Data Preprocessing

Lecturer: Dr. Nguyen Ngoc Thao
Department of Computer Science, FIT, HCMUS

Slides adapted from Jiawei Han, Micheline Kamber and Jian Pei

Outline

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

Outline

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

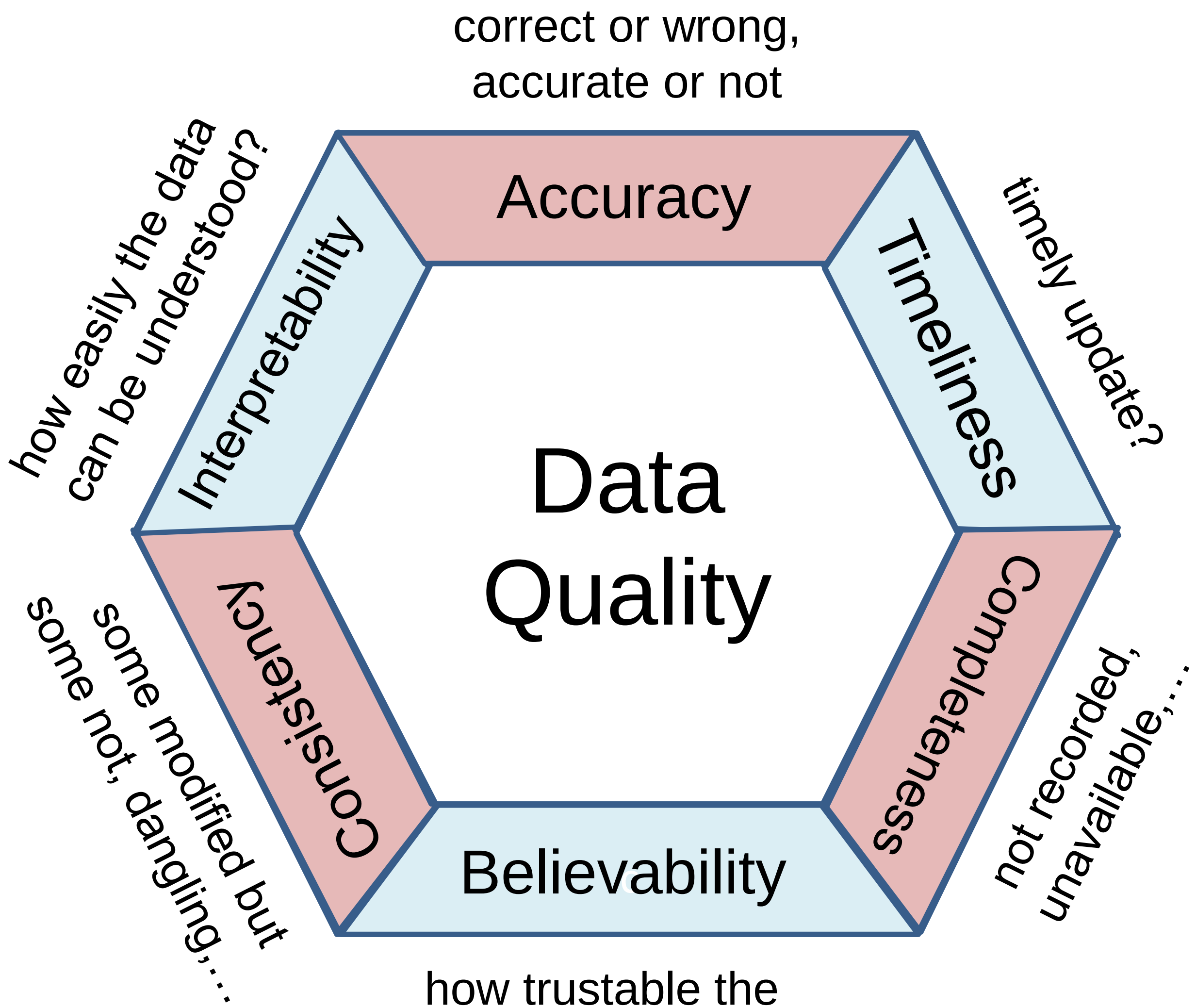
An Fictitious Example

- A manager has been charged to analyze the sales data in his branch
- He inspects the company's database and data warehouse to select the attributes (e.g., item, price, and units sold) to be included in the analysis

- BUT the database has many problems
  - Several attributes of various tuples have no recorded value
  - Information needed for the analysis has not been recorded
  - Many errors, unusual values, and inconsistencies in the data recorded for some transactions have been reported

Inaccurate, incomplete, and inconsistent data are commonplace properties of large real-world databases and data warehouses

Data Quality: Why Preprocess the Data?



Data Quality: Accuracy

- Faulty data collection instruments
- Incorrect data values submitted for mandatory fields
  - Users do not wish to submit personal information
    - E.g., choosing the default value “January 1” displayed for birthday
- Errors in data transmission due to technology limitations
  - E.g., limited buffer size for coordinating synchronized data transfer

Data Quality: Completeness

- Attributes of interest may not always be available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data 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

Data Quality: Consistency

- Inconsistencies in naming conventions or data codes, or inconsistent formats for input fields
  - E.g., 2016/01/21 or 21/01/2016, USA or US
- Duplicate tuples also require data cleaning

Data Quality: Timeliness

- Suppose you are overseeing the data of monthly sales
- For a period of time following each month: the data stored in the database are incomplete
  - Several sales representatives fail to submit their sales records on time at the end of the month
  - There are also a number of corrections and adjustments flowing in after the month’s end
- However, once all of the data are received, it is correct
- The fact that the month-end data are not updated in a timely fashion has a negative impact on the data quality

Data Quality: Believability & Intepretablity

- Suppose that a database, at one point, had several errors, all of which have since been corrected
- Believability: how trustable the data are correct?
  - E.g., the past errors had caused many problems for sales department users, and so they no longer trust the data
- Intepretability: how easily the data can be understood?
  - E.g., the data also use many accounting codes, which the sales department does not know how to interpret.

Data Quality: Why Preprocess the Data?

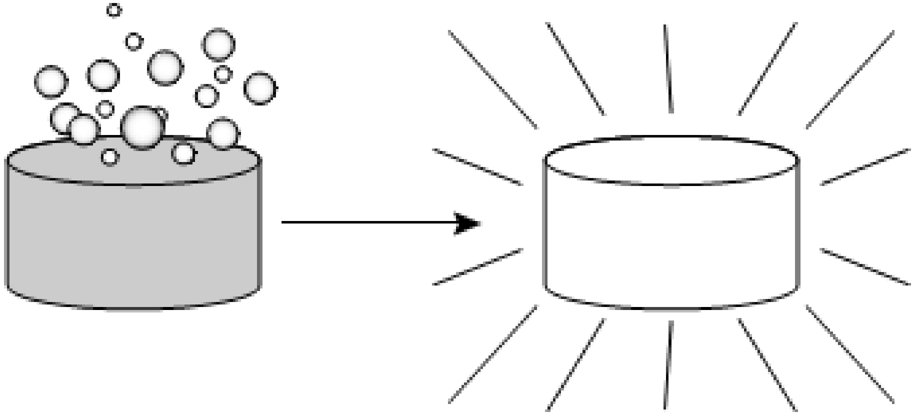
- The data quality depends on the intended use of the data
- For example,
  - In the database mentioned before, some of the addresses are outdated or incorrect, yet overall, 80% of the address are accurate
  - The marketing analyst who needs a list of customer addresses considers this to be a large customer database for target marketing purposes and he is pleased with the database’s accuracy
  - Meanwhile, the sales manager considers the database inaccurate and is not pleased

Major Tasks in Data Preprocessing

- Data cleaning
- Data integration
- Data reduction
- Data compression
- Data transformation and data discretization

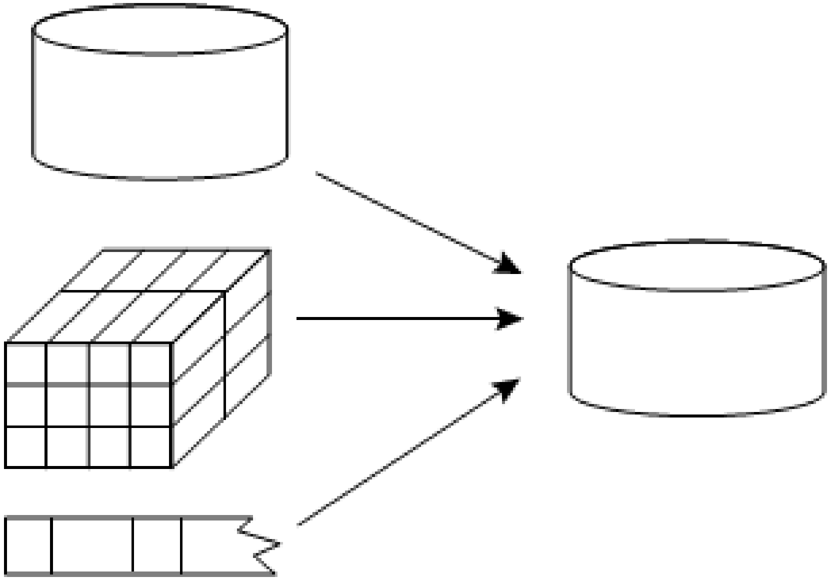
Major Tasks: Data Cleaning

- Objectives: *Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies*
- Users are unlikely to trust the results of any data mining that has been applied on the "dirty" data
- Most mining routines have procedures for dealing with incomplete or noisy data, yet they are not always robust



Major Tasks: Data Integration

- Objectives: *Integration of multiple databases, data cubes, or files*



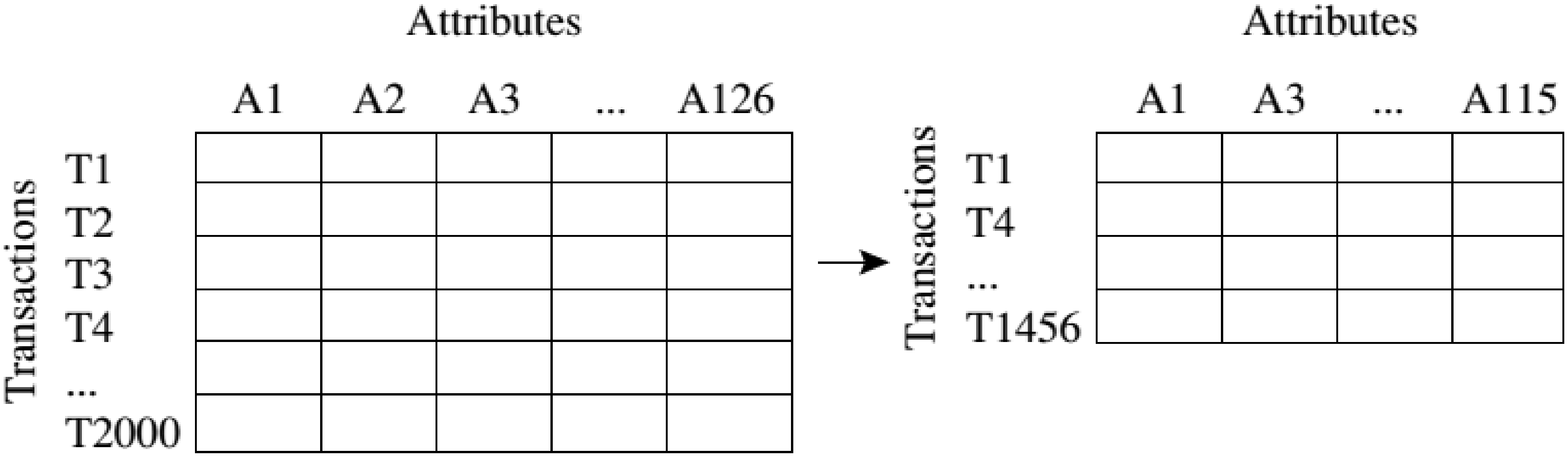
- Some attributes representing given concept may have different names in different databases
  - E.g., customer_id vs. cust_id, Bill vs. Willilam vs. B.
- Some attributes may be inferred from others
  - E.g., annual revenue

Major Tasks: Data Reduction

- Objectives: *Reduce the data set in volume so that the same (or almost the same) analytical results are still achieved*
- Dimensionality reduction:
  - Data encoding schemes are applied to obtain a compressed representation of the original data
  - Data compression (wavelet transforms, PCA), attribute subset selection, and attribute construction
- Numerosity reduction
  - The data are replaced by alternative, smaller representations
  - Parametric models (regression, log-linear models), nonparametric models (histograms, clusters, data aggregation, sampling)
- Data compression

Major Tasks: Data Reduction

- Objectives: *Reduce the data set in volume so that the same (or almost the same) analytical results are still achieved*



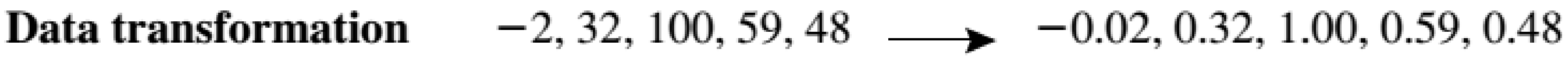
	A1	A2	A3	...	A126
T1					
T2					
T3					
T4					
...					
T2000					

→

	A1	A3	...	A115
T1				
T4				
...				
T1456				

Data Transformation and Data Discretization

- Objectives: *Allow data mining at multiple abstraction levels and the use of distance-based mining algorithms*
- Normalization:
  - Values are scaled to a smaller range such as [0.0, 1.0]
    - E.g., distance measurements taken on annual salary and age
- Discretization and concept hierarchy generation:
  - Raw data values are replaced by ranges or higher conceptual levels
    - E.g., raw values for age may be replaced by higher-level concepts, such as youth, adult, or senior



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

Outline

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

Data Cleaning

- Real-world data tend to be incomplete, noisy, and inconsistent
  - Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., Occupation = " " (missing data)
  - Noisy: containing noise, errors, or outliers
    - e.g., Salary = "−10" (an error)
  - Inconsistent: containing discrepancies in codes or names,
    - e.g., Age = "42", Birthday = "03/07/2010", was rating "1, 2, 3", now rating "A, B, C", discrepancy between duplicate records
  - Intentional (e.g., disguised missing data)
    - E.g., 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 smooth each bin by its mean, median, or 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

- An iteration of data discrepancy detection and data transformation (to correct discrepancies)
- Data discrepancy can be caused by several factors
  - Data entry forms having too many optional fields, human error in data entry, deliberate errors (e.g., respondents not wanting to divulge information about themselves), and data decay (e.g., outdated addresses)
  - Inconsistent data representations and inconsistent use of codes
  - Errors in data recording devices and system errors
  - Data used for purposes other than originally intended
  - Inconsistencies due to data integration (e.g., an attribute can have different names in different databases)

Data Cleaning as a Process

- Data discrepancy detection
  - Use metadata (e.g., domain, range, dependency, distribution)
  - Check inconsistent use of codes and any inconsistent data representations
  - Check field overloading
    - Result when squeezing new attribute definitions into unused (bit) portions of already defined attributes
  - 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 Cleaning as a Process

- 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)

Outline

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

Data Integration

- Combines data from multiple sources into a coherent store
- Entity identification problem:
  - Identify real world entities from multiple data sources
    - E.g., Bill Clinton = William Clinton
  - Special attention must be paid to the structure of the data when matching attributes from one database to another
    - E.g., in one system, a discount may be applied to the order, whereas in another system it is applied to each individual line item within the order

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
- Correlation analysis: χ^2(chi-square) test (for nominal data) and correlation coefficient – covariance (for numeric data)
- Careful integration may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Correlation Analysis: Nominal Data

- Suppose attribute A has c distinct values, namely $a_1, a_2, \ldots,$ a_c, attribute B has r distinct values, namely $b_1, b_2, \ldots,$ b_r
- Let (A_i, B_j) denote the joint event that $A = a_i$ và $B = b_i$
- χ^2(chi-square) statistic tests the hypothesis that A and B are independent

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

  - o_{ij}: observed frequency (i.e. actual count) of (A_i, B_j)
  - e_{ij}: expected frequency of (A_i, B_j) $e_{ij} = {}^{count(A=a_i)\times count(B=b_j)}/_{n}$
    - n: nuber of data tuples

Correlation Analysis: Nominal Data

- χ^2(chi-square) statistic tests the hypothesis that A and B are independent

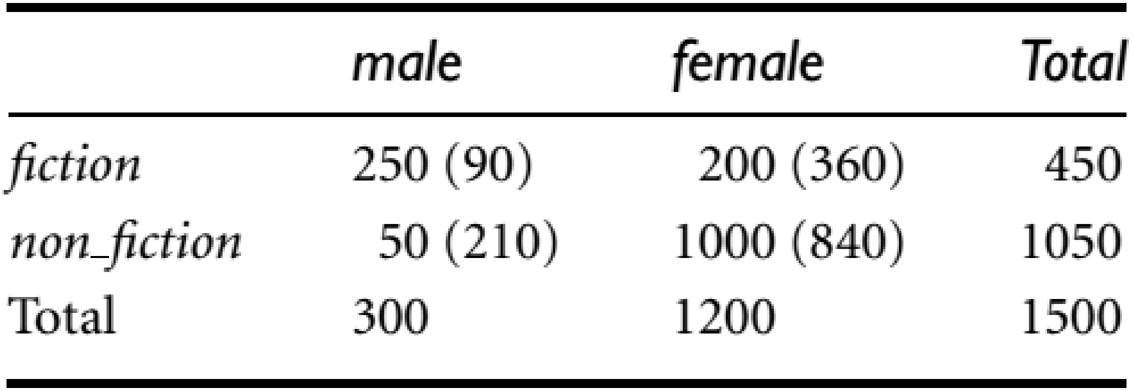
$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{\left(o_{ij} - e_{ij}\right)^2}{e_{ij}}$$

  - The larger the χ^2 value, the more likely the variables are related

- 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

- Contingency table



	male	*female*	*Total*
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

(Numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

- Are *gender* and *preferred_reading* correlated?

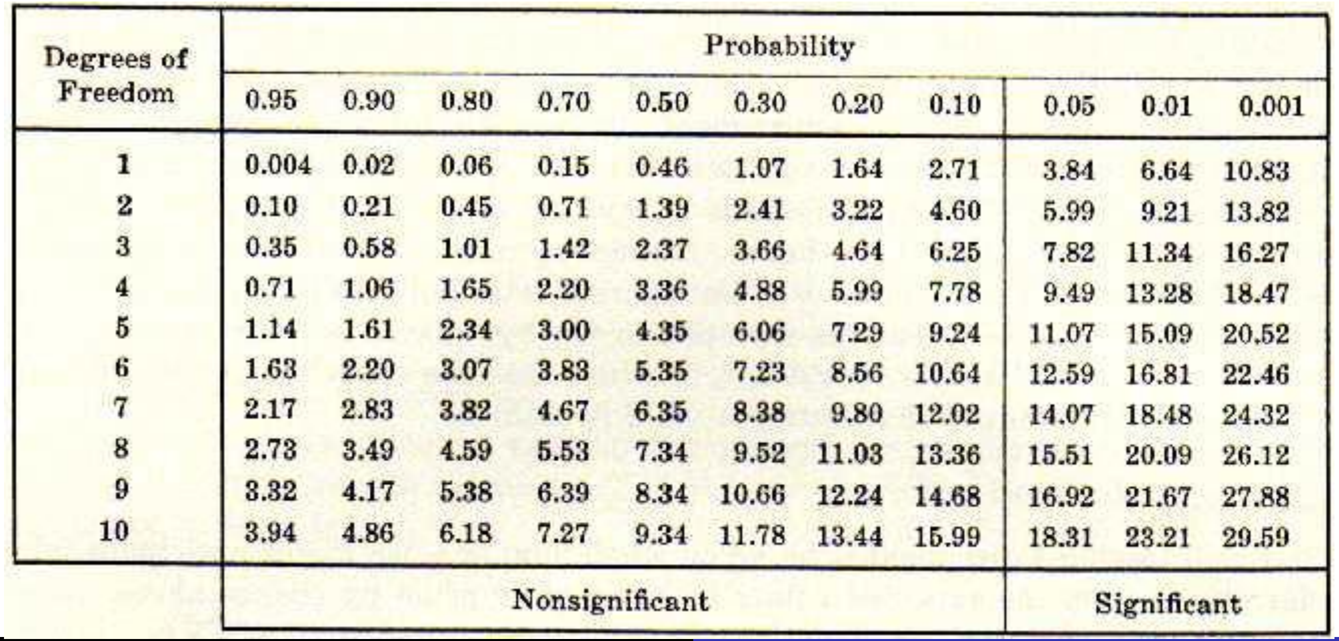
$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that two attributes are (strongly) correlated for the given group of people

Chi-Square Calculation: An Example

- The test is based on a significance level, with $(r-1)\times(c-1)$ degrees of freedom (DOF).
- If the hypothesis can be rejected, then we say that A and B are statistically correlated.

- In the previous example, DOF = 1, the χ^2 value needed to reject the hypothesis at the 0.001 significance level is 10.828



Degrees of Freedom	Probability										
	0.95	0.90	0.80	0.70	0.50	0.30	0.20	0.10	0.05	0.01	0.001
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.64	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.60	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.82	11.34	16.27
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88
10	3.94	4.86	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59
	Nonsignificant								Significant		

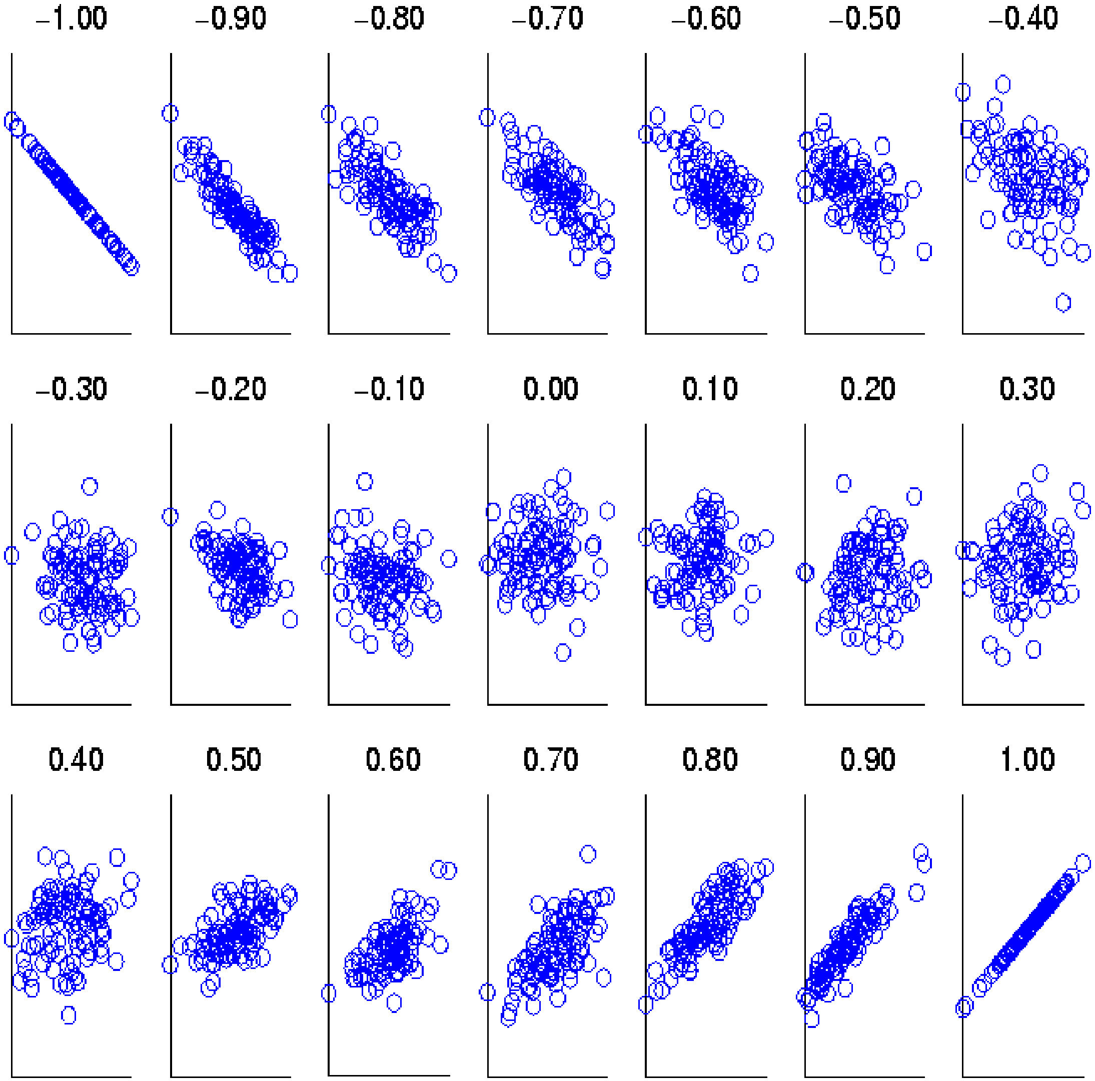
Correlation Analysis (Numeric Data)

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

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

  - n: the number of tuples,
  - $\bar{A}, \bar{B}, \sigma_A, \sigma_B$: means and standard deviations of A and B, respectively
  - $\Sigma a_i b_i$: sum of the AB cross-product
  - $r_{A,B} > 0$: A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation
  - $r_{A,B} = 0$: A and B are independent
  - $r_{A,B} < 0$: A and B are negatively correlated

Visually Evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

Covariance Analysis (Numeric Data)

- Consider two numeric attributes A and B, and a set of n observations $\{(a_1, b_1), .., (a_n, b_n)\}$
- Expected values on A and B:

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

- Covariance between A and B:

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

- Covariance vs. correlation: $r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$

Covariance Analysis (Numeric Data)

- Covariance between A and B:

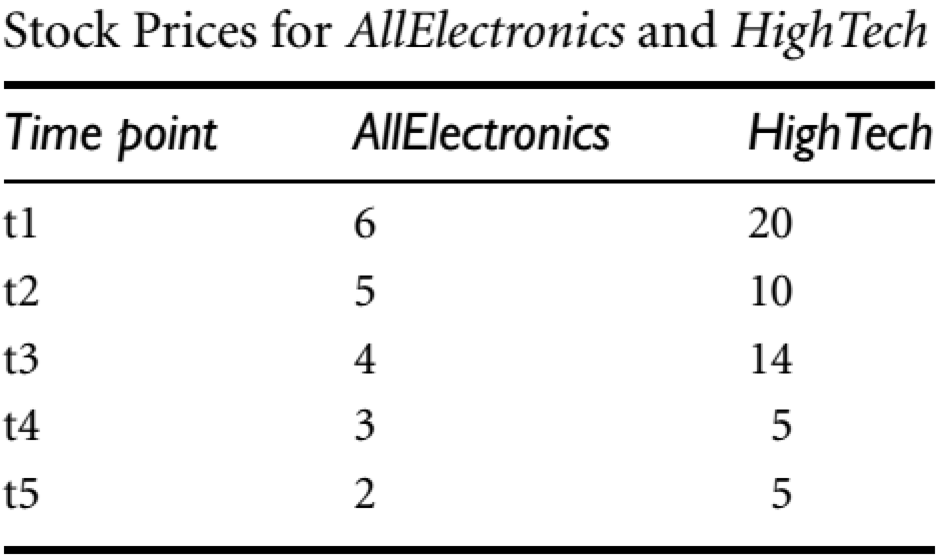
$$Cov(A,B) = E\left(\left(A-\bar{A}\right)\left(B-\bar{B}\right)\right) = \frac{\sum_{i=1}^{n}\left(a_i-\bar{A}\right)\left(b_i-\bar{B}\right)}{n}$$

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

  - $Cov(A,B) > 0$: Positive covariance, A and B both tend to be larger than their expected values.
  - $Cov(A,B) < 0$: Negative covariance, if A is larger than its expected value, B is likely to be smaller than its expected value.
  - $Cov(A,B) = 0$: Independence, but the converse is not true
    - Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

Covariance: An Example

- If the stocks are affected by the same industry trends, will their prices rise or fall together?



Stock Prices for *AllElectronics* and *HighTech*

Time point	*AllElectronics*	*HighTech*
t1	6	20
t2	5	10
t3	4	14
t4	3	5
t5	2	5

- $E(AllElectronics) = \frac{6+5+4+3+2}{5} = \frac{20}{5} = \$4$
- $E(HighTech) = \frac{20+10+14+5+5}{5} = \frac{54}{5} = \$10.80$
- $Cov(AllElectronics, HighTech) = \frac{6\times20+5\times10+4\times14+3\times5+2\times5}{5} - 4 \times 10.80 = 7$
- Therefore, given the positive covariance, stock prices for both companies rise together

Tuple Duplication

- Duplication should also be detected at the tuple level
  - E.g., two or more identical tuples for a given unique data entry case
- The use of denormalized tables (often done to improve performance by avoiding joins operation)
  - E.g., when a purchase order database contains attributes for the purchaser's name and address instead of a key to this information in a purchaser database, the same purchaser's name may with different addresses within the purchase order database

Data Value Conflict Detection and Resolution

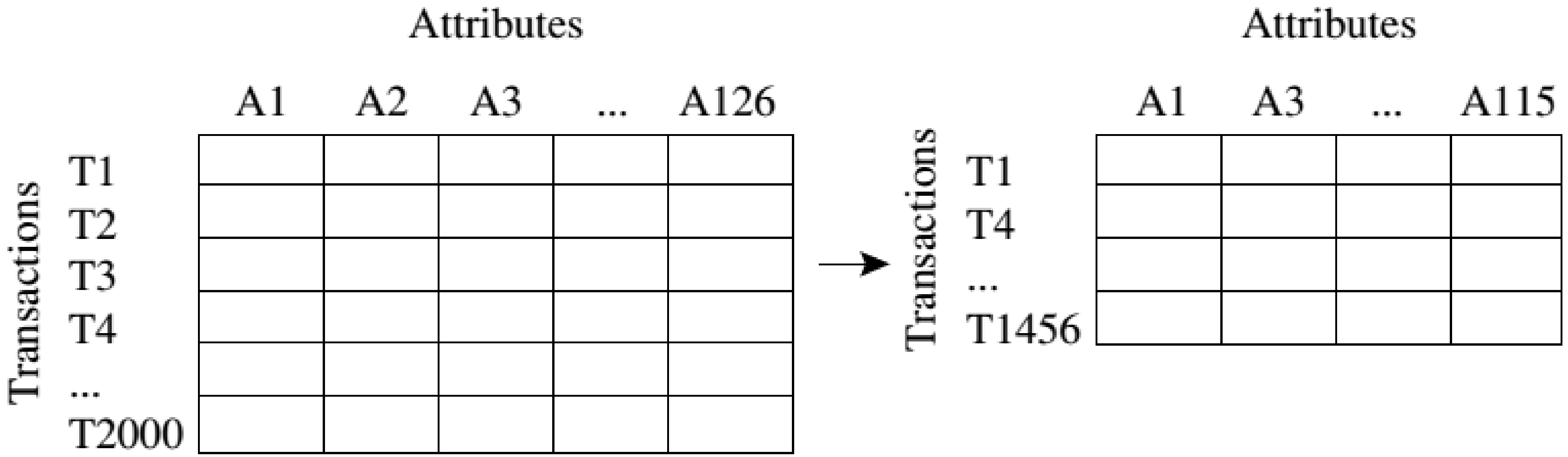
- For the same real world entity, attribute values from different sources are different
  - Due to differences in representation, scaling, or encoding
  - E.g., metric units in British system and other systems, currencies, grading scheme between schools
- Attributes may also differ on the abstraction level
  - An attribute in one system is recorded at, say, a lower abstraction level than the “same” attribute in another.
  - E.g., the total sales may refer to one branch of All Electronics or the total sales for AllElectronics stores in a given region

Outline

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

Data Reduction

- Objective: 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

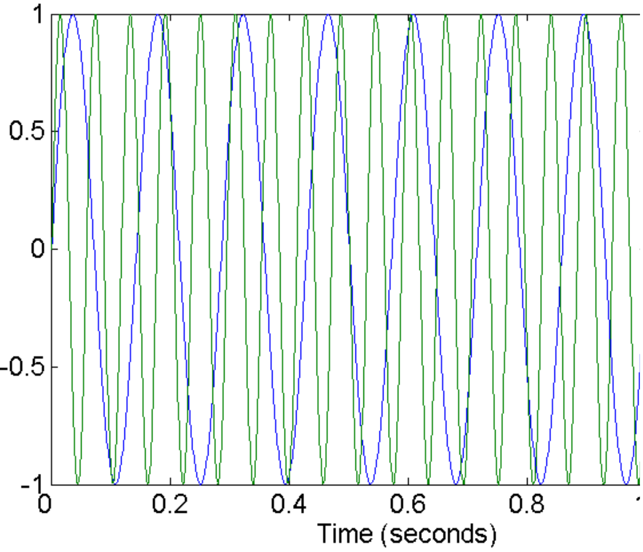
Strategy 1: Dimensionality Reduction

- 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

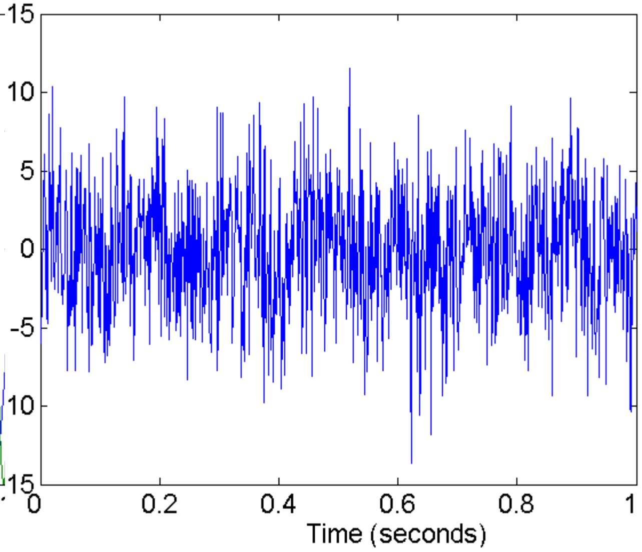
- Dimensionality reduction
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization

Mapping Data to a New Space

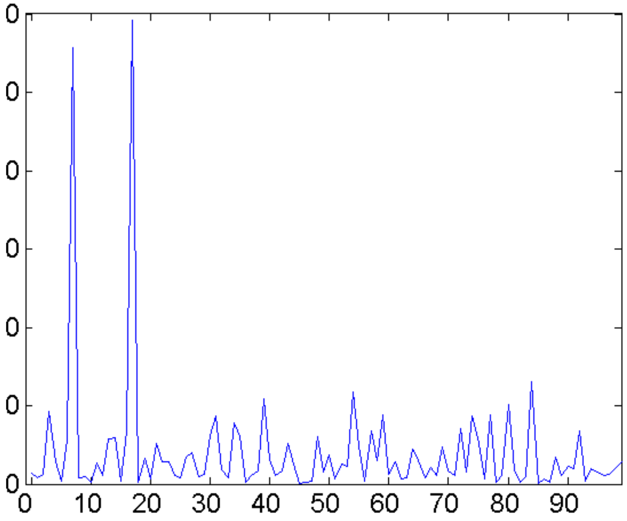
- Fourier transform
- Wavelet transform



Two Sine Waves



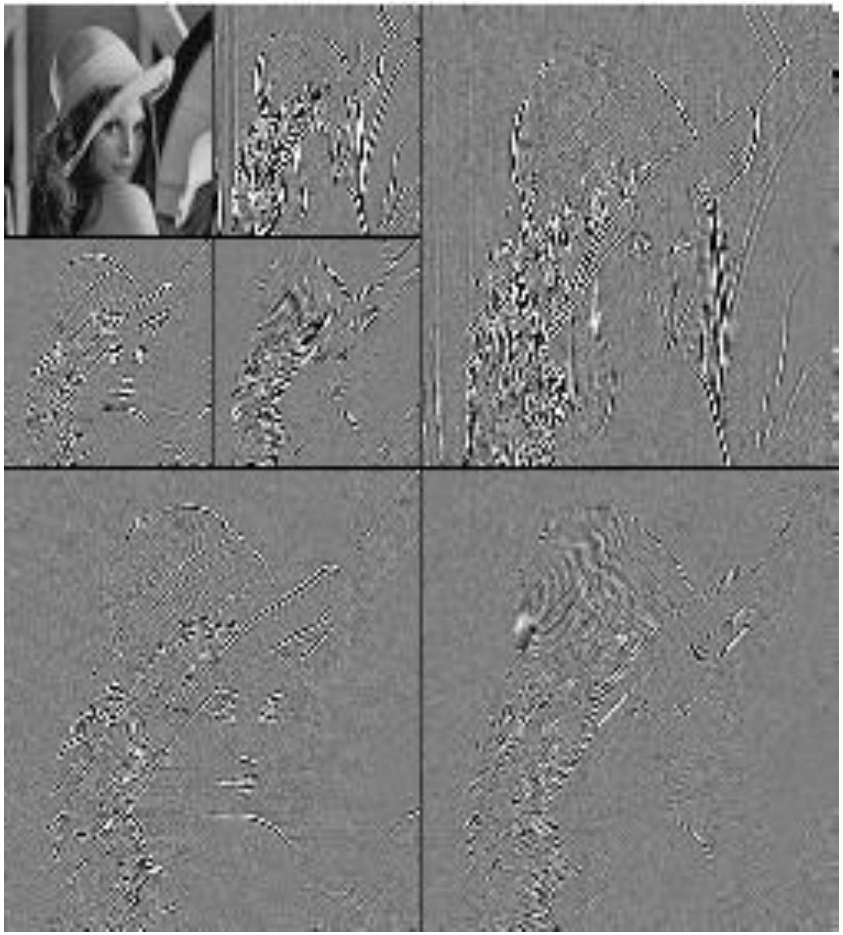
Two Sine Waves + Noise



Frequency

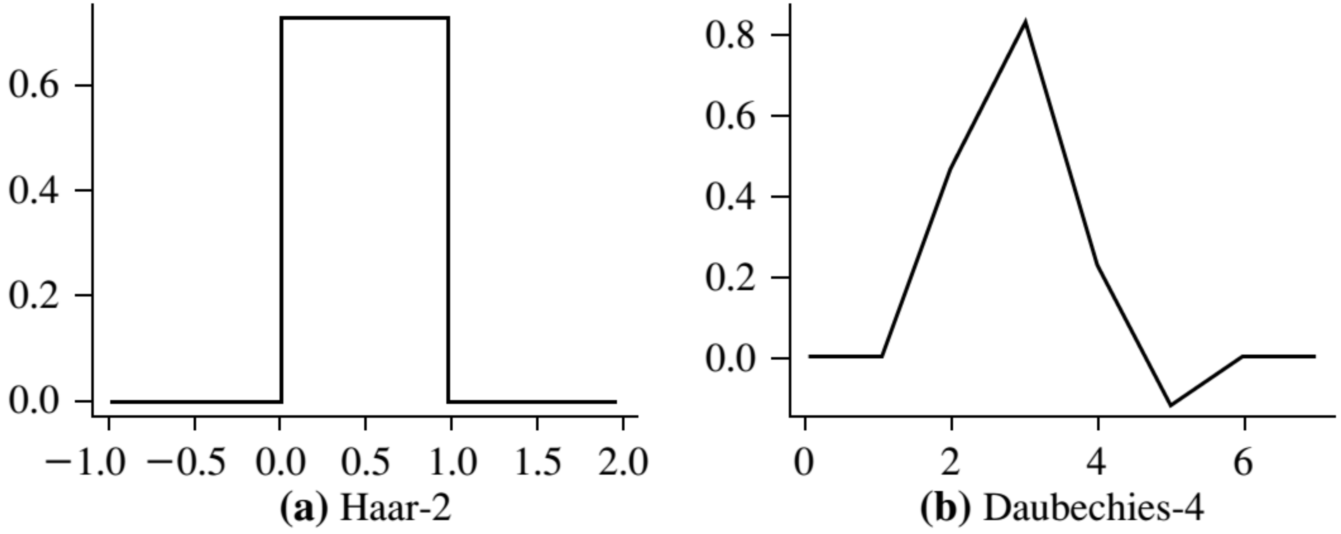
Wavelet Transform

- Decomposes a (n-dimensional) signal into different frequency subbands
- 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 Transform

- 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



Wavelet Transform Method

Consider each tuple as an n-dimensional data vector, $X = (x_1, x_2, \dots, x_n)$

1. The length, L, of the input data vector must be an integer power of 2, padding zeros as necessary ($L \geq n$)
2. For pairs of data points in X, i.e. (x_{2i}, x_{2i+1}), apply data smoothing (e.g., such as a sum or weighted average) and a weighted difference, which acts to bring out the detailed features of the data
3. The two functions are recursively applied to the data sets obtained in the previous loop, until the resulting data sets obtained are of length 2.
4. Selected values from the data sets obtained in the previous iterations are designated the wavelet coefficients of the transformed data

Wavelet Decomposition

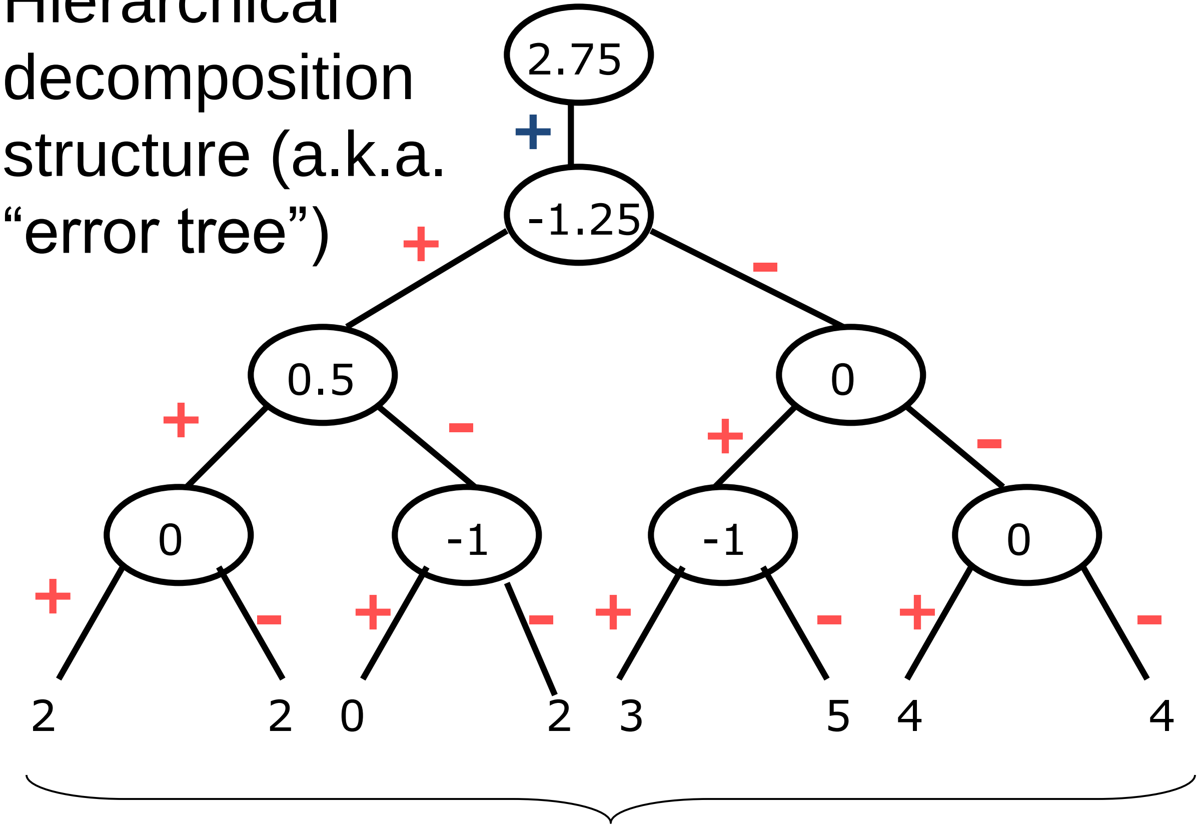
- Wavelets: A math tool for space-efficient hierarchical decomposition of functions
- $S = [2, 2, 0, 2, 3, 5, 4, 4]$ can be transformed to $S\hat{} = [23/4, -11/4, 1/2, 0, 0, -1, -1, 0]$

Resolution	Averages	Detail Coefficients
8	[2, 2, 0, 2, 3, 5, 4, 4]	
4	[2, 1, 4, 4]	[0, -1, -1, 0]
2	[1.5, 4]	[0.5, 0]
1	[2.75]	[-1.25]

- Compression: small detail coefficients can be replaced by 0's, and only significant coefficients are retained

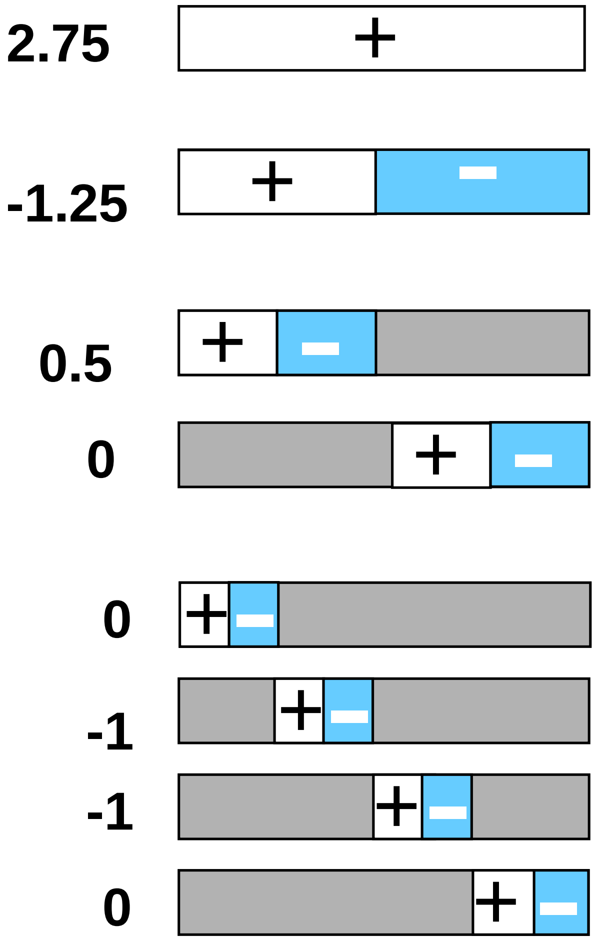
Haar Wavelet Coefficients

Hierarchical decomposition structure (a.k.a. “error tree”)



Original frequency distribution

Coefficient “Supports”

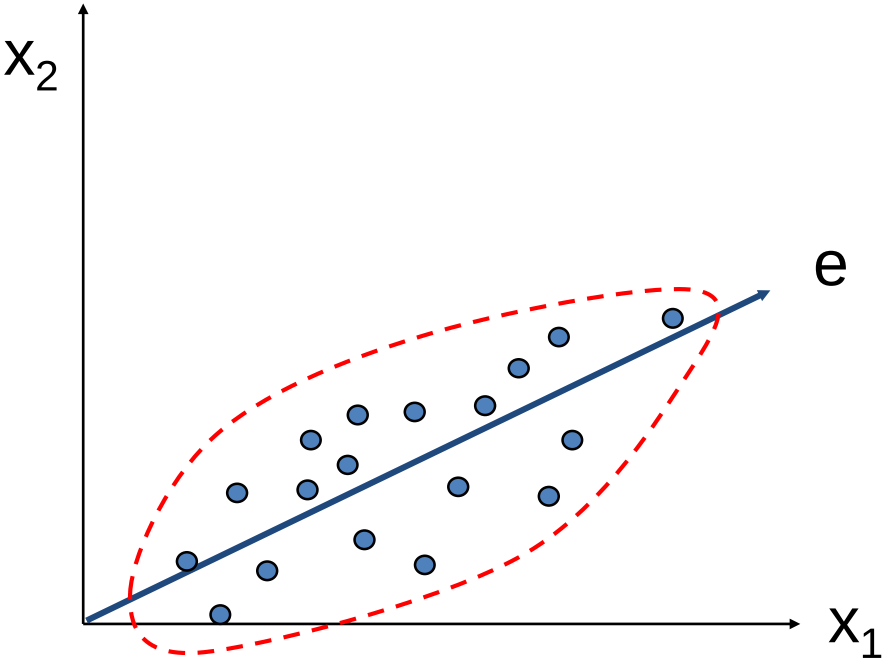


Why Wavelet Transform?

- 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)

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



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
  - Normalize input data: each attribute falls within the same range
  - Compute k orthonormal (unit) vectors, i.e., principal components
  - Each input vector is a linear combination of k principal component vectors
  - Principal components are sorted in order of decreasing “significance” or strength
  - Eliminate weak components, i.e., those with low variance, to reduce the size of the data and reconstruct a good approximation of the original data with the strongest principal components
- Works for numeric data only

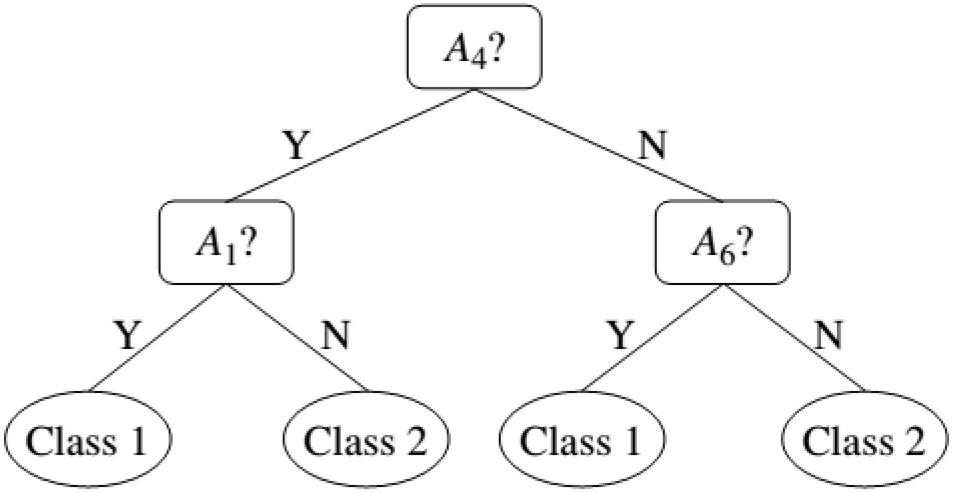
Attribute Subset Selection

- 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 Selection: Greedy Methods

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ => $\{A_1\}$ => $\{A_1, A_4\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  A_4? Y N A_1? A_6? Y N Y N Class 1 Class 2 Class 1 Class 2 => Reduced attribute set: $\{A_1, A_4, A_6\}$

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

Strategy 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)
  - E.g., log-linear models obtain value at a point in m-D space as the product on appropriate marginal subspaces
- Non-parametric methods
  - Do not assume models
  - Major families: histograms, clustering, sampling

Regression and Log-Linear Models

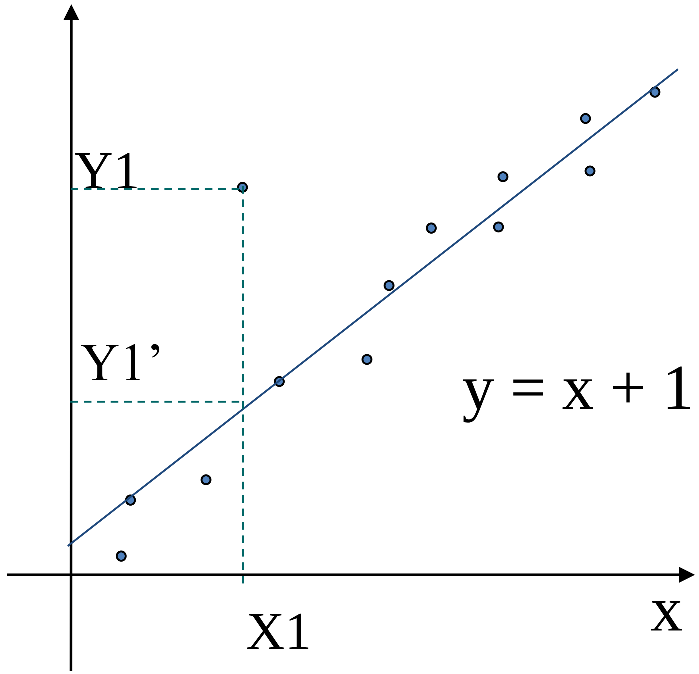
- Linear regression
  - Data modeled to fit a straight line
  - Often uses the least-square method to fit the line
- Multiple regression
  - Allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- Log-linear model
  - Approximates discrete multidimensional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

Regression Analysis

- 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

Regression Analysis

- 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: $Y = wX + b$
  - 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: $Y = b_0 + b_1 X_1 + b_2 X_2$
  - 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

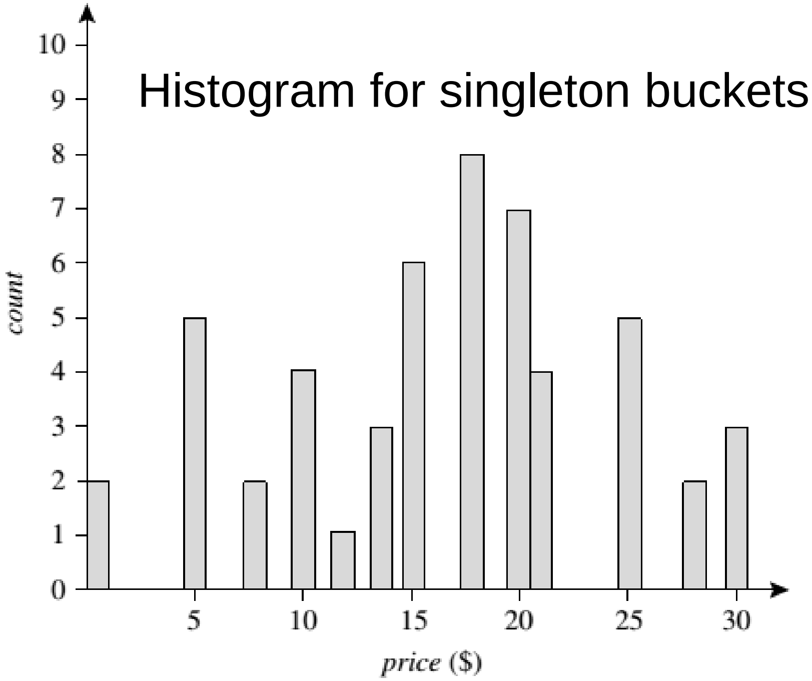
Histogram Analysis

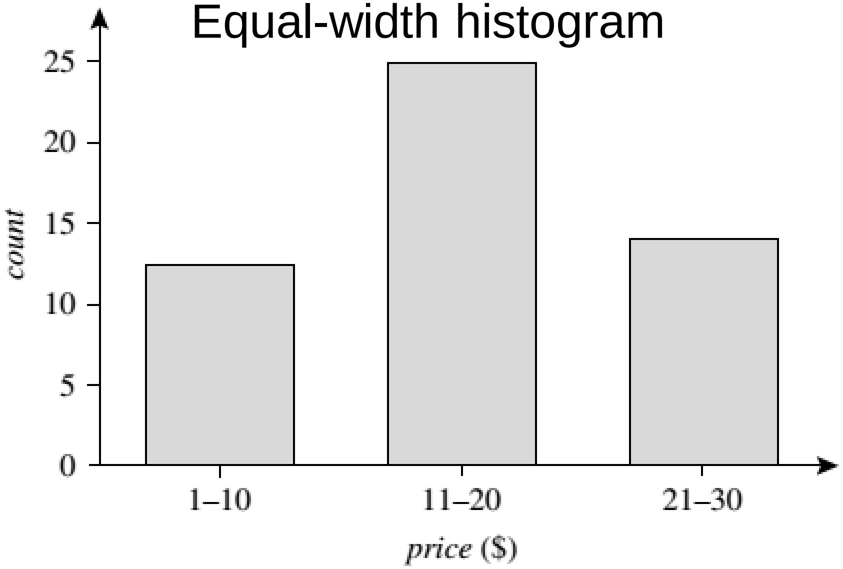
- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
  - Equal-width: equal bucket range
  - Equal-frequency (or equal-depth)

Histogram Analysis: An Example

- The following data are a list of AllElectronics prices for commonly sold items (rounded to the nearest dollar)

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, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30


Histogram for singleton buckets


Equal-width histogram

Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms

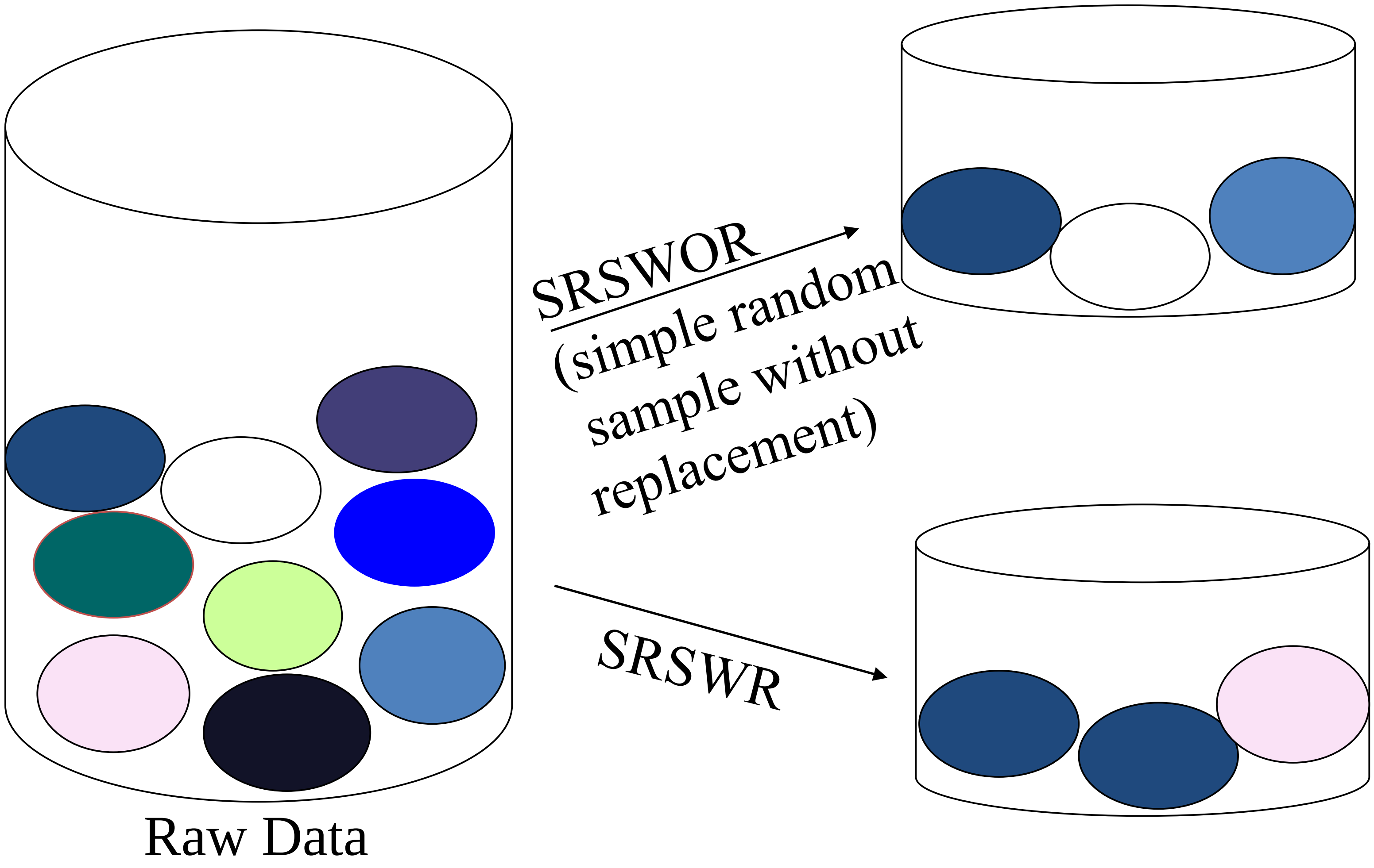
Sampling

- Obtain 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

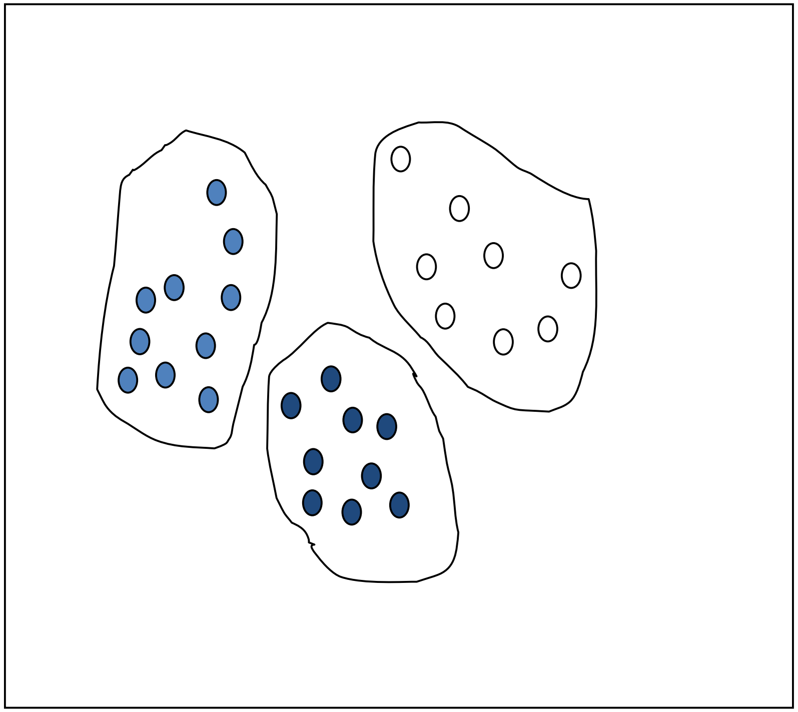
- Simple random sampling
  - There is an equal probability of selecting any particular item
- Sampling without replacement
  - Once an object is selected, it is removed from the population
- Sampling with replacement
  - A selected object is not removed from the population
- Stratified sampling:
  -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

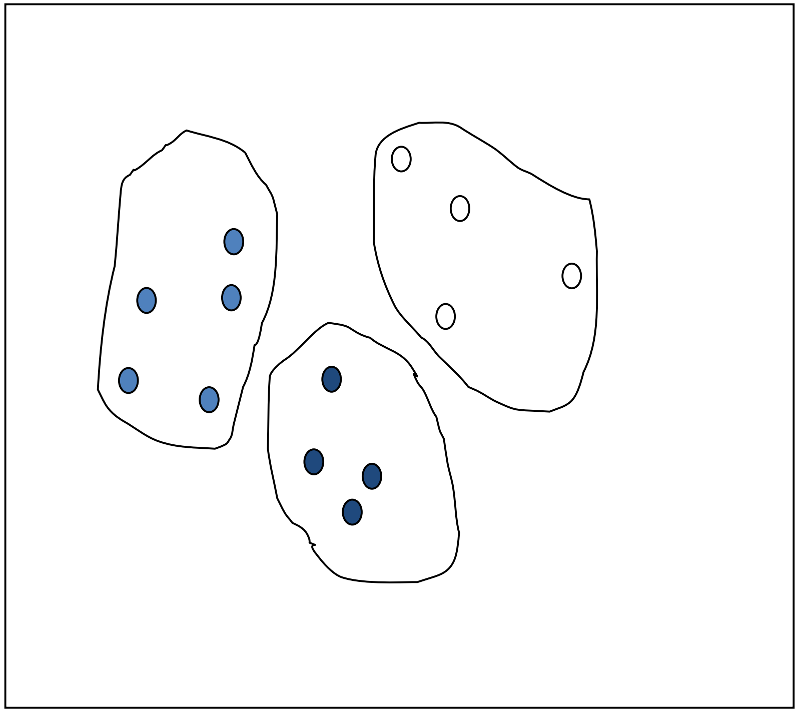


Sampling: Cluster or Stratified Sampling

Raw Data

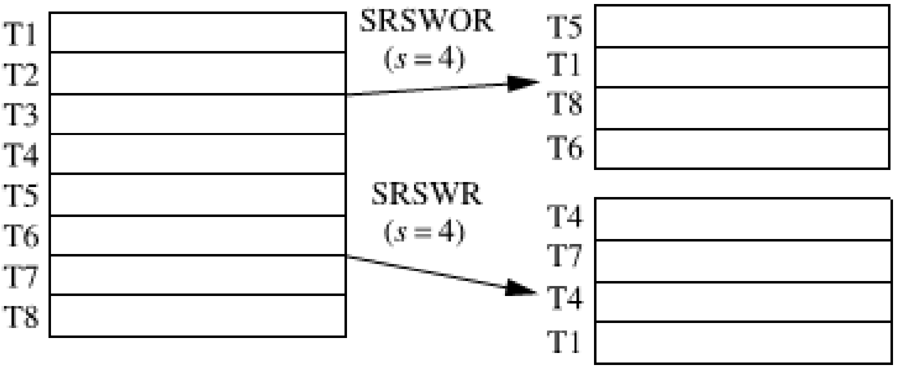


Cluster/Stratified Sample



69

Types of Sampling: An Example

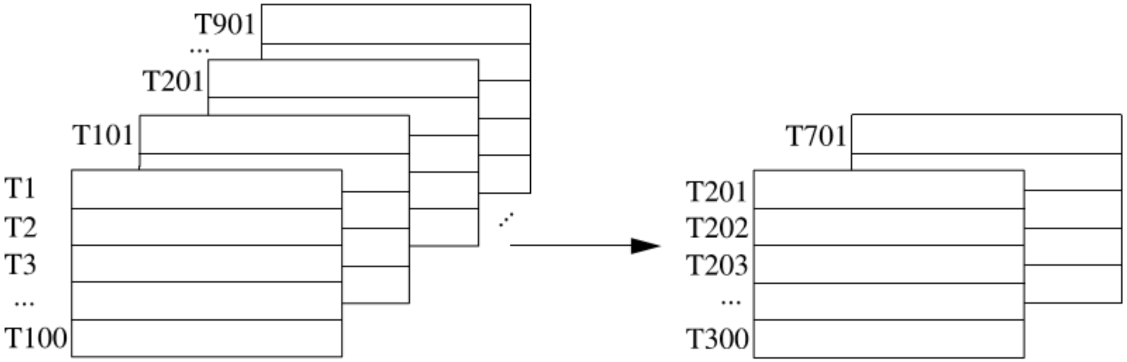


Startified sample
(according to *age*)

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

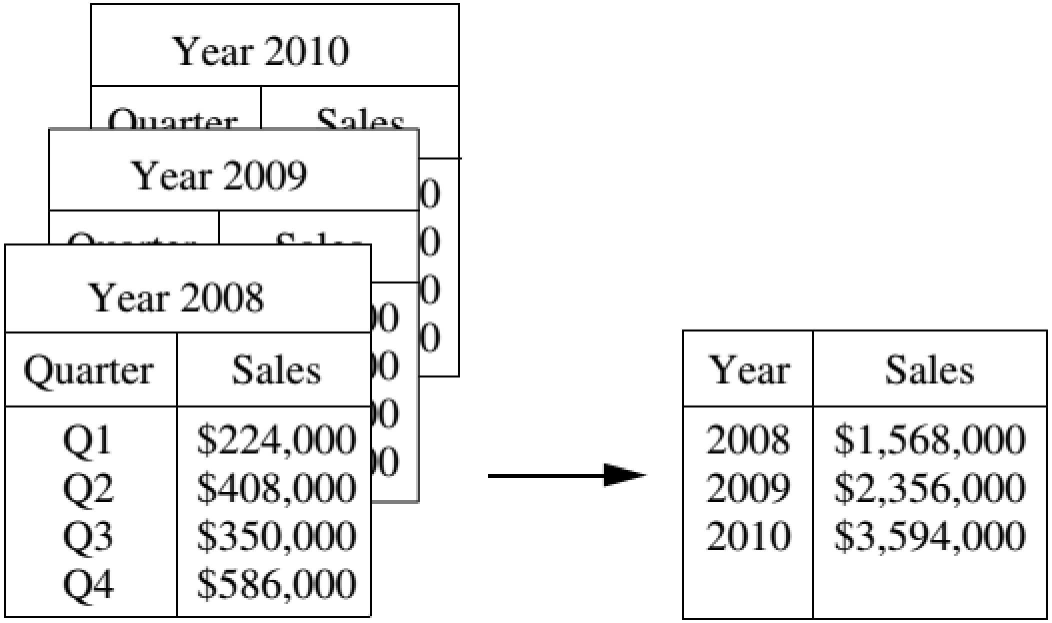
T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

Cluster sample
(s = 2)



Data Cube Aggregation

- An example of data aggregation:
  - Sales data for a given branch of AllElectronics for the years 2008 through 2010 are aggregated to provide the annual sales



Year 2010

Year 2009

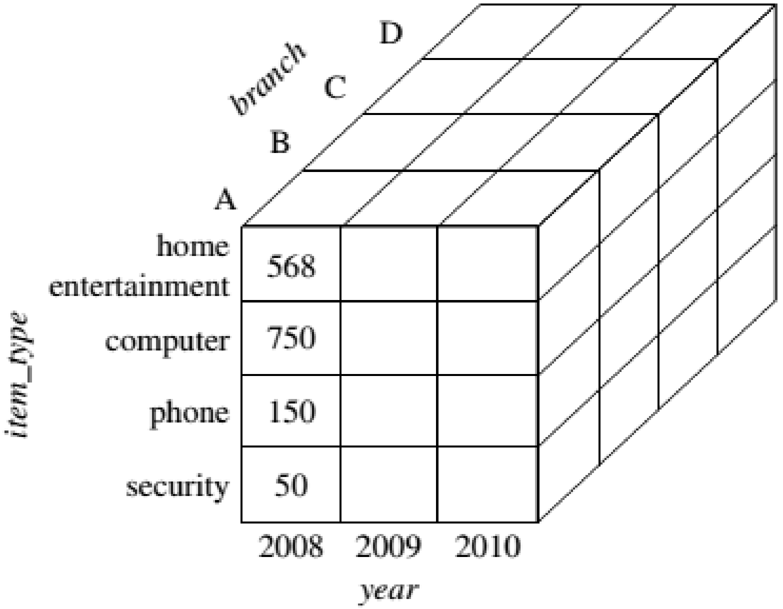
Year 2008

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

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

Data Cube Aggregation

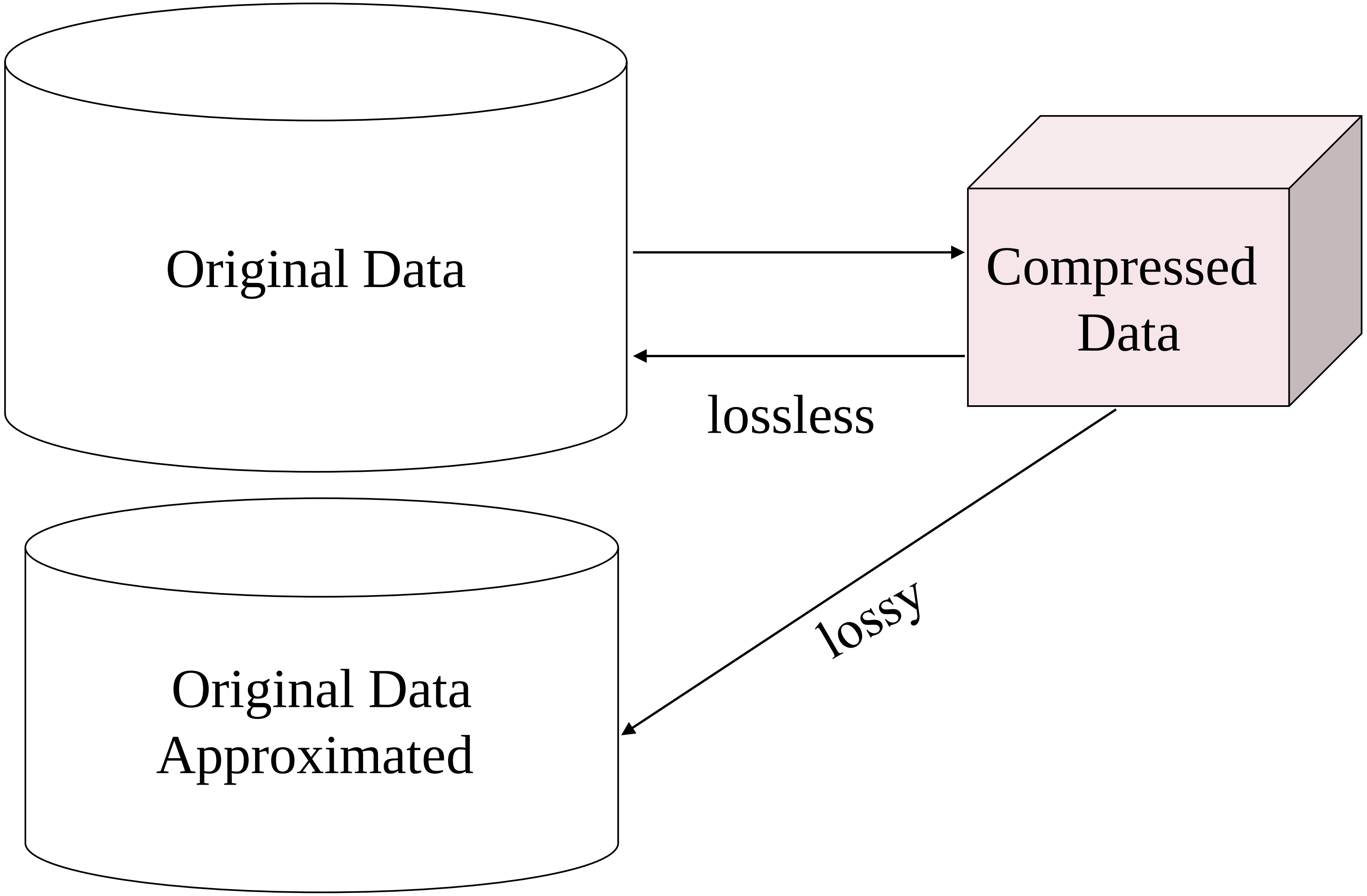
- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the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



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



Original Data

Compressed Data

lossless

lossy

Original Data Approximated

Outline

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

Data Transformation

- Map the entire set of values of a given attribute to a new set of replacement values so that 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

- Let A be a numeric attribute with n observed values, $v_1, v_2, \dots, v_n$
- Min-max normalization: to [new_min$_A$, new_max$_A$]

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

  - E.g., let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then \$73,000 is mapped to $\frac{73,600-12,000}{98,000-12,000}(1.0 - 0) + 0 = 0.716$

Normalization

- Z-score normalization:

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A}$$

  - where $\bar{A}$ and σ_A are the mean and standard deviation, respectively, of attribute A
  - E.g., suppose that the mean and standard deviation of the values for the attribute income are \$54,000 and \$16,000, respectively. Then \$73,000 is mapped to $\frac{73,600-54,000}{16,000} = 1.225$
  - Another variation: replace σ_A by $s_A = \frac{1}{n}\left(\left|v_1 - \bar{A}\right| + \left|v_2 - \bar{A}\right| + \cdots +\right.$

Normalization

- Decimal scaling:

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

  - where j is the smallest integer such that $\max(|v_i'|) < 1$
  - Normalize by moving the decimal point of values of A, the number of decimal points moved depends on the maximum absolute value of A
  - E.g., suppose that the recorded values of A range from −986 to 917. The maximum absolute value of A is 986, therefore divide each value by 1000 (i.e., j = 3). −986 normalizes to −0.986 and 917 normalizes to 0.917

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

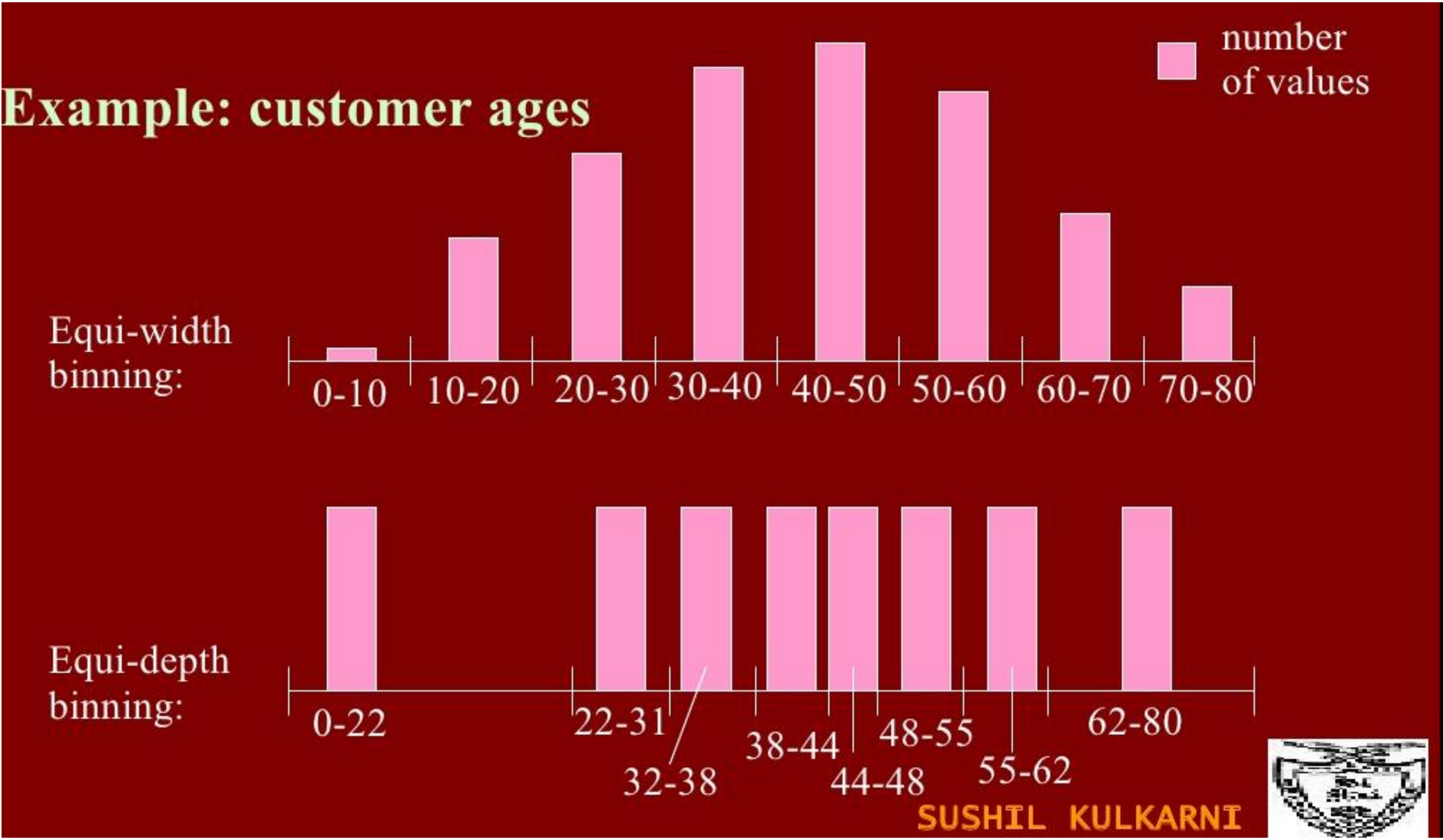
Data Discretization Methods

- Typical methods: can be applied recursively
    - Binning
        - Top-down split, unsupervised
    - Histogram analysis
        - Top-down split, unsupervised
    - Clustering analysis
        - Top-down split or bottom-up merge, unsupervised
    - Decision-tree analysis
        - Top-down split, supervised
    - Correlation (e.g., χ^2) analysis
        - Bottom-up merge, unsupervised

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

Equal-width vs. Equal-depth



Binning Method: An Example

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

Partition into equal-frequency bins:
Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

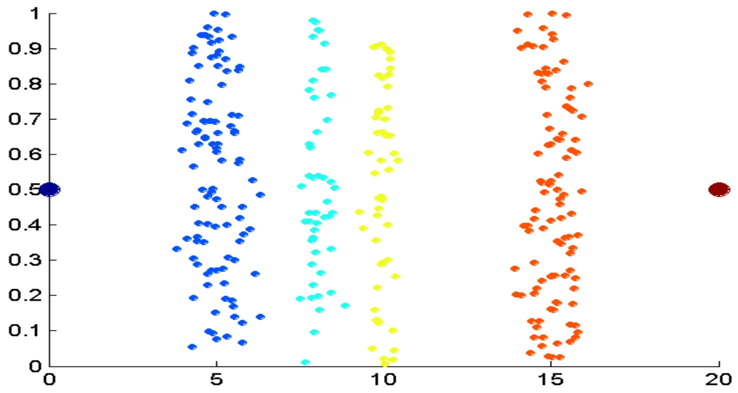
Smoothing by bin means:
Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin medians:
Bin 1: 8, 8, 8
Bin 2: 21, 21, 21
Bin 3: 28, 28, 28

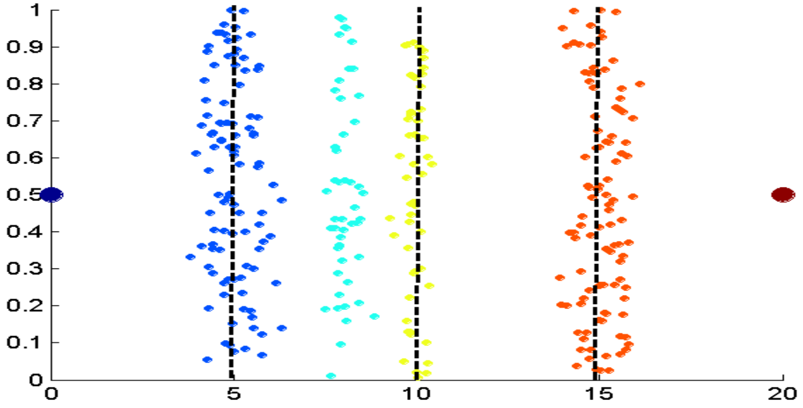
Smoothing by bin boundaries:
Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Discretization Without Using Class Labels

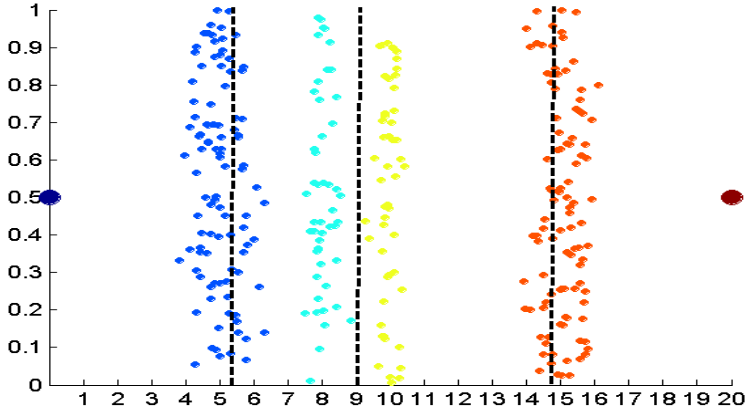
- Binning vs. Clustering



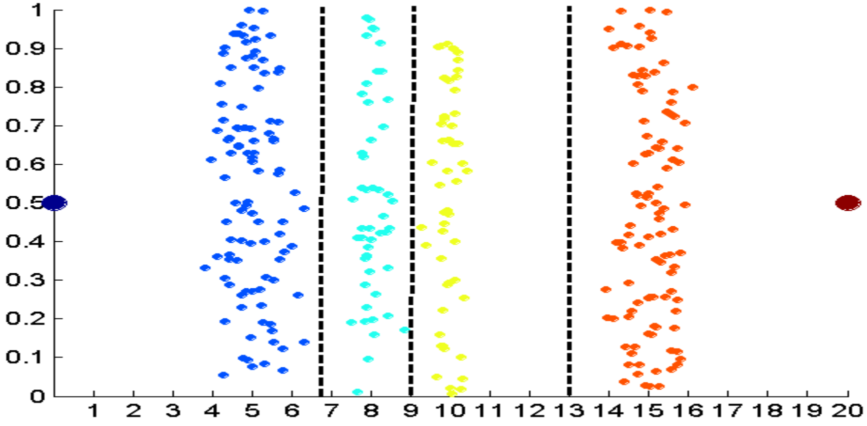
Data



Equal-width binning



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

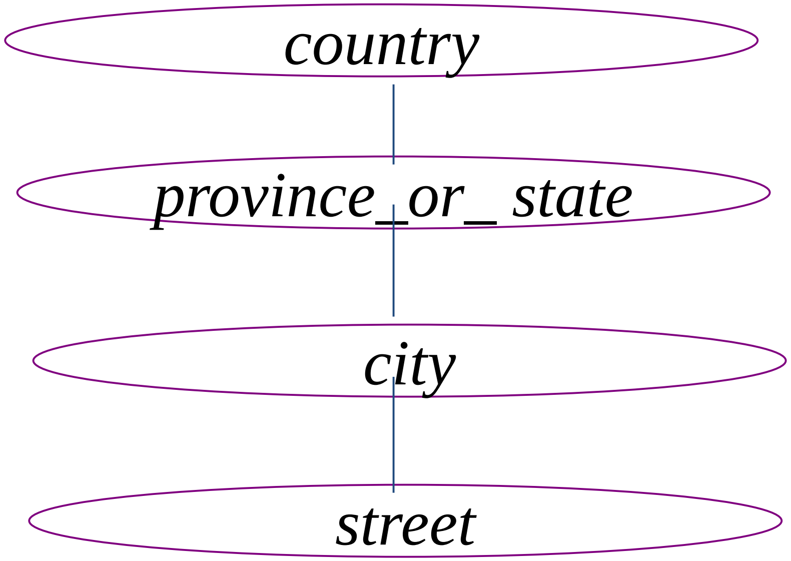
- Organize concepts (i.e., attribute values) hierarchically
- Facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: recursively reduce the data by collecting and replacing low level concepts by higher level concepts
    - E.g., numeric values for age to {youth, adult, or senior}
- Automatically formed for both numeric and nominal data. For numeric data, use discretization methods shown.

Concept Hierarchy Generation

- 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

Outline

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

Summary

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

Recommended Reference Books

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