COMP5121
Data Mining and Data Warehousing Applications

Week 3: Data Preprocessing

Dr. Fengmei Jin

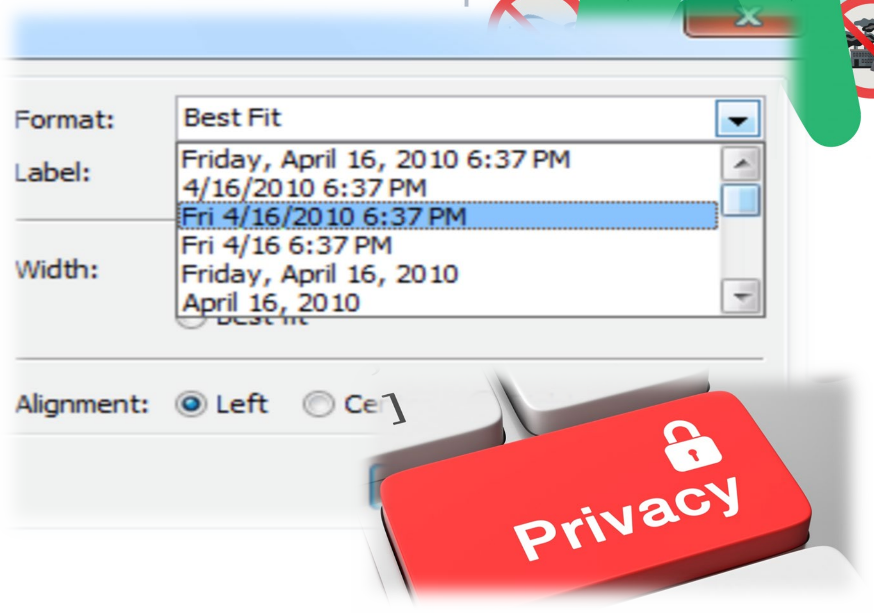
- Email: fengmei.jin@polyu.edu.hk
- Office: PQ747 (+852 3400 3327)
- Consultation Hours: 2:30-4:30pm every Thursday

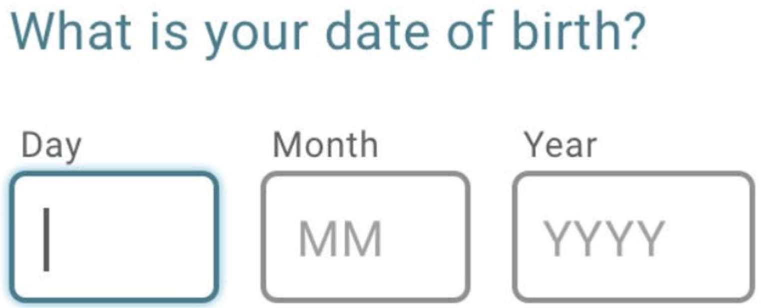
Outline

- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation
- Summary

Why Preprocess the Data? Data Quality!

- Data quality depends on the intended use of data.
- Multidimensional views of data quality:
  - **Accuracy**: data must correctly reflect real-world scenarios without errors or noise.
  - **Completeness**: all required data fields should be present and valid.
  - **Consistency**: data should follow the same rules and format across all records.
  - **Timeliness**: data should be up-to-date.
  - **Believability**: data should be credible and from trusted sources.
  - **Interpretability**: data should be clear and understandable.

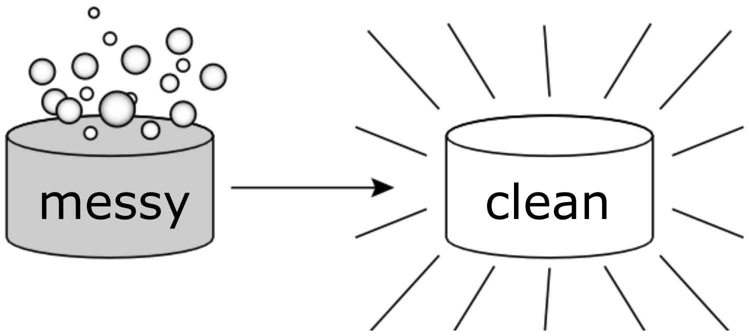




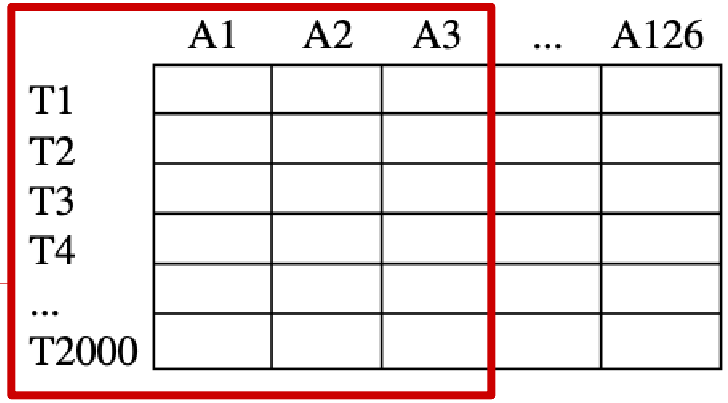
Common Sources of Low Data Quality

- Data Collection Issues
  - Human errors or misreporting during manual data entry
  - Lack of validation during input
- Data Duplications
  - Multiple entries of same information
  - Redundant record keeping
  - Merged datasets without deduplication
- Format Inconsistencies
  - Different input formats (MM/DD/YY vs. DD/MM/YY)
  - Varying units of measurement
  - Inconsistent naming conventions

Major Tasks of Data Preprocessing



- Data **Cleaning**
  - To fill in missing data, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data **Integration** (e.g., Bill Gates, William Gates, B. Gates, …)
  - To merge multiple databases into a coherent data store
- Data **Reduction** (efficiency of mining process)
  - To obtain a reduced representation of the data with similar results
- Data **Transformation**
  - To normalize data for similarity-based mining (e.g., age vs salary)



incomplete, missing, noisy, inconsistent, intentional, …

DATA CLEANING

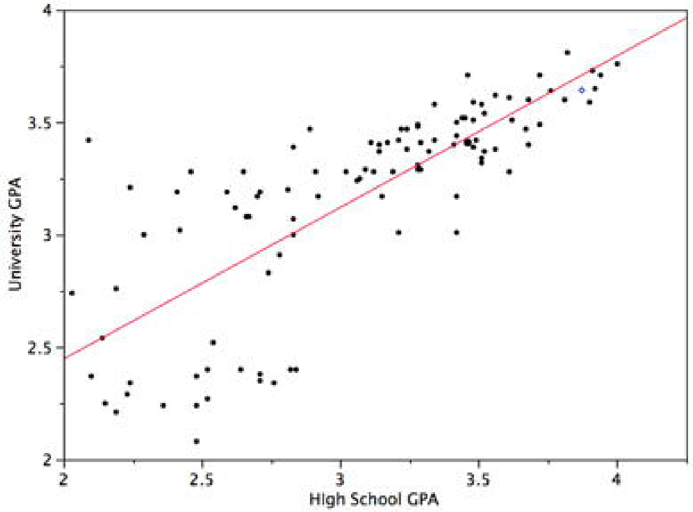
Data Cleaning

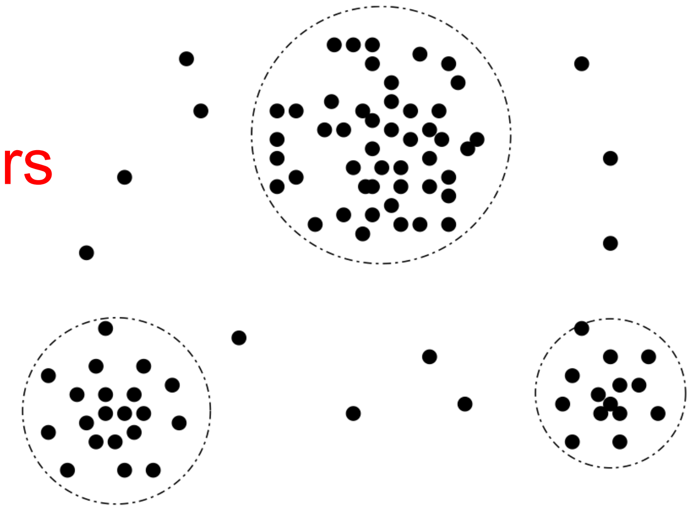
Data in the real world is Dirty!

- Lots of potentially incorrect data due to faulty collection, human or computer errors, transmission errors, etc.
  - **Inaccurate**: containing noise, errors, or outliers
    - e.g., Salary = “−10” – error
  - **Incomplete**: lacking attribute values or attributes of interest
    - e.g., Occupation = “ ” – missing values
  - **Inconsistent**: containing discrepancies in attribute values
    - Age = “20”, but Birthday = “01/01/1970”
    - Used to rate via “1, 2, 3”, now rating via “A, B, C”
    - Discrepancy between duplicate records: “Bill Gates” vs “B. GATES”
  - **Intentional**: e.g., setting 01/01/1970 as everyone’s birthday

(1) How to Handle Inaccurate (Noisy) Data?

- Noise: random error or variance in a measured variable
- **Data Smoothing** – *discretization*
  - **Binning**
    - First sort data and partition into bins
    - Then one can smooth by bin means / median / boundaries, etc.
  - **Regression**: smooth by fitting data into regression functions
  - **Clustering**: to detect and remove outliers





(2) How to Handle Incomplete (Missing) Data?

- **Ignore the tuple:** *could have been useful to the task*
- **Fill in the missing value manually**: *costly and infeasible*
- **Fill in it automatically with:**
  - Global constant: *“unknown”, infinity, or a new class label*
  - The attribute’s mean/median/mode: *suitable for symmetric data*
  - That for all samples belonging to the same class: *smarter*
    - a tuple with missing income → customers with the same credit risk
  - The most probable value through inference-based such as Bayesian formula or decision tree induction

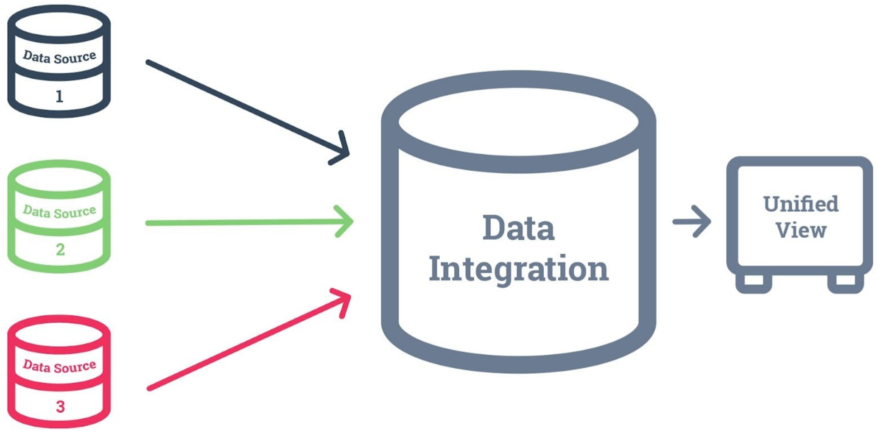
Data Cleaning as a Process

- Detection Steps:
  - **Metadata Analysis**: any knowledge you may already have regarding properties of the data – *"data about data"*
    - e.g., type, domain, range, central tendency, dependency, distribution
  - **DB Structure Validation**
    - A single field in a DB schema may store multiple pieces of information, e.g., address = "78 Staff House Rd, Brisbane, QLD 4072"
  - **Data Migration**: mixed formats like "Male", "M", "1", "MALE"
  - **Rule Checking**: unique / consecutive / null rules in DB
  - **Domain Knowledge**: e.g., postal code, spell-checking, rule/relationship discovery to detect violators

entity identification, redundancy, correlation, duplication, …

DATA INTEGRATION

Data Integration



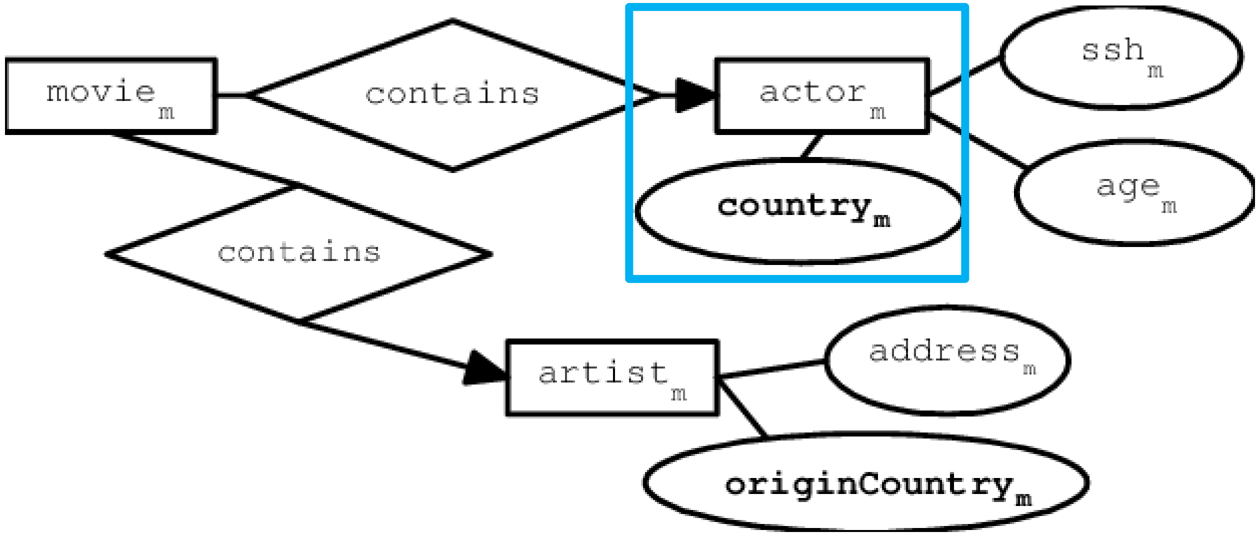
- To merge data from multiple sources into a single unified view
  - **Schema Integration** – *redundancy*
    - Integrate metadata (e.g., "user-ID" in *DB1* and "user-#" in *DB2*)
  - **Entity Matching** – *duplication*
    - Identify different representations of the same real-world entities (e.g., "Bill Gates" and "William Gates")
  - **Conflict Resolution**
    - Inconsistent values from multiple sources for the same entity, potentially caused by different representations or scales (e.g., "mile" for British units vs. "meter" for metric)

Handling Redundancy in Data Integration

- Redundancy occurs often when data integration.
  - Simple: Same information with different names across DBs
  - Complicated: One attribute may be derived from another attribute or set of attributes, e.g., "birthday" → "age"

Correlation Analysis: to measure *how strongly one attribute implies the other*, based on available data

- Nominal data: χ^2 (chi-square) test
- Numeric data: correlation coefficient and covariance → *how one attribute's values vary from those of another.*



Correlation Analysis for Nominal Data

- χ^2 (chi-square) Test for two nominal attributes A and B
  - **Input**: all data tuples about A and B, A with c distinct values $\{a_1, a_2, \dots, a_c\}$, B with r distinct values: $\{b_1, b_2, \dots, b_r\}$.
  - Let (A_i, B_j) represents a joint event: $A = a_i, B = b_j$.
    - o_{ij} is the observed frequency of (A_i, B_j); e_{ij} is the expected frequency.
    - n is # data tuples in total; $count(A = a_i)$ is # tuples with value a_i for A.
  - Range: $\chi^2 \geq 0$
    - Higher χ^2 → more likely A and B are correlated
    - Lower χ^2 → higher independence between A and B

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

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

Note: correlation does not imply causality.

- *Coffee Consumption* and Programmer Productivity in a company are correlated.
- Both are causally linked to the third variable: **work hours**

Example: Calculation of χ^2 Chi-Square

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

- The expected frequency of *play_chess* and *like_science_fiction*

$$= \frac{450 \times 300}{1500} = 90$$

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

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

→ *like_science_fiction* and *play_chess* are correlated in this group.

Correlation Analysis for Numeric Data

- Correlation Coefficient between two variables A and B based on a set of n tuples $\{(a_1, b_1), (a_2, b_2), \dots, (a_n, b_n)\}$

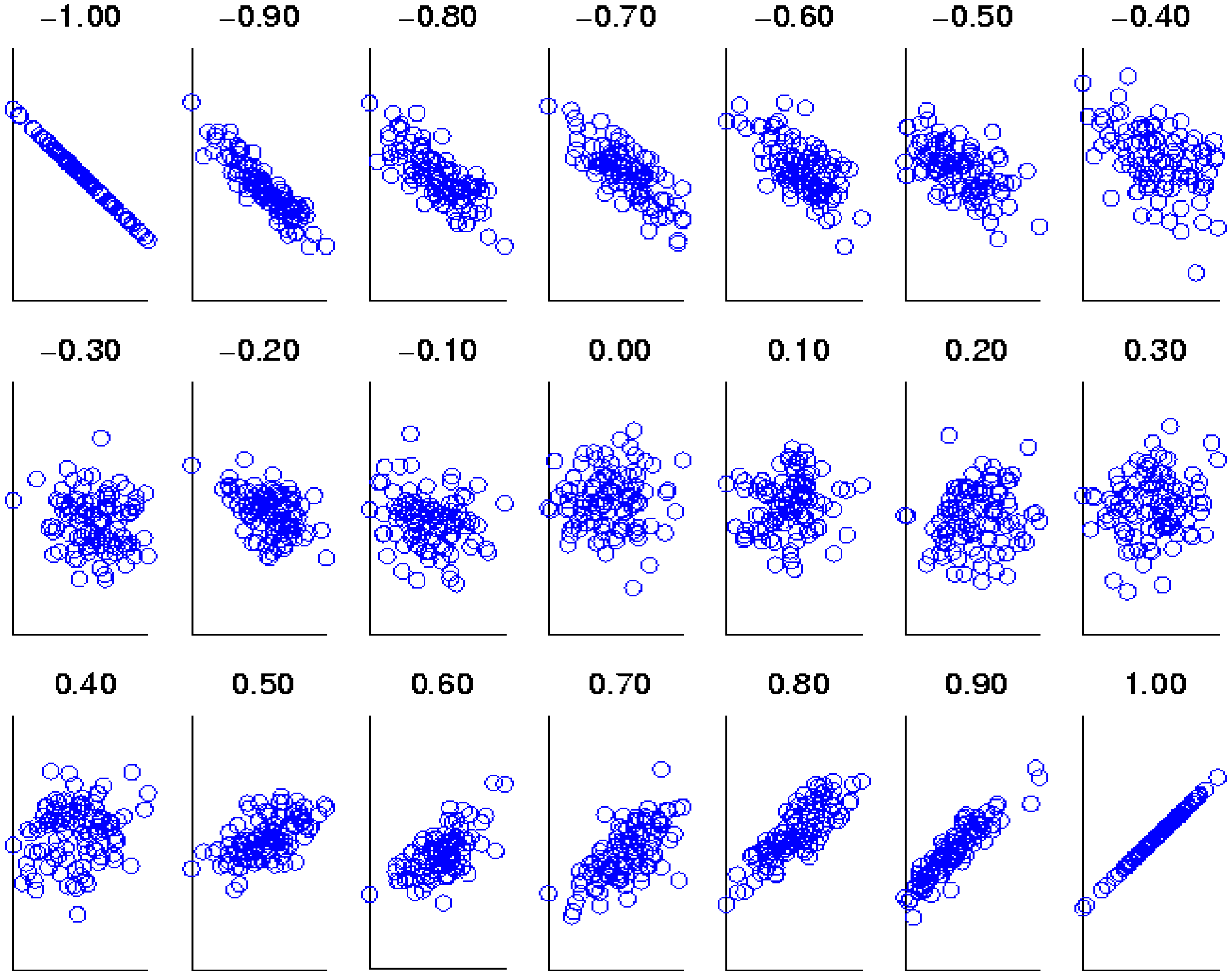
$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B}$$

- $\bar{A}$ and $\bar{B}$ are their respective mean values
- σ_A and σ_B are the respective standard deviations of A and B
- $\sum(a_i b_i)$ is the sum of AB cross-product

- Range: $-1 \le r_{A,B} \le 1$
  - If $r_{A,B} > 0$, positively correlated (i.e., A increase as B).
    - The higher $r_{A,B}$, the stronger correlation. → A or B might be removed.
  - If $r_{A,B} = 0$, no linear correlation.
  - If $r_{A,B} < 0$, negatively correlated.

Visualizing Changes of **Correlation Coefficient**

- Correlation coefficient value range: [–1, 1]



Covariance Analysis for Numeric Data

- Covariance: how much two attributes change together

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

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

$$\boxed{r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}}$$

- If they tend to change together, i.e., if A is larger than $\bar{A}$, then B is likely to be larger than $\bar{B}$ → $Cov(A,B) > 0$
- Differently, if an attribute tends to be above its mean yet the other attribute is below its mean → $Cov(A,B) < 0$
- If they are independent → $Cov(A,B) = 0$
  - $Cov(A,B) = 0$ suggests no linear relationship.

← But the converse is not true!

Example: Calculation of Covariance

- Suppose two stocks A and B have the following values in one week: $(2,5),(3,8),(5,10),(4,11),(6,16)$
- Q: If the stocks are affected by some trends, will their prices rise or fall together?
- **Covariance:** $Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$
  - $\bar{A} = (2+3+5+4+6)/5 = 20/5 = 4$
  - $\bar{B} = (5+8+10+11+16)/5 = 50/5 = 10$
  - $Cov(A,B) = (10+2+0+0+12)/5 = 4.8 > 0$
- Thus, A and B change together due to the positive covariance.

dimensionality reduction, numerosity reduction, data compression

DATA REDUCTION

Data Reduction

- To obtain a reduced representation of the data set
  - much smaller in volume, but almost the same analytical results
- **Why?**
  - Handle large-scale datasets, reduce complexity, minimize storage costs, speed up analysis, focus on most relevant info, …
- **Strategies for Data Reduction**
  - Dimensionality reduction: reduce # variables under consideration
  - Numerosity reduction: replace the original data volume by alternative, smaller forms of data representations
  - Data compression: lossless or lossy

(1) Dimensionality Reduction

- Curse of dimensionality
  - As dimensionality increases, data becomes increasingly sparse.
  - *Density* and *distance* → less meaningful
  - The possible combinations of subspaces will grow exponentially.
- To reduce # random variables under consideration by obtaining a set of principal variables
  - Avoid the curse of dimensionality
  - Help eliminate *irrelevant* features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization

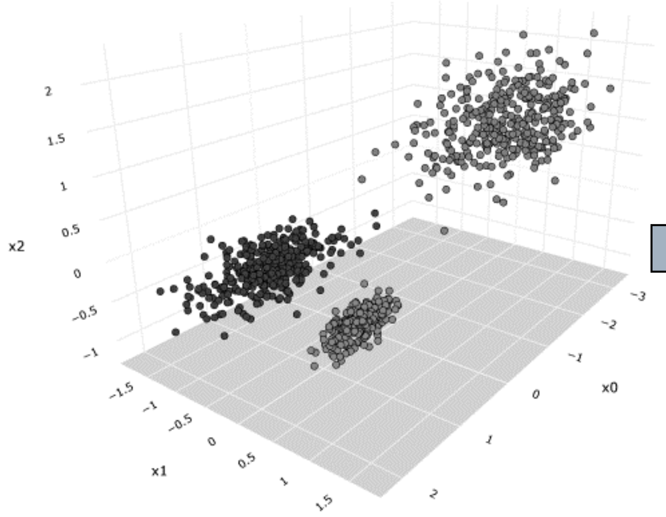
Dimensionality Reduction Techniques

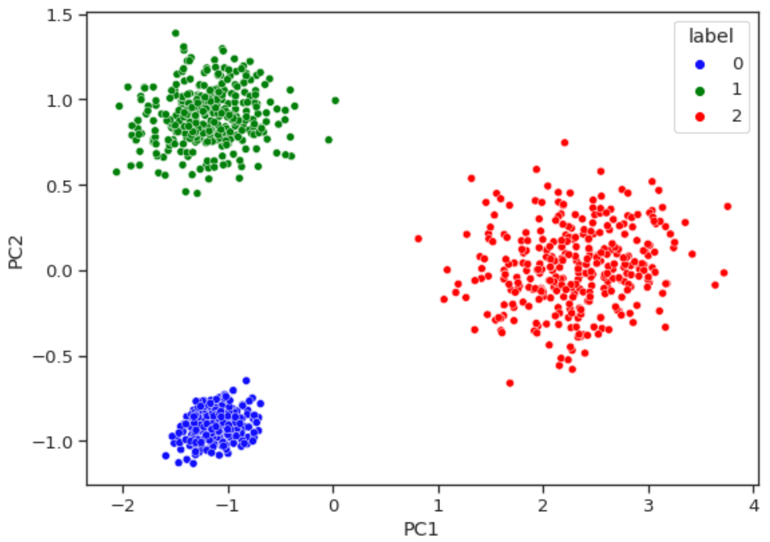
- **Feature extraction:** To transform original data to a new *lower-dimensional* space
- **Feature selection:** To find a subset of the original variables
- **Feature aggregation**: To combine related variables

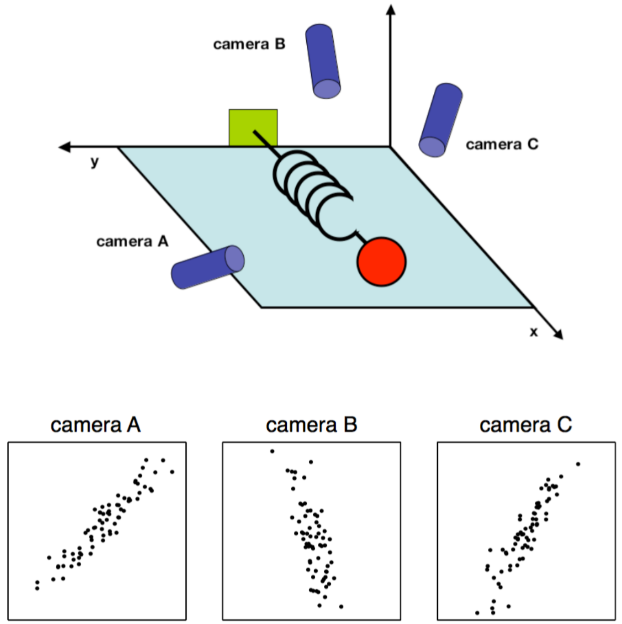
- Some typical methods
  - Principal Component Analysis: 100 stock prices → 3 market factors
  - Attribute subset selection
  - Attribute creation / construction: height and weight → area

Principal Component Analysis (PCA)

- A statistical procedure via orthogonal transformation
  - **Original**: a set of observations of possibly correlated variables
  - **Projected**: a set of values of linearly uncorrelated variables, called principal components (PC)





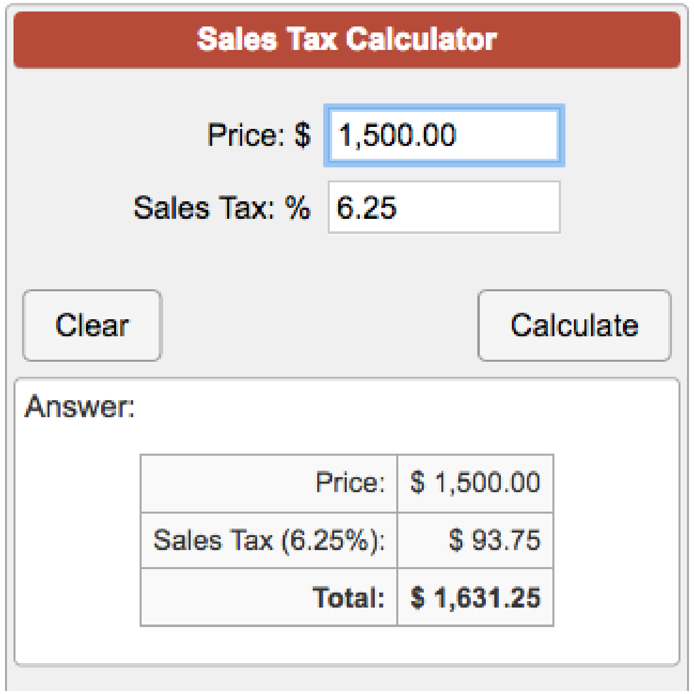


Principal Component Analysis (Method)

- Input: N data vectors from d-dimensions
- Output: Find $k \leq d$ orthogonal vectors that best represent data
  - Normalization: each attribute should fall within comparable range
  - Transformation: compute orthonormal (unit) vectors, so that each input vector is a linear combination of these vectors
  - Sorting: vectors are sorted by decreasing “importance” or strength
  - Reduction: keep top-k strongest components and discard those weak components with lower variance
    - Use the strongest PCs to reconstruct a good approximation of the original data and distinguish data points from one another
- **Works for numeric data only!**

Attribute Subset Selection

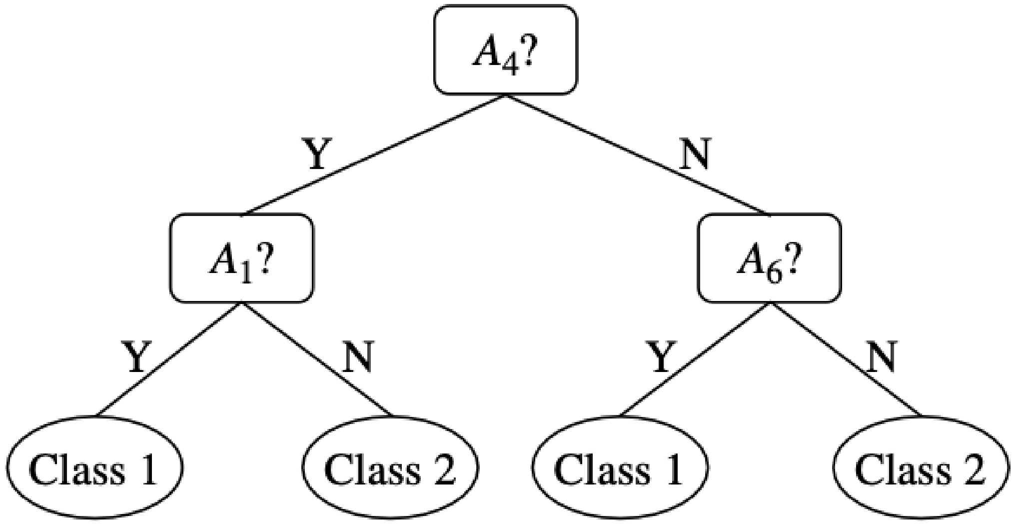
- To keep a ‘good’ subset of original features
  - Redundant attributes
    - duplicate information from other attributes
    - e.g., product price vs. the amount of tax paid
  - Irrelevant attributes
    - no impact on the target task
    - e.g., student ID vs. predicting GPA, telephone number vs. credit risk



Attribute Subset Selection by Heuristic Search

- An exhaustive search is expensive and impossible.
  - There are 2^d possible attribute combinations of d attributes.
- Typical heuristic (greedy) attribute selection methods:
  - Forward selection: 1) an empty set of attribute initially, 2) select the best of the remaining attributes at each iteration
  - Backward elimination: 2) full set of attributes; 2) at each step, remove the worst one remaining in the set
  - Decision tree induction
    - each non-leaf node → a test on an attribute
    - each branch → an outcome of the test
    - each leaf node → a class prediction

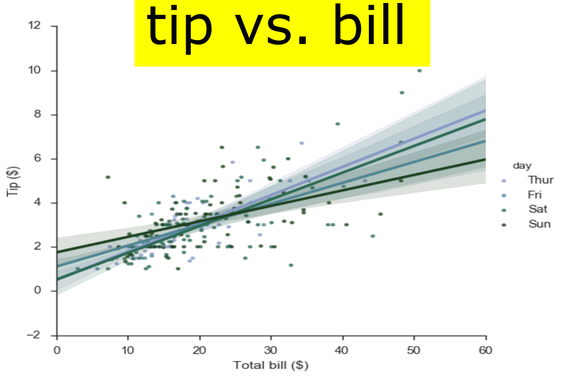
Attribute Subset Selection by Heuristic Search

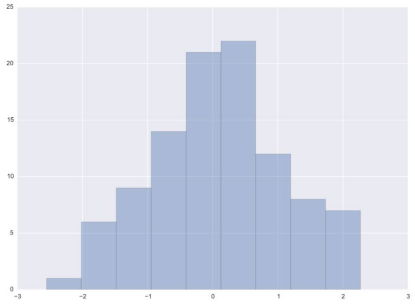
Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ => $\{A_1\}$ => $\{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  => Reduced attribute set: $\{A_1, A_4, A_6\}$

(2) Numerosity Reduction

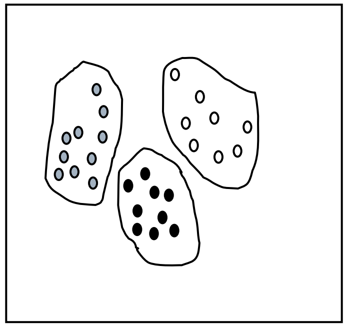
- Reduce data volume by choosing alternative, smaller forms of data representation
- **Parametric** methods
  - Assume the data fits mathematical model, estimate model parameters, store only the parameters, and discard the data (except possible *outliers*)
- **Non-parametric** methods
  - Do not assume data shape or models
  - e.g., histograms, clustering, sampling, …

tip vs. bill





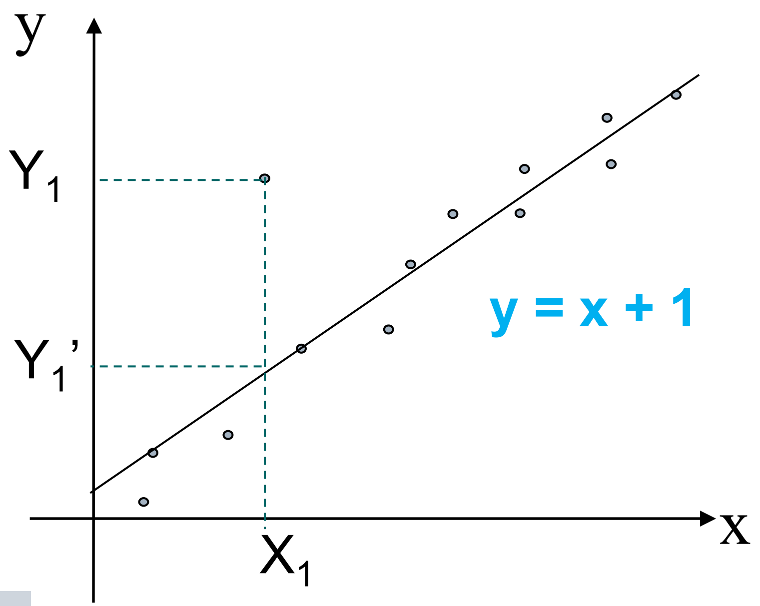
Histogram



Clustering

Numerosity Reduction – Parametric

- Regression analysis
  - A collective name for techniques for the modeling and analysis of numeric data
  - Parameters are estimated by minimizing differences between prediction and actual values so as to give a 'best fit' of the data.

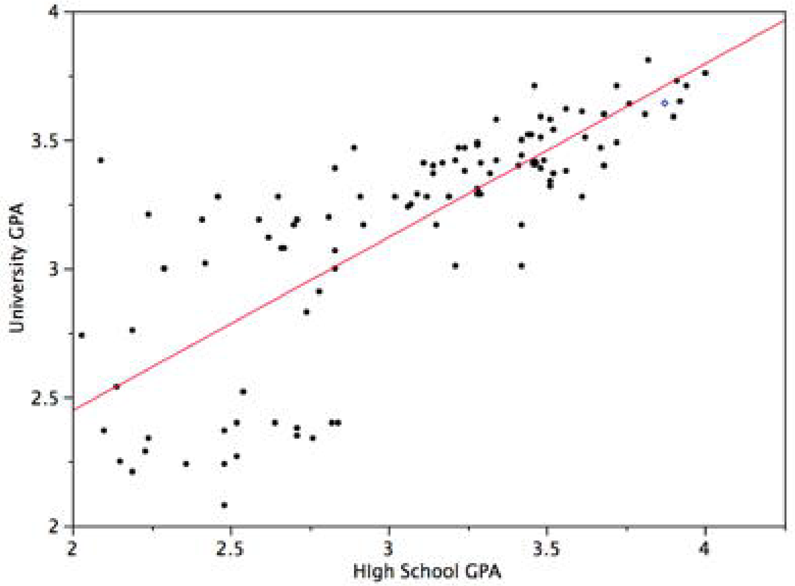


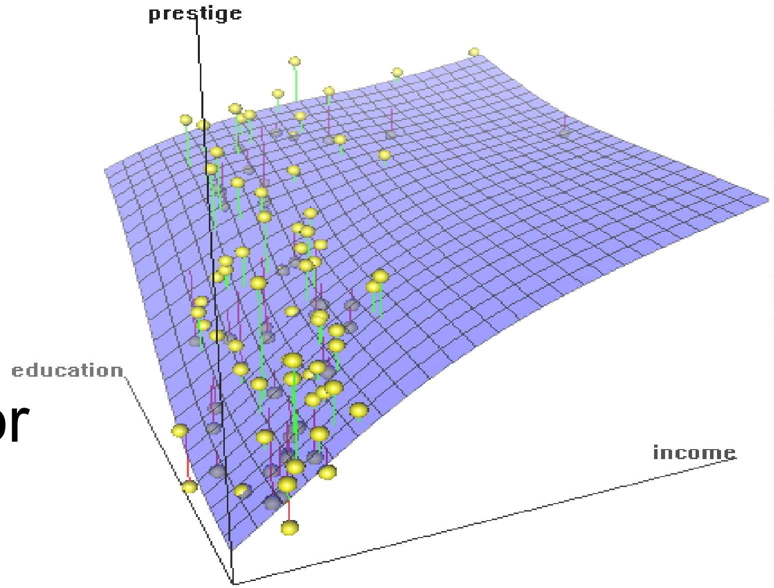
Other practical applications:

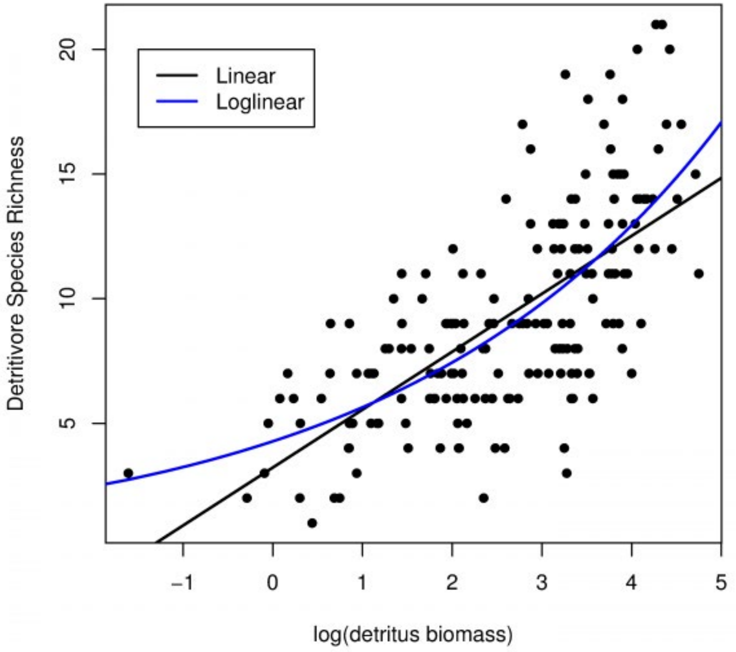
- Prediction: What will happen next?
- Inference: Understanding relationships
- Hypothesis testing: Testing assumptions
- Causal modeling: Understanding cause/effect

Linear and Multiple Regression

- **Linear regression**: $Y = wX + b$
  - Model the data to best fit a straight line
  - Often uses the least-square method to fit it
  - The regression coefficients, w and b, specify the line – estimated by using the data at hand
  - Using the least squares criterion to the data
- **Multiple regression**: $Y = b_0 + b_1X_1 + b_2X_2$
  - Allow Y to be modeled as a linear function of multi-dim feature vector





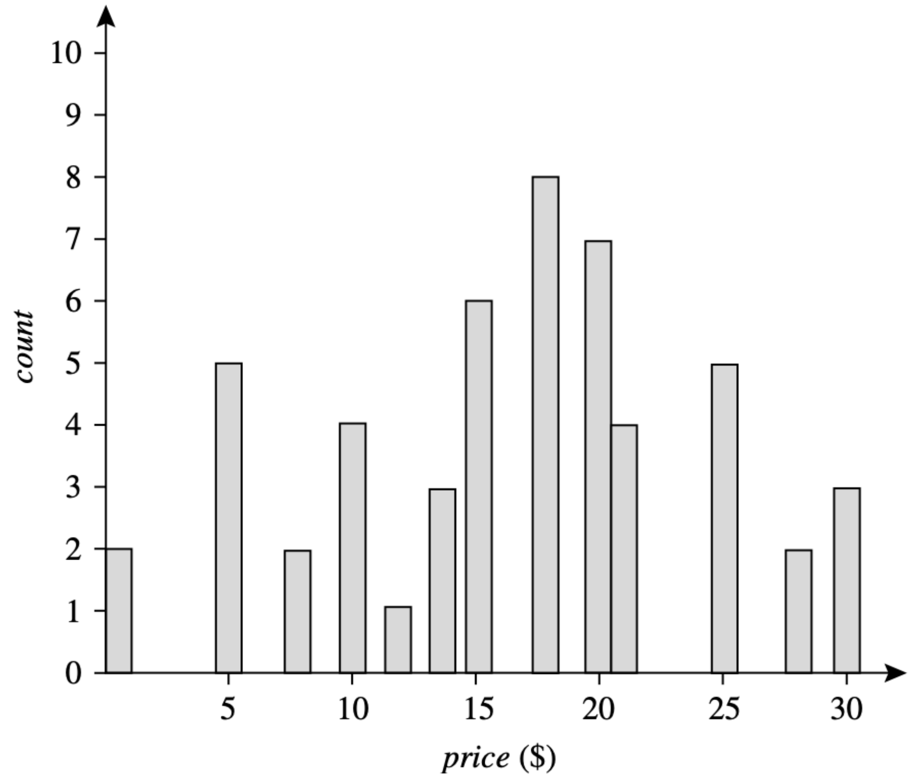


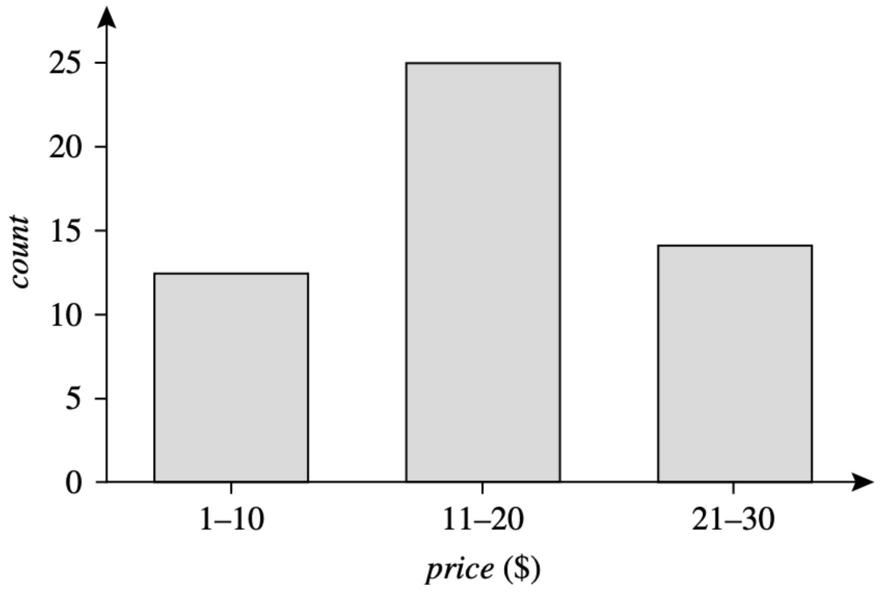
Histogram Analysis

- To approximate data distributions
  - divide data into disjoint buckets (or bins) and store the average (or sum) for each

Sales data:
1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

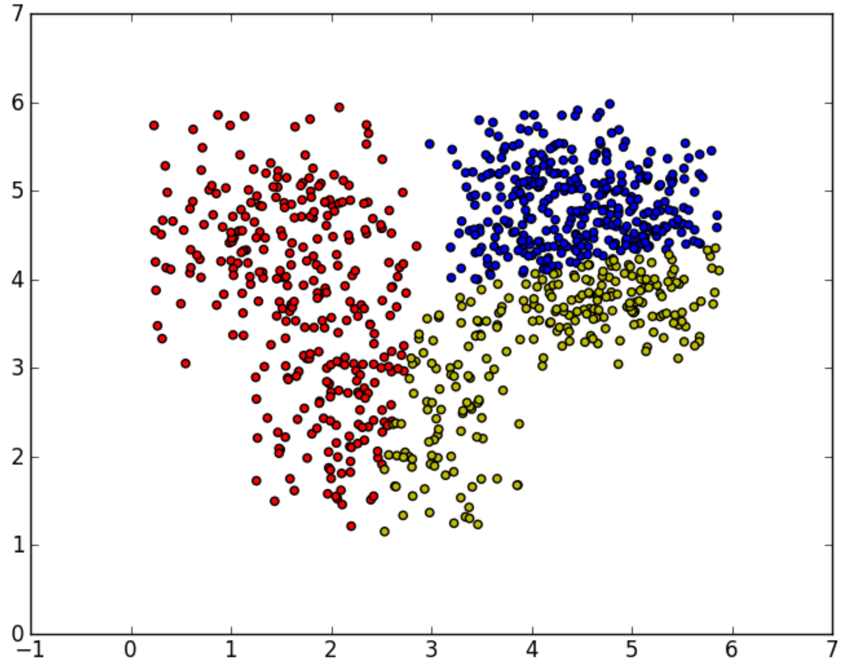
- **Partitioning rules**
  - Equal-width: equal bucket range
  - Equal-frequency: equal # items per bin





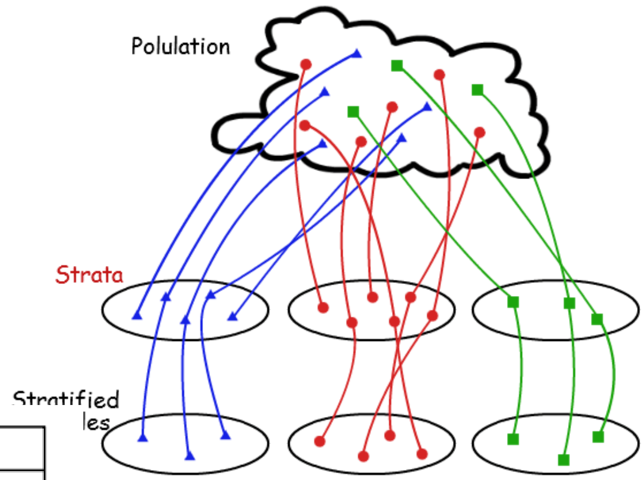
Clustering

- Partition data set into clusters based on similarity
- Store cluster representations (e.g., centroid) only
  - Objects within a cluster are *similar* and *dissimilar* to objects in other clusters.



Numerosity Reduction by Sampling

- To obtain a small sample s to represent the whole data set N
- **Key**: sample a **representative** subset of the data



Stratified sampling

Simple random sampling:
- Equal probability of selecting each object
- Sampling with / without replacement: a selected object is / is not removed
- Poor performance in *skewed* data

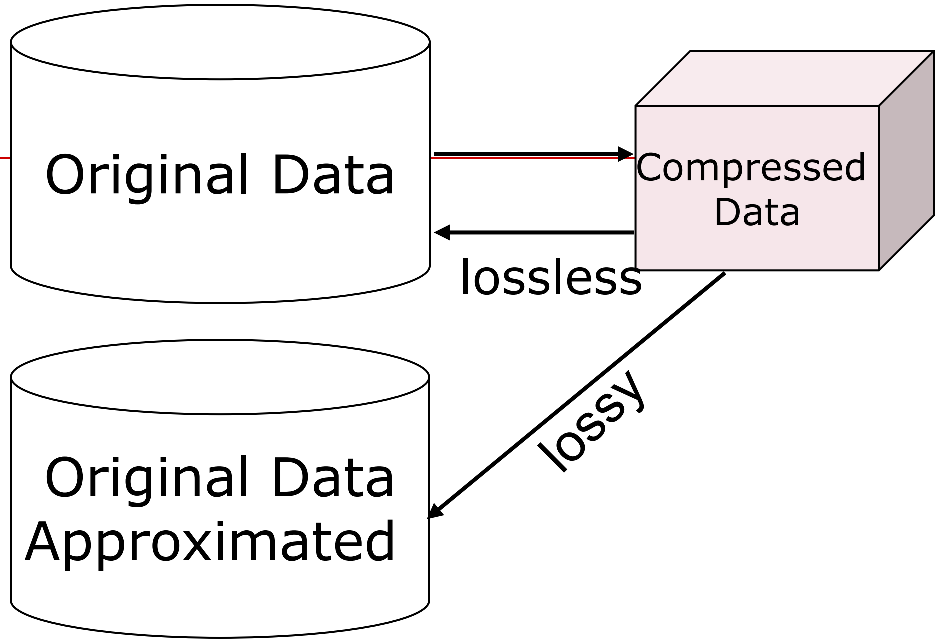
Adaptive sampling (stratified sampling)
- Partition the data and draw samples from each cluster
- Sampling probability is proportional to each strata size

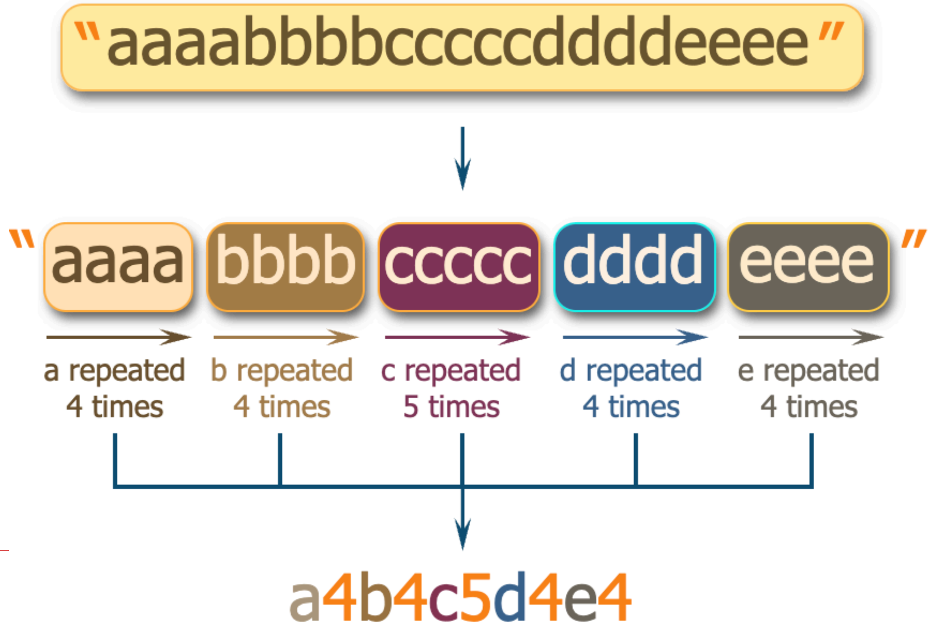
T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

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

(3) Data Compression

- To obtain a reduced or "compressed" representation of the original data.
  - **Lossless**: if the original data can be reconstructed from the compressed data without any information loss
    - e.g., string compression
  - **Lossy**: only an approximation of the original data can be reconstructed
    - e.g., audio/video compression





normalization, discretization, …

DATA TRANSFORMATION

Data Transformation

- To map the entire set of data → a new set of replacement values *s.t.* each old value can be identified with new values

- Strategies:
  - **Normalization**: scaled to fall within a smaller range, e.g., [0,1]
  - **Discretization**: concept hierarchy climbing (also for *reduction*)
  - **Smoothing**: remove noise from data (also for *cleaning*)
  - **Aggregation**: summarization in data cube (also for reduction)
  - **Attribute construction**: existing attributes → new attributes, e.g., amount and unit price → total cost (also for *reduction*)

Min-Max Normalization

- Range: $[\mathrm{minA}, \mathrm{maxA}] \rightarrow [\mathrm{new\_minA}, \mathrm{new\_maxA}]$

$$v' = \frac{\mathrm{v} - \mathrm{min}A}{\mathrm{max}A - \mathrm{min}A} \times (\mathrm{new\_max}A - \mathrm{new\_min}A) + \mathrm{new\_min}A$$

- For example, normalize range: [$12,000, $98,000] → [0.0, 1.0]
  - Then, $73,600 is mapped to: $\frac{73{,}600-12{,}000}{98{,}000-12{,}000} \times (1.0 - 0) + 0 = 0.716$

Z-score Normalization

- Rely on μ (mean) and σ (standard deviation)

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

Z-score: The distance between the raw score and the population mean in the unit of the standard deviation.

- For example, let $\mu = 54{,}000$, $\sigma = 16{,}000$.
  - Then, \$73,600 is mapped to: $\frac{73{,}600-54{,}000}{16{,}000} = 1.225$.

Normalization by Decimal Scaling

- Find the scaling factor j as the smallest integer *s.t.* $\max(|v'|) < 1$ for all normalized v':

$$v' = \frac{v}{10^j}$$

- Bounded range within $[-1, 1]$
- For example, given a data set with the range of $[-986, 917]$:
  - $j = 3$ because the max absolute value is 986
  - New Range: $[-0.986, 0.917]$

Discretization

- Three types of attributes
  - **Nominal**: values from an unordered set, e.g., color, marital status
  - **Ordinal**: values from an ordered set, e.g., drink size, profession
  - **Numeric**: real numbers, e.g., age, height, weight

- To divide the range of a continuous attribute into distinct intervals
  - → Interval labels can then be used to replace actual data values
  - → Reduce data size

Simple Discretization: Binning

- Equal-width (distance) partitioning
  - Divides the range into N intervals of equal size: **uniform partition**
  - The width of intervals: $W = (B - A)/N$, where A and B are the lowest and highest values of the attribute.
    - 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

Example: Binning Methods

- Sorted data for price (in dollars)
  - {4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34}

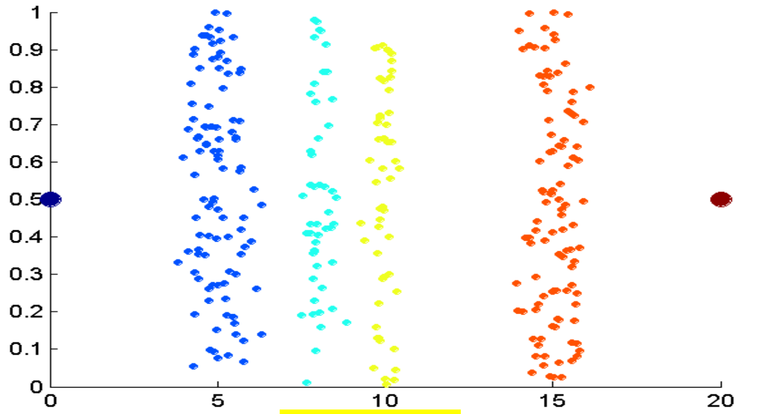
Equal-width partition (width=10):
- Bin 1: 4, 8, 9
- Bin 2: 15, 21, 21
- Bin 3: 24, 25, 26, 28, 29, 34

Smoothing by *bin boundaries*:
- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

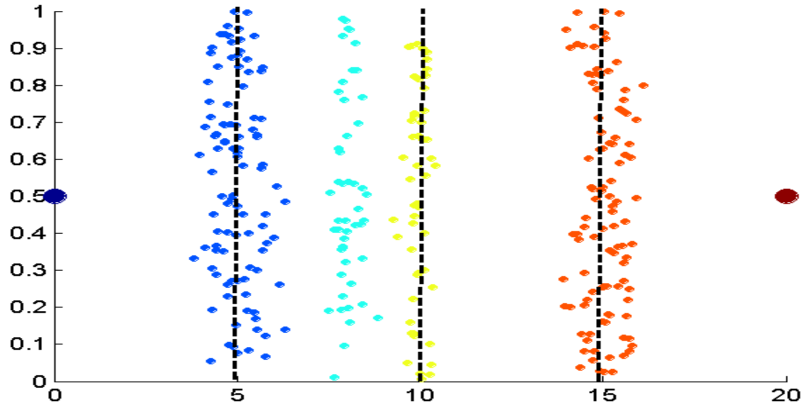
Equal-frequency partition:
- Bin 1: 4, 8, 9, 15
- Bin 2: 21, 21, 24, 25
- Bin 3: 26, 28, 29, 34

Smoothing by *bin means*:
- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

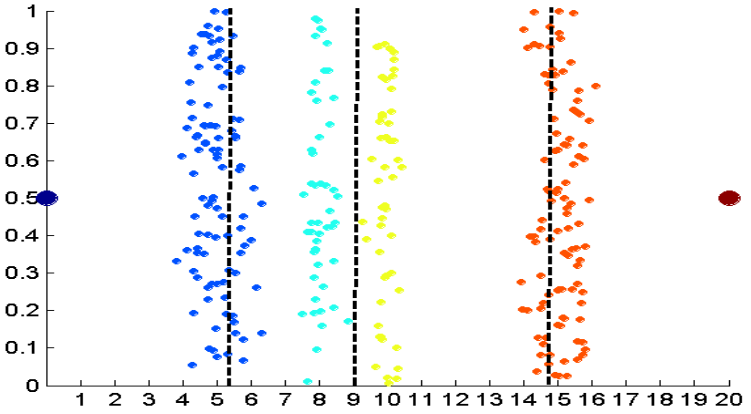
Discretization Without Supervision: Binning vs. Clustering



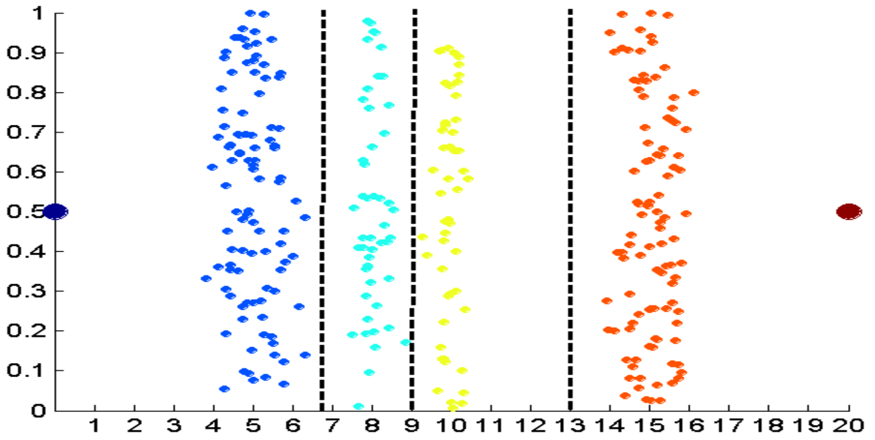
Data



Equal width (distance) binning



Equal depth (frequency) (binning)



K-means clustering leads to better results

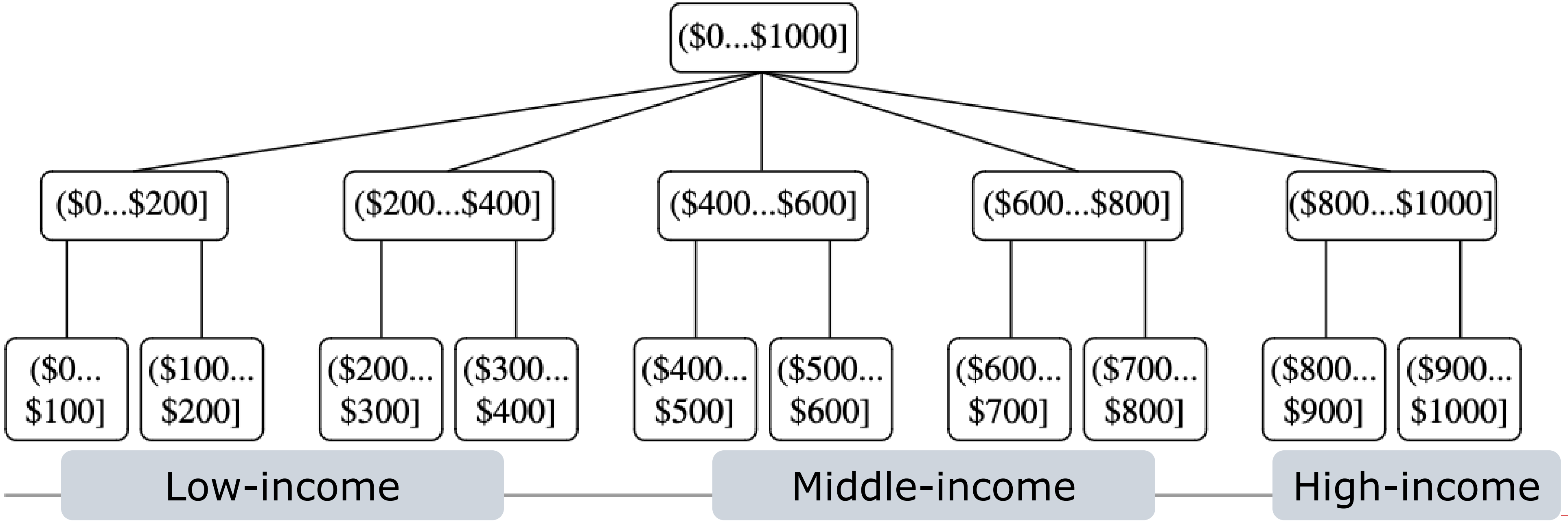
Discretization with Supervision

- **Classification-based** (e.g., decision tree)
  - Key: use class labels to guide boundary selection (split points) and create intervals that maximize class discrimination
  - Common metrics: information gain, entropy, …

- **Correlation-based**
  - Key: use statistical dependencies to determine intervals while preserving relationships between variables
  - Classic algorithm: *ChiMerge*, a χ^2-based method

Concept Hierarchy

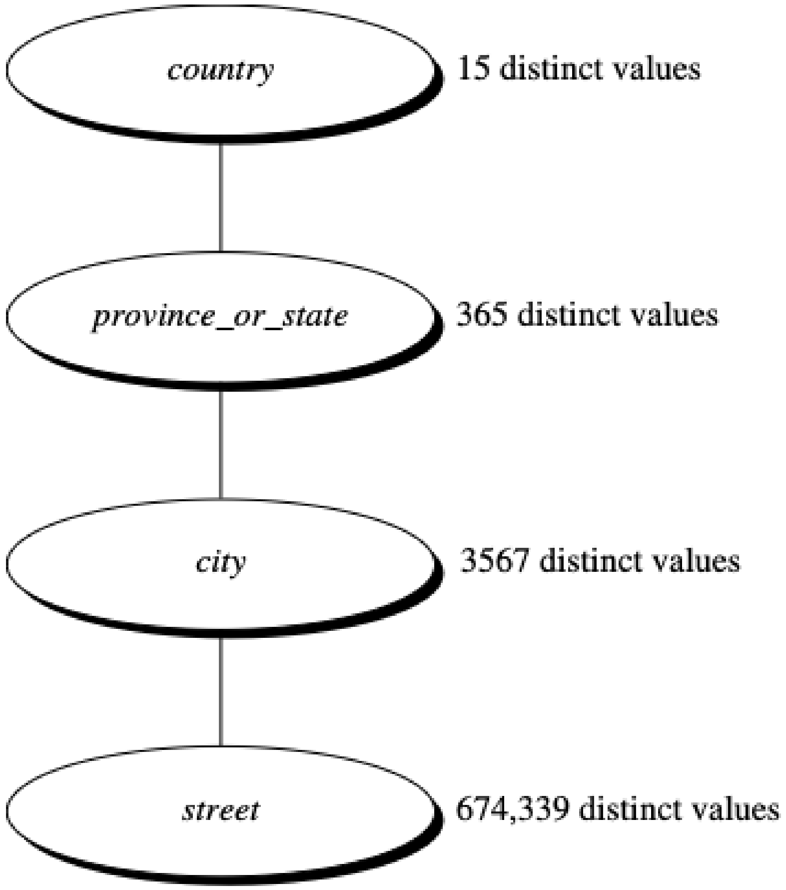
- To organizes concepts (i.e., attribute values) hierarchically
  - **Formation**: Recursively reduce the data by collecting and replacing low level concepts (e.g., numeric values for age) by higher level concepts (e.g., youth, adult, or senior)



A concept hierarchy for the attribute *price*, where an interval ($$X \ldots \$Y$] denotes the range from $$X$ (exclusive) to $$Y$ (inclusive).

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on # 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



Summary

- Data quality based on the intended use of the data: accuracy, completeness, consistency, timeliness, believability, interpretability
- Data cleaning: to fill in missing values, smooth out noise, identify outliers, and correct inconsistencies
- Data integration: to combine multi-source data as a coherent data store (duplication, redundancy, conflicts)
- Data reduction: to obtain a reduced representation of the data while minimizing the loss of information content
- Data transformation: to convert the data into appropriate forms
- Data discretization: to transform continuous data to interval or labels

Email: fengmei.jin@polyu.edu.hk

Office: PQ747

THANK YOU!