Data Mining:
Concepts and Techniques

— Chapter 2 —

Data Preprocessing

Kiran Bhowmick

Chapter 2: Data Preprocessing

- Why preprocess the data?
- Descriptive data summarization
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., occupation=“ ”
  - noisy: containing errors or outliers
    - e.g., Salary=“-10”
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age=“42” Birthday=“03/07/1997”
    - e.g., Was rating “1,2,3”, now rating “A, B, C”
    - e.g., discrepancy between duplicate records

Why Is Data Dirty?

- Incomplete data may come from
  - "Not applicable" data value when collected
  - Different considerations between the time when the data was collected and when it is analyzed.
  - Human/hardware/software problems
- Noisy data (incorrect values) may come from
  - Faulty data collection instruments
  - Human or computer error at data entry
  - Errors in data transmission
- Inconsistent data may come from
  - Different data sources
  - Functional dependency violation (e.g., modify some linked data)
- Duplicate records also need data cleaning

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

Multi-Dimensional Measure of Data Quality

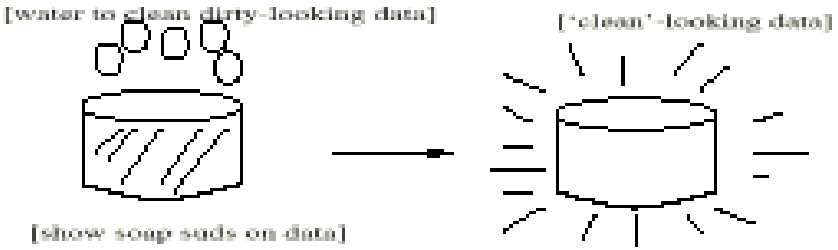
- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility
- Broad categories:
  - Intrinsic, contextual, representational, and accessibility

Major Tasks in Data Preprocessing

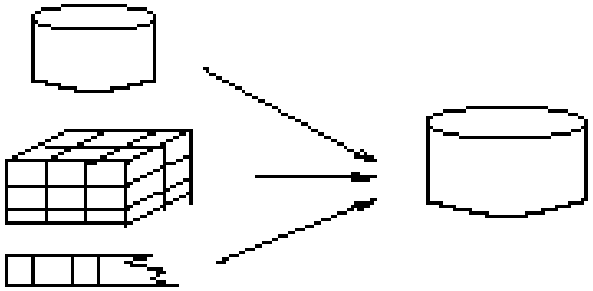
- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data

Forms of Data Preprocessing

Data Cleaning



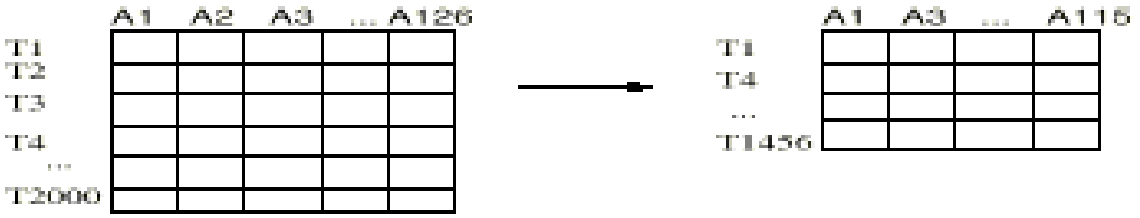
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction



Data Cleaning

- Real world data
  - Incomplete
  - Noisy
  - inconsistent
- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data
  - Resolve redundancy caused by data integration

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 (assuming the tasks in classification—not effective when the percentage 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: regression, 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 requires data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

How to Handle Noisy Data?

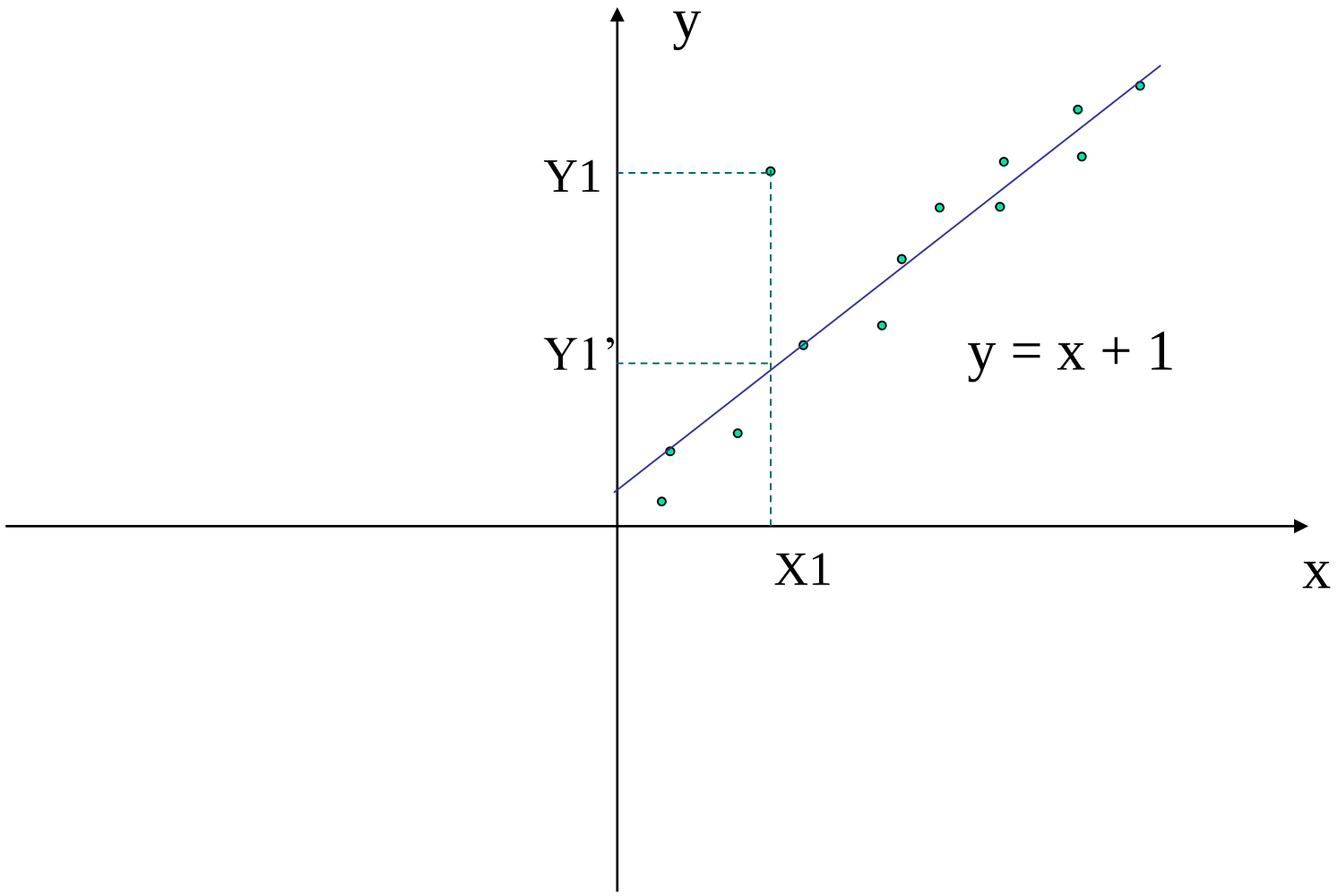
- Binning
  - is a data pre-processing method used to minimize the effects of small observation errors
  - Smooth the data by consulting its neighborhood
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
  - This has a smoothing effect on the input data and may also reduce the chances of overfitting in the case of small datasets
- 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)

Binning Methods for Data Smoothing

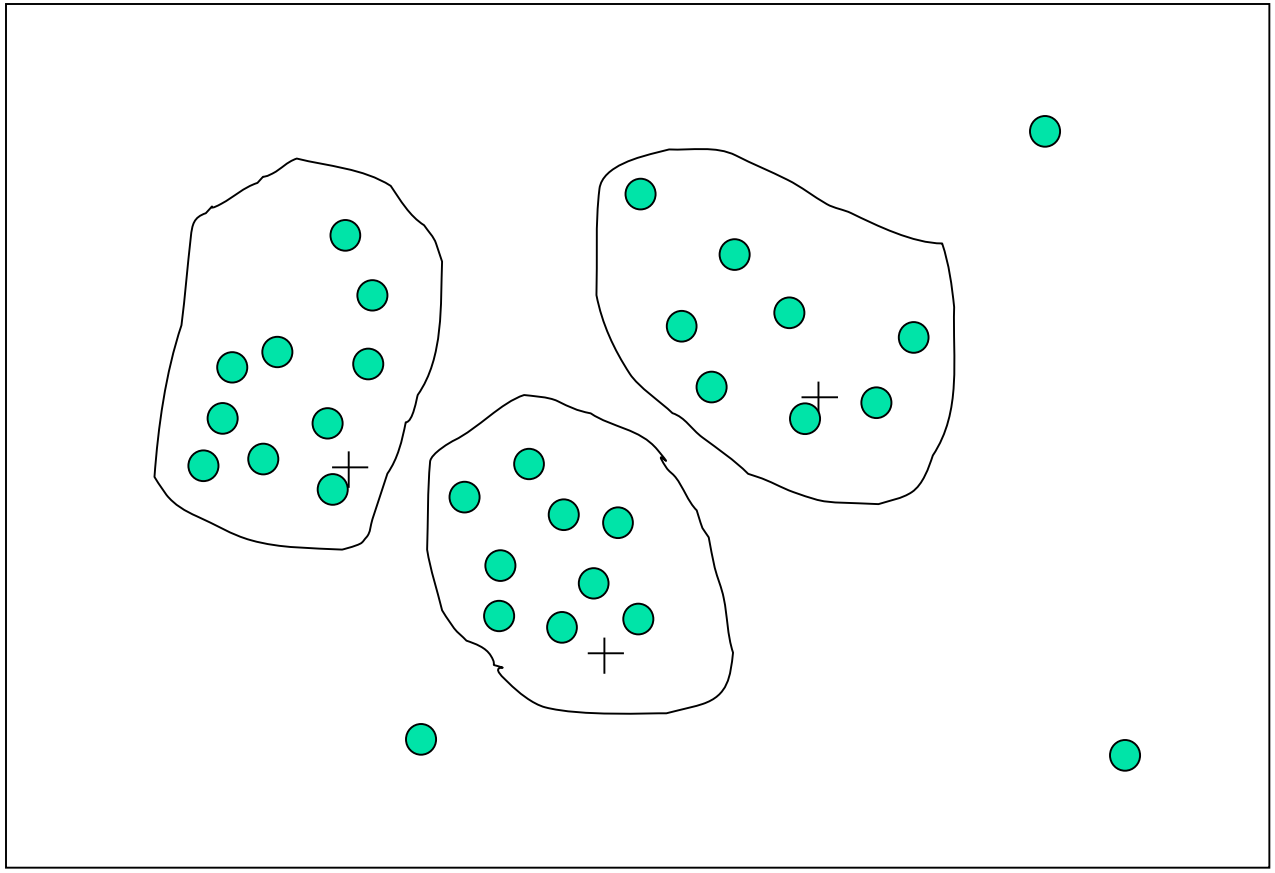
- Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* 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

Regression



Cluster Analysis



Data Cleaning as a Process

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

Data Integration

- Data integration:
  - Combines data from multiple sources into a coherent store
  - Should avoid redundancies and inconsistencies
  - Help improve accuracy and speed of the data mining process
- Entity identification problem:
- Schema integration and object matching:
  - e.g., A.cust-id ≡ B.cust-#
  - Pay_band; 'H','S' ≡ 1 and 2
- Integrate metadata from different sources
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - *Object identification*: The same attribute or object may have different names in different databases
  - *Derivable data:* One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by *correlation analysis*
- Given two attributes, correlation analysis implies how strongly one attribute implies the other based on the available data.

Correlation Analysis (Categorical Data)

- X^2 (chi-square) test

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

- The larger the X^2 value, the more likely the variables are related
- The cells that contribute the most to the X^2 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

Correlation Analysis (Numerical Data)

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

$$r_{A,B} = \frac{\sum (A - \overline{A})(B - \overline{B})}{(n-1)\sigma_A \sigma_B} = \frac{\sum (AB) - n\overline{A}\,\overline{B}}{(n-1)\sigma_A \sigma_B}$$

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 Σ(AB) 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; $r_{A,B} < 0$: negatively correlated

Covariance of Numeric Data

- Consider two numeric attributes A and B with n observations each.
- The expected value $E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n}$
- $E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}$
- The covariance between A & B is
- $Cov(A, B) = E((A - \bar{A})\ (B - \bar{B})) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})\ (b_i - \bar{B})}{n}$
- Correlation is $r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$

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

Chapter 2: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Data Reduction Strategies

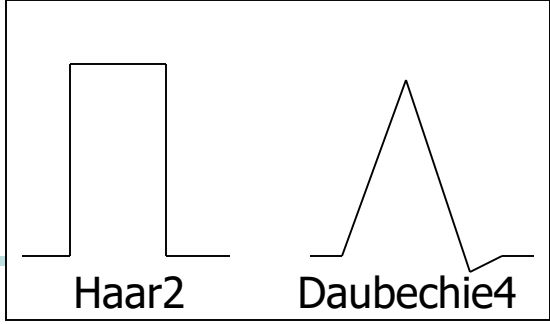
- Why data reduction?
  - A database/data warehouse may store terabytes of data
  - Complex data analysis/mining may take a very long time to run on the complete data set

- Data reduction
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results

Data Reduction Strategies

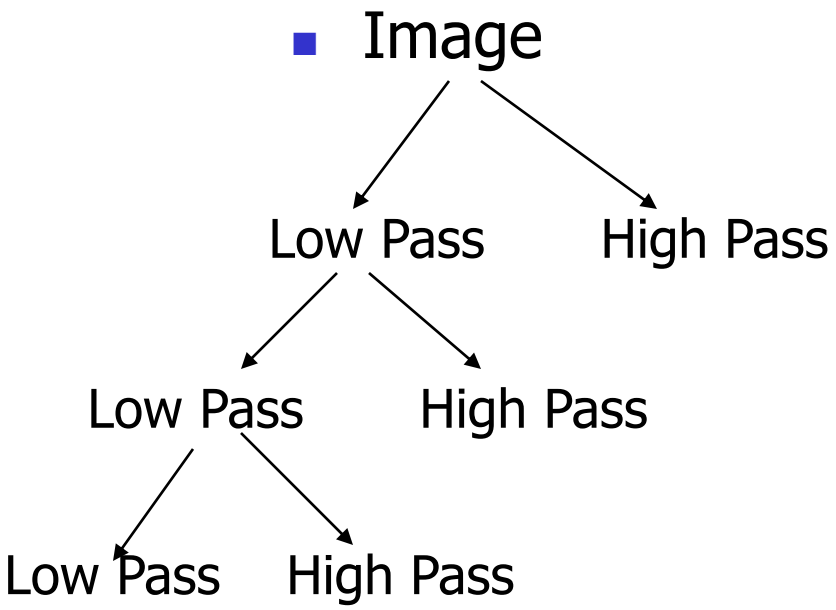
- Data reduction strategies
  - Dimensionality reduction — e.g., remove unimportant attributes
  - Wavelet Transforms
  - Principal Components Analysis
  - Attribute Subset Selection
  - Regression and Log-Linear Models: Parametric Data Reduction
  - Histograms
  - Clustering
  - Sampling
  - Data cube aggregation

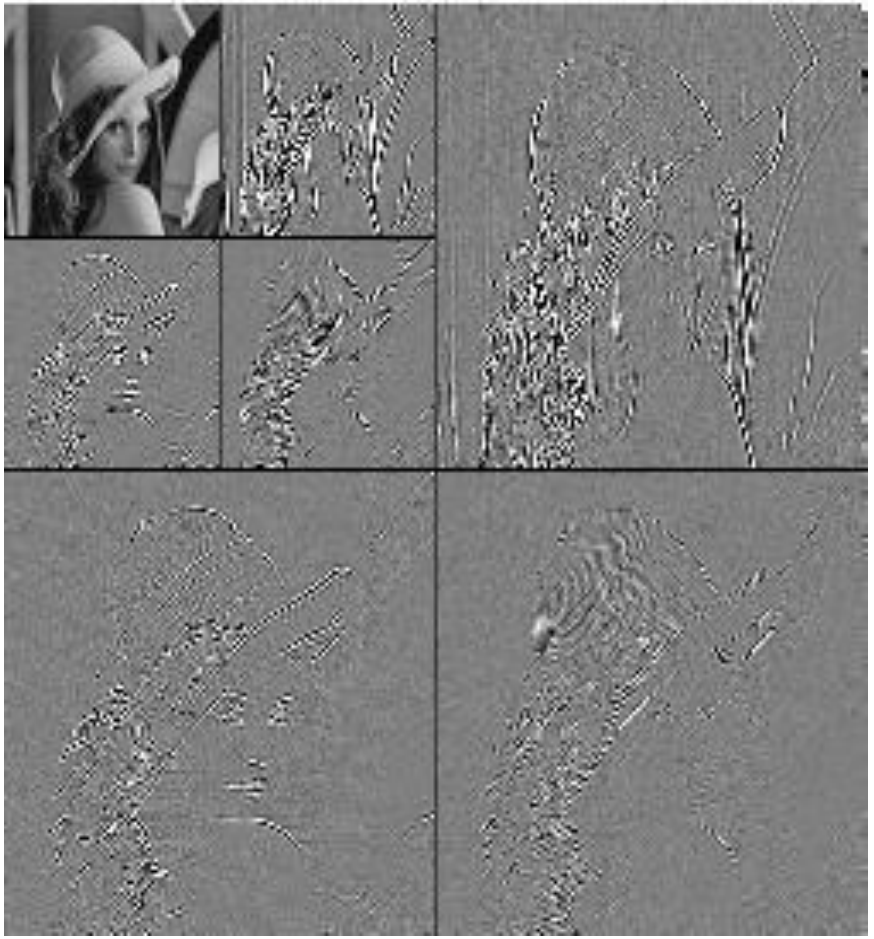
Dimensionality Reduction: Wavelet Transformation



- Discrete wavelet transform (DWT): linear signal processing, multi-resolutional 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

DWT for Image Compression

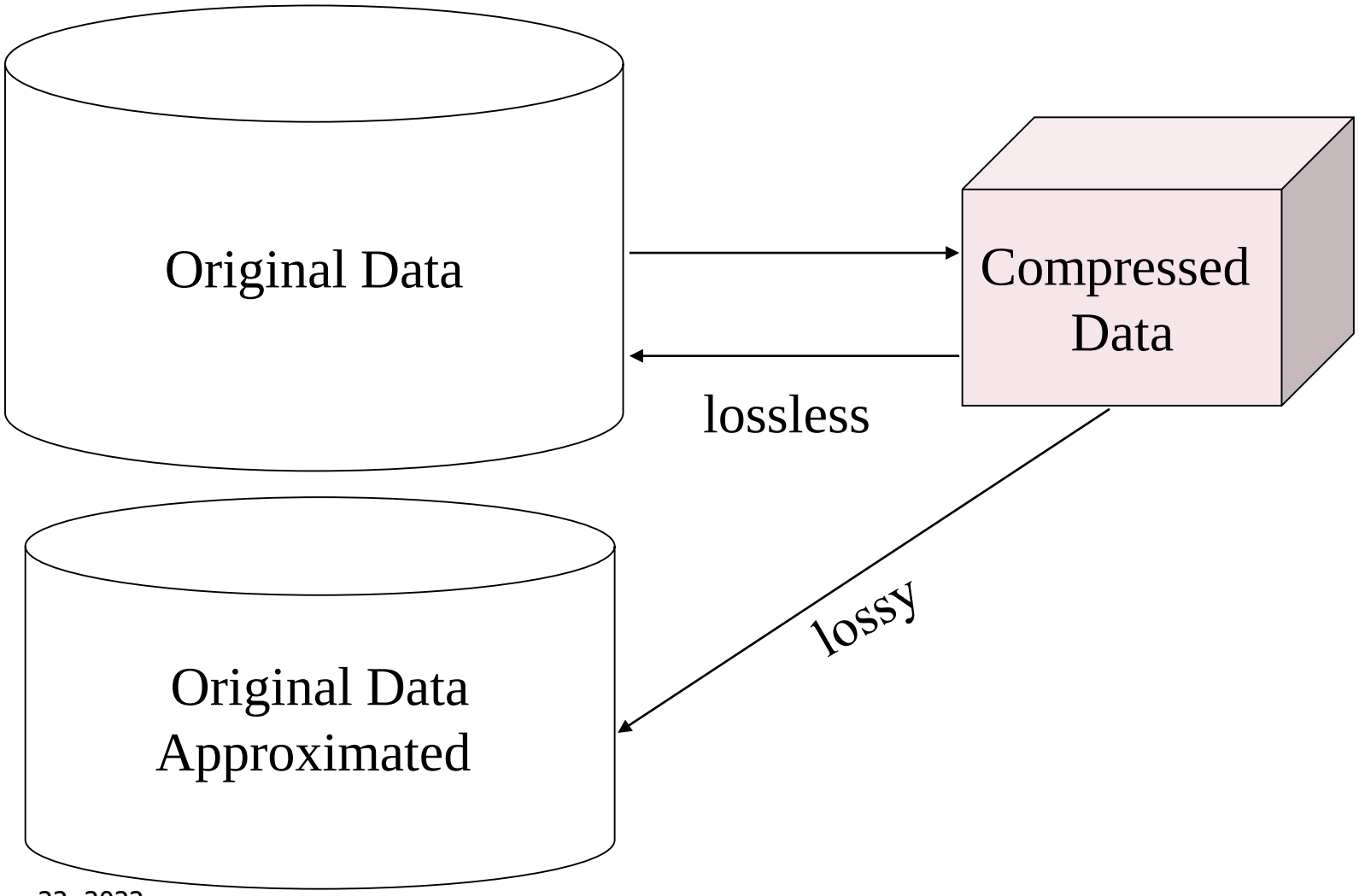




Data Compression

- 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

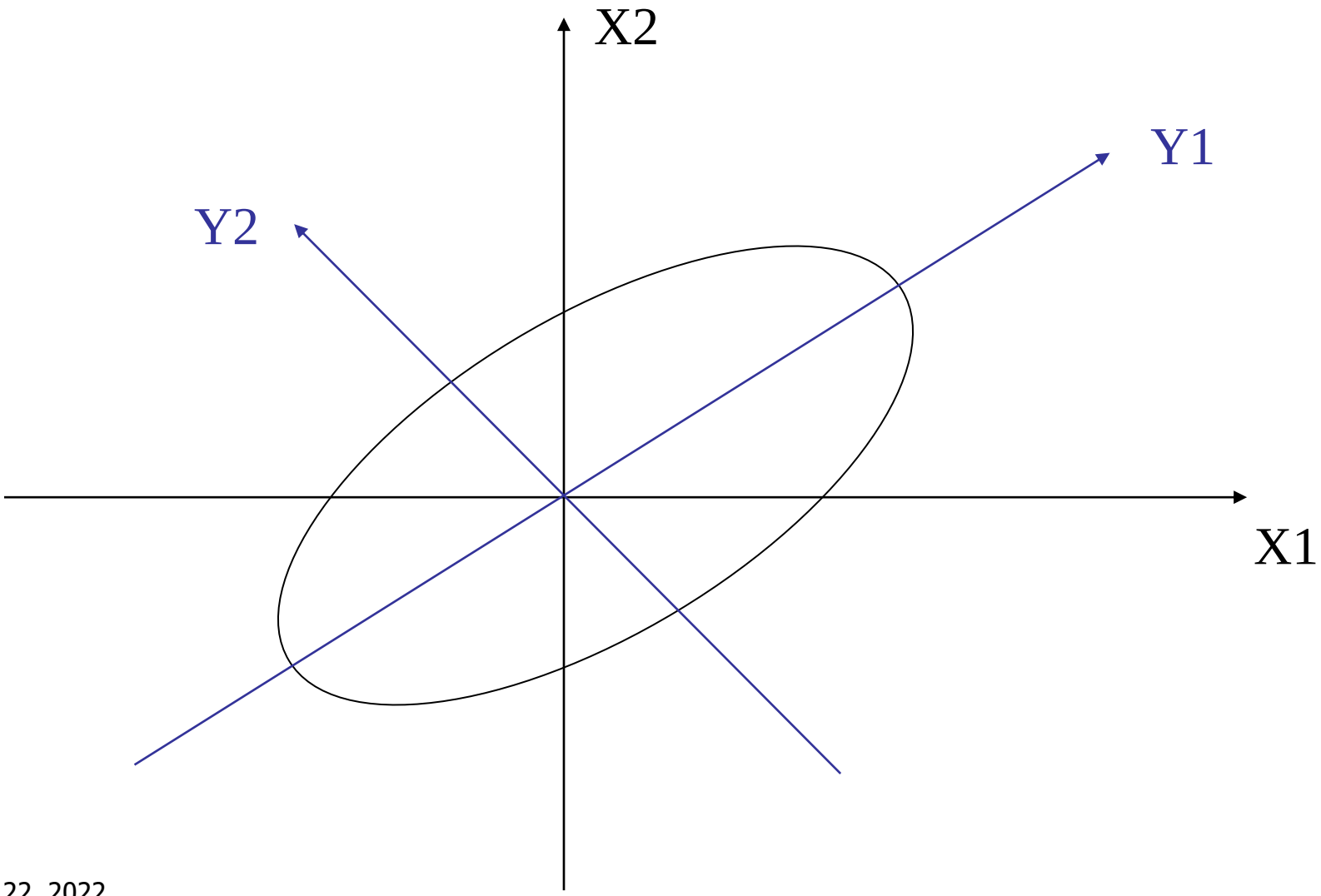
Data Compression



Dimensionality Reduction: Principal Component Analysis (PCA)

- Given *N* data vectors from *n*-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
- Steps
  - 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
- Used when the number of dimensions is large

Principal Component Analysis

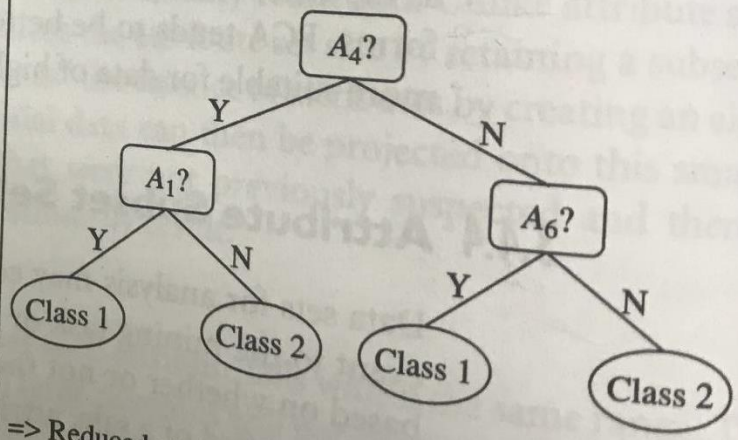


Attribute Subset Selection

- Feature selection (i.e., attribute subset selection):
  - Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
  - reduce # of patterns in the patterns, easier to understand
- Heuristic methods (due to exponential # of choices):
  - Step-wise forward selection
  - Step-wise backward elimination
  - Combining forward selection and backward elimination
  - Decision-tree induction

Heuristic Feature Selection Methods

- There are 2^d possible sub-features of d features
- Several heuristic feature selection methods:
  - Best single features under the feature independence assumption: choose by significance tests
  - Best step-wise feature selection:
    - The best single-feature is picked first
    - Then next best feature condition to the first, ...
  - Step-wise feature elimination:
    - Repeatedly eliminate the worst feature
  - Best combined feature selection and elimination
  - Optimal branch and bound:
    - Use feature elimination and backtracking

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: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$

3.6 Greedy (heuristic) methods for attribute subset selection.

[4] In machine l...

Data Reduction Method: Parametric Data Reduction

- Reduce data volume by choosing alternative, smaller forms of data representation
- Parametric methods
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Example: 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

Data Reduction Method (1): Regression and Log-Linear Models

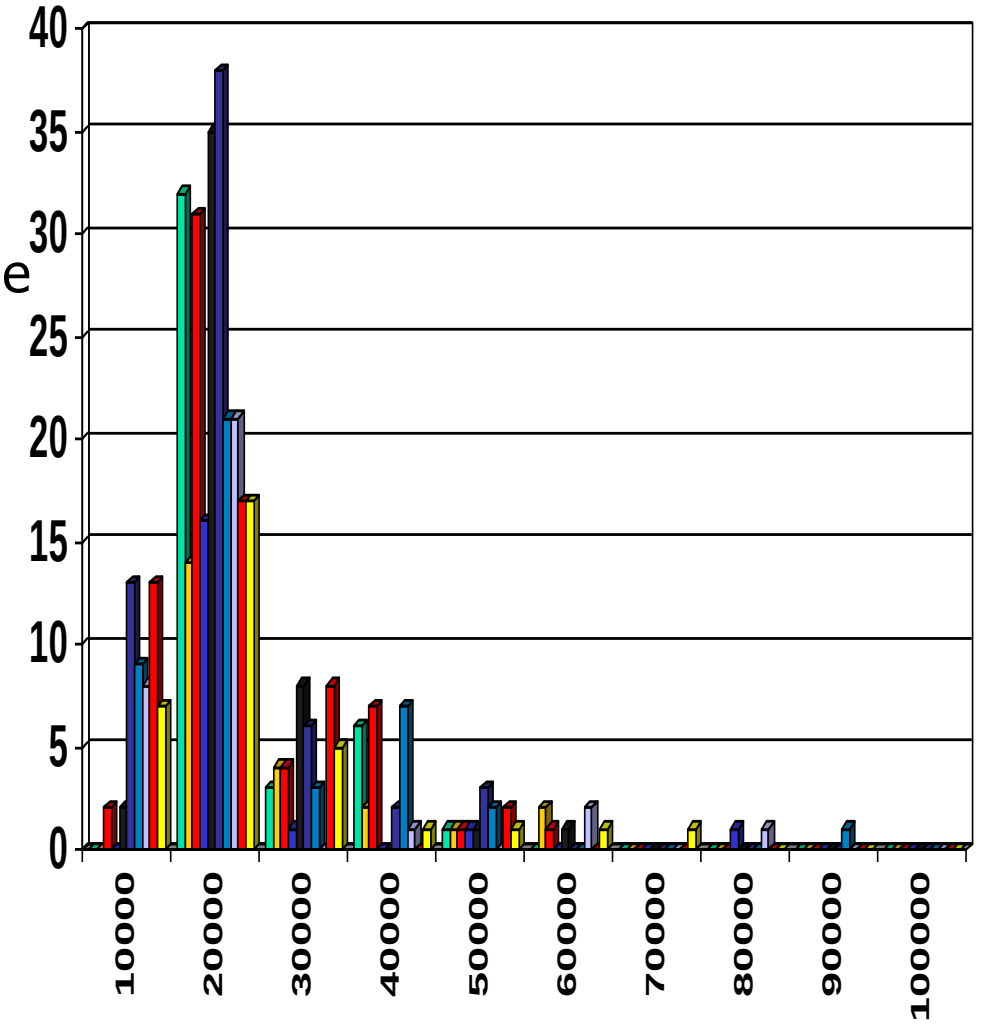
- Linear regression: Data are modeled to fit a straight line
  - Often uses the least-square method to fit the line
- Multiple regression: allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- Log-linear model: approximates discrete multidimensional probability distributions

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 $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.
- Multiple regression: $Y = b0 + b1 X1 + b2 X2.$
  - Many nonlinear functions can be transformed into the above
- Log-linear models:
  - Estimate probability of each point in a multidimensional space for a set of discretized attributes based on a smaller subset of dimensional combinations.
  - Allows a higher dimensional data space to be constructed from lower-dimensional spaces.

Data Reduction Method (2): Histograms

- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
  - Equal-width: equal bucket range
  - Equal-frequency (or equal-depth)



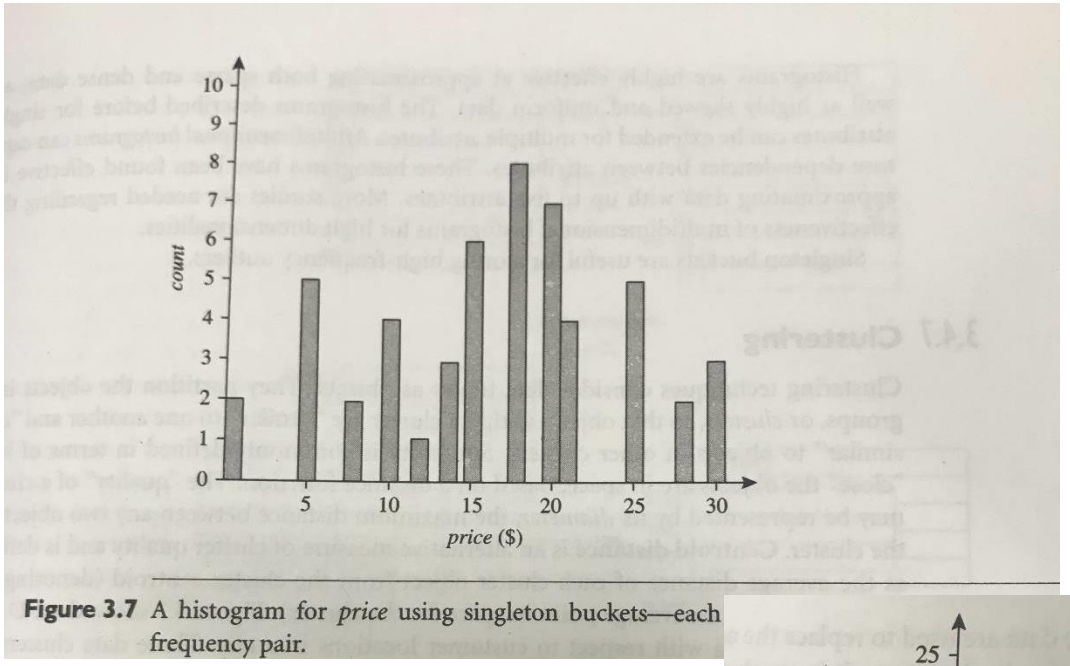


Figure 3.7 A histogram for *price* using singleton buckets—each frequency pair.

1,1,5,5,5,5,5,8,8,10,10,10,10,12,14,14,14,15,15,15,15,15,15,18,18,18,18,18,18,18,18,20,20,20,20,20,20,21,21,21,21,25,25,25,25,25,28,28,30,30,30

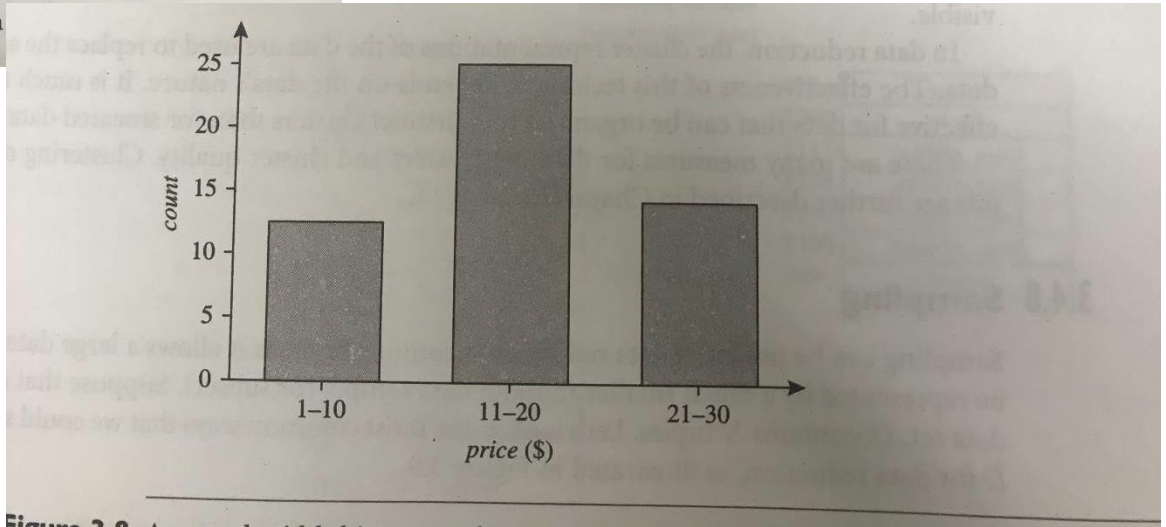


Figure 3.8 An equal-width histogram for *price*, where values are aggregated so that each bucke

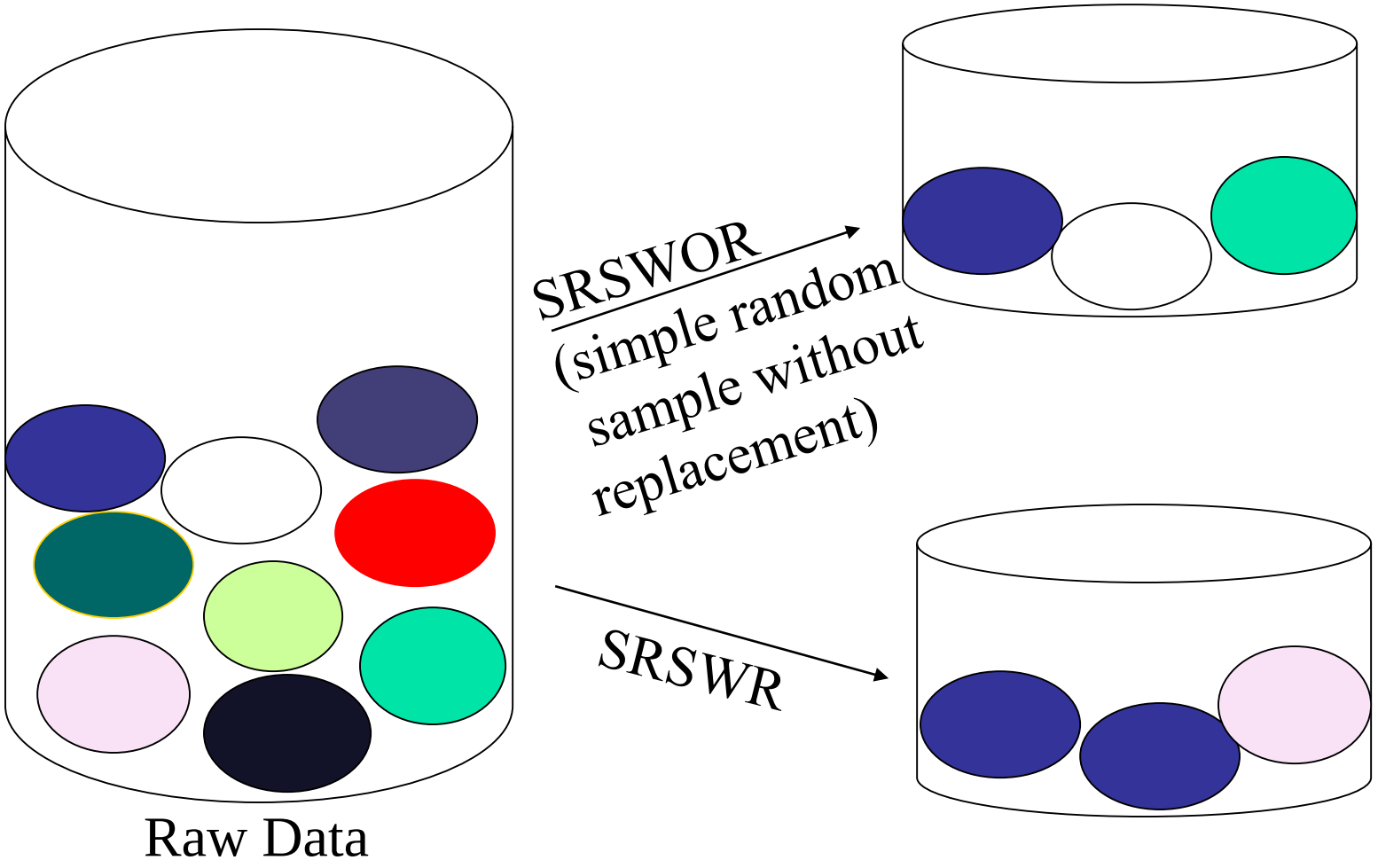
Data Reduction Method (3): Clustering

- 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 7

Data Reduction Method (4): Sampling

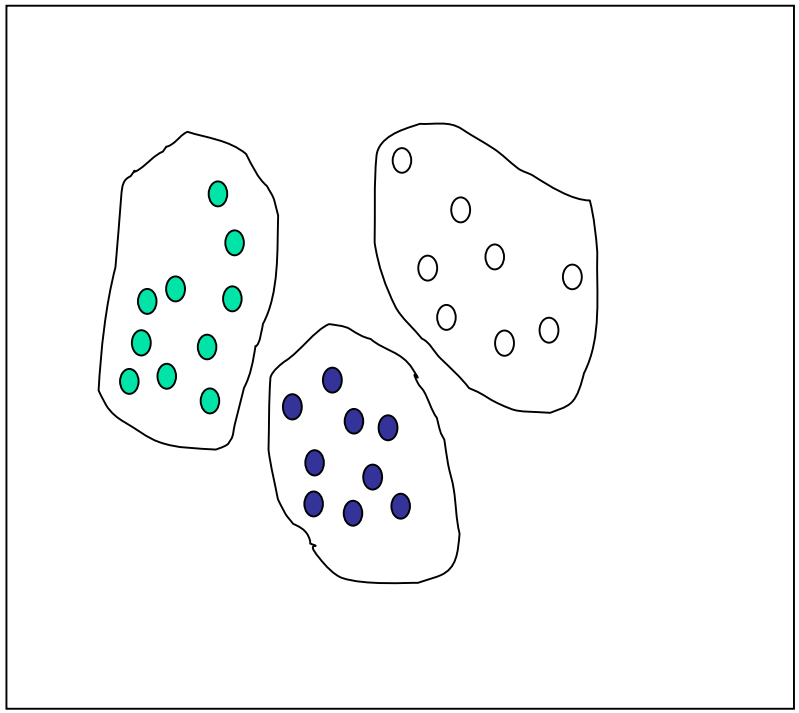
- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Used in conjunction with skewed data
- Note: Sampling may not reduce database I/Os (page at a time)

Sampling: with or without Replacement

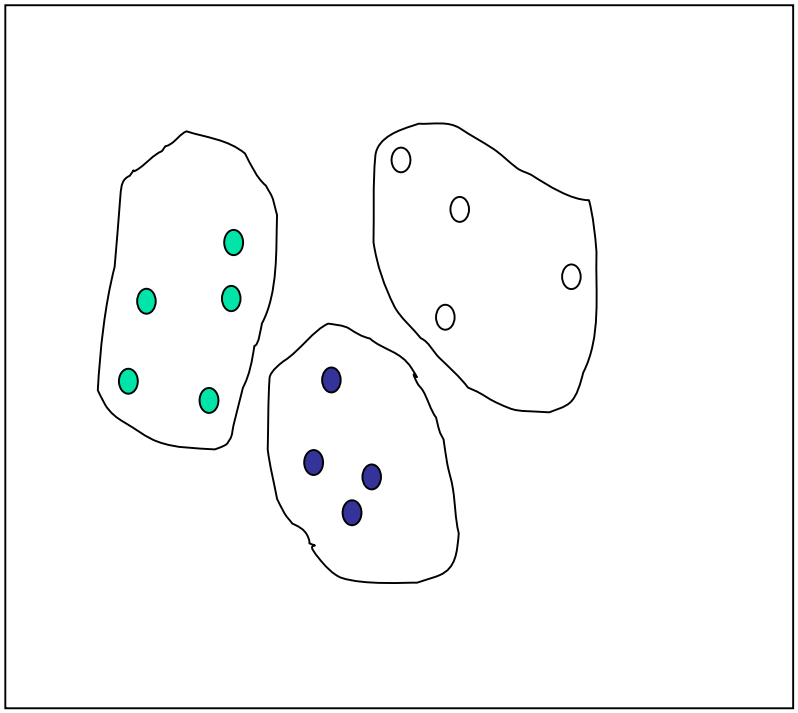


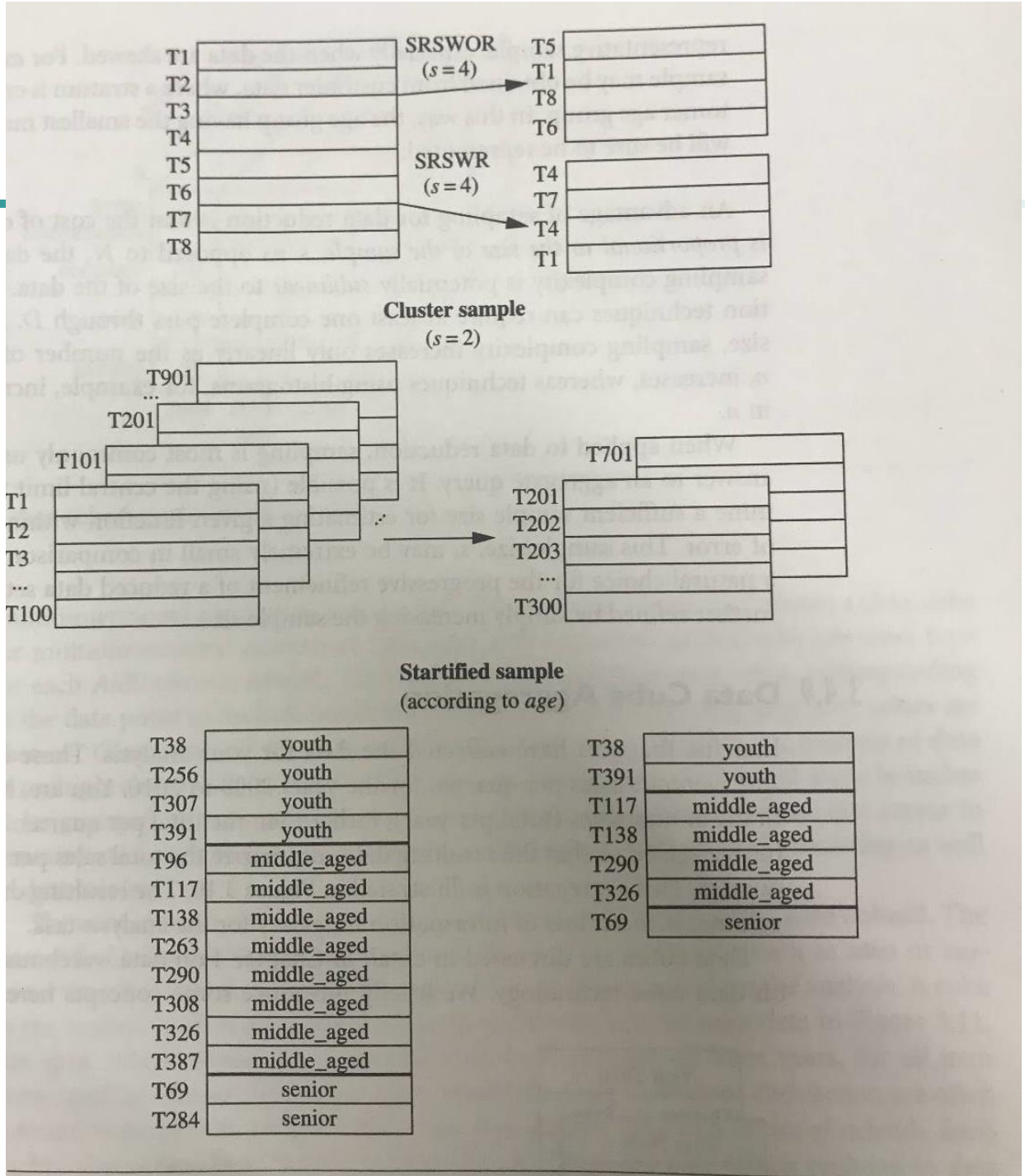
Sampling: Cluster or Stratified Sampling

Raw Data



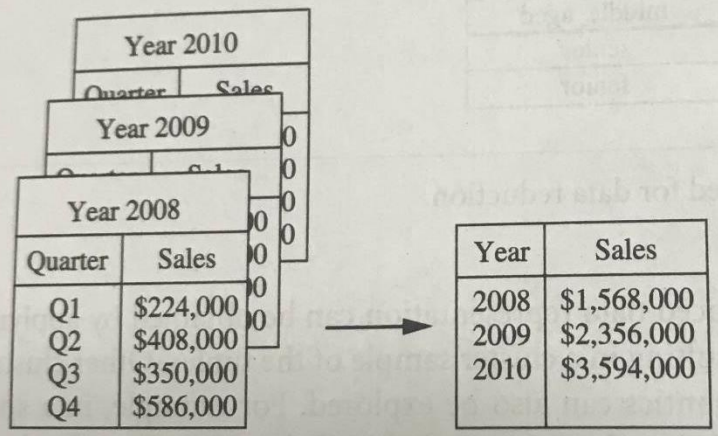
Cluster/Stratified Sample



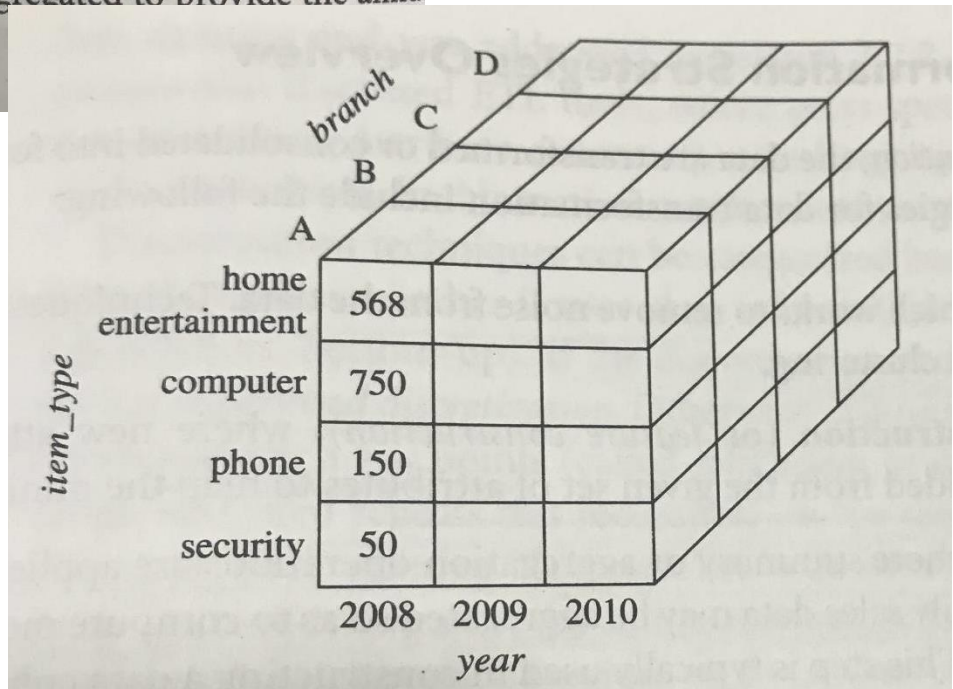

T1
T2
T3
T4
T5
T6
T7
T8
SRSWOR
(s = 4)
T5
T1
T8
T6
SRSWR
(s = 4)
T4
T7
T4
T1
Cluster sample
(s = 2)
T901
...
T201
T101
T1
T2
T3
...
T100
T701
T201
T202
T203
...
T300
Startified sample
(according to age)
T38 youth
T256 youth
T307 youth
T391 youth
T96 middle_aged
T117 middle_aged
T138 middle_aged
T263 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

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



e 3.10 Sales data for a given branch of *AllElectronics* for the years 2008 through 2010. On the *lef* the sales are shown per quarter. On the *right*, the data are aggregated to provide the annu sales.



A data cube for sales at *AllElectronics*.

Data Transformation

- 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 small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Discretization: raw values are replaced by interval labels
- Generalization: concept hierarchy climbing

Data Transformation: 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):

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

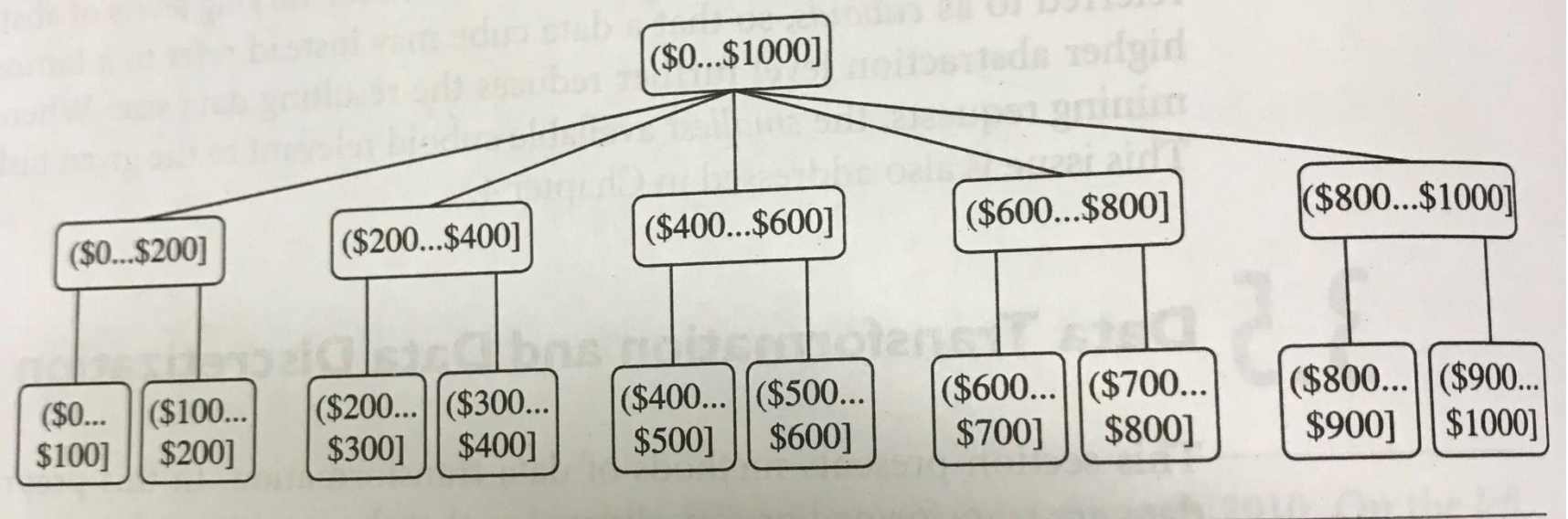
  - Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- Normalization by decimal scaling

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

Discretization

- Raw values of numeric attribute (e.g. age) are replaced by interval labels (e.g. 0-10, 11-20 etc) or conceptual labels (e.g. youth, middle aged, senior)
- The labels can be then recursively organized into higher-level concepts resulting in concept hierarchy



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

Discretization

- Discretization techniques are categorized based on how the discretization is performed.
- Using class information – supervised otherwise unsupervised
- If starts by splitting entire attribute to result into a specific interval – top down discretization or splitting
- If starts by merging values to form intervals – bottom up discretization

Discretization and Concept Hierarchy Generation for Numeric Data

- Typical methods: All the methods can be applied recursively
  - Binning (covered above)
    - Top-down split, unsupervised,
  - Histogram analysis (covered above)
    - Top-down split, unsupervised
  - Clustering analysis (covered above)
    - Either top-down split or bottom-up merge, unsupervised
  - Entropy-based discretization: supervised, top-down split
  - Interval merging by χ^2 Analysis: unsupervised, bottom-up merge
  - Segmentation by natural partitioning: top-down split, unsupervised

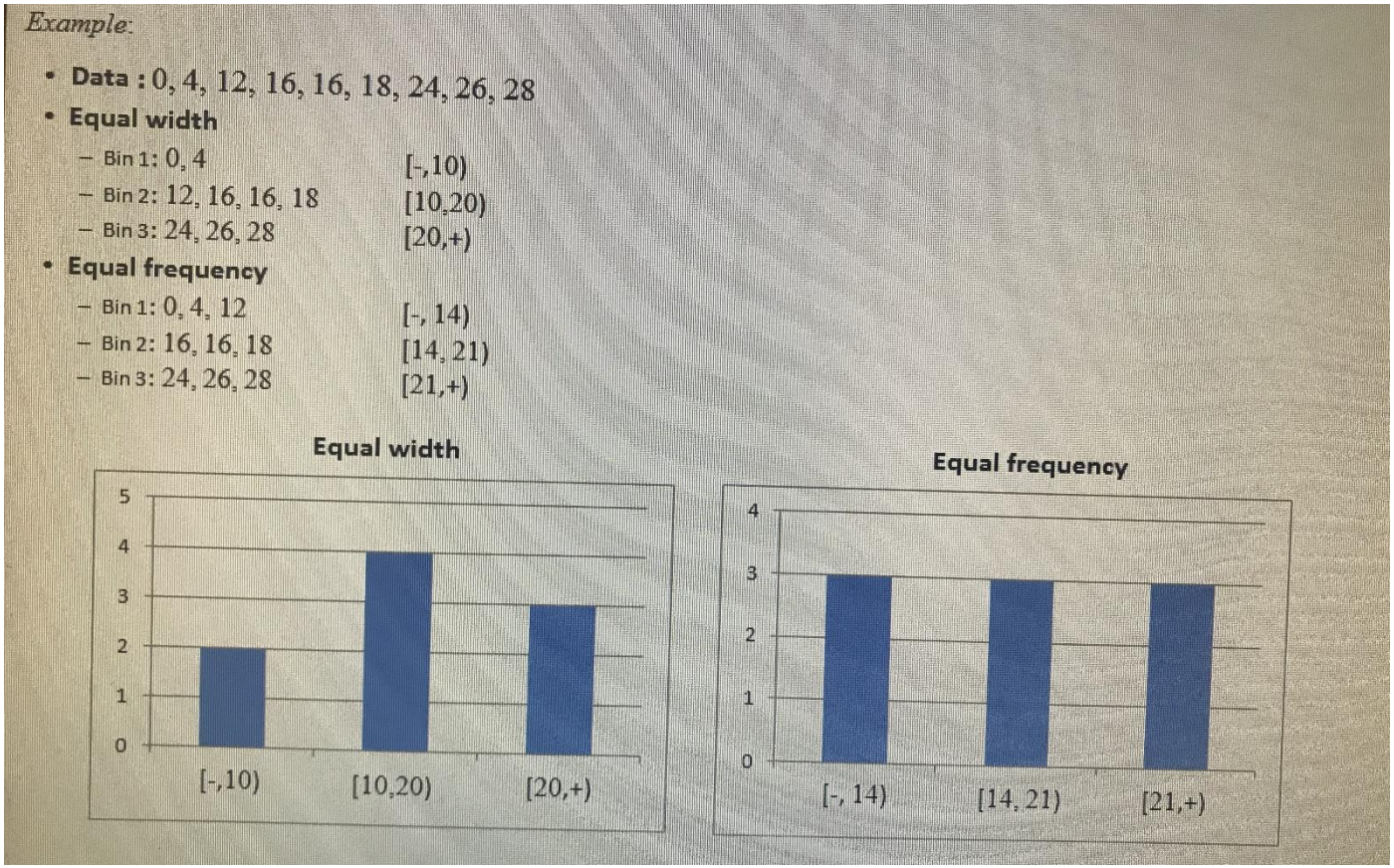
Discretization by binning

- Binning is a top-down splitting technique based on specified number of bins
- Attribute values are discretized by either Equal-width or equal-frequency binning, and then replacing each bin value by the bin mean or median as in smoothing
- Unsupervised discretization technique as it does not use class information
- It is sensitive to user-specified number of bins as well as the presence of outliers.

Example of binning

Example:

- **Data :** 0, 4, 12, 16, 16, 18, 24, 26, 28
- **Equal width**
  - Bin 1: 0, 4 [-,10)
  - Bin 2: 12, 16, 16, 18 [10,20)
  - Bin 3: 24, 26, 28 [20,+)
- **Equal frequency**
  - Bin 1: 0, 4, 12 [-, 14)
  - Bin 2: 16, 16, 18 [14, 21)
  - Bin 3: 24, 26, 28 [21,+)



Discretization by Histogram Analysis

- Like binning, histogram analysis is an unsupervised discretization technique
- Histogram partitions the values of an attribute into disjoint ranges called buckets or bins.
- Various partitioning rules can be used to define histograms like equal-width or equal-frequency
- The histogram analysis is then applied recursively to each partition in order to automatically generate multilevel concept hierarchy.
- The recursive procedure terminates when a pre specified number of concept levels has been reached.

Discretization by Clustering

- Clustering is a popular discretization method which is applied to numeric attribute.
- Clustering can be used to generate concept hierarchy by either top down splitting strategy or bottom up merging strategy.
- Clustering takes the distribution into consideration and also the closeness of the data points and hence is able to produce high-quality discretization results.

Discretization by classification

- Techniques to generate decision trees for classification can be applied to discretization.
- These techniques employ a top-down splitting approach and are supervised in nature.
- Class distribution information is used in the calculation and determination of split-points.
- The split-points are selected such that the each resulting partition contains instances of the same class.
- Entropy measure is used to select the split-point

Entropy-Based Discretization

- Given a set of samples S, if S is partitioned into two intervals S_1 and S_2 using boundary T, the information gain after partitioning is

$$I(S,T) = \frac{|S_1|}{|S|} Entropy(S_1) + \frac{|S_2|}{|S|} Entropy(S_2)$$

- Entropy is calculated based on class distribution of the samples in the set. Given *m* classes, the entropy of S_1 is

$$Entropy(S_1) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

where p_i is the probability of class *i* in S_1

- The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization
- The process is recursively applied to partitions obtained until some stopping criterion is met
- Such a boundary may reduce data size and improve classification accuracy

Discretization of correlation

- Measure of correlation can be used for discretization.
- ChiMerge is a χ^2 – based discretization method.
- It employs bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals
- Initially, each distinct value of a numerical attr. A is considered to be one interval
  - χ^2 tests are performed for every pair of adjacent intervals
  - Adjacent intervals with the least χ^2 values are merged together, since low χ^2 values for a pair indicate similar class distributions
  - This merge process proceeds recursively until a predefined stopping criterion is met (such as significance level, max-interval, max inconsistency, etc.)

Interval Merge by χ^2 Analysis

- Merging-based (bottom-up) vs. splitting-based methods
- Merge: Find the best neighboring intervals and merge them to form larger intervals recursively
- ChiMerge [Kerber AAAI 1992, See also Liu et al. DMKD 2002]
  - Initially, each distinct value of a numerical attribute A is considered to be one interval
  - χ^2 tests are performed for every pair of adjacent intervals
  - Adjacent intervals with the least χ^2 values are merged together, since low χ^2 values for a pair indicate similar class distributions
  - This merge process proceeds recursively until a predefined stopping criterion is met (such as significance level, max-interval, max inconsistency, etc.)

Concept Hierarchy Generation for Nominal Data

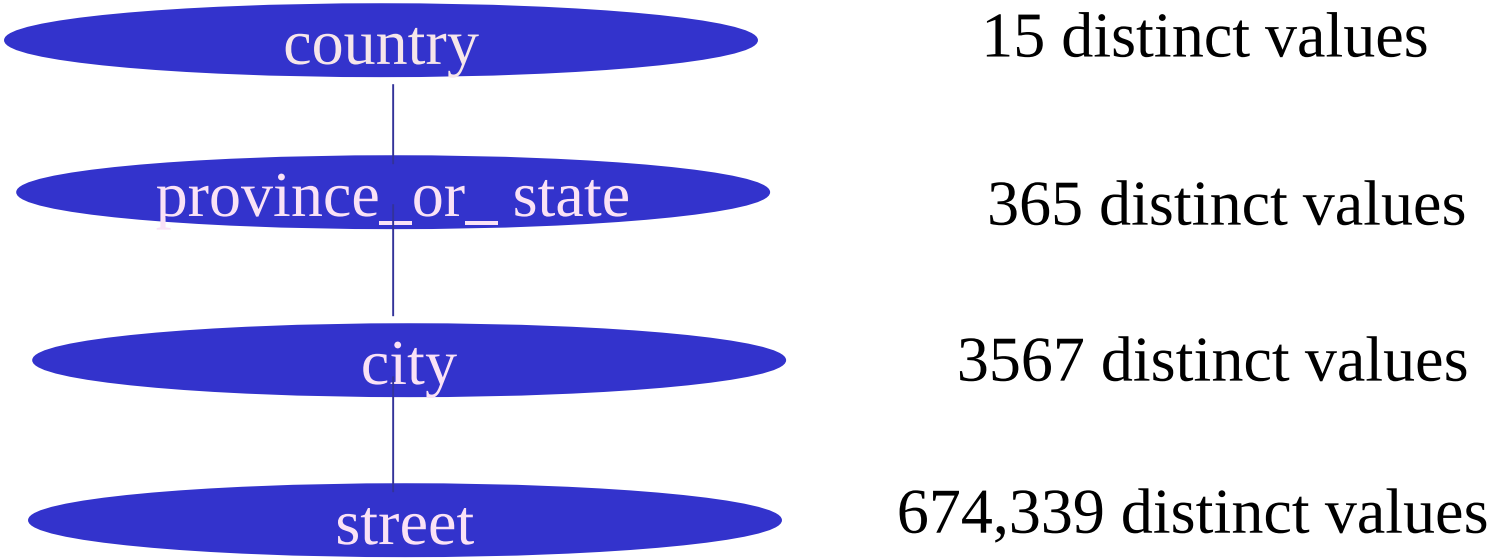
- Nominal data have distinct values and no ordering among the values.
- Manual definition of concept hierarchies can be a tedious and time-consuming task.
- However, many hierarchies are implicit within the database schema and can be automatically defined at the schema level.
- The concept hierarchies can be used to transform the data into multiple levels of granularity.
- Four methods for generation of concept hierarchies are as follows:

Concept Hierarchy Generation for Categorical 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 portion of a hierarchy by explicit data grouping
  - {Maharashtra, Gujarat, Rajasthan, Goa} C West India
  - {West India, East India} C India
- Specification of a set of attributes, but not of their partial orderings
  - Specify set of attributes forming concept hierarchy but omit to explicitly state their partial ordering
  - 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}
- Specification of only a partial set of attributes
  - E.g., only street < city, not others

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

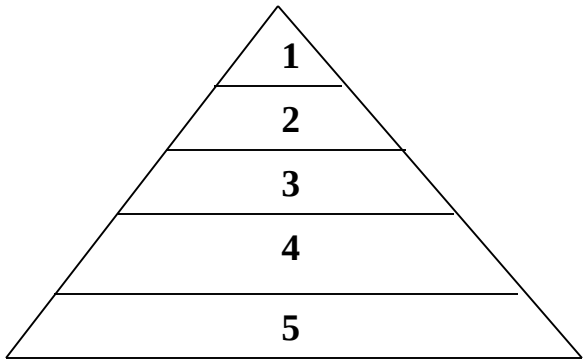


What is Concept Description?

- Descriptive vs. predictive data mining
  - Descriptive mining: describes concepts or task-relevant data sets in concise, summarative, informative, discriminative forms
  - Predictive mining: Based on data and analysis, constructs models for the database, and predicts the trend and properties of unknown data
- Concept description:
  - Characterization: provides a concise and succinct summarization of the given collection of data
  - Comparison: provides descriptions comparing two or more collections of data

Data Generalization and Summarization-based Characterization

- Data generalization
  - A process which abstracts a large set of task-relevant data in a database from a low conceptual levels to higher ones.



Conceptual levels

  - Approaches:
    - Data cube approach(OLAP approach)
    - Attribute-oriented induction approach

Concept Description vs. OLAP

- Similarity:
  - Data generalization
  - Presentation of data summarization at multiple levels of abstraction.
  - Interactive drilling, pivoting, slicing and dicing.
- Differences:
  - Can handle complex data types of the attributes and their aggregations
  - Automated desired level allocation.
  - Dimension relevance analysis and ranking when there are many relevant dimensions.
  - Sophisticated typing on dimensions and measures.
  - Analytical characterization: data dispersion analysis

Attribute-Oriented Induction

- Proposed in 1989 (KDD '89 workshop)
- Not confined to categorical data nor particular measures
- How it is done?
  - Collect the task-relevant data (*initial relation*) using a relational database query
  - Perform generalization by attribute removal or attribute generalization
  - Apply aggregation by merging identical, generalized tuples and accumulating their respective counts
  - Interactive presentation with users

Basic Principles of Attribute-Oriented Induction

- Data focusing: task-relevant data, including dimensions, and the result is the *initial relation*
- Attribute-removal: remove attribute *A* if there is a large set of distinct values for *A* but (1) there is no generalization operator on *A*, or (2) *A*'s higher level concepts are expressed in terms of other attributes
- Attribute-generalization: If there is a large set of distinct values for *A*, and there exists a set of generalization operators on *A*, then select an operator and generalize *A*
- Attribute-threshold control: typical 2-8, specified/default
- Generalized relation threshold control: control the final relation/rule size

Attribute-Oriented Induction: Basic Algorithm

- InitialRel: Query processing of task-relevant data, deriving the *initial relation*.
- PreGen: Based on the analysis of the number of distinct values in each attribute, determine generalization plan for each attribute: removal? or how high to generalize?
- PrimeGen: Based on the PreGen plan, perform generalization to the right level to derive a "prime generalized relation", accumulating the counts.
- Presentation: User interaction: (1) adjust levels by drilling, (2) pivoting, (3) mapping into rules, cross tabs, visualization presentations.

Example

- DMQL: Describe general characteristics of graduate students in the Big-University database

  use Big_University_DB

  mine characteristics as "Science_Students"

  in relevance to name, gender, major, birth_place, birth_date, residence, phone#, gpa

  from student

  where status in "graduate"

- Corresponding SQL statement:

  Select name, gender, major, birth_place, birth_date, residence, phone#, gpa

  from student

  where status in {"Msc", "MBA", "PhD" }

Class Characterization: An Example

Initial Relation

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Jim Woodman	M	CS	Vancouver,BC, Canada	8-12-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83
…	…	…	…	…	…	…	…
Removed	Retained	Sci,Eng, Bus	Country	Age range	City	Removed	Excl, VG,..

Prime Generalized Relation

Gender	Major	Birth_region	Age_range	Residence	GPA	Count
M	Science	Canada	20-25	Richmond	Very-good	16
F	Science	Foreign	25-30	Burnaby	Excellent	22
…	…	…	…	…	…	…

Gender \ Birth_Region	Canada	Foreign	Total
M	16	14	30
F	10	22	32
Total	26	36	62

Presentation of Generalized Results

- Generalized relation:
  - Relations where some or all attributes are generalized, with counts or other aggregation values accumulated.
- Cross tabulation:
  - Mapping results into cross tabulation form (similar to contingency tables).
  - Visualization techniques:
  - Pie charts, bar charts, curves, cubes, and other visual forms.
- Quantitative characteristic rules:
  - Mapping generalized result into characteristic rules with quantitative information associated with it, e.g.,

$$grad(x) \wedge male(x) \Rightarrow$$
$$birth\_region(x) = \text{"}Canada\text{"}[t:53\%] \vee birth\_region(x) = \text{"}foreign\text{"}[t:47\%].$$

Mining Class Comparisons

- Comparison: Comparing two or more classes
- Method:
  - Partition the set of relevant data into the target class and the contrasting class(es)
  - Generalize both classes to the same high level concepts
  - Compare tuples with the same high level descriptions
  - Present for every tuple its description and two measures
    - support - distribution within single class
    - comparison - distribution between classes
  - Highlight the tuples with strong discriminant features
- Relevance Analysis:
  - Find attributes (features) which best distinguish different classes

Quantitative Discriminant Rules

- Cj = target class
- q_a = a generalized tuple covers some tuples of class
  - but can also cover some tuples of contrasting class
- d-weight
  - range: [0, 1]

$$d-weight = \frac{count(q_a \in C_j)}{\sum_{i=1}^{m} count(q_a \in C_i)}$$

- quantitative discriminant rule form

$$\forall X,\ target\_class(X) \Leftarrow condition(X)\quad [d : d\_weight]$$

Example: Quantitative Discriminant Rule

Status	Birth_country	Age_range	Gpa	Count
Graduate	Canada	25-30	Good	90
Undergraduate	Canada	25-30	Good	210

Count distribution between graduate and undergraduate students for a generalized tuple

- Quantitative discriminant rule

$$\forall X,\ graduate\_student(X) \Leftarrow$$
$$birth\_country(X) = "Canada" \wedge age\_range(X) = "25-30" \wedge gpa(X) = "good" \quad [d : 30\%]$$

  - where 90/(90 + 210) = 30%

Class Description

- Quantitative characteristic rule

$$\forall X,\ target\_class(X) \Rightarrow condition(X)\ \ [t : t\_weight]$$

  - necessary

- Quantitative discriminant rule

$$\forall X,\ target\_class(X) \Leftarrow condition(X)\ \ [d : d\_weight]$$

  - sufficient

- Quantitative description rule

$$\forall X,\ target\_class(X) \Leftrightarrow$$

$$condition_1(X)\,[t : w_1, d : w'_1] \vee \ldots \vee condition_n(X)\,[t : w_n, d : w'_n]$$

  - necessary and sufficient

Example: Quantitative Description Rule

Location/item		TV			Computer			Both_items	
	Count	*t-wt*	*d-wt*	*Count*	*t-wt*	*d-wt*	*Count*	*t-wt*	*d-wt*
Europe	80	25%	40%	240	75%	30%	320	100%	32%
N_Am	120	17.65%	60%	560	82.35%	70%	680	100%	68%
Both_ regions	200	20%	100%	800	80%	100%	1000	100%	100%

Crosstab showing associated t-weight, d-weight values and total number (in thousands) of TVs and computers sold at AllElectronics in 1998

- Quantitative description rule for target class *Europe*

$$\forall X, Europe(X) \Leftrightarrow$$

$$(item(X) = "TV") [t: 25\%, d: 40\%] \vee (item(X) = "computer") [t: 75\%, d: 30\%]$$

Summary

- Data preparation or preprocessing is a big issue for both data warehousing and data mining
- Discriptive data summarization is need for quality data preprocessing
- Data preparation includes
  - Data cleaning and data integration
  - Data reduction and feature selection
  - Discretization
- A lot a methods have been developed but data preprocessing still an active area of research

References

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Communications of ACM, 42:73-78, 1999
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley & Sons, 2003
- T. Dasu, T. Johnson, S. Muthukrishnan, V. Shkapenyuk. Mining Database Structure; Or, How to Build a Data Quality Browser. SIGMOD'02.
- H.V. Jagadish et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), December 1997
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Bulletin of the Technical Committee on Data Engineering. Vol.23, No.4*
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'2001
- T. Redman. Data Quality: Management and Technology. Bantam Books, 1992
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. Communications of ACM, 39:86-95, 1996
- 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