Data Mining

Unit II

- **Data Preprocessing :**
- Data Preprocessing: An Overview
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization

Data Preprocessing: An Overview

- Data Quality: Why Preprocess the Data?

Data quality measures:

Accuracy

Completeness

Consistency

Timeliness

Believability

Interpretability.

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?

Data Preprocessing: An Overview

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
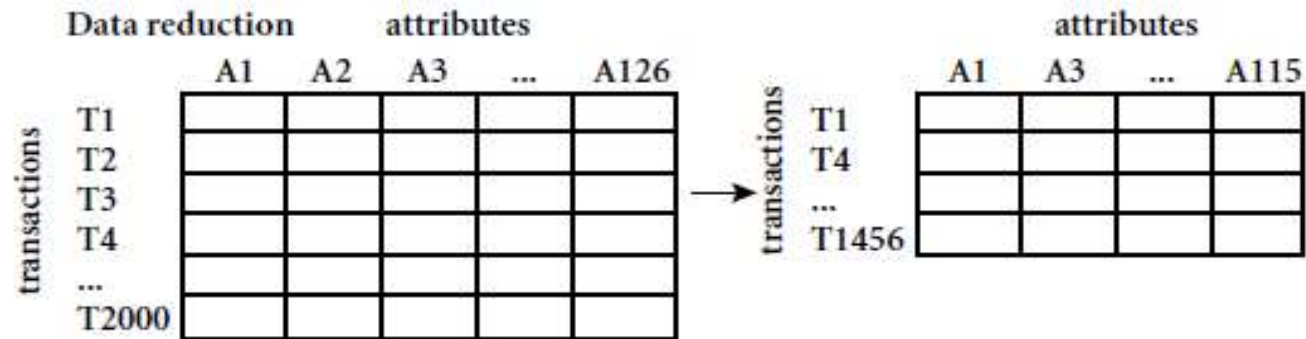 - Concept hierarchy generation

Data cleaning
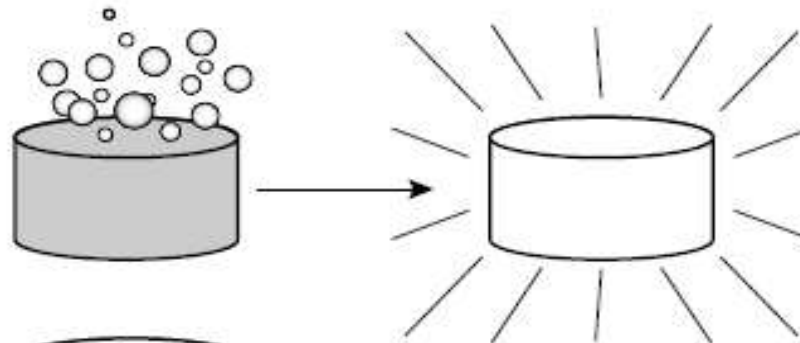
Data integration
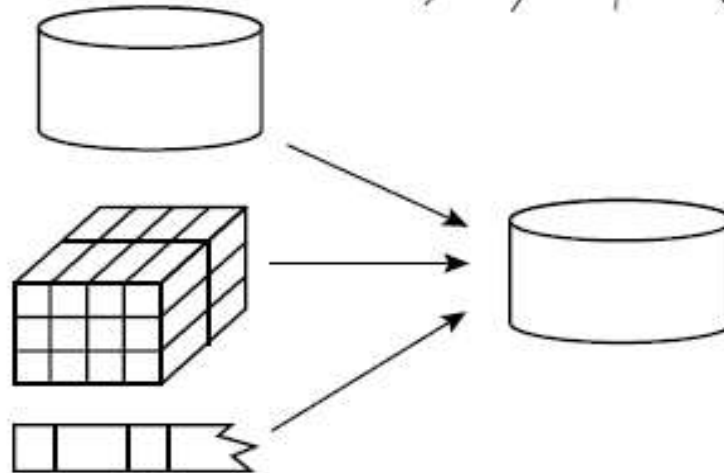
Data transformation $-2, 32, 100, 59, 48 \longrightarrow -0.02, 0.32, 1.00, 0.59, 0.48$

Data reduction

	attributes				
transactions	A1	A2	A3	...	A126
T1					
T2					
T3					
T4					
...					
T2000					

	attributes			
transactions	A1	A3	...	A115
T1				
T4				
...				
T1456				

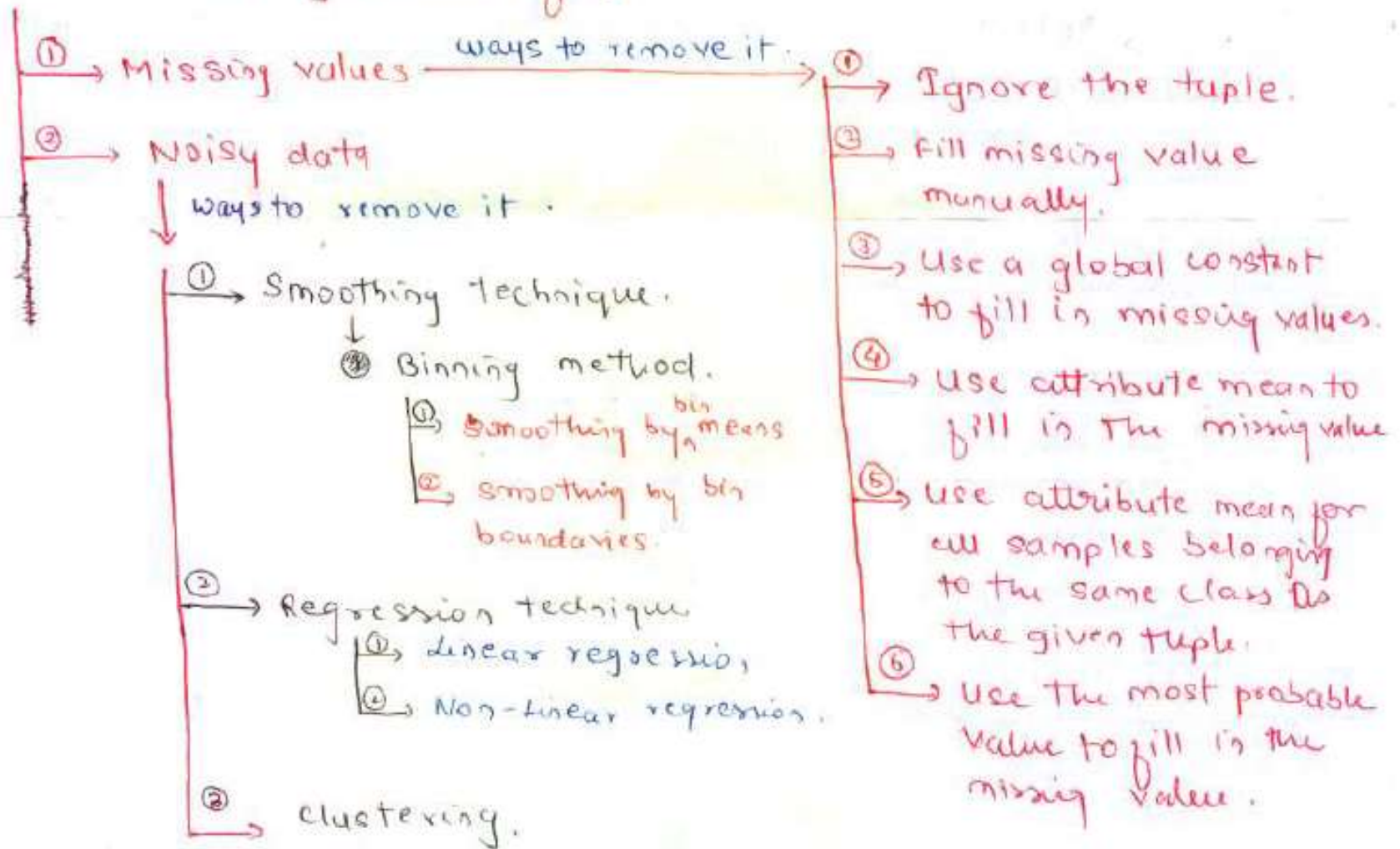**re 2.1** Forms of data preprocessing.

Why Is Data Preprocessing Important?

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
  - Data warehouse needs consistent integration of quality data
- Data extraction, cleaning, and transformation comprises the majority of the work of building a data warehouse

Data Cleaning

(A) Data cleaning (Need of it).

(1) → Missing values —— ways to remove it. → (1) → Ignore the tuple.

(2) → Noisy data
  ↓ ways to remove it.

  (1) → Smoothing technique.
    ↓
    (B) Binning method.
      (a) Smoothing by bin means
      (b) smoothing by bin boundaries.

  (2) → Regression technique
    (1) → Linear regression,
    (2) → Non-linear regression.

  (3) → clustering.

(1) → Ignore the tuple.

(2) → Fill missing value manually.

(3) → Use a global constant to fill in missing values.

(4) → Use attribute mean to fill in the missing value

(5) → Use attribute mean for all samples belonging to the same class As the given tuple.

(6) → Use the most probable value to fill in the missing value.

Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - <u>incomplete</u>: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*=" " (missing data)
  - <u>noisy</u>: containing noise, errors, or outliers
    - e.g., *Salary*="−10" (an error)
  - <u>inconsistent</u>: 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
  - <u>Intentional</u> (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: smarter
  - 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
    - 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 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
  - Iterative and interactive (e.g., Potter's Wheels)

Binning

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

* Partition into (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
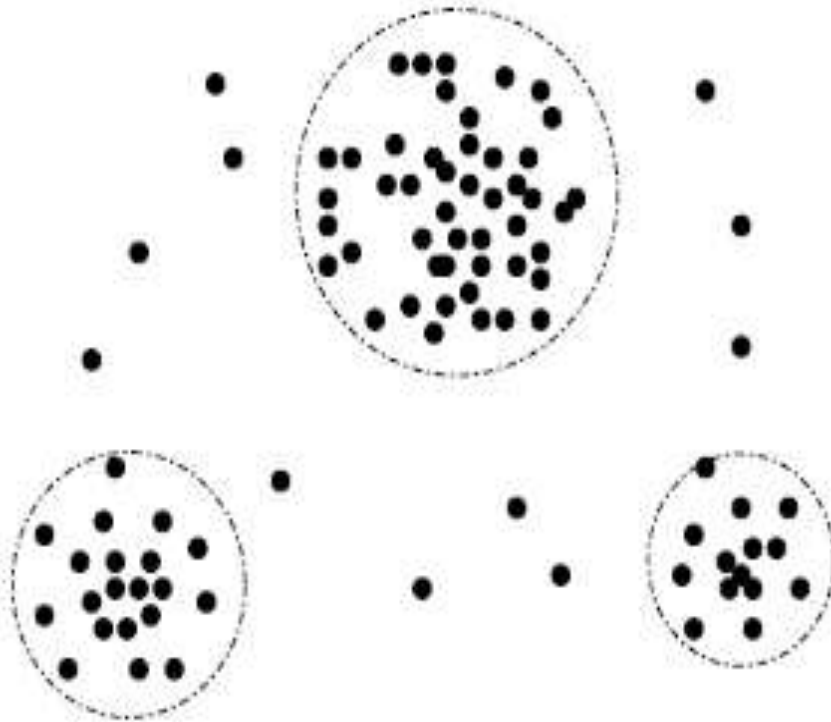- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

Clustering

gure 3.3 A 2-D customer data plot with respect to customer locations in a city, showing three data clusters. Outliers may be detected as values that fall outside of the cluster sets.

Data Integration

1 → Schema Integration and ———→ Use of Metadata.
   object matching

2 → Redundancy ————————→ Correlation Analysis.
                                        ↓
                                   Two Types
                                   ① For numerical attribute
                                        → correlation coefficient.
                                   ② For categorical attribute.
                                        → correlation using x^2

                                   (x^2 called chi square)

③ → Detection and ———→ Special attention to struc
   resolution of   of data.
   data value conflicts

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

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

Correlation Analysis (Nominal Data)

- **X² (chi-square) test**

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

- 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

- X^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)

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

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

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

Mapping Data to a New Space

- **Fourier transform**
- **Wavelet transform**
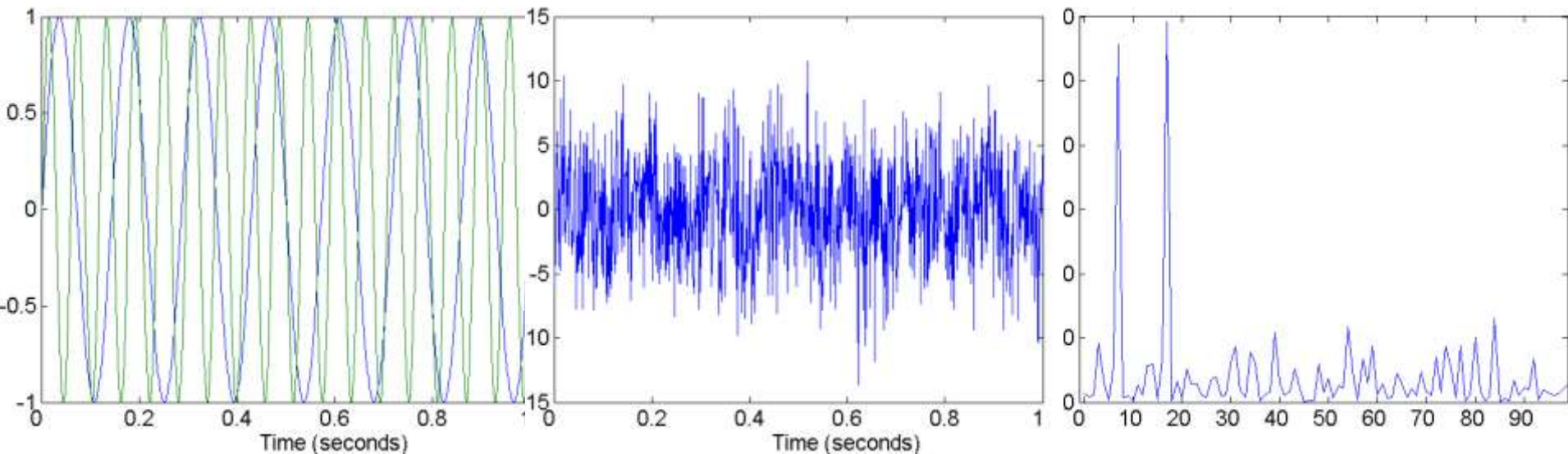
Two Sine Waves   **Two Sine Waves + Noise**   **Frequency**

What Is Wavelet Transform?

- Decomposes a signal into different frequency subbands
  - Applicable to n-dimensional signals
- Data are transformed to preserve relative distance between objects at different levels of resolution
- Allow natural clusters to become more distinguishable
- Used for image compression

Wavelet Transformation

Haar2   Daubechie4
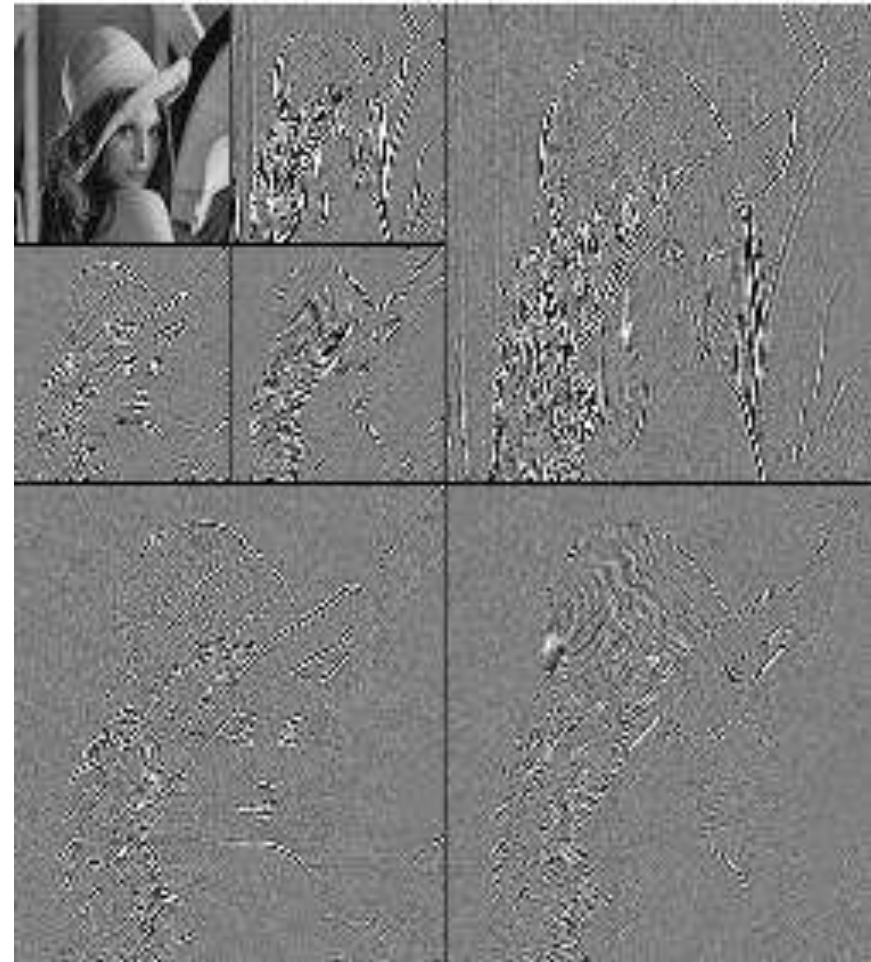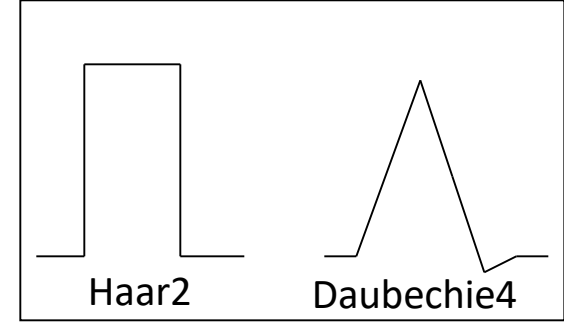
- Discrete wavelet transform (DWT) for linear signal processing, multi-resolution analysis

- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients

- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space

- 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?

- Use hat-shape filters
  - Emphasize region where points cluster
  - Suppress weaker information in their boundaries
- Effective removal of outliers
  - Insensitive to noise, insensitive to input order
- Multi-resolution
  - Detect arbitrary shaped clusters at different scales
- Efficient
  - Complexity O(N)
- Only applicable to low dimensional data

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
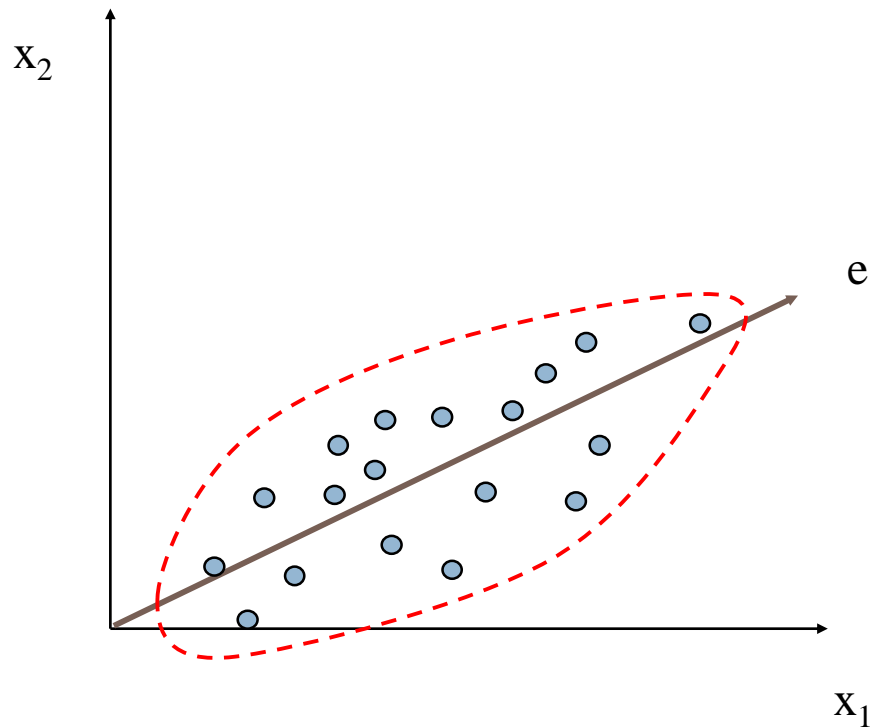
Principal Component Analysis (Steps)

- 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

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:
    - 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
- 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 7)
    - 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 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

$y = x + 1$

- 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
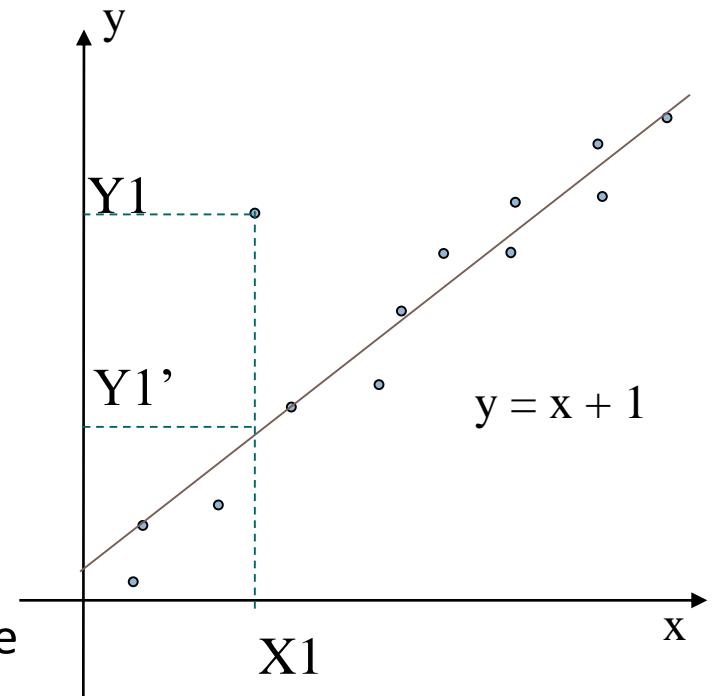
- Used for prediction (including forecasting of time-series data), inference, hypothesis testing, and modeling of causal relationships

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 $Y_1, Y_2, ..., X_1, X_2, ...$
- Multiple regression: $Y = b_0 + b_1 X_1 + b_2 X_2$
  - Many nonlinear functions can be transformed into the above
- 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

Histogram Analysis

- Divide data into buckets and store average (sum) for each bucket

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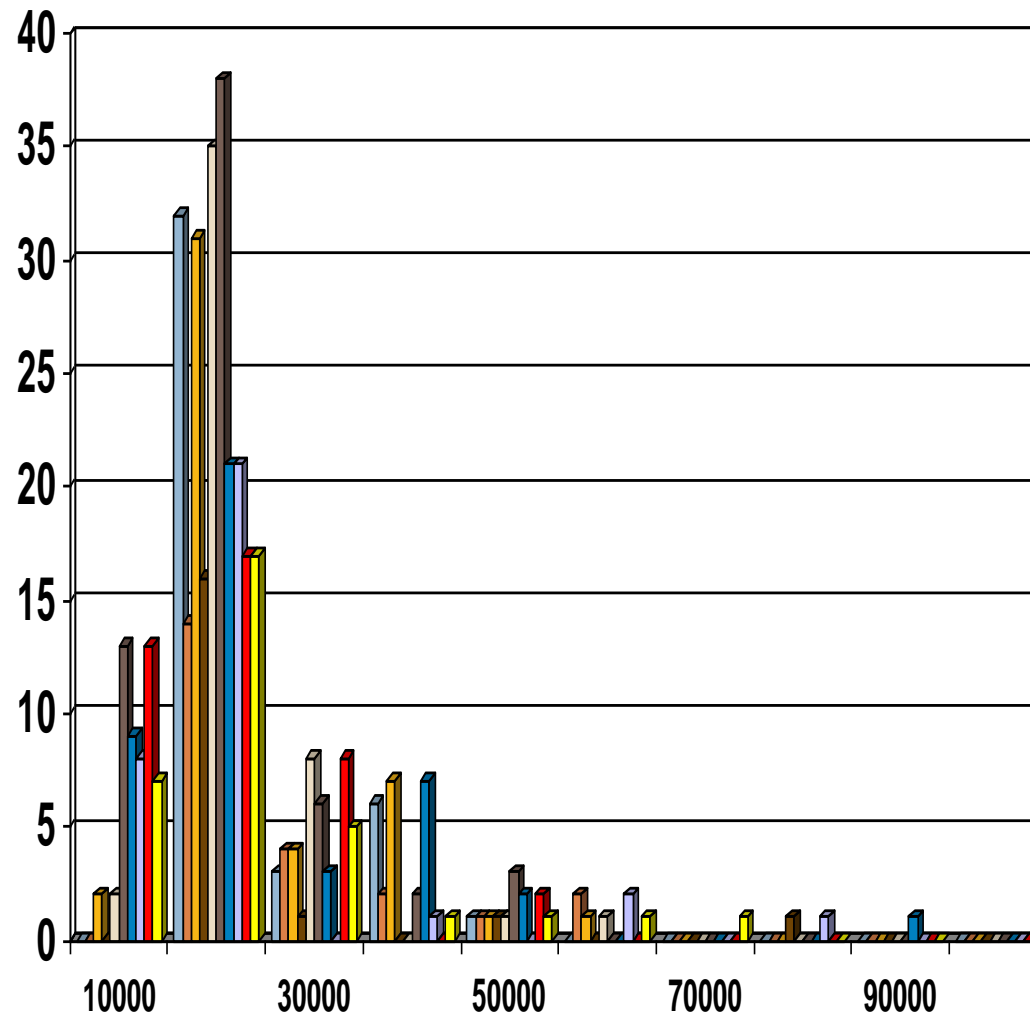
Clustering

- 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

- Cluster analysis will be studied in depth in Chapter 10

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

Sampling: With or without Replacement

SRSWOR (simple random sample without replacement)
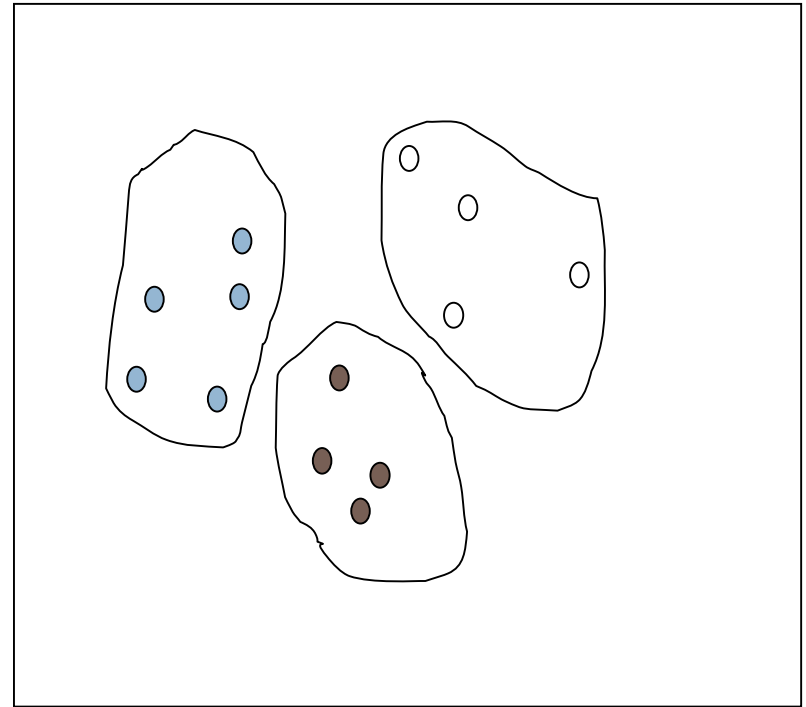
SRSWR
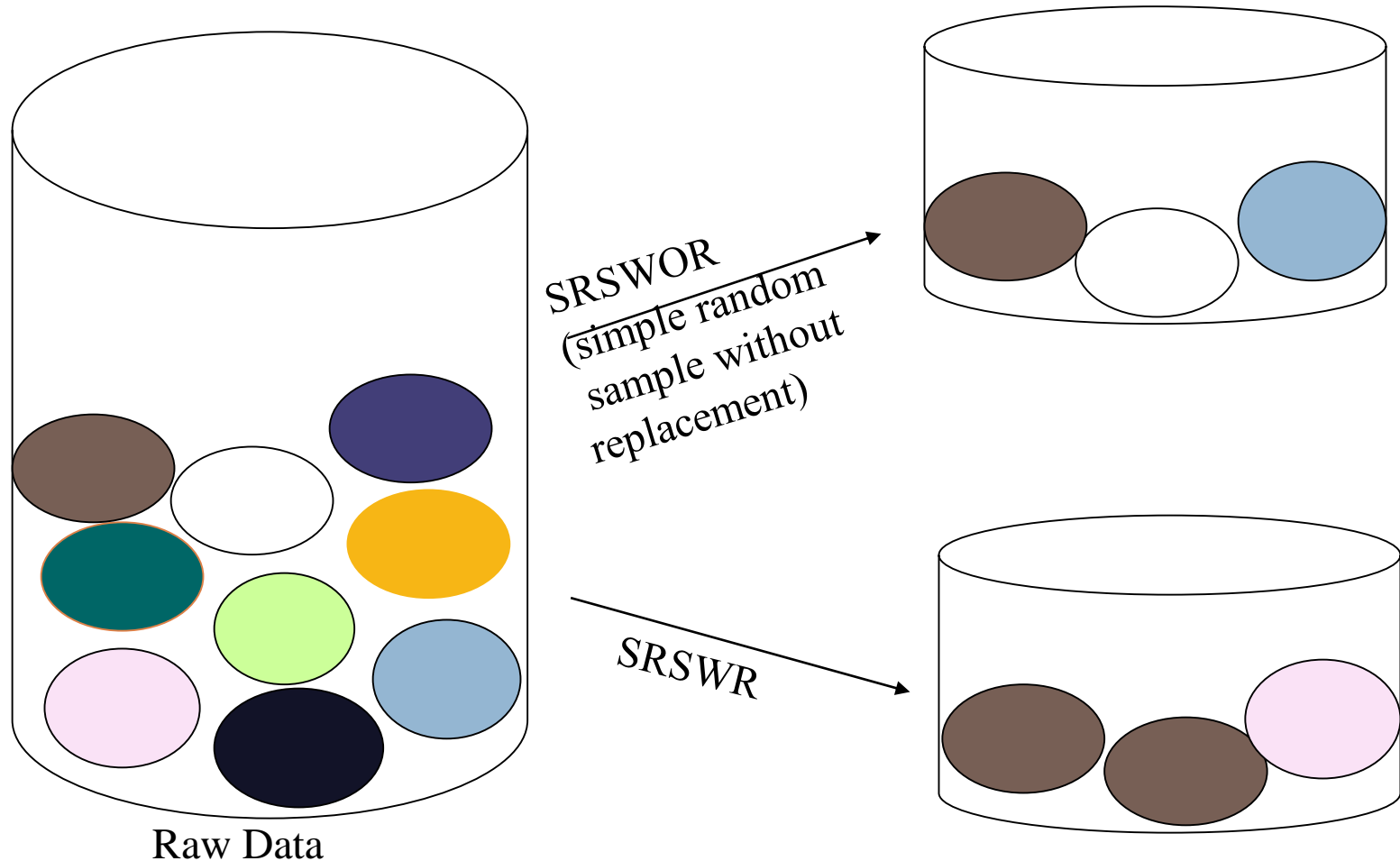
Raw Data

Sampling: Cluster or Stratified Sampling

Raw Data
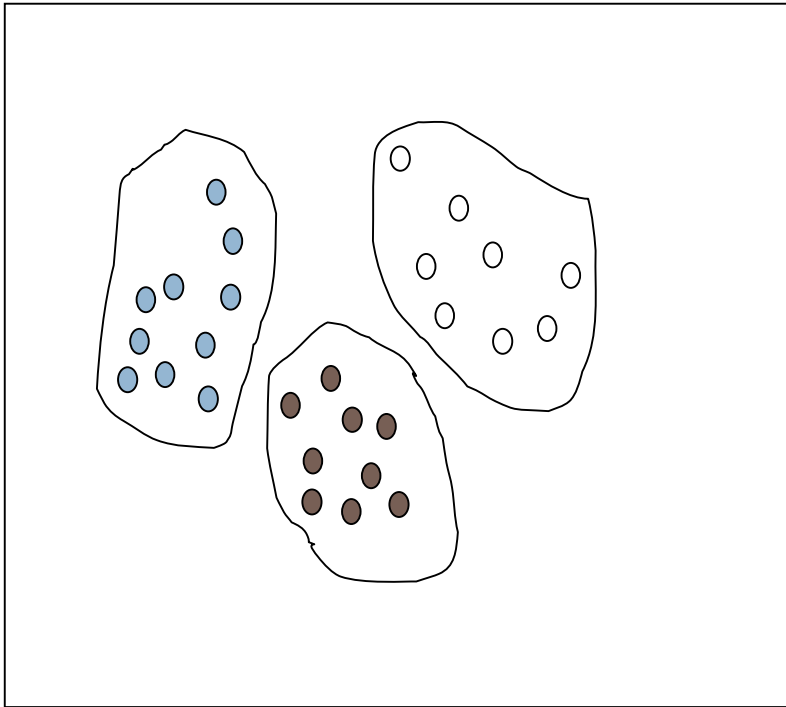
Cluster/Stratified Sample

Data Cube Aggregation

- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- 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 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
- 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
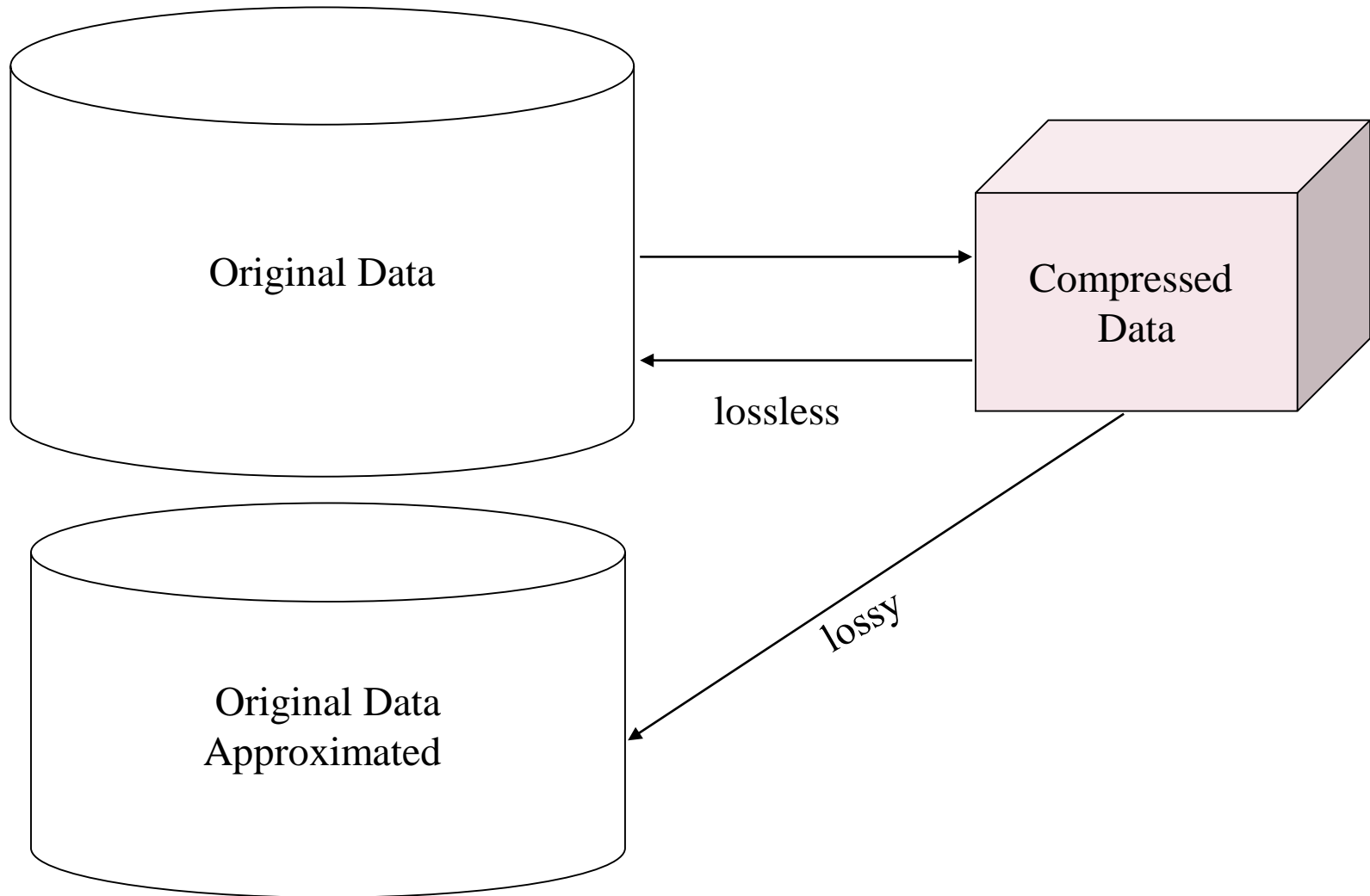
Data Transformation

- 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 \$73,000 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 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

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
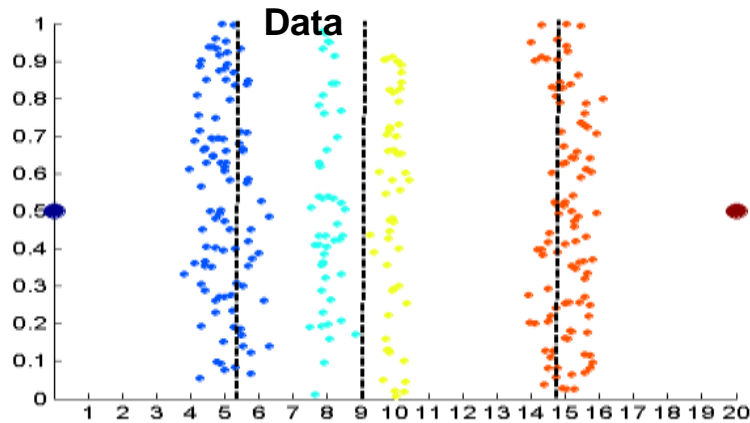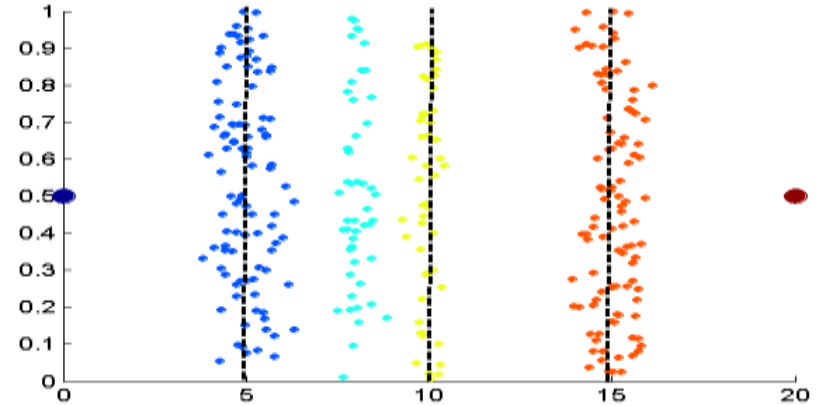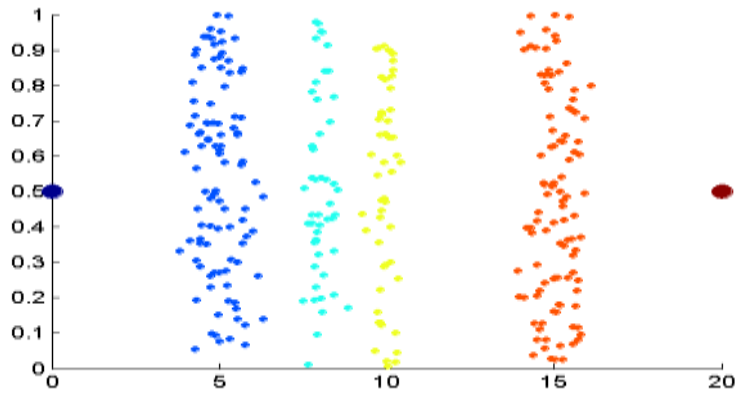
    - Bin 3: 26, 28, 29, 34

* Smoothing by **bin means**:

    - Bin 1: 9, 9, 9, 9
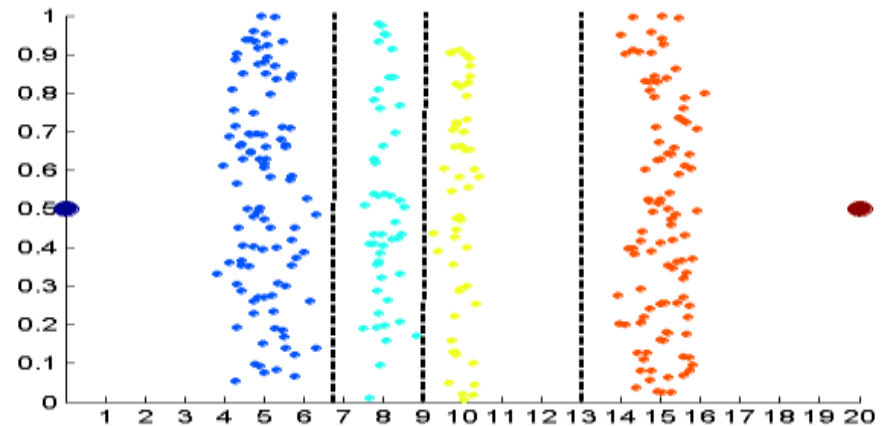
    - Bin 2: 23, 23, 23, 23

    - 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 Without Using Class Labels
(Binning vs. Clustering)

Equal frequency (binning)

K-means clustering leads to better results

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 <u>drilling and rolling</u> 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
  - {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

country	15 distinct values
province_or_ state	365 distinct values
city	3567 distinct values
street	674,339 distinct values