric Data)

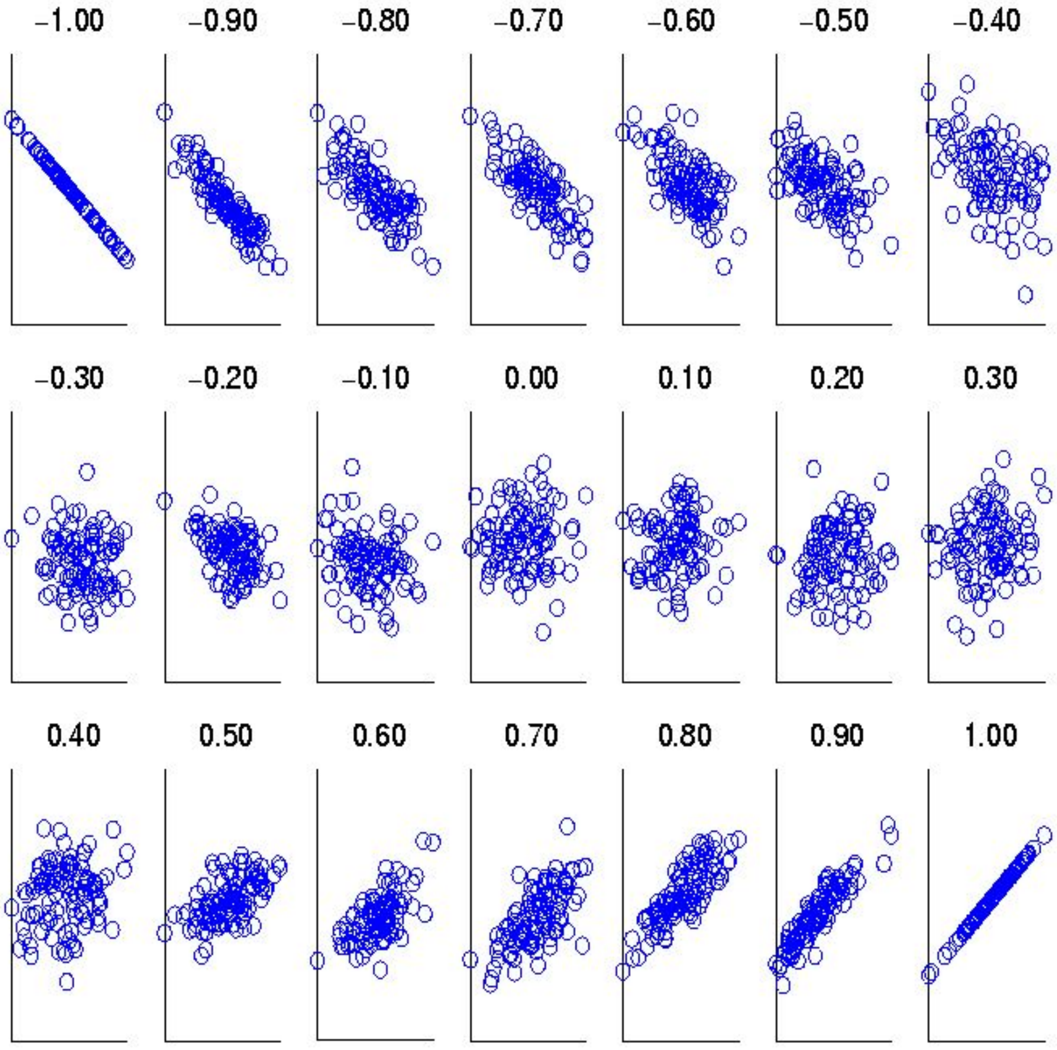
- Correlation coefficient (also called Pearson's product moment coefficient)

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

$$r(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

  - where n is the number of tuples, $\overline{A}$ and $\overline{B}$ 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

Visually Evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

Covariance (Numeric Data)

- Covariance is a statistical measure that quantifies the degree to which two variables change together.
- It indicates the direction of the linear relationship between variables.
- Covariance is the foundation for calculating the correlation coefficient, which is a standardized measure of linear relationship and easier to interpret because it is unitless.

- **Positive Covariance**: If two variables increase or decrease together, their covariance is positive.
- **Negative Covariance**: If one variable increases while the other decreases, their covariance is negative.
- **Zero Covariance**: If there is no relationship between the two variables, their covariance is close to zero.

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, $\overline{A}$ and $\overline{B}$ 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**: $Cov_{A,B} = 0$
- **Variance is special case of Covariance**

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) = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4
- Thus, A and B rise together since Cov(A, B) > 0.

Covariance

- Covariance tells you whether two variables **increase/decrease together**
  - **Positive covariance** → when X increases, Y also increases
  - **Negative covariance** → when X increases, Y decreases
  - **Zero covariance** → no linear relationship
- Problem
  - Covariance **depends on the scale** of the variables.
  - Example
    - Covariance between **height (cm)** and **weight (kg)** will be much larger in absolute value than if height is measured in **meters.**

Covariance and Correlation

- CoVariance:
  - Relation between X and Y (Direction)
  - Its magnitude is not easily interpretable due to the unit dependence.

- Correlation
  - How strongly X and Y are related?

Correlation

- Correlation = **normalized covariance.**
- Interpretation
  - **+1** → perfect positive linear relationship
  - **0** → no linear relationship
  - **−1** → perfect negative linear relationship
- Advantages
  - Removes scale effects
  - Unitless
  - Comparable across datasets
  - Always between −1 and +1

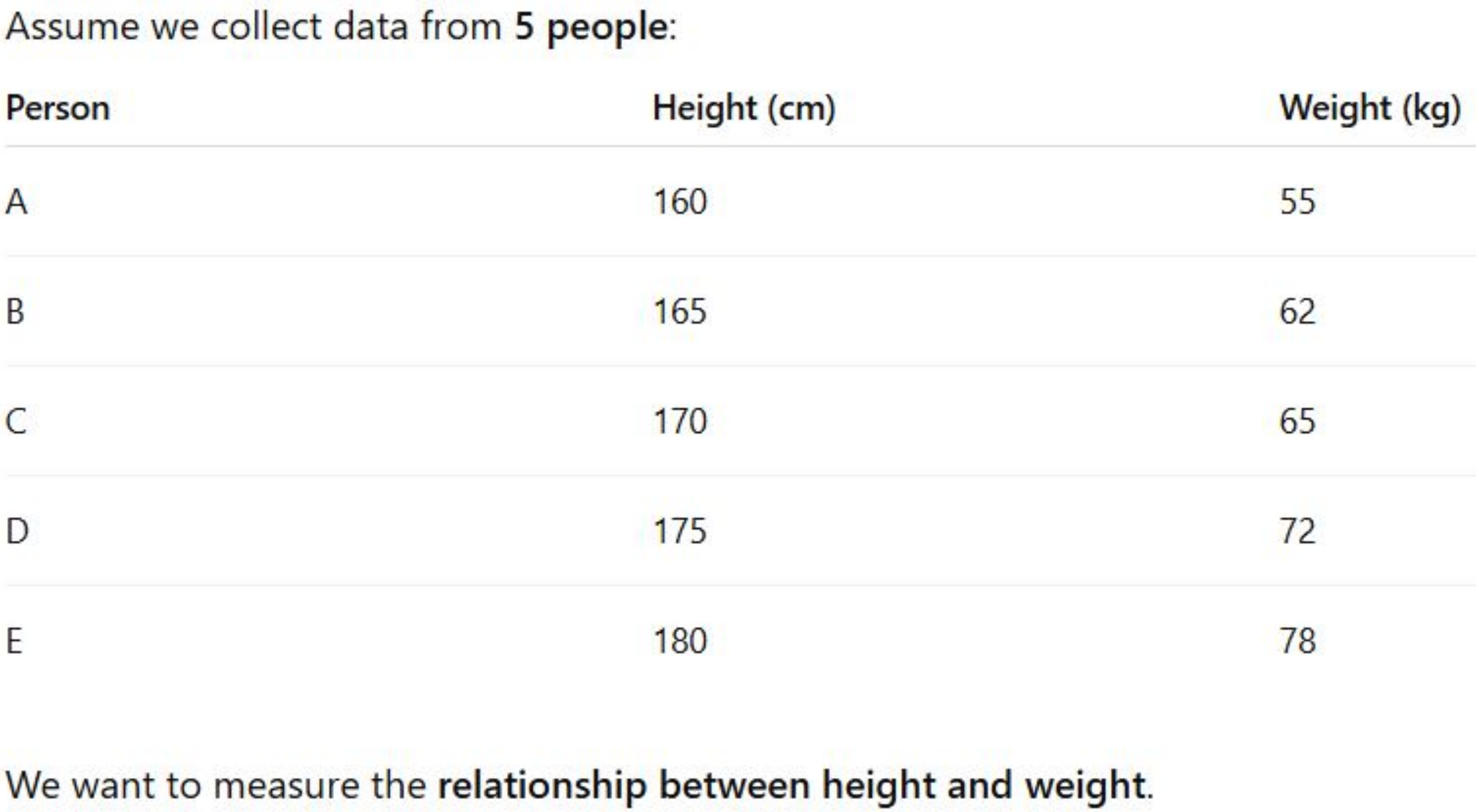
$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

Where

σX = standard deviation of X

σY = standard deviation of Y

Calculate Covariance and Correlation

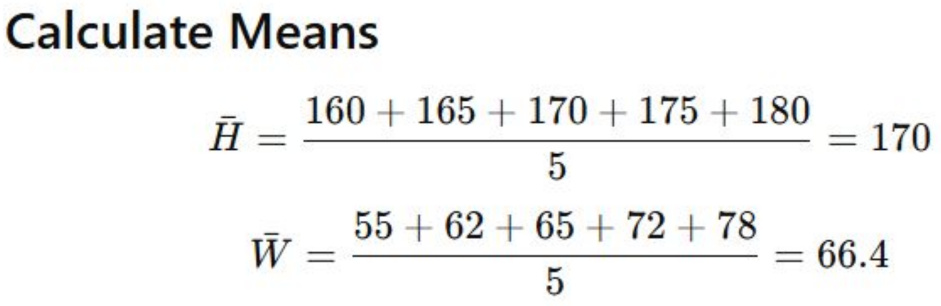


Assume we collect data from **5 people**:

Person	Height (cm)	Weight (kg)
A	160	55
B	165	62
C	170	65
D	175	72
E	180	78

We want to measure the **relationship between height and weight**.

Step 1 : Calculate Means

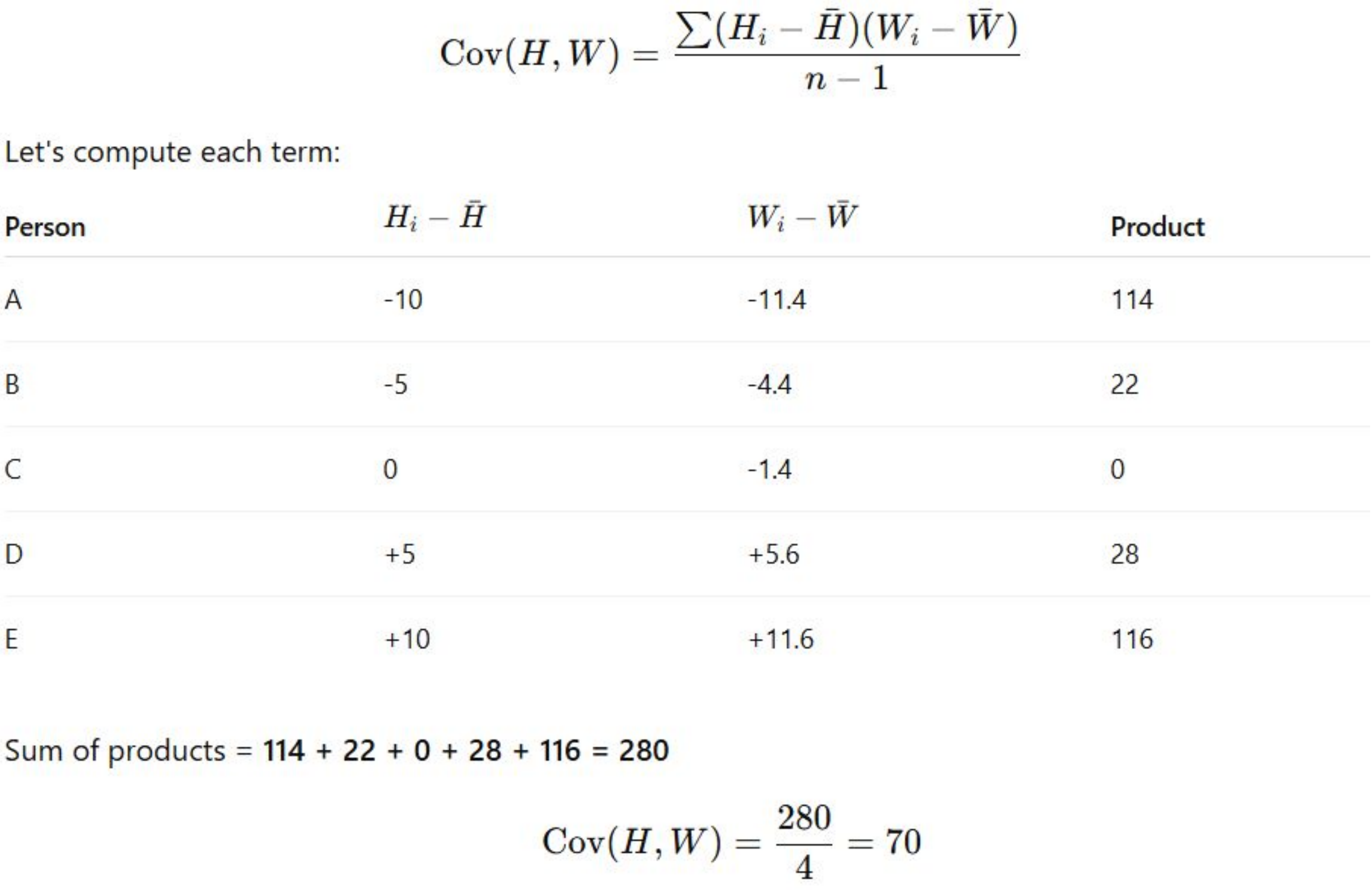


Calculate Means

$$\bar{H} = \frac{160 + 165 + 170 + 175 + 180}{5} = 170$$

$$\bar{W} = \frac{55 + 62 + 65 + 72 + 78}{5} = 66.4$$

Step 2 : Calculate Covariance



$$\text{Cov}(H,W) = \frac{\sum(H_i - \bar{H})(W_i - \bar{W})}{n-1}$$

Let's compute each term:

Person	$H_i - \bar{H}$	$W_i - \bar{W}$	Product
A	-10	-11.4	114
B	-5	-4.4	22
C	0	-1.4	0
D	+5	+5.6	28
E	+10	+11.6	116

Sum of products = **114 + 22 + 0 + 28 + 116 = 280**

$$\text{Cov}(H,W) = \frac{280}{4} = 70$$

Interpretation of Covariance = **+70.** Positive ⇒ *height and weight increase together*

Compute Standard Deviation

Variance of Height:

$$\mathrm{Var}(H) = \frac{(-10)^2 + (-5)^2 + 0^2 + 5^2 + 10^2}{4} = \frac{250}{4} = 62.5$$

$$\sigma_H = \sqrt{62.5} = 7.9057$$

Variance of Weight:

$$\mathrm{Var}(W) = \frac{(-11.4)^2 + (-4.4)^2 + (-1.4)^2 + 5.6^2 + 11.6^2}{4}$$

$$= \frac{129.96 + 19.36 + 1.96 + 31.36 + 134.56}{4} = \frac{317.2}{4} = 79.3$$

$$\sigma_W = \sqrt{79.3} = 8.906$$

Calculate Correlation

$$\rho = \frac{70}{(7.9057)(8.906)} = \frac{70}{70.42} = 0.9959$$

Interpretation of Correlation = 0.996

Very strong positive relationship
Taller people tend to weigh more
Since correlation is close to **1**, the relationship is almost perfectly linear

3. Tuple Duplication

- Due to denormalized tables

4. Data value conflict detection and resolution

- Detection and resolution of data value conflict.
  - E.g. height in cm and height in inch/feet
  - temp in .C and temp in .F
- Difference due to abstract level
  - “**Total_sales**” at branch level
  - “**Total_sales**” at region level

Chapter 3: Data Preprocessing

- 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

- **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.
- **Curse of dimensionality**

Data Reduction 1: 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

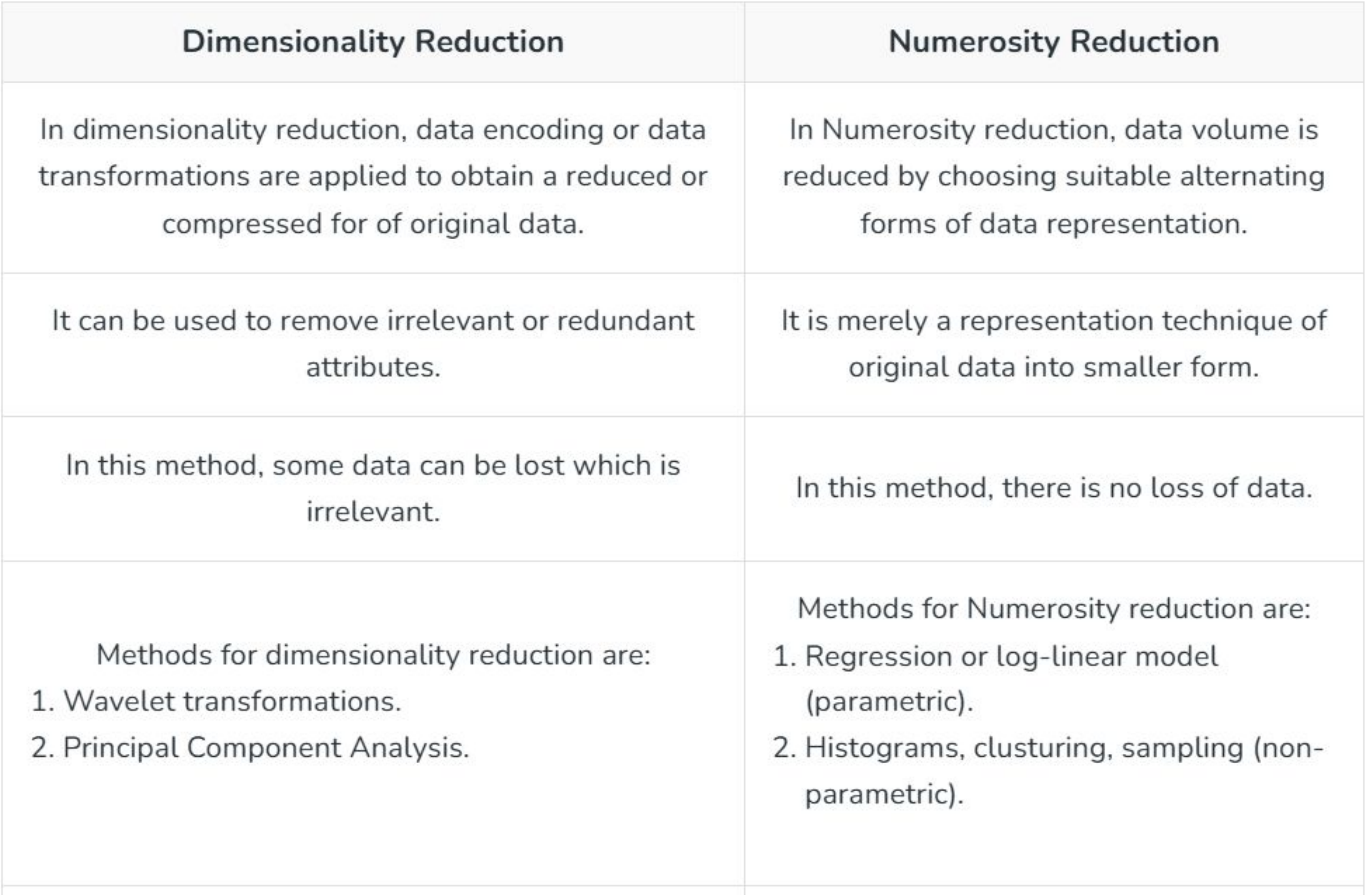
Data Reduction Strategies

- 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 Strategies

- Dimensionality Reduction
  - In dimensionality reduction, data encoding or transformations are used to access a reduced or “compressed” depiction of the original data.
  - If the original data can be regenerated from the compressed data without any loss of data, the data reduction is known as lossless.
  - If data reconstructed is only approximated of the original data, then the data reduction is called lossy.
- Numerosity Reduction
  - In the numerosity reduction, the data volume is decreased by selecting an alternative, smaller form of data representation.
  - These techniques can be parametric or nonparametric.
  - For parametric methods, a model can estimate the data, so that only the data parameters need to be saved, instead of the actual data, for example, Log-linear models. Non-parametric methods are used for storing a reduced representation of the data which include histograms, clustering, and sampling.

Data Reduction Strategies



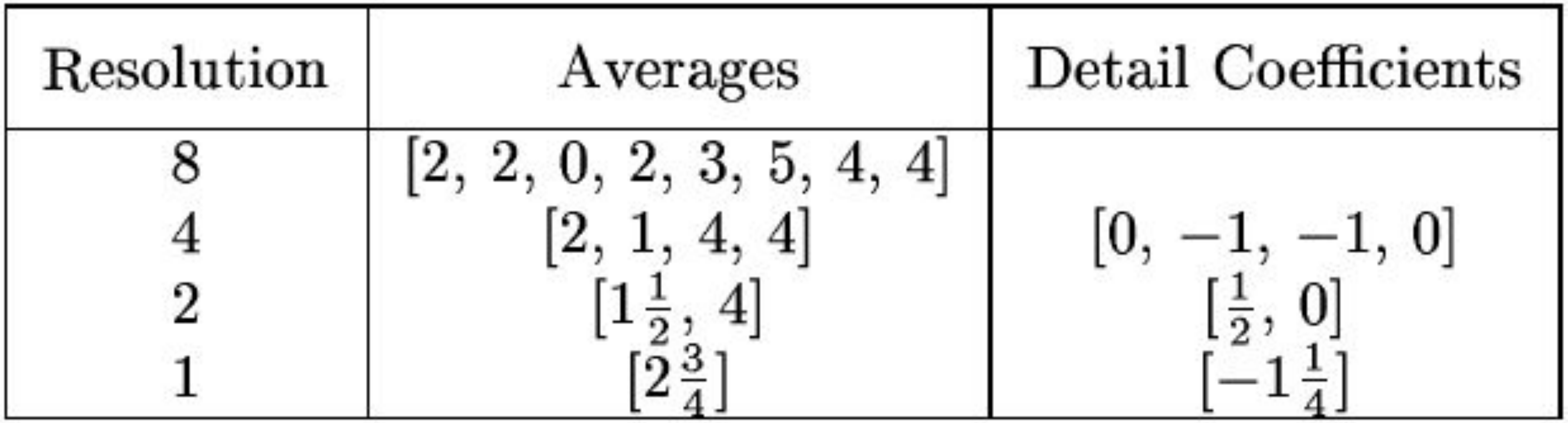
Dimensionality Reduction	Numerosity Reduction
In dimensionality reduction, data encoding or data transformations are applied to obtain a reduced or compressed for of original data.	In Numerosity reduction, data volume is reduced by choosing suitable alternating forms of data representation.
It can be used to remove irrelevant or redundant attributes.	It is merely a representation technique of original data into smaller form.
In this method, some data can be lost which is irrelevant.	In this method, there is no loss of data.
Methods for dimensionality reduction are: 1. Wavelet transformations. 2. Principal Component Analysis.	Methods for Numerosity reduction are: 1. Regression or log-linear model (parametric). 2. Histograms, clusturing, sampling (non-parametric).

What Is Wavelet Transform?

- Signal processing technique that, when applied to a data vector X, transforms it to a numerically different vector X’ of wavelet coefficients.
- Both are of same length
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Store only a small fraction of the strongest of the wavelet coefficients
- Used for image compression

Wavelet Decomposition

- Wavelets: A math tool for space-efficient hierarchical decomposition of functions
- S = [2, 2, 0, 2, 3, 5, 4, 4] can be transformed to $S_\wedge$ = $[2^3/_4, -1^1/_4, {}^1/_2, 0, 0, -1, -1, 0]$
- Compression: many small detail coefficients can be replaced by 0’s, and only the significant coefficients are retained



Resolution	Averages	Detail Coefficients
8	$[2, 2, 0, 2, 3, 5, 4, 4]$	
4	$[2, 1, 4, 4]$	$[0, -1, -1, 0]$
2	$[1\frac{1}{2}, 4]$	$[\frac{1}{2}, 0]$
1	$[2\frac{3}{4}]$	$[-1\frac{1}{4}]$

Wavelet Transformation

- Method:
  - Length, L, must be an integer power of 2 (padding with 0's, when necessary)
  - Each transform has 2 functions: smoothing, difference
  - Applies to pairs of data, resulting in two set of data of length L/2
  - Applies two functions recursively, until reaches the desired length

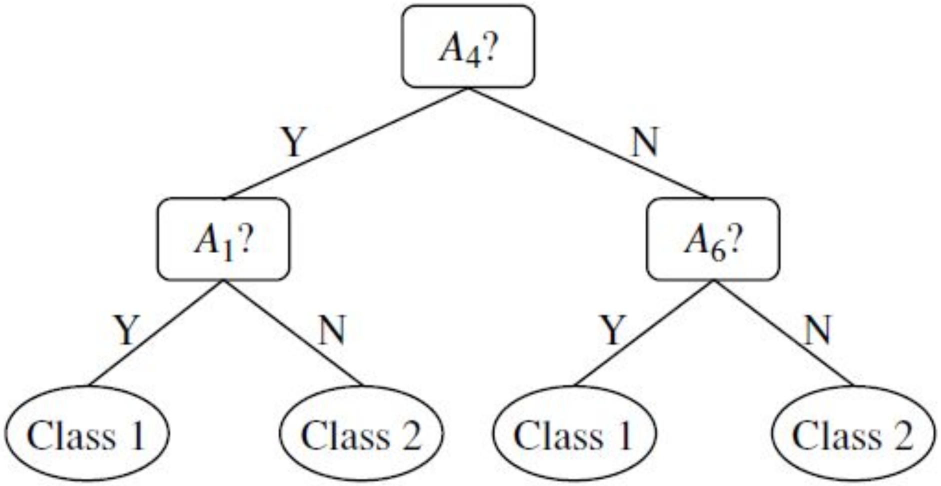
Why Wavelet Transform?

- Effective removal of outliers
  - Insensitive to noise, insensitive to input order
- Efficient
  - Complexity O(N)
- Only applicable to low dimensional data

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 GST 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 methods in Attribute Selection

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: { } => $\{A_1\}$ => $\{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  A_4? Y N A_1? A_6? Y N Y N Class 1 Class 2 Class 1 Class 2 => Reduced attribute set: $\{A_1, A_4, A_6\}$

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 forward selection:**
    - The best single-attribute is picked first
    - Then next best attribute condition to the first, ...
  - **Step-wise attribute elimination:**
    - Repeatedly eliminate the worst attribute
  - **Best combined attribute selection and elimination**
  - **Optimal branch and bound:**
    - Use attribute elimination and backtracking

Attribute Creation (Feature Generation)

- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones

- E.g. Add **area** attribute based on **height** and **width** of the shape.

2. Numerosity Reduction

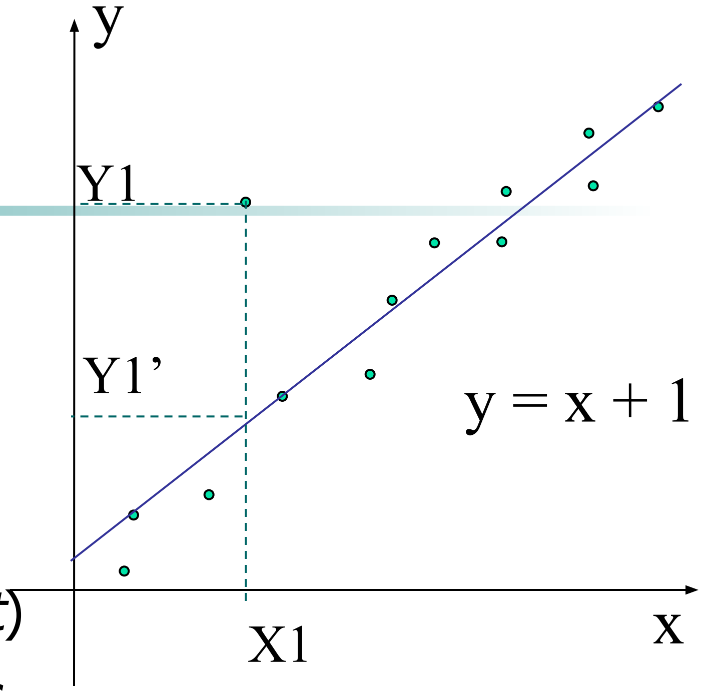
Data Reduction 2: Numerosity Reduction

- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods** (e.g., regression)
  - Assume the data fits some model,
  - estimate 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

Regress Analysis and Log-Linear Models

- Linear regression: $Y = w X + b$
  - Two regression coefficients, *w* and *b*, specify the line and are to be estimated by using the data at hand
  - Using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.
  - *Example: Predicting house price based on square feet only*
- Multiple regression: $Y = b_0 + b_1 X_1 + b_2 X_2$
  - Many nonlinear functions can be transformed into the above
  - Example: Predicting house price based on square feet, location, age of house etc.
- Log-linear models:
  - Approximate discrete multidimensional probability distributions
  - Estimate the probability of each point (tuple) in a multi-dimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations
  - Useful for dimensionality reduction and data smoothing

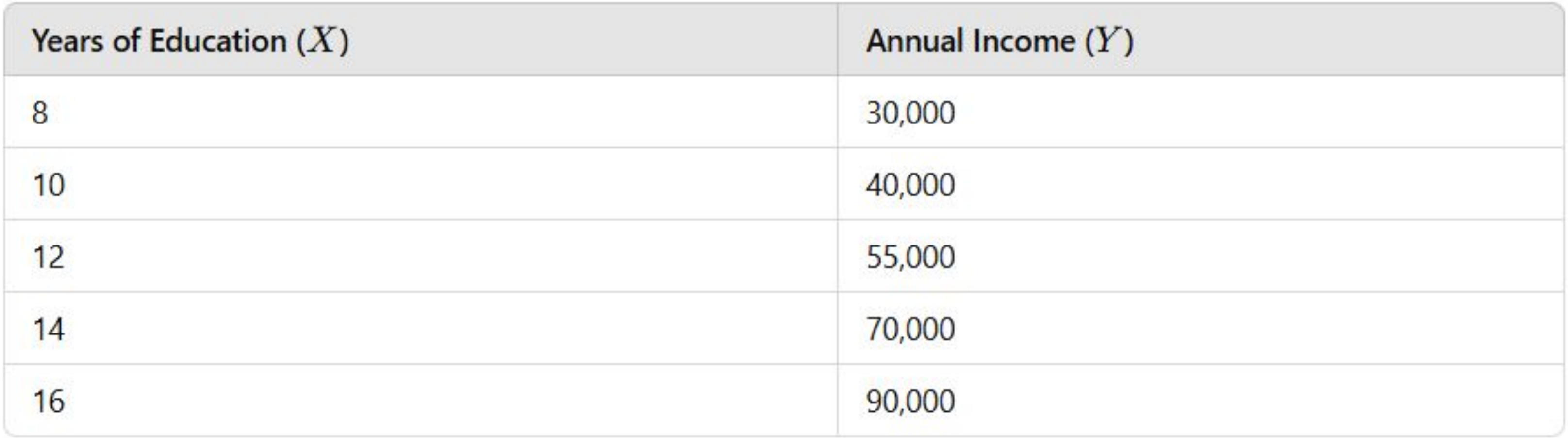
Regress Analysis and Log-Linear Models

- *These models are commonly used in economics, finance, biology, and other fields when the response variable exhibits exponential growth or follows a non-linear trend that can be linearized by applying a logarithmic transformatio*

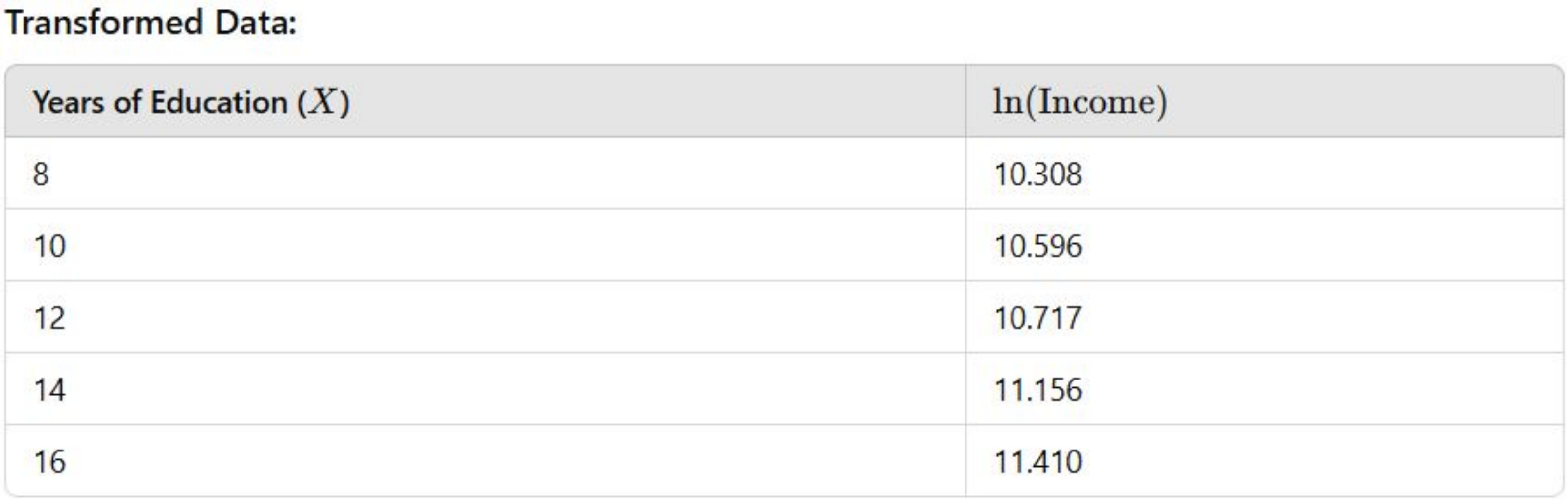
$$\ln(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

- *Example*
  - *Study how the level of education (in years) affects annual income.*
  - *However, the relationship between education and income is not linear; instead, income tends to grow exponentially with additional years of*

Regress Analysis and Log-Linear Models



Years of Education (X)	Annual Income (Y)
8	30,000
10	40,000
12	55,000
14	70,000
16	90,000



Transformed Data:

Years of Education (X)	$\ln(\text{Income})$
8	10.308
10	10.596
12	10.717
14	11.156
16	11.410

Regress Analysis and Log-Linear Models

- *The model which fits the data is (where B0 = 7.0, B1 = 0.2)*

$$\ln(\text{Income}) = 7.0 + 0.2 \times \text{Education}$$

- *For each additional year of education, the estimated percentage change in income is proportional to $e^{0.2}$ ≈1.221 or about 22.1% increase.*

$$Y = e^{\ln(\hat{\text{Income}})} = e^{7.0+0.2\times\text{Education}}$$

- *To get predictions for Y (actual income) instead of ln(Y), transform back using the exponential function:*

Histogram Analysis

- Partitioning rules:
  - Equal-width: equal bucket range
  - Equal-frequency (or equal-depth)

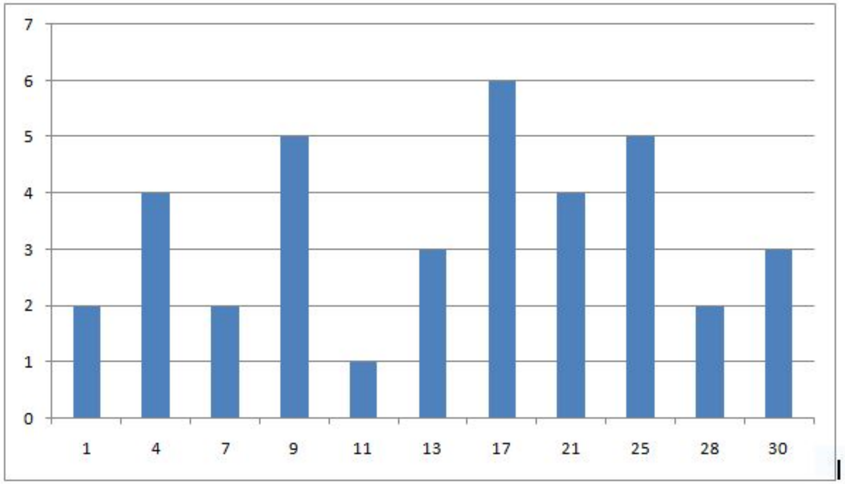


Figure 1 Histogram using price where one bucket represents one value

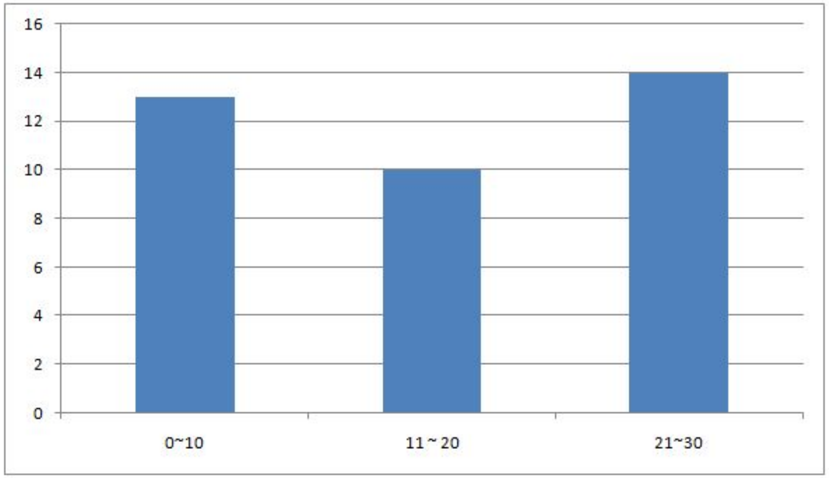


Figure 2 : Equal width Histogram

Histogram Analysis

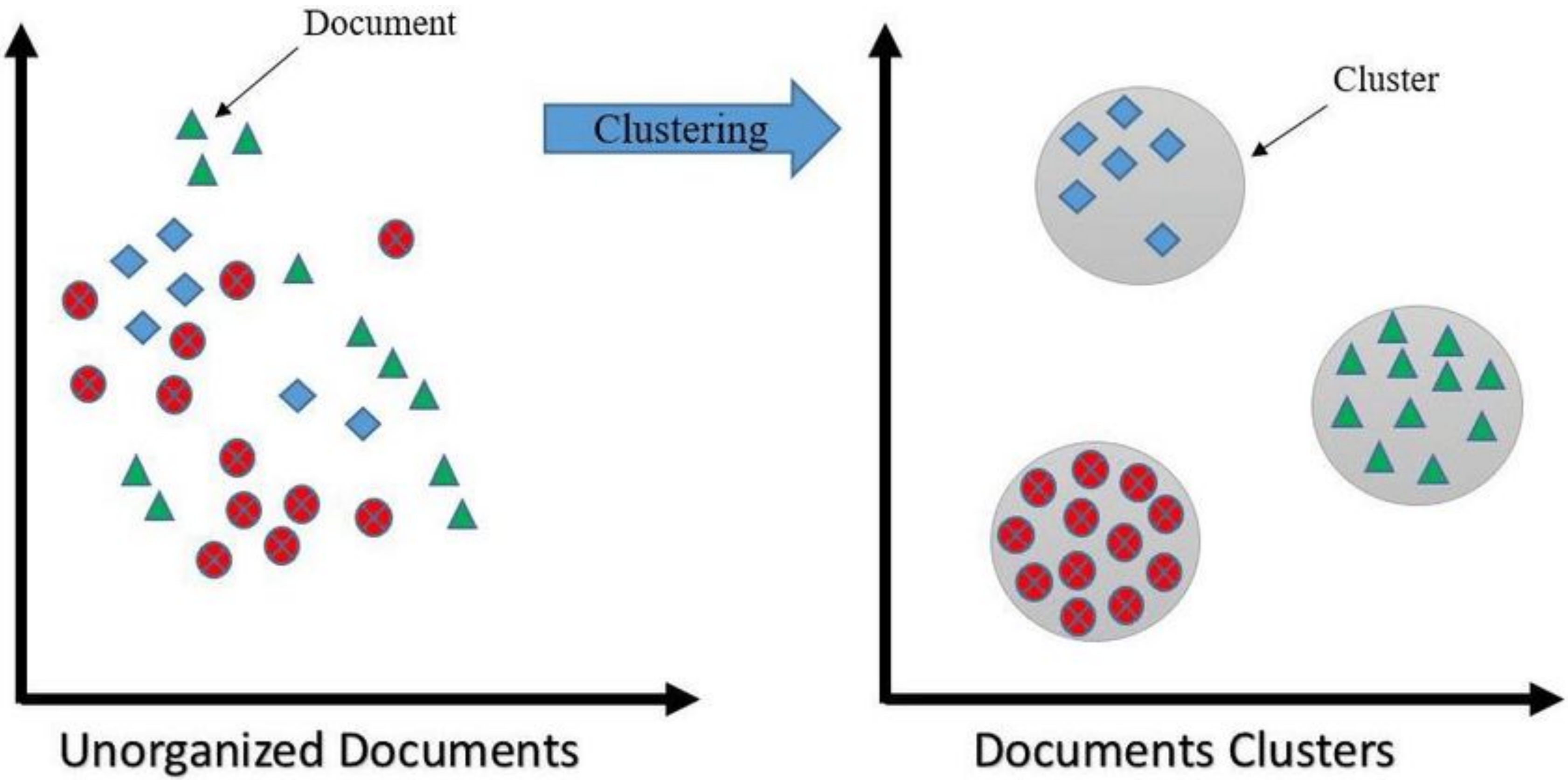
- Equal Frequency Binning
  - Equal-frequency binning divides the values into bins that have the same number of observations or frequency. For example, if we have 100 observations and we want 10 bins, each bin will have 10 observations.

- If the data is symmetric and evenly distributed, equal-width binning may be more suitable; if the data is skewed or has outliers, equal-frequency binning may be more appropriate.

Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms
- Cluster analysis will be studied in depth in Chapter 10

Clustering

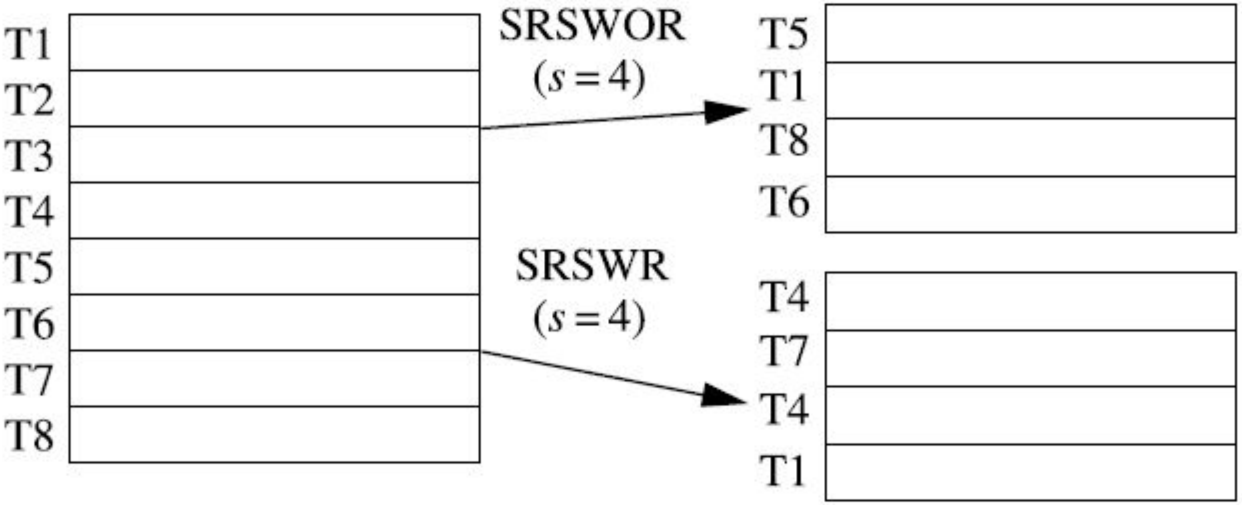

Document
Clustering
Cluster
Unorganized Documents
Documents Clusters

Sampling

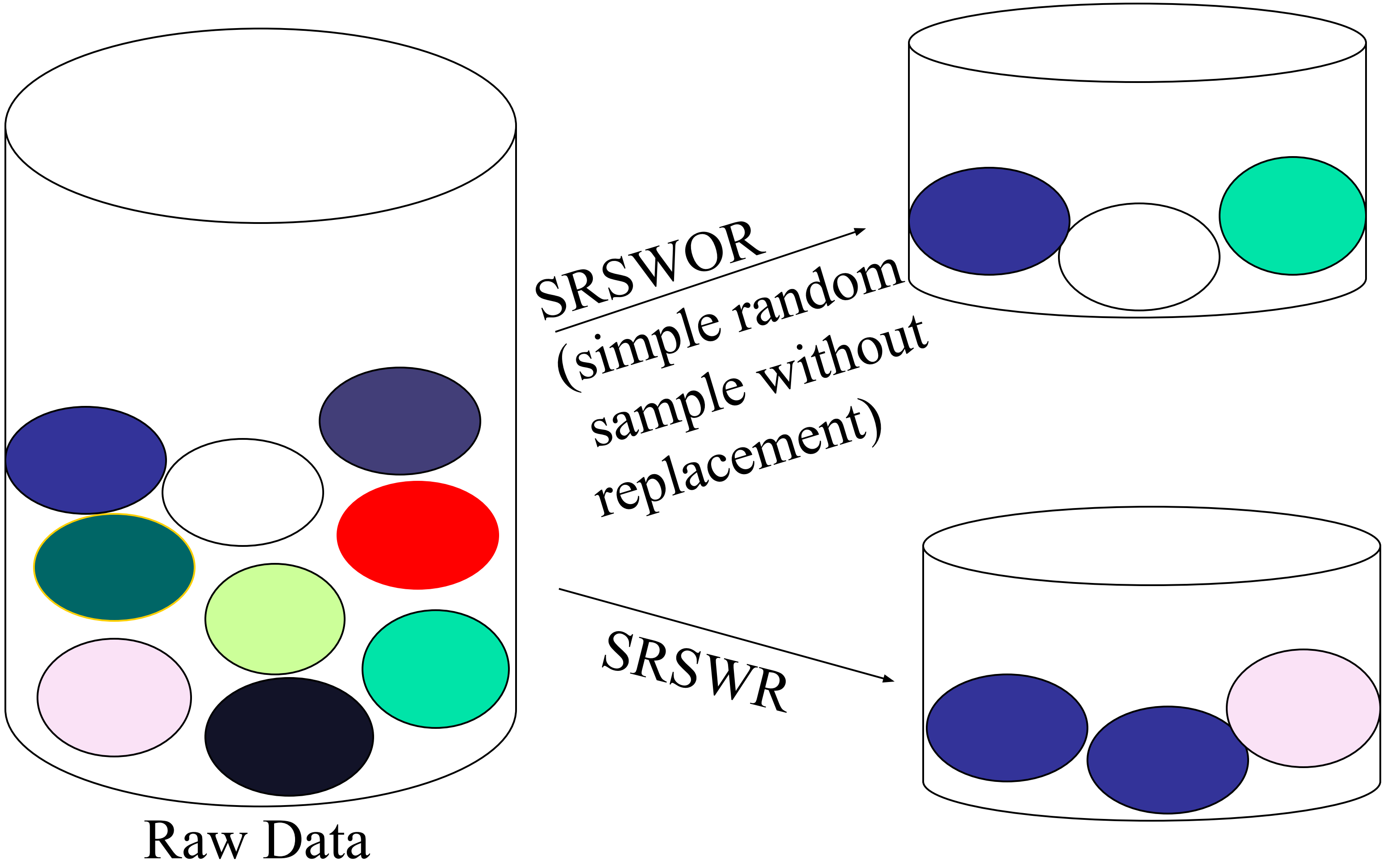
- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a representative subset of the data
    - Simple random sampling may have very poor performance in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

Types of Sampling

- **Simple random sample without replacement (SRSWOR) of size:**
  - Once an object is selected, it is removed from the population
- **Simple random sample with replacement (SRSWR) of size**
  - A selected object is not removed from the population

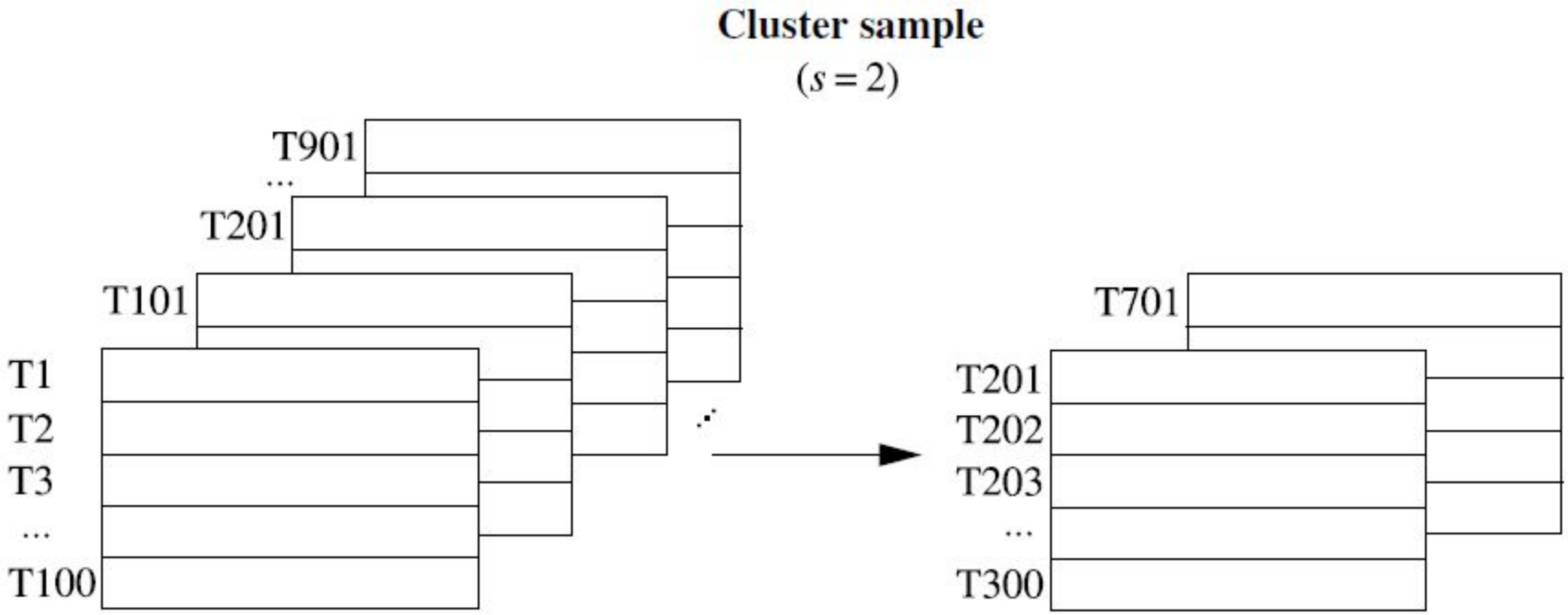


Sampling: With or without Replacement



Types of Sampling

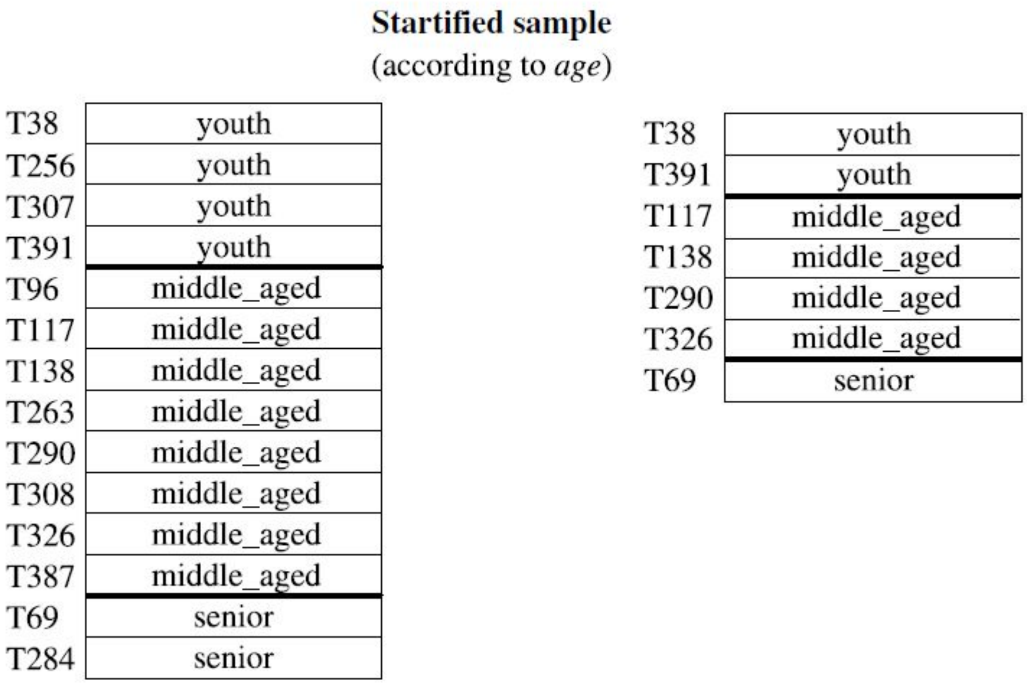
- **Cluster sample**:



Imagine a researcher wants to analyze the water usage of households in a large city. Surveying every household Individually would be costly and time-consuming. Therefore, divide city into clusters based on neighborhoods. Each neighborhood forms a cluster containing several households.

Types of Sampling

- **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



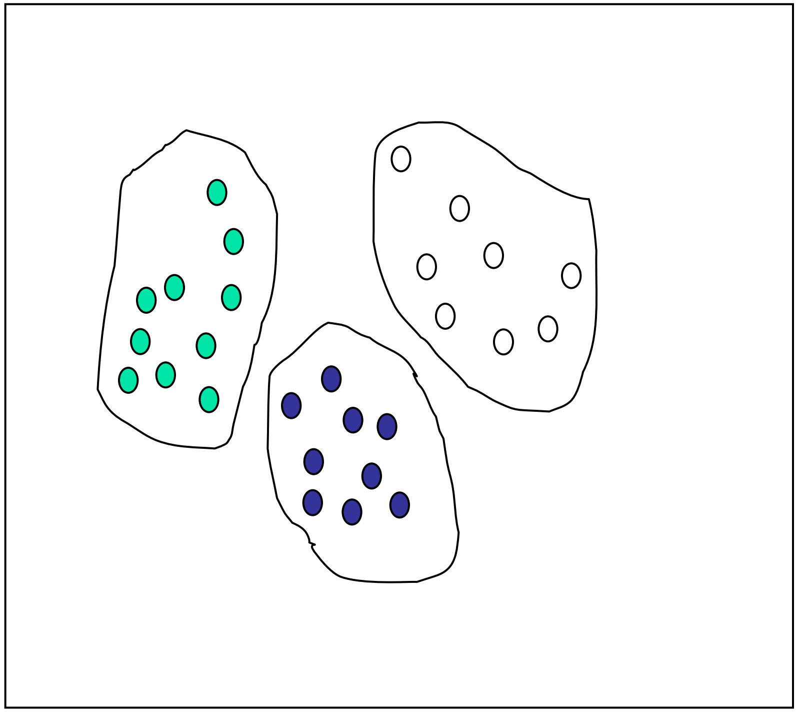
Startified sample
(according to *age*)

T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

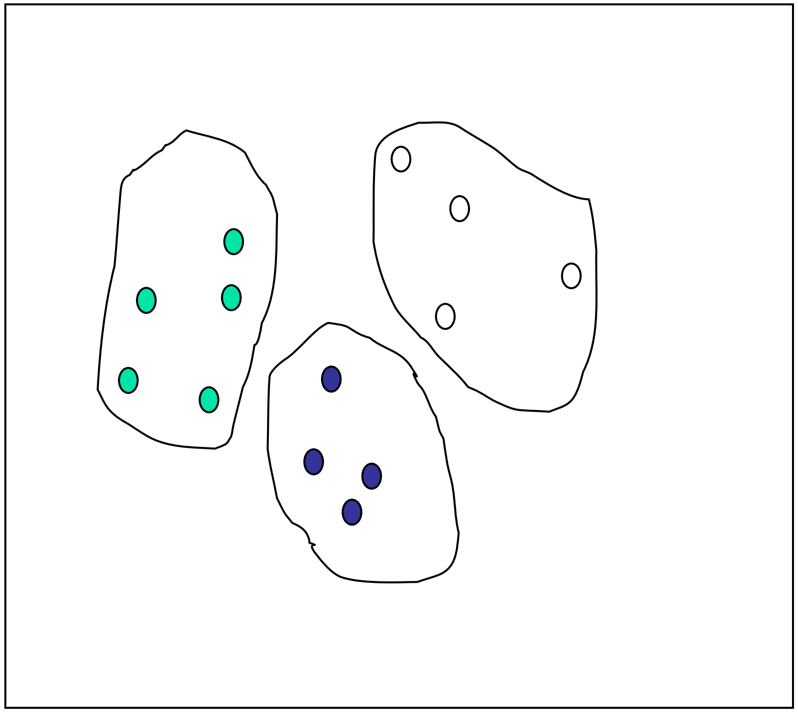
T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

Sampling: Stratified Sampling

Raw Data



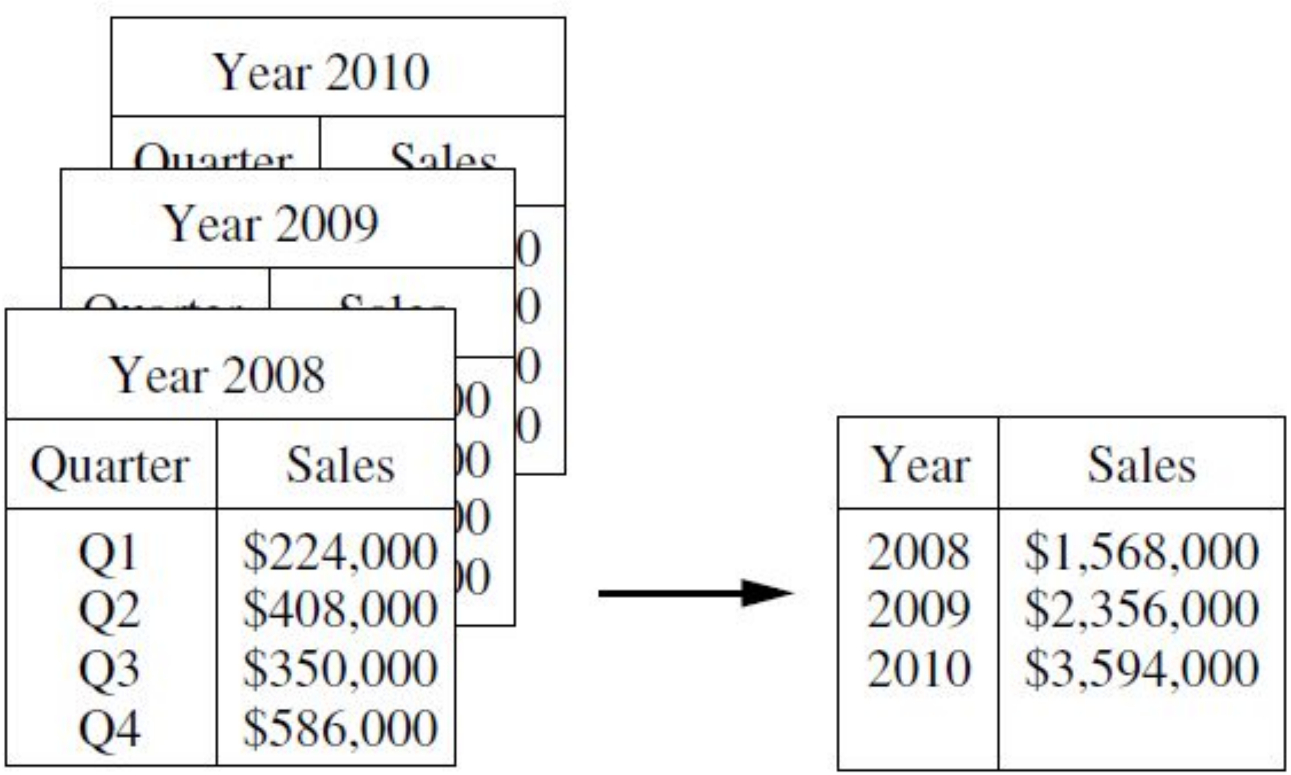
Stratified Sample



Data Cube Aggregation

- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

Data Aggregation



Year 2010

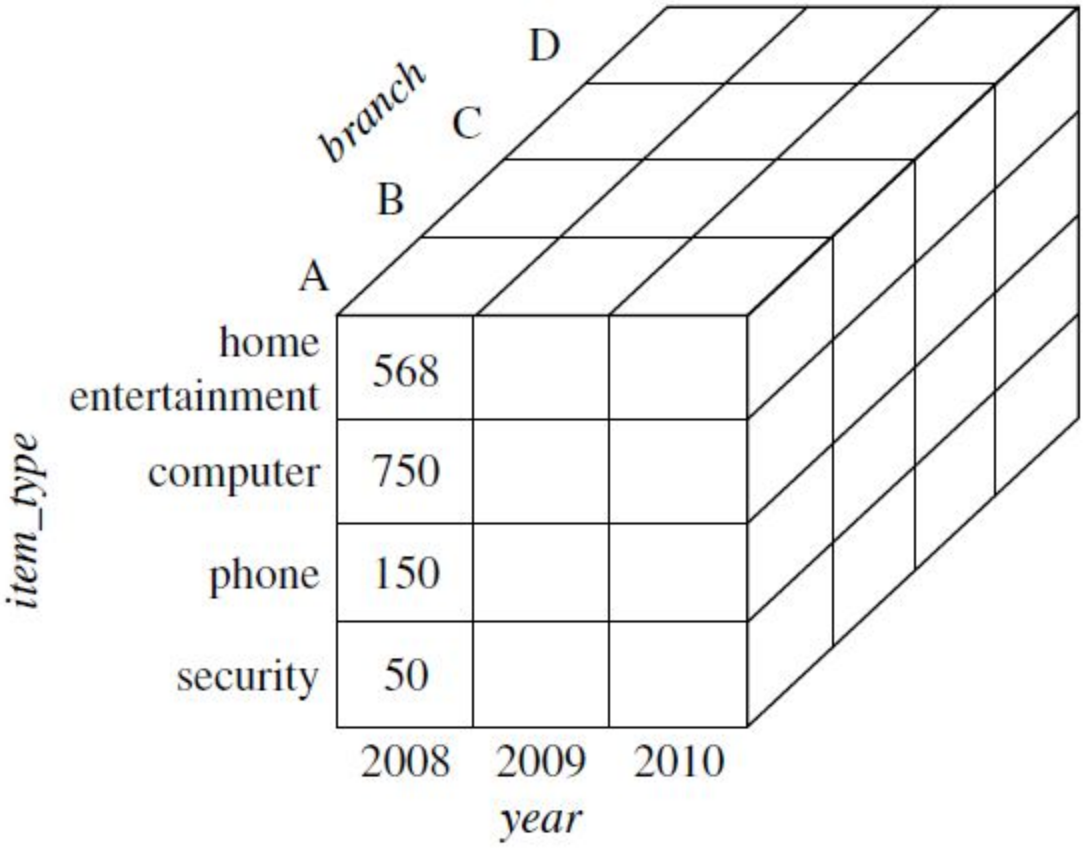
Year 2009

Year 2008

Quarter	Sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

Year	Sales
2008	$1,568,000
2009	$2,356,000
2010	$3,594,000

Data cube Aggregation

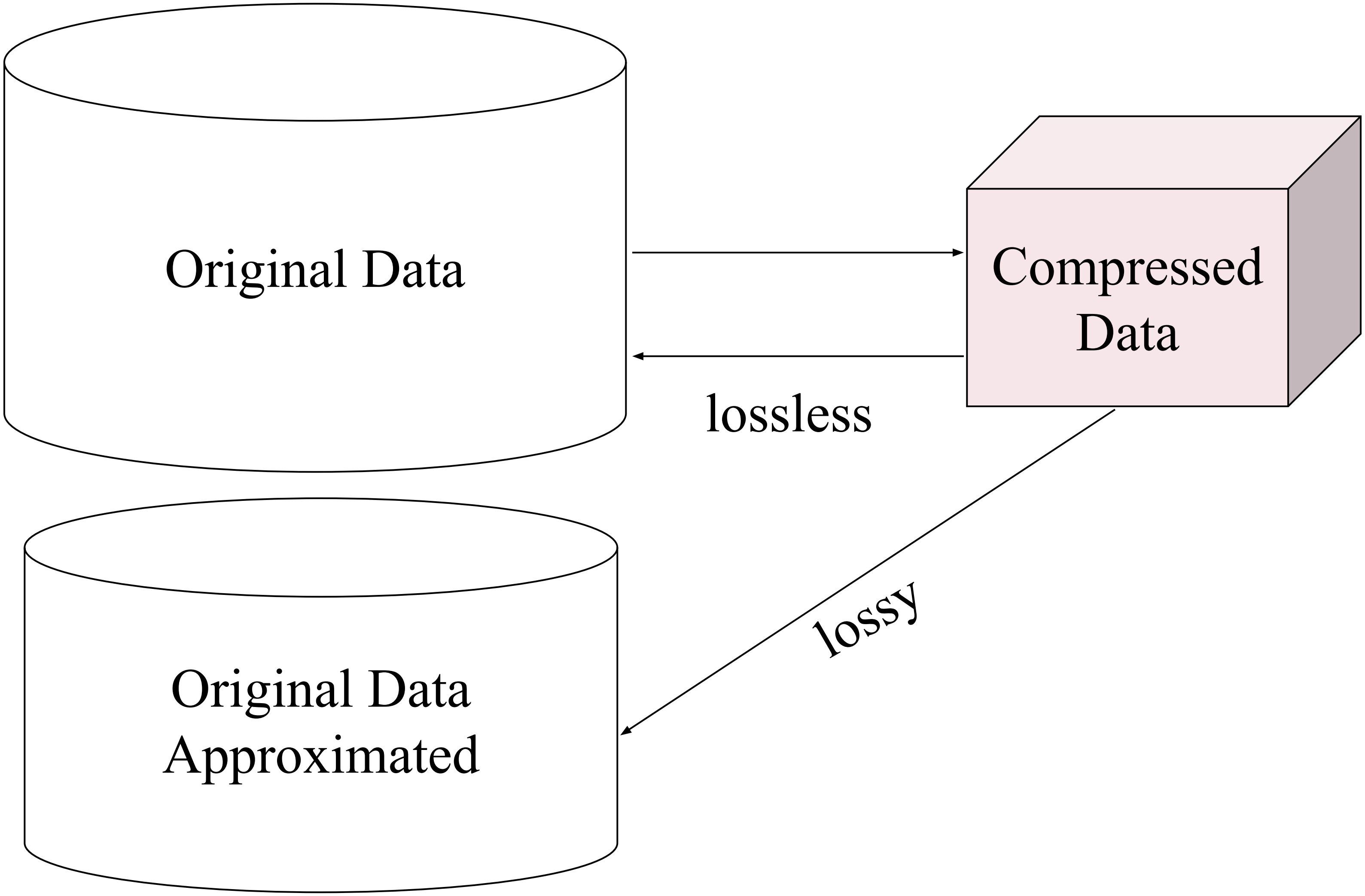


	2008	2009	2010
home entertainment	568		
computer	750		
phone	150		
security	50		

Data Reduction 3: 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
- Dimensionality and numerosity reduction may also be considered as forms of data compression

Data Compression



Chapter 3: Data Preprocessing

- 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
- 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 [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 \$73,600 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 μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- **Normalization by decimal scaling**

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

  - Ex. A range from -986 to 917. So|-986| = 968 so, V' = -986/1000 = -0.986

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 7
- 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

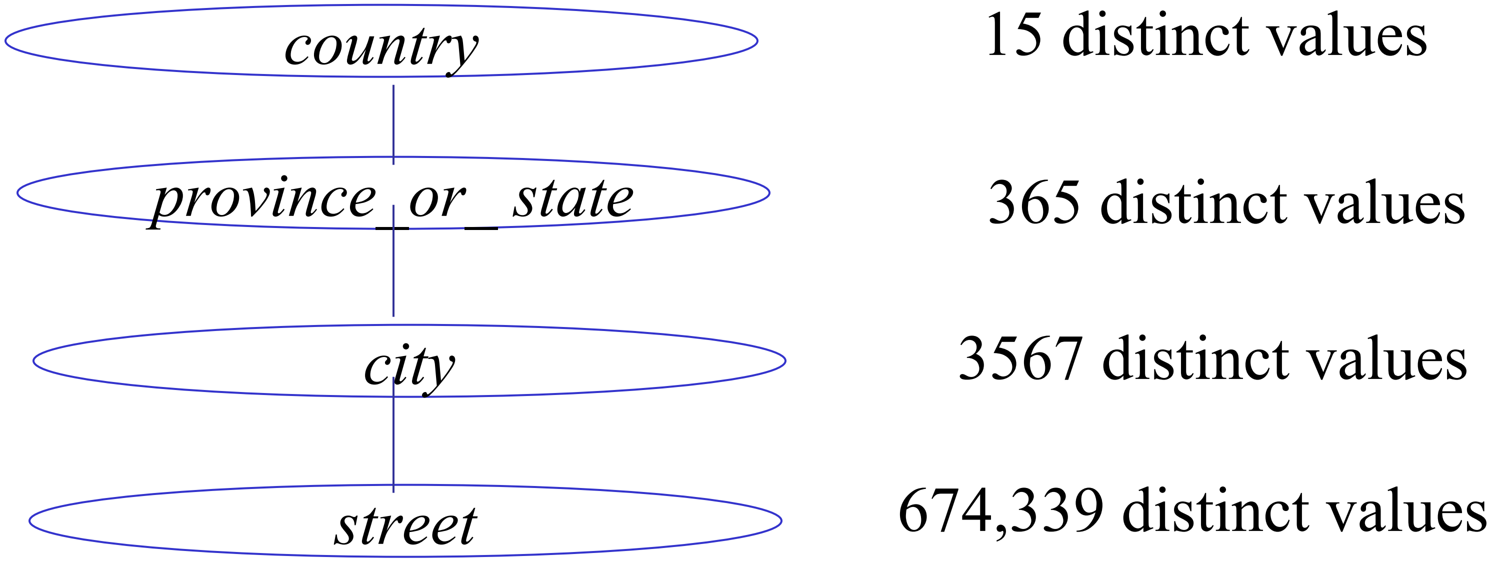
- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: Recursively 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. For numeric data, use discretization methods shown.

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
  - Price range grouping or grouping by location i.e. north INDIA, SOUTH INDIA etc.
- 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



country — 15 distinct values

province_or_state — 365 distinct values

city — 3567 distinct values

street — 674,339 distinct values

Chapter 3: Data Preprocessing

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

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

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Comm. of ACM, 42:73-78, 1999
- A. Bruce, D. Donoho, and H.-Y. Gao. Wavelet analysis. *IEEE Spectrum*, Oct 1996
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley, 2003
- J. Devore and R. Peck. *Statistics: The Exploration and Analysis of Data*. Duxbury Press, 1997.
- H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. *VLDB'01*
- M. Hua and J. Pei. Cleaning disguised missing data: A heuristic approach. *KDD'07*
- H. V. Jagadish, et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), Dec. 1997
- H. Liu and H. Motoda (eds.). *Feature Extraction, Construction, and Selection: A Data Mining Perspective*. Kluwer Academic, 1998
- J. E. Olson. *Data Quality: The Accuracy Dimension*. Morgan Kaufmann, 2003
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'2001
- T. Redman. *Data Quality: The Field Guide*. Digital Press (Elsevier), 2001
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995