

# **Data Science** **(CS3206)**

**Data Preprocessing**

# In today's discussion...

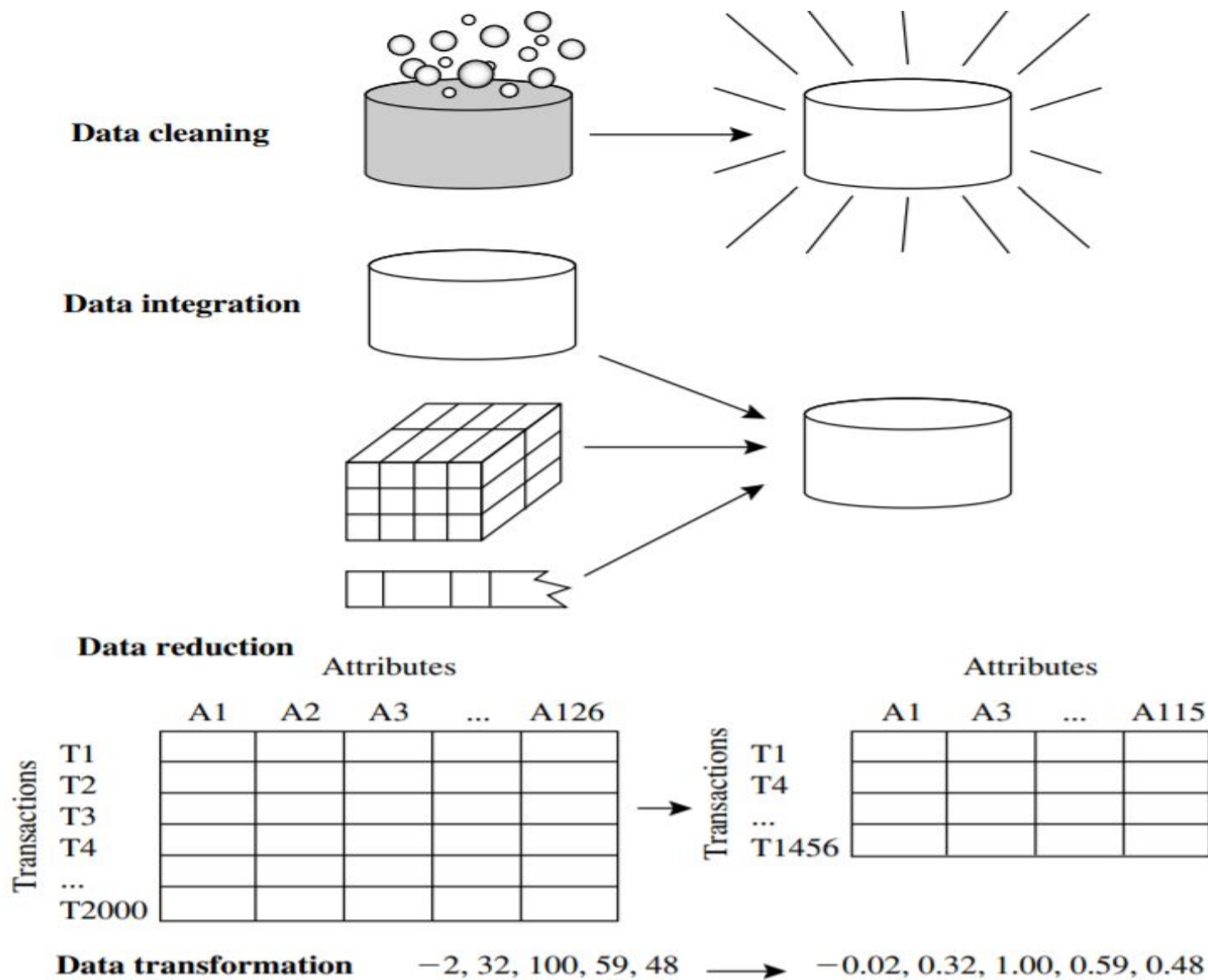
- Data Preprocessing: An Overview
- Data Cleaning
- Data Integration
- Data Reduction
- Data Transformation and Data Discretization

# Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
  - Accuracy: containing errors, or values that deviate from values expected ( 1<sup>st</sup> June DOB)
  - Completeness: not recorded, unavailable, aggregate data
  - Consistency: some modified but some not, dangling
  - Timeliness: timely update? weakly, monthly
  - Believability: how trustable the data are correct?
  - Interpretability: how easily the data can be understood?

# Major Tasks in Data Preprocessing

- **Data cleaning**
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
  - Integration of multiple databases, data cubes, or files  
(customer\_id/cust\_id)
- **Data reduction**
  - Dimensionality reduction(wavelet transforms, PCA, attribute subset selection)
  - Numerosity reduction(regression, histogram, cluster, sampling, aggregation)
  - Data Compression
- **Data transformation and data discretization**
  - Normalization (scaled to smaller range)
  - Concept hierarchy generation( age values replaced by youth, adult or senior)



Forms of data preprocessing.

# Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., *Occupation*=" " (missing data)
  - noisy: containing noise, errors, or outliers
    - e.g., *Salary*="−10" (an error)
  - inconsistent: containing discrepancies in codes or names, e.g.,
    - *Age*="42", *Birthday*="03/07/2010"
    - Was rating "1, 2, 3", now rating "A, B, C"
    - discrepancy between duplicate records
  - Intentional (e.g., *disguised missing* data)
    - Jan. 1 as everyone's birthday?

# Data Cleaning

- Fill in missing values
- Smooth out noise identifying outliers
- Correct inconsistencies in data

# 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 large dataset?
- Fill in it automatically with
  - a global constant : e.g., “unknown”- not foolproof
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
  - the most probable value: regression, inference-based tools

# 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.

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

**Partition into (equal-frequency) bins:**

Bin 1: 4, 8, 15

Bin 2: 21, 21, 24

Bin 3: 25, 28, 34

**Smoothing by bin means:**

Bin 1: 9, 9, 9

Bin 2: 22, 22, 22

Bin 3: 29, 29, 29

**Smoothing by bin boundaries:**

Bin 1: 4, 4, 15

Bin 2: 21, 21, 24

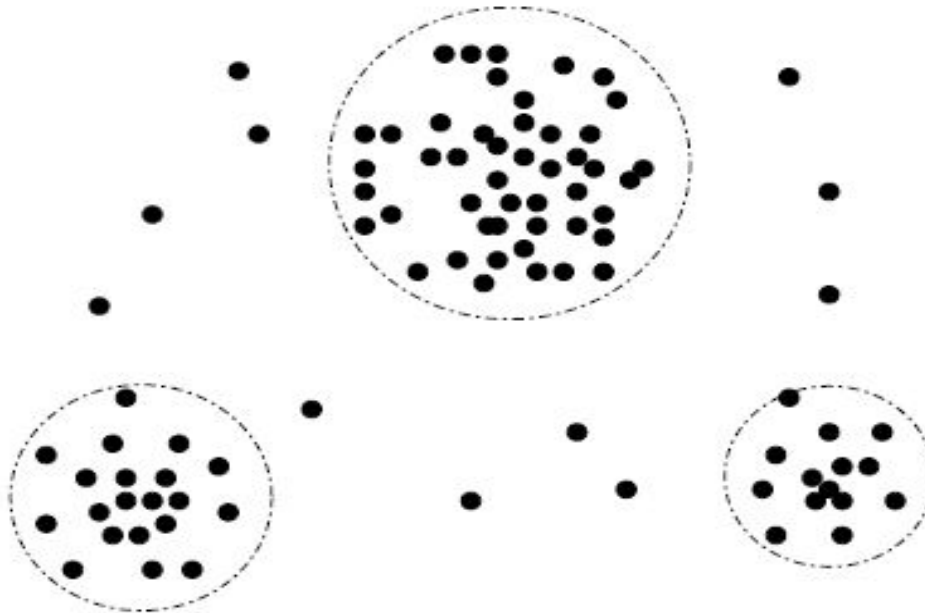
Bin 3: 25, 25, 34

# How to Handle Noisy Data?

- Regression
  - smooth by fitting the data into regression functions
  - Linear regression involves finding the “best” line to fit two attributes (or variables) so that one attribute can be used to predict the other
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

# How to Handle Noisy Data?

- Clustering
  - detect and remove outliers



---

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 Cleaning as a Process

- Data discrepancy detection
  - Use metadata (e.g., data type, domain, range, dependency, distribution)
  - Check field overloading (unused bits of an attribute 31 out of 32)
  - 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

# 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

# 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



# Correlation Analysis (Nominal Data)

- **$\chi^2$  (chi-square) test**

- To discover correlation between two attributes A and B

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

- The larger the  $\chi^2$  value, the more likely the variables are related
- The cells that contribute the most to the  $\chi^2$  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 (Nominal Data)

- The  $\chi^2$  statistic tests the hypothesis that *A and B are independent, that is, there is no correlation between them.*
- The test is based on a significance level, with  $(r-1)*(c-1)$  degrees of freedom
- If the hypothesis can be rejected, then we say that *A and B are statistically correlated.*

# Chi-Square Calculation: An Example

Suppose that a group of 1500 people was surveyed.

The gender of each person was noted.

Each person was polled as to whether his or her preferred type of reading material was fiction or nonfiction

**Are gender and preferred reading correlated?**

	male	female	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

Expected frequency calculation

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

$$e_{11} = \frac{300 \times 450}{1500} = 90$$

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

# Chi-Square Calculation: An Example

**Are gender and preferred reading correlated?**

	male	female	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

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

In any row, the sum of the expected frequencies must equal the total observed frequency for that row, and the sum of the expected frequencies in any column must also equal the total observed frequency for that column.

- The test is based on a significance level , with  $(r-1)*(c-1)$  degree of freedom
- Degree of freedom  $= (2-1)*(2-1)=1$
- For 1 degree of freedom, the  $\chi^2$  value needed to reject the hypothesis at the 0.001 significance level is 10.828 .
- computed value is above this, we can reject the hypothesis that gender and preferred reading are independent and conclude that the two attributes are (strongly) correlated for the given group of people

# Chi Square distribution table.

probability level (alpha)

Df	0.5	0.10	0.05	0.02	0.01	0.001
1	0.455	2.706	3.841	5.412	6.635	10.827
2	1.386	4.605	5.991	7.824	9.210	13.815
3	2.366	6.251	7.815	9.837	11.345	16.268
4	3.357	7.779	9.488	11.668	13.277	18.465
5	4.351	9.236	11.070	13.388	15.086	20.517

# Chi Square Calculation : Exercise

- The table given below shows the data obtained during the outbreak of smallpox:

	Attacked	Not Attacked	Sum (row)
Vaccinated	31	469	500
Not Vaccinated	185	1315	1500
Sum(col.)	216	1784	2000

- Test the effectiveness of vaccination in preventing the attack from smallpox at 5% level of significance

$$e_{ij} = \frac{\text{count}(A = a_i) * \text{count}(B = b_j)}{N}$$

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

Chi Square distribution table.  
probability level (alpha)

Df	0.5	0.10	0.05	0.02	0.01	0.001
1	0.455	2.706	3.841	5.412	6.635	10.827
2	1.386	4.605	5.991	7.824	9.210	13.815
3	2.366	6.251	7.815	9.837	11.345	16.268
4	3.357	7.779	9.488	11.668	13.277	18.465
5	4.351	9.236	11.070	13.388	15.086	20.517

# Correlation Analysis (Numeric Data)

- Correlation coefficient (also called **Pearson's product moment coefficient**) to evaluate correlation between two attributes, *A* and *B*,

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

- where  $n$  is the number of tuples,  $\bar{A}$  and  $\bar{B}$  are the respective means of *A* and *B*,  $\sigma_A$  and  $\sigma_B$  are the respective standard deviation of *A* and *B*, and  $\sum(a_i b_i)$  is the sum of the *AB* cross-product. (i.e., for each tuple, the value for *A* is multiplied by the value for *B* in that tuple)
- 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

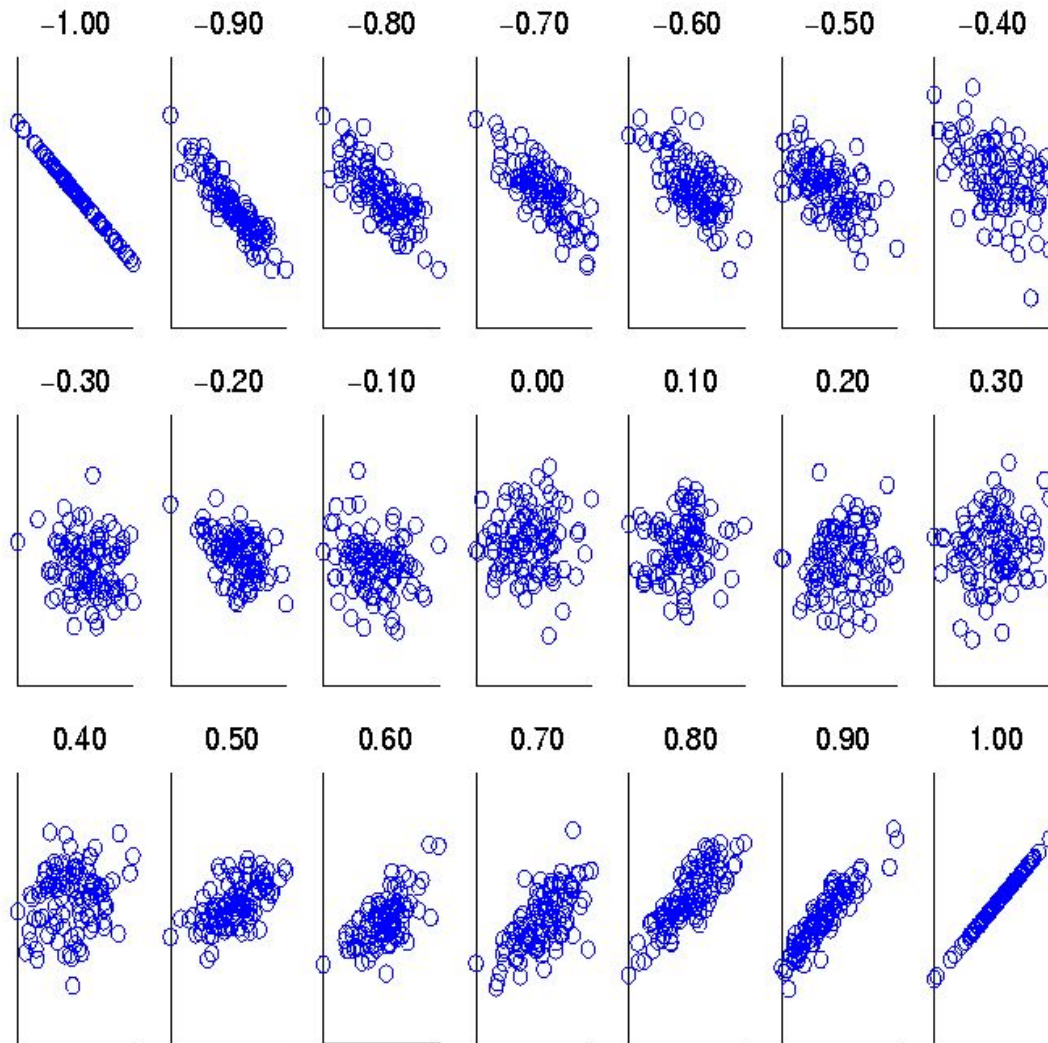
# Example

<i>age</i>	23	23	27	27	39	41	47	49	50
<i>%fat</i>	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
<i>age</i>	52	54	54	56	57	58	58	60	61
<i>%fat</i>	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

Calculate the correlation coefficient (Person's product moment coefficient). Are these two variables positively or negatively correlated?



# Visually Evaluating Correlation



**Scatter plots  
showing the  
similarity from  
-1 to 1.**

# Correlation (viewed as linear relationship)

- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects,  $A$  and  $B$ , and then take their dot product

$$a'_k = (a_k - \text{mean}(A)) / \text{std}(A)$$

$$b'_k = (b_k - \text{mean}(B)) / \text{std}(B)$$

$$\text{correlation}(A, B) = A' \bullet B'$$

# Covariance (Numeric Data)

- Covariance is similar to correlation

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

Correlation coefficient:  $r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B}$

where  $n$  is the number of tuples,  $\bar{A}$  and  $\bar{B}$  are the respective mean or **expected values** of  $A$  and  $B$ ,  $\sigma_A$  and  $\sigma_B$  are the respective standard deviation of  $A$  and  $B$ .


- Positive covariance:** If  $Cov_{A,B} > 0$ , then  $A$  and  $B$  both tend to be larger than their expected values.
- Negative covariance:** If  $Cov_{A,B} < 0$  then if  $A$  is larger than its expected value,  $B$  is likely to be smaller than its expected value.
- Independence:**  $Cov_{A,B} = 0$  but the converse is not true:

# Co-Variance: An Example

- $$Cov(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{n}$$
  
It can be simplified in computation as

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

- Suppose two stocks A and B have the following values in one week:  
(2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?
  - $E(A) = (2 + 3 + 5 + 4 + 6) / 5 = 20 / 5 = 4$
  - $E(B) = (5 + 8 + 10 + 11 + 14) / 5 = 48 / 5 = 9.6$
  - $Cov(A, B) = (2 \times 5 + 3 \times 8 + 5 \times 10 + 4 \times 11 + 6 \times 14) / 5 - 4 \times 9.6 = 4$
- Thus, A and B rise together since  $Cov(A, B) > 0$ .



# **Data Science** **(CS3206)**

**Data Preprocessing**

# In today's discussion...

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

# 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 (lossles,lossy)

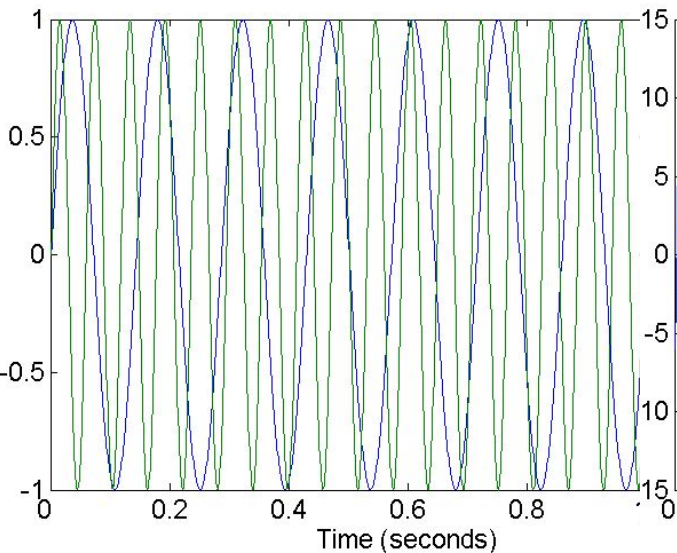
# Data Reduction 1: Dimensionality Reduction

- **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 analytics
  - Allow easier visualization
- **Dimensionality reduction techniques**
  - Wavelet transforms(DWT,DFT)
  - Principal Component Analysis
  - Supervised and nonlinear techniques (e.g., feature selection)

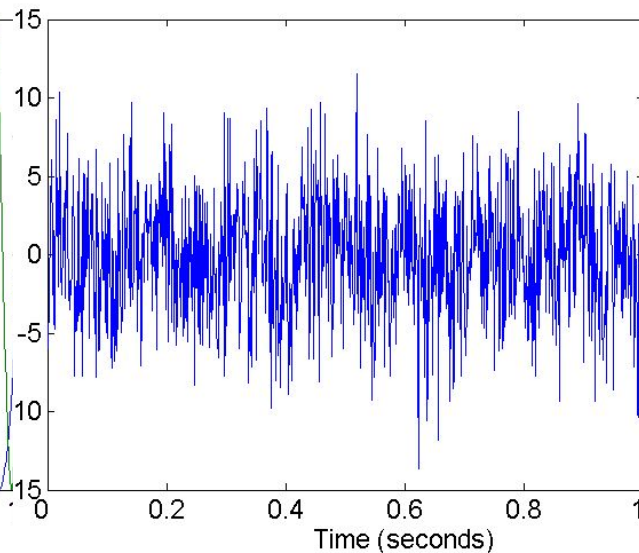


# Mapping Data to a New Space

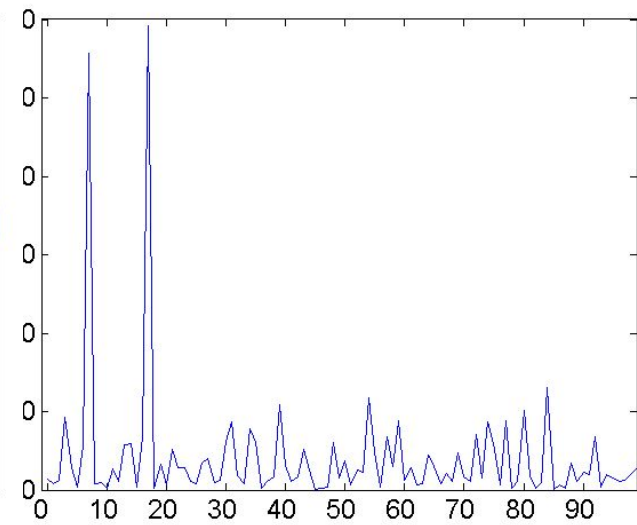
- **Fourier transform**
- **Wavelet transform**



**Two Sine Waves**



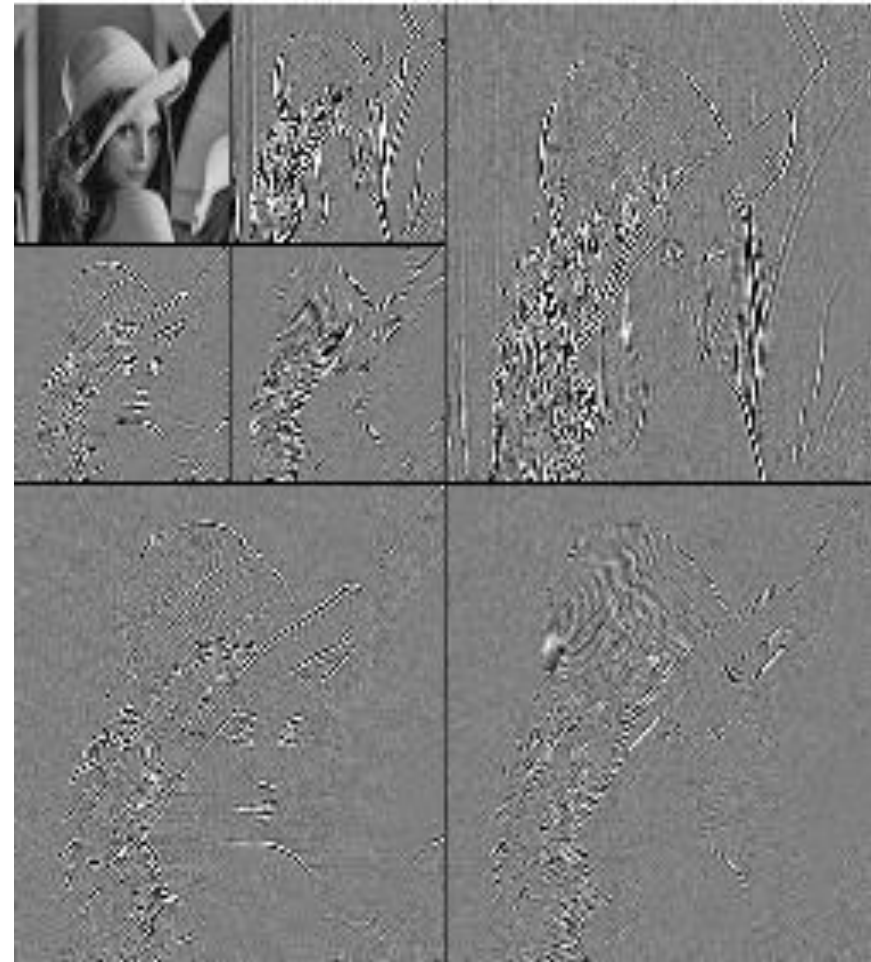
**Two Sine Waves + Noise**



**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

- 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

# Wavelet Decomposition

- Wavelets: A math tool for space-efficient hierarchical decomposition of functions
- Haar wavelets: recursive pair-wise averaging and differencing at different resolutions
  - Simplest wavelet basis, easy to implement

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

$$(a, b) \rightarrow (d, s)$$

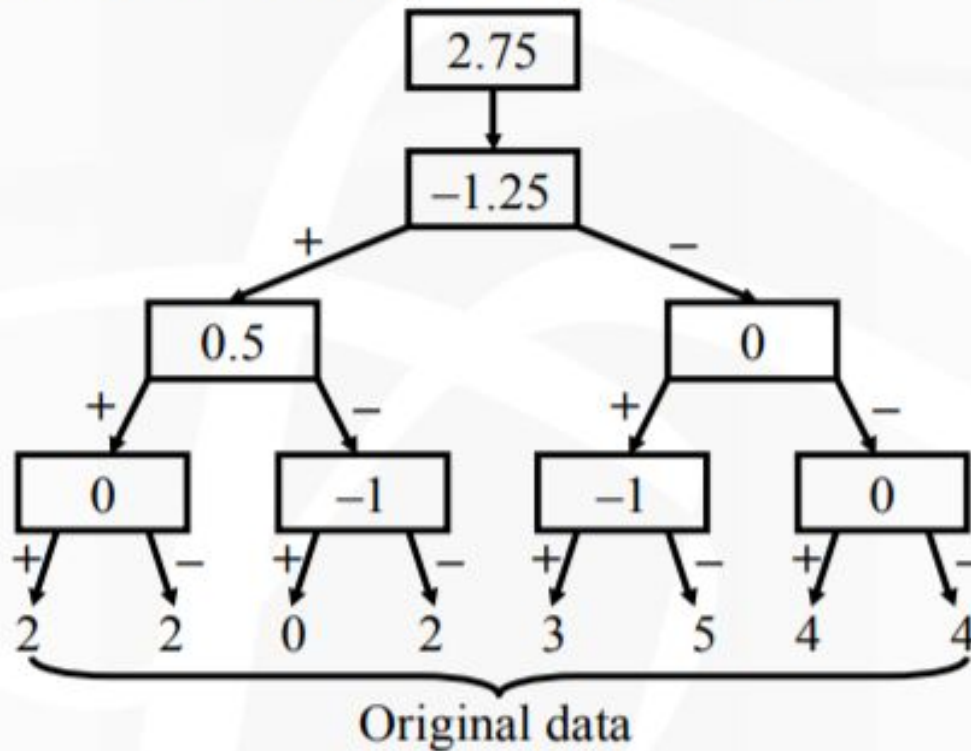
$$s = \frac{a + b}{2},$$

$$d = a - s.$$

Haar wavelet decomposition: [2.75, -1.25, 0.5, 0, 0, -1, -1, 0]

# Haar Wavelet Coefficients

- Hierarchical decomposition structure

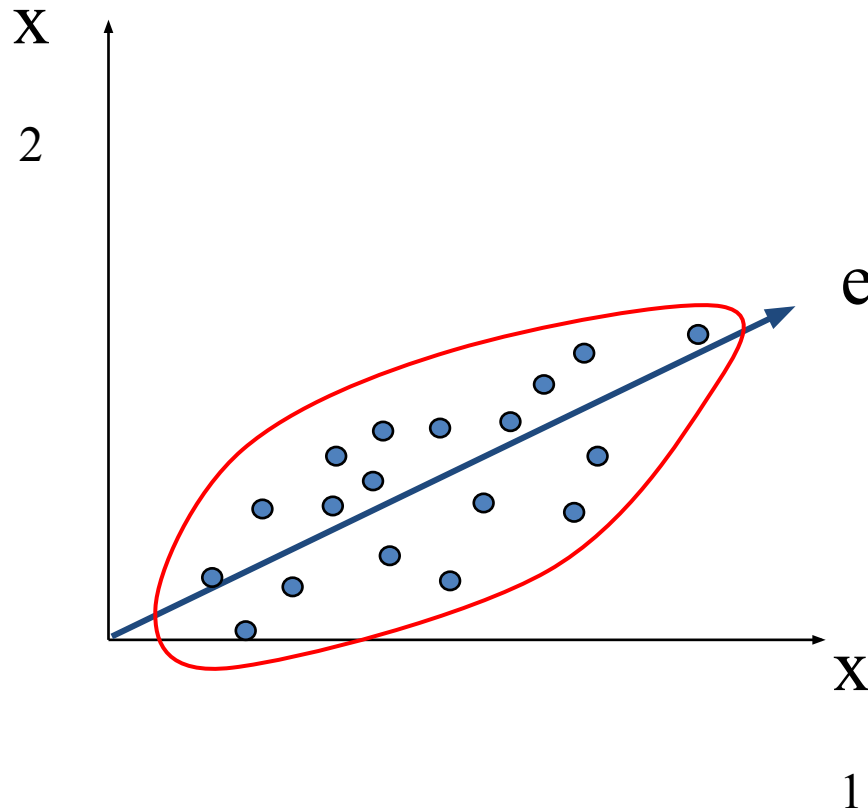


# Why Wavelet Transform?

- 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  $O(N)$

# 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



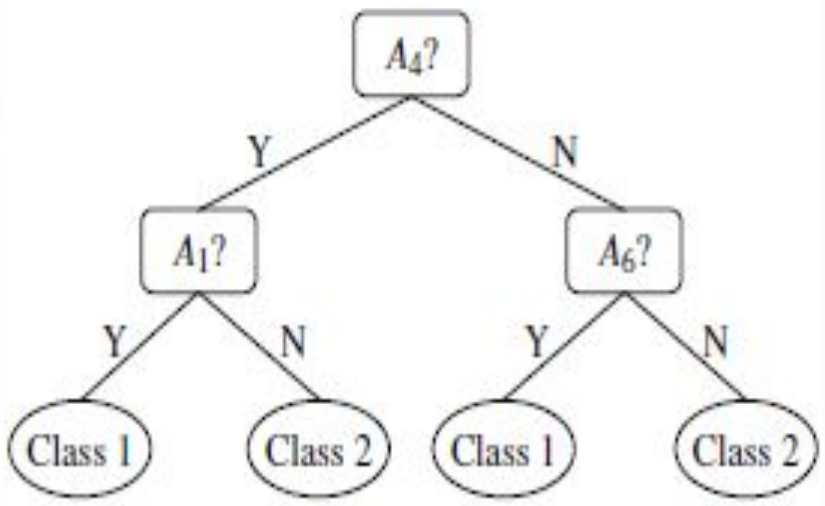
# Attribute Subset Selection

- 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 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

# Heuristic Search in Attribute Selection

Forward selection	Backward elimination	Decision tree induction
<p>Initial attribute set:  <math>\{A_1, A_2, A_3, A_4, A_5, A_6\}</math></p> <p>Initial reduced set:  <math>\{\}</math>  <math>\Rightarrow \{A_1\}</math>  <math>\Rightarrow \{A_1, A_4\}</math>  <math>\Rightarrow</math> Reduced attribute set:  <math>\{A_1, A_4, A_6\}</math></p>	<p>Initial attribute set:  <math>\{A_1, A_2, A_3, A_4, A_5, A_6\}</math>  <math>\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}</math>  <math>\Rightarrow \{A_1, A_4, A_5, A_6\}</math>  <math>\Rightarrow</math> Reduced attribute set:  <math>\{A_1, A_4, A_6\}</math></p>	<p>Initial attribute set:  <math>\{A_1, A_2, A_3, A_4, A_5, A_6\}</math></p>  <pre> graph TD     A4["A4?"] -- Y --&gt; A1["A1?"]     A4 -- N --&gt; A6["A6?"]     A1 -- Y --&gt; C1_1((Class 1))     A1 -- N --&gt; C2_1((Class 2))     A6 -- Y --&gt; C1_2((Class 1))     A6 -- N --&gt; C2_2((Class 2))     </pre> <p><math>\Rightarrow</math> Reduced attribute set:  <math>\{A_1, A_4, A_6\}</math></p>

# 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 approach
  - Attribute construction
    - Combining features
    - 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

# 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  $Y_1, Y_2, \dots, X_1, X_2, \dots$
- 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:
  - Singleton bucket: bucket of single attribute-value pair
  - Equal-width: equal bucket range
  - Equal-frequency (or equal-depth)
- Multi-dimensional histograms to capture dependencies between the attributes



# Histogram Example

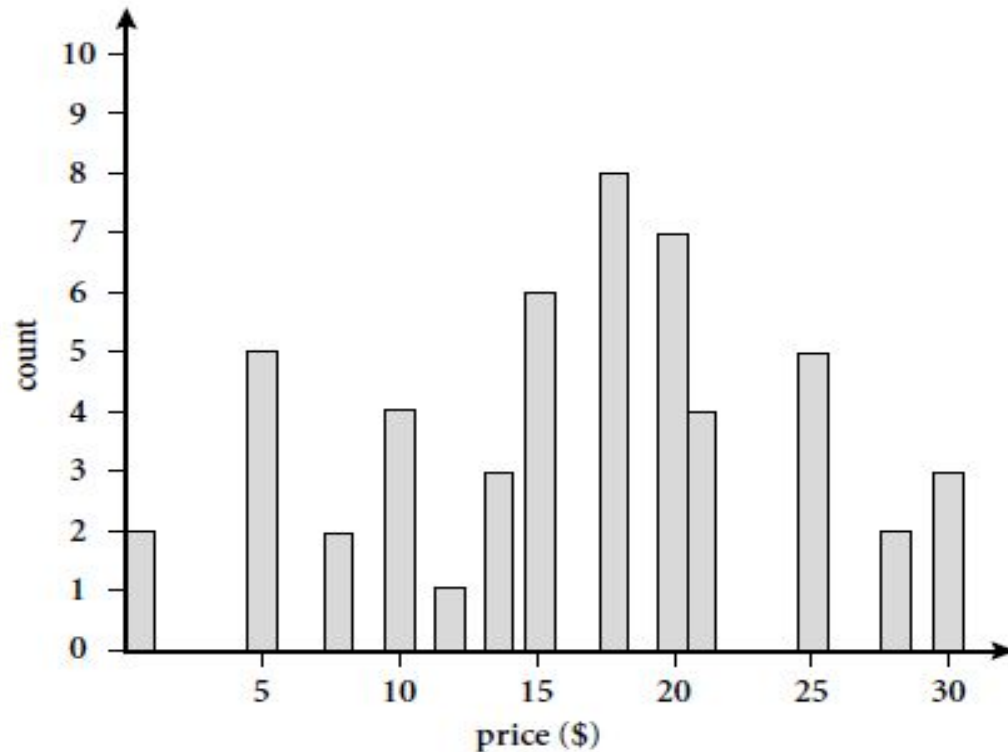
Data: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 12, 14, 14, 14,  
15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18,  
20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25,  
25, 25, 28, 28, 30, 30, 30

Value Frequency Pairs:

(1,2),(5,5),(8,2),(10,3),(12,1),(14,3),(15,6),(18,8),(20,7),  
(21,4),(25,5),(28,2),(30,3)

# Histogram Example

Singleton Bucket:

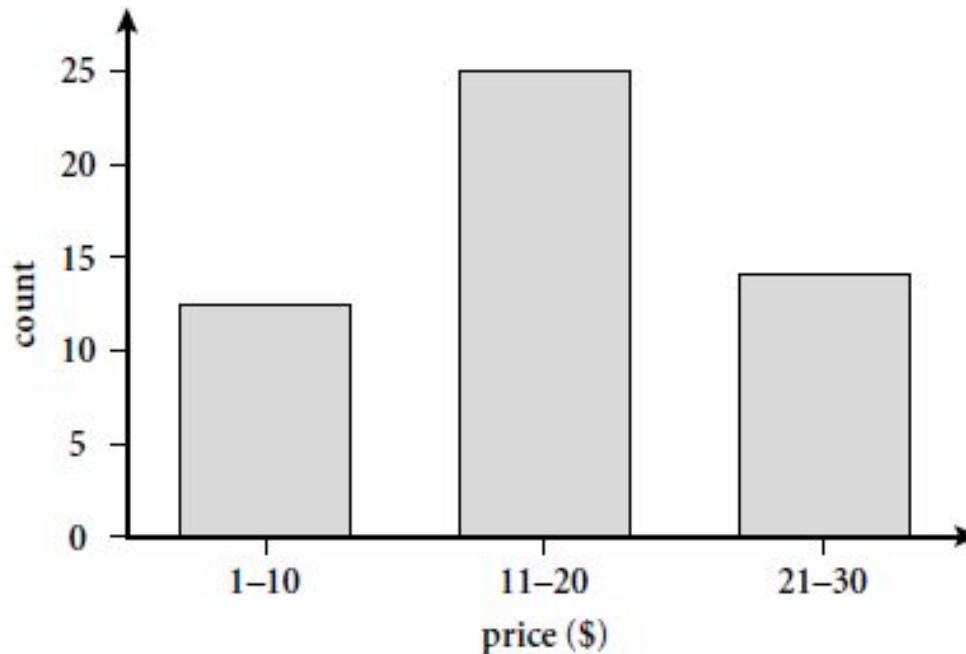


---

A histogram for *price* using singleton buckets—each bucket represents one price-value/frequency pair.

# Histogram Example

Equal-Width



---

An equal-width histogram for *price*, where values are aggregated so that each bucket has a uniform width of \$10.

# Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Parameters to be considered:
  - Intra-cluster similarity
  - Inter-cluster dissimilarity

} Based on a distance function defining 'closeness' of the objects
- Quality parameters
  - Diameter: the maximum distance between any two objects in the cluster
  - Centroid distance: the average distance of each cluster object from the cluster centroid

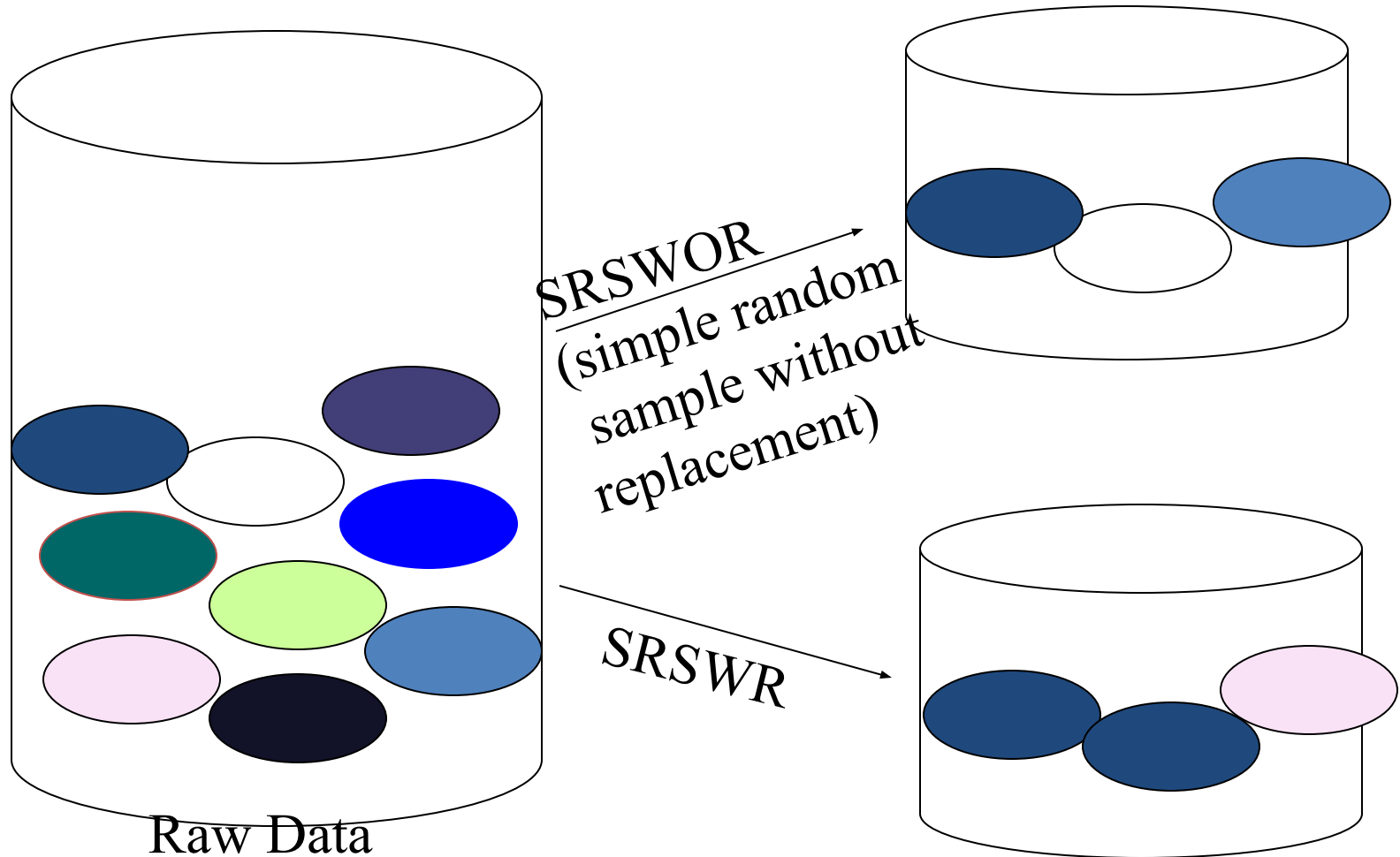
# Sampling

- Sampling: obtaining a small sample  $s$  to represent the whole data set  $N$
- Key principle: Choose a **representative** subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
  - Develop adaptive sampling methods, e.g., stratified sampling:

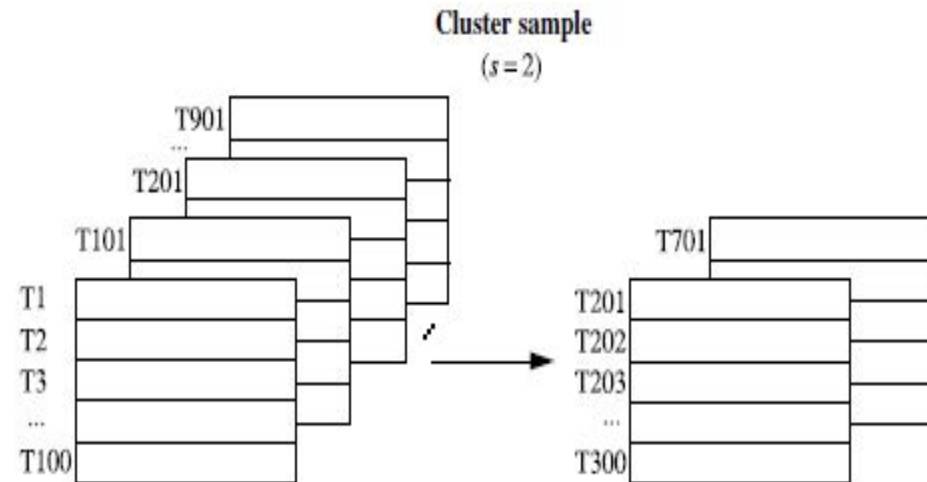
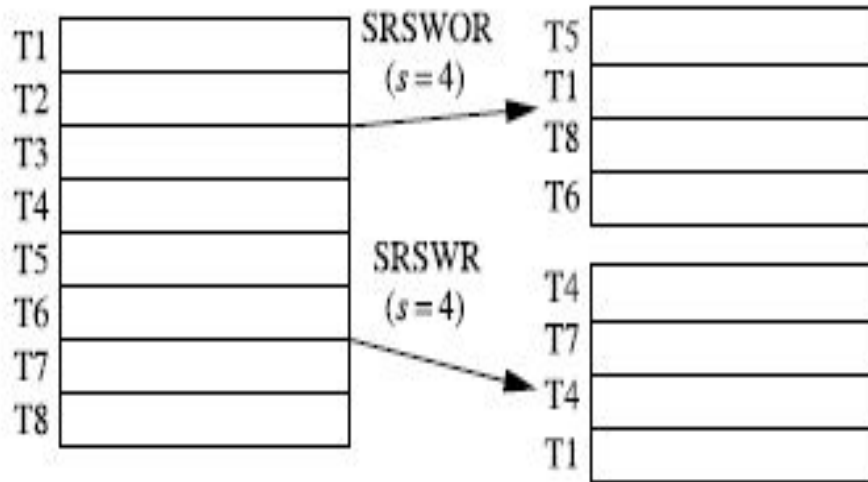
# Types of Sampling

- Sampling: obtaining a small sample  $s$  to represent the whole data set  $N$
- **Simple random sample without replacement (SRSWOR)**
  - All tuples have equal probability to be sampled
  - Once an object is selected, it is removed from the population
- **Simple Random Sample With Replacement (SRSWR)**
  - A selected object is not removed from the population
- **Cluster or 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



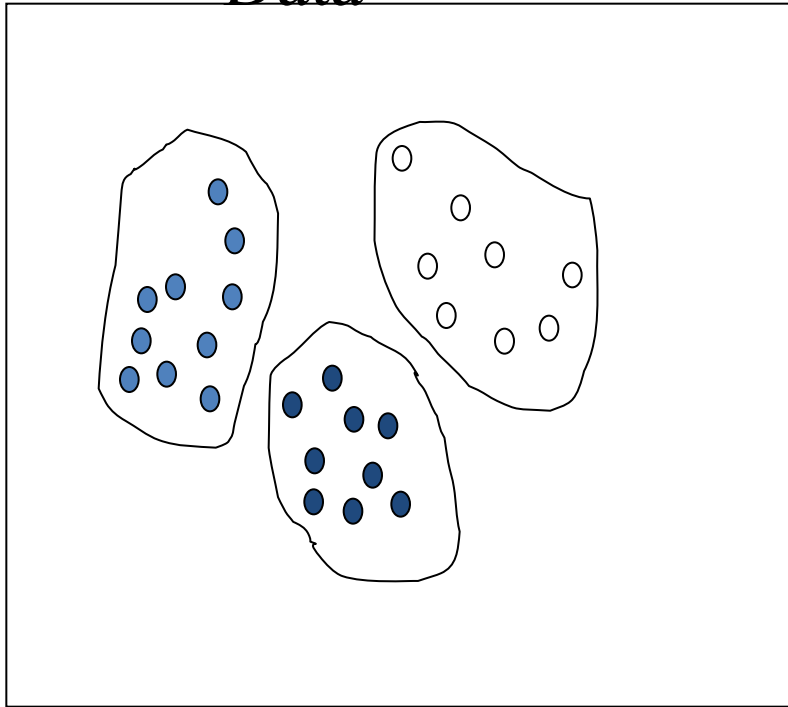
# Sampling: With or without Replacement



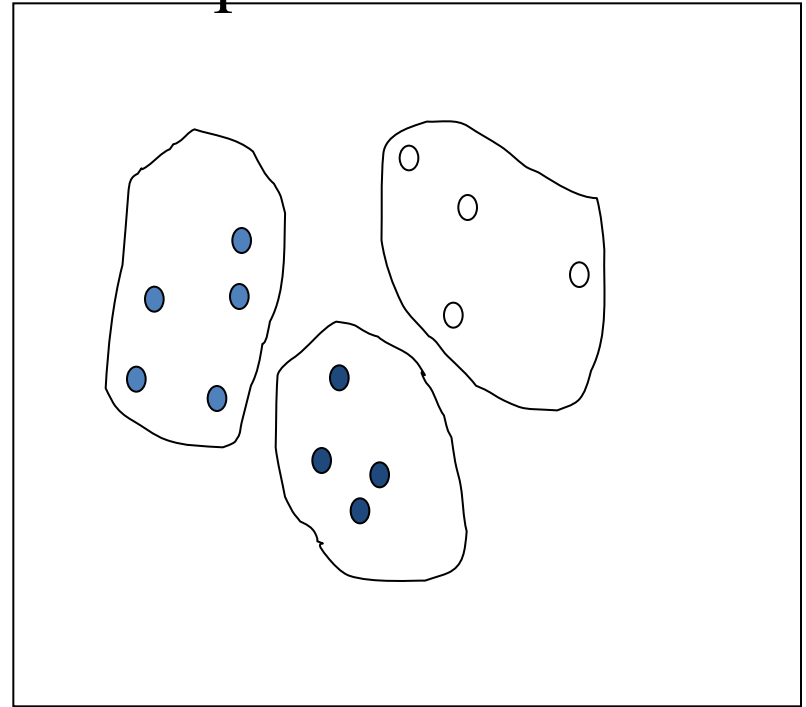


# Sampling: Cluster or Stratified Sampling

Raw  
Data



Cluster/Stratified  
Sample



# Sampling: Cluster or Stratified Sampling

Stratified 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

# Data Cube Aggregation

- Reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data

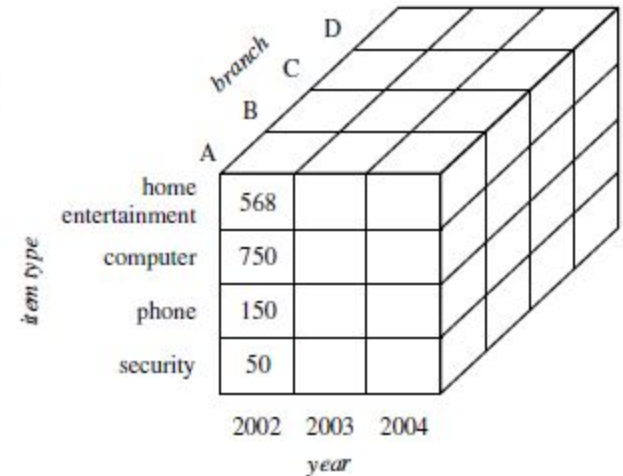
Year 2004	
Quarter	Sales
Q1	0
Q2	0
Q3	0
Q4	0

Year 2003	
Quarter	Sales
Q1	0
Q2	0
Q3	0
Q4	0

Year 2002	
Quarter	Sales
Q1	\$224,000
Q2	\$408,000
Q3	\$350,000
Q4	\$586,000

Year	Sales
2002	\$1,568,000
2003	\$2,356,000
2004	\$3,594,000

Sales data for a given branch of *AllElectronics* for the years 2002 to 2004. On the left, the sales are shown per quarter. On the right, the data are aggregated to provide the annual sales.

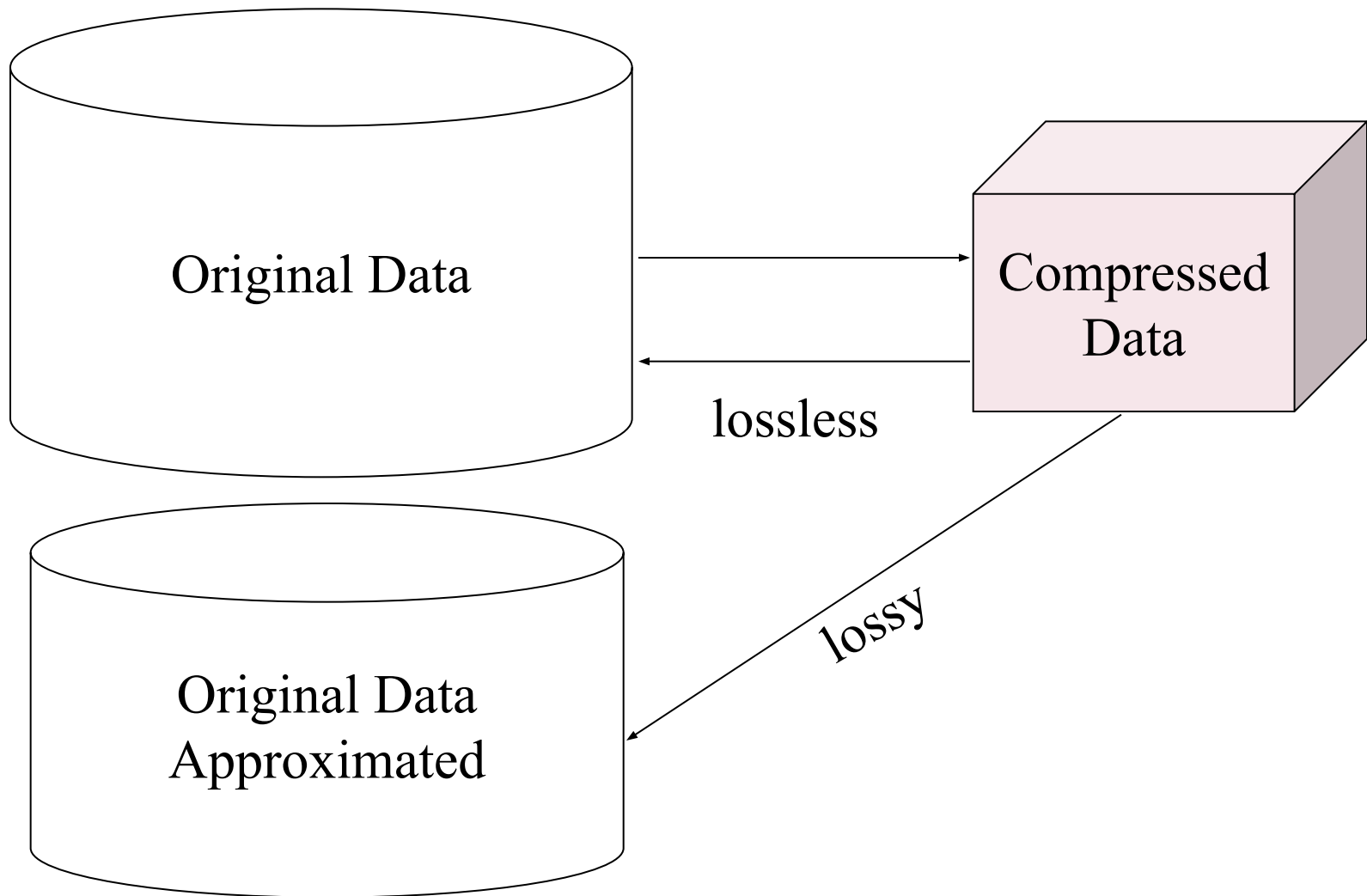


A data cube for sales at *AllElectronics*.

# 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; binning, regression, clustering
  - 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

# Data Transformation: Normalization

- Min-max normalization: to  $[\text{new\_min}_A, \text{new\_max}_A]$

$$v' = \frac{v - \text{min}_A}{\text{max}_A - \text{min}_A} (\text{new\_max}_A - \text{new\_min}_A) + \text{new\_min}_A$$

- Ex. Let income range \$12,000 to \$98,000 normalized to  $[0.0, 1.0]$ .

Then \$73,600 is mapped to  $\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716$

- Z-score normalization ( $\mu$ : mean,  $\sigma$ : standard deviation):

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

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



# 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
  - Discretization can be performed recursively on an attribute

# Concept Hierarchy Generation

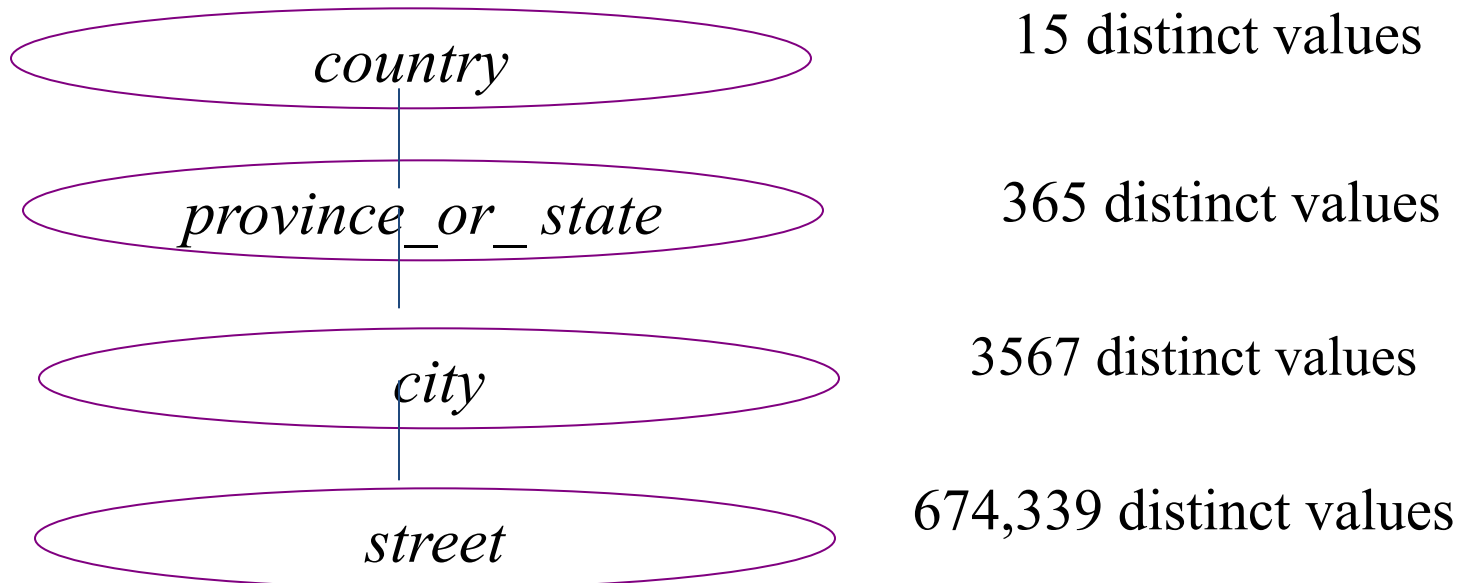
- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with dimension
- Concept hierarchies facilitate to view data in multiple granularity
- Concept hierarchy formation: Recursively reduce the data by collecting and replacing low level concepts (such 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 designers
- Concept hierarchy can be automatically formed for both numeric and nominal data.

# 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



# 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 transformation**
  - Normalization; Concept hierarchy generation