D Y PATIL
UNIVERSITY
NAVI MUMBAI

Data Mining & Business Intelligence

Module 2:Data Preprocessing

Index -

Lecture 8 - Measuring similarity and dissimilarity.	3
Lecture 9 - Data Pre-processing	24
Lecture 10 - Histograms, Clustering and Sampling, Data Transformation	52
Lecture 11-Normalization, Binning, Histogram analysis and concept hierarchy generation	63

Lecture 8

Measuring Similarity and Dissimilarity

Similarity and Dissimilarity

- **Similarity**
  - Numerical measure of how alike two data objects are
  - Value is higher when objects are more alike
  - Often falls in the range [0,1]
- **Dissimilarity** (e.g., distance)
  - Numerical measure of how different two data objects are
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- **Proximity** refers to a similarity or dissimilarity

Data Matrix and Dissimilarity Matrix

- Data matrix
  - n data points with p dimensions
  - Two modes

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

- Dissimilarity matrix
  - n data points, but registers only the distance
  - A triangular matrix/singular
  - Single mode
  - Weights are associated
  - Distance function is used

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ : & : & : & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Distance Measure(Jaccard Distance)

  - **The Jaccard similarity** of two **sets** is the size of their intersection divided by the size of their union:
    ***sim*$(C_1, C_2) = |C_1 \cap C_2|/|C_1 \cup C_2$**
  - **Jaccard distance: *d*$(C_1, C_2) = 1 - |C_1 \cap C_2|/|C_1 \cup C_2|$**
- Verify the function:
- d(x,y) is nonnegative: becoz size of intersection cannot exceed the size of union.
- d(x,y)=0 is strictly positive.
- d(x,y)=d(y,x) : both union and intersections are symmetric.
- For triangle inequality SIM(x,y)is the probability a random minhash function maps x and y to the same value.

Distance Measure(Euclidian Distance)

- N dimensional space is one where points are vectors of n real numbers.
- It verify first three points:
- Distance between two points cannot be negative
- Aii squares of real numbers are nonnegative (xi!yi)
- If xi=yi then the distance is clearly zero
- The sum of the lengths of any two sides of a triangles is no less than the length of the third side.

Distance Measure(Cosine Distance)

- The cosine distance makes sense in space(dimension)
- Where points are vectors with integer components or Boolean components.
- Cosine distance between two points is the angle that the vectors to those points make.
- The angle can be in range 0 to 180 degrees.
- Calculate the cosine distance first compute the cosine of angles.
- Then applying the arc-cosine function to translate to an angle in the 0 – 180 degree range.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

Distance Measure(Edit Distance)

- The distance makes sence when points are strings.
- The distance between two strings x=x1x2---------xn and y=y1y2----------ym
- The smallest number of insertion and deletion of single characters that will convert x to y.
- Edit distance can be negative only two identical strings have an edit distance of 0
- Edit distance is symmetric: when th sequence of insertion and deletion can be reversed.
- The triangle inequality is also straightforward.

$D(x,y)=|x|+|y|-2|LCS(x,y)|$

Distance Measure(Hamming Distance)

- Hamming distance cannot be negative
- If it is zero, vectors are identical
- The distance does not depends upon which two vector we consider first.
- The triangle inequality should also evident
- If x and z differ in m components and z and y differ in n components , then x and y cannot differ in more than m+n components.
- Hamming distance is used when the vectors are Boolean; they consist values of 0’s and 1’s.

Proximity Measure for Nominal Attributes

- Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Method 1: Simple matching
  - *m*: # of matches, *p*: total # of variables

$$d(i,j)=\frac{p-m}{p}$$

- Method 2: Use a large number of binary attributes
  - creating a new binary attribute for each of the *M* nominal states

Proximity Measure for Binary Attributes

- A contingency table for binary data

		Object *j*		
		1	0	sum
Object *i*	1	q	r	$q+r$
	0	s	t	$s+t$
	sum	$q+s$	$r+t$	p

- Distance measure for symmetric binary variables:

$$d(i, j) = \frac{r+s}{q+r+s+t}$$

- Distance measure for asymmetric binary variables:

$$d(i, j) = \frac{r+s}{q+r+s}$$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$sim_{Jaccard}(i, j) = \frac{q}{q+r+s}$$

- Note: Jaccard coefficient is the same as “coherence”:

$$coherence(i, j) = \frac{sup(i, j)}{sup(i) + sup(j) - sup(i, j)} = \frac{q}{(q+r)+(q+s)-q}$$

Dissimilarity between Binary Variables

- Example

Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

  - Gender is a symmetric attribute
  - The remaining attributes are asymmetric binary
  - Let the values Y and P be 1, and the value N 0

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

Standardizing Numeric Data

- Z-score:
  - X: raw score to be standardized, μ: mean of the population, σ: standard deviation
  - the distance between the raw score and the population mean in units of the standard deviation
  - negative when the raw score is below the mean, "+" when above
- An alternative way: Calculate the mean absolute deviation

$$z = \frac{x-\mu}{\sigma}$$

where

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

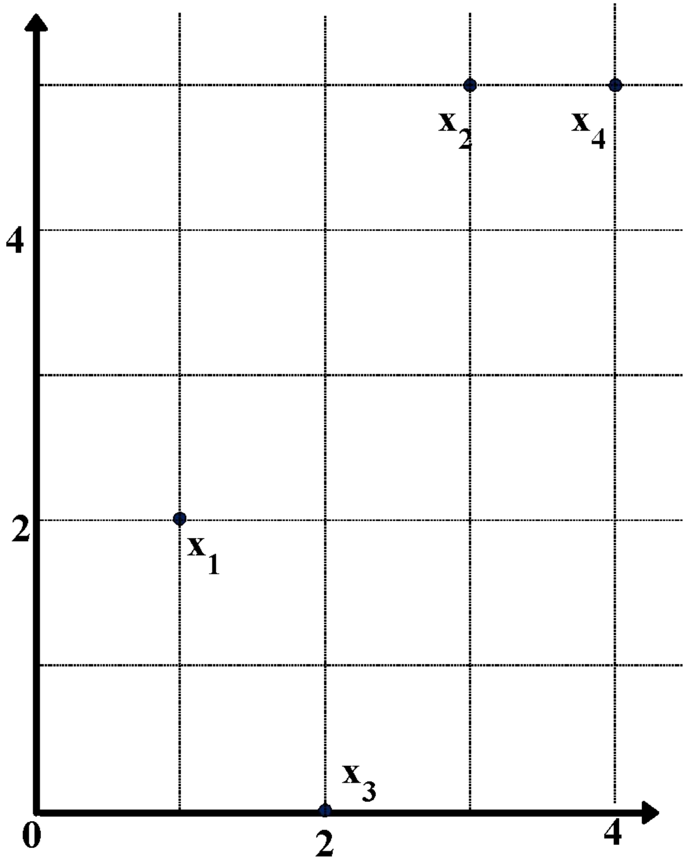
$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + ... + x_{nf}).$$

  - standardized measure (*z-score*):

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation

Example:
Data Matrix and Dissimilarity Matrix



Data Matrix

point	attribute1	attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Dissimilarity Matrix

(with Euclidean Distance)

	x1	*x2*	*x3*	*x4*
x1	0			
x2	3.61	0		
x3	5.1	5.1	0	
x4	4.24	1	5.39	0

Distance on Numeric Data: Minkowski Distance

- *Minkowski distance*: A popular distance measure

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ are two p-dimensional data objects, and h is the order (the distance so defined is also called L-h norm)

- Properties
  - d(i, j) > 0 if i ≠ j, and d(i, i) = 0 (Positive definiteness)
  - d(i, j) = d(j, i) (Symmetry)
  - d(i, j) ≤ d(i, k) + d(k, j) (Triangle Inequality)
- A distance that satisfies these properties is a metric

Special Cases of Minkowski Distance

- h = 1: Manhattan (city block, L_1 norm) distance
  - E.g., the Hamming distance: the number of bits that are different between two binary vectors

- h = 2: (L_2 norm) Euclidean distance

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+\ldots+|x_{i_p}-x_{j_p}|^2)}$$

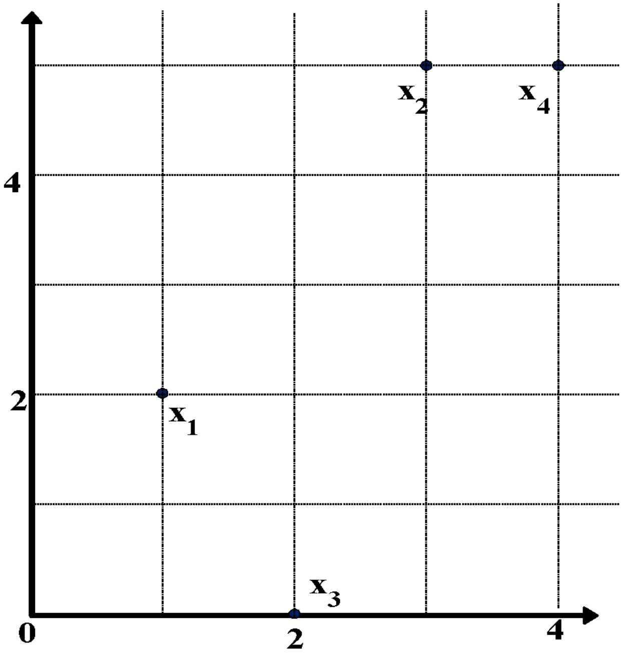
$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+\ldots+|x_{i_p}-x_{j_p}|$$

- h → ∞. “supremum” (L_{max} norm, L_{∞} norm) distance.
  - This is the maximum difference between any component (attribute) of the vectors

$$d(i,j)=\lim_{h\to\infty}\left(\sum_{f=1}^{p}|x_{if}-x_{jf}|^h\right)^{\frac{1}{h}}=\max_{f}^{p}|x_{if}-x_{jf}|$$

Example: Minkowski Distance

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



Dissimilarity Matrices

Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidean (L_2)

L2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Supremum

L_∞	x1	x2	x3	x4
x1	0			
x2	3	0		
x3	2	5	0	
x4	3	1	5	0

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank(e.g freshman,sophomore,junior,senior)
- Can be treated like interval-scaled
  - replace x_{if} by their rank $r_{if} \in \{1, \ldots, M_f\}$
  - map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - compute the dissimilarity using methods for interval-scaled variables
  - Freshman:0sophomore:1/3;junior:2/3;senior:1
  - Distance:d(freshman,senior)=1,d(junior,senior)=1/3

Attributes of Mixed Type

A database may contain all attribute types

Nominal, symmetric binary, asymmetric binary, numeric, ordinal

One may use a weighted formula to combine their effects

$$d(i,j) = \frac{\Sigma_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\Sigma_{f=1}^{p} \delta_{ij}^{(f)}}$$

f is binary or nominal:

$d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise

f is numeric: use the normalized distance

f is ordinal

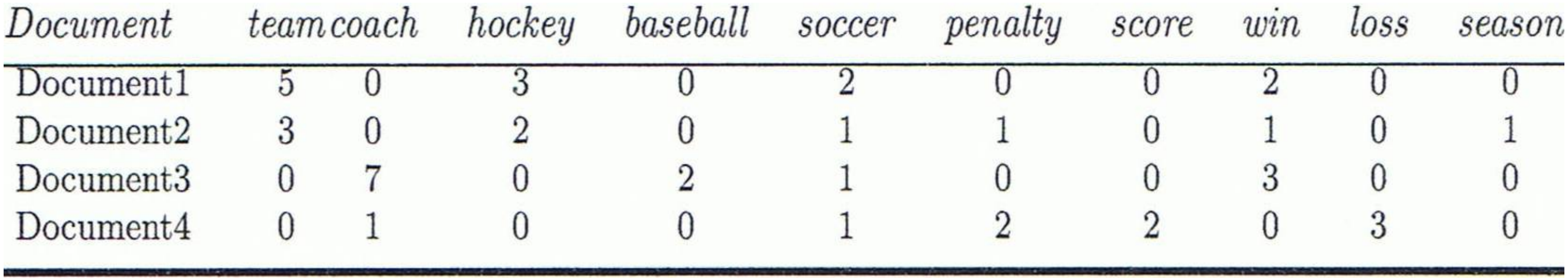
Compute ranks r_{if} and

Treat z_{if} as interval-scaled

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

Cosine Similarity

- A **document** can be represented by thousands of attributes, each recording the *frequency* of a particular word (such as keywords) or phrase in the document.



Document	*team*	*coach*	*hockey*	*baseball*	*soccer*	*penalty*	*score*	*win*	*loss*	*season*
Document1	5	0	3	0	2	0	0	2	0	0
Document2	3	0	2	0	1	1	0	1	0	1
Document3	0	7	0	2	1	0	0	3	0	0
Document4	0	1	0	0	1	2	2	0	3	0

- Applications: information retrieval, biologic taxonomy, gene feature mapping, ...
- Cosine measure: If d_1 and d_2 are two vectors (e.g., term-frequency vectors), then
- $\cos(d_1, d_2) = (d_1 \cdot d_2) / ||d_1|| \, ||d_2||$,
  - where · indicates vector dot product, $||d||$: the length of vector d

Example: Cosine Similarity

- $\cos(d_1, d_2) = (d_1 \cdot d_2) / ||d_1|| \, ||d_2||$,
  where · indicates vector dot product, $||d|$: the length of vector d

- Ex: Find the **similarity** between documents 1 and 2.

  d_1 **=** (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)
  d_2 **=** (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)

  $d_1 \cdot d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$
  $||d_1|| = (5*5+0*0+3*3+0*0+2*2+0*0+0*0+2*2+0*0+0*0)^{\mathbf{0.5}}=(42)^{\mathbf{0.5}} = 6.481$
  $||d_2|| = (3*3+0*0+2*2+0*0+1*1+1*1+0*0+1*1+0*0+1*1)^{\mathbf{0.5}}=(17)^{\mathbf{0.5}} = 4.12$
  $\cos(d_1, d_2) = 0.94$

Lecture 9

Data Pre-processing

Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
  - Accuracy: correct or wrong, accurate or not
  - Completeness: not recorded, unavailable, …
  - Consistency: some modified but some not, dangling, …
  - Timeliness: timely update?
  - Believability: how trustable the data are correct?
  - Interpretability: how easily the data can be understood?

Major Tasks in Data Preprocessing

- **Data cleaning**
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
  - Integration of multiple databases, data cubes, or files
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction
  - Data compression
- **Data transformation and data discretization**
  - Normalization
  - Concept hierarchy generation

Data Cleaning

- Data 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?

Incomplete (Missing) Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g., “unknown”, a new class?!
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
  - the most probable value: inference-based such as Bayesian formula or decision tree

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which require data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

How to Handle Noisy Data?

- Binning
  - first sort data and partition into (equal-frequency) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
  - smooth by fitting the data into regression functions
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)

Data Cleaning as a Process

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

Data Integration

- **Data integration**:
  - Combines data from multiple sources into a coherent store
- Schema integration: e.g., A.cust-id ≡ 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

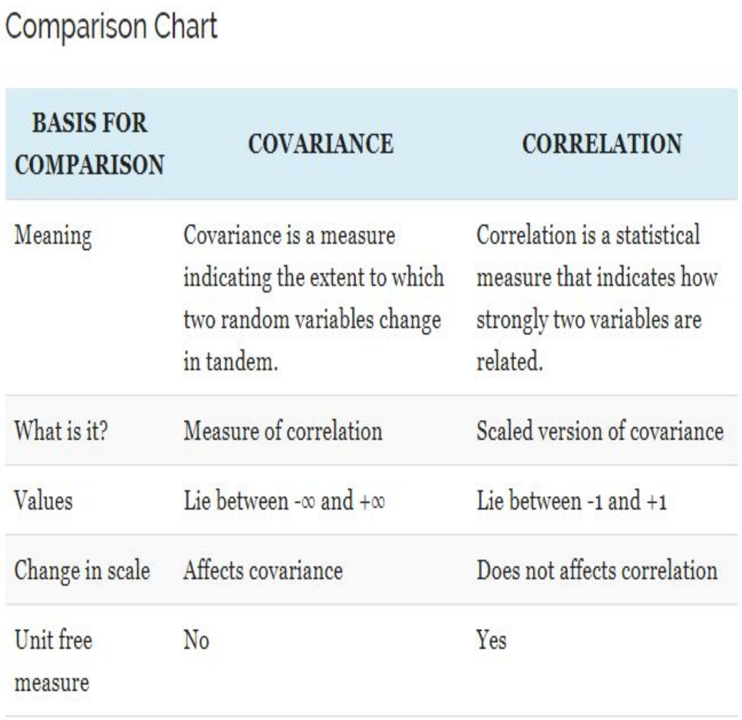
32

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

Covariance and Correlation

- **Covariance** and **Correlation** are two mathematical concepts which are quite commonly used in business statistics.
- Both of these two determine the relationship and measures the dependency between two random variables.
- Despite, some similarities between these two mathematical terms, they are different from each other.
- Correlation is when the change in one item may result in the change in another item
- Correlation is considered as the best tool for for measuring and expressing the quantitative relationship between two variables in formula
- . On the other hand, covariance is when two items vary together.



Comparison Chart

BASIS FOR COMPARISON	COVARIANCE	CORRELATION
Meaning	Covariance is a measure indicating the extent to which two random variables change in tandem.	Correlation is a statistical measure that indicates how strongly two variables are related.
What is it?	Measure of correlation	Scaled version of covariance
Values	Lie between -∞ and +∞	Lie between -1 and +1
Change in scale	Affects covariance	Does not affects correlation
Unit free measure	No	Yes

Correlation Analysis (Nominal 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

Chi-Square Calculation: An Example

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

- X^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

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

- It shows that like_science_fiction and play_chess are correlated in the group

Correlation Analysis (Numeric Data)

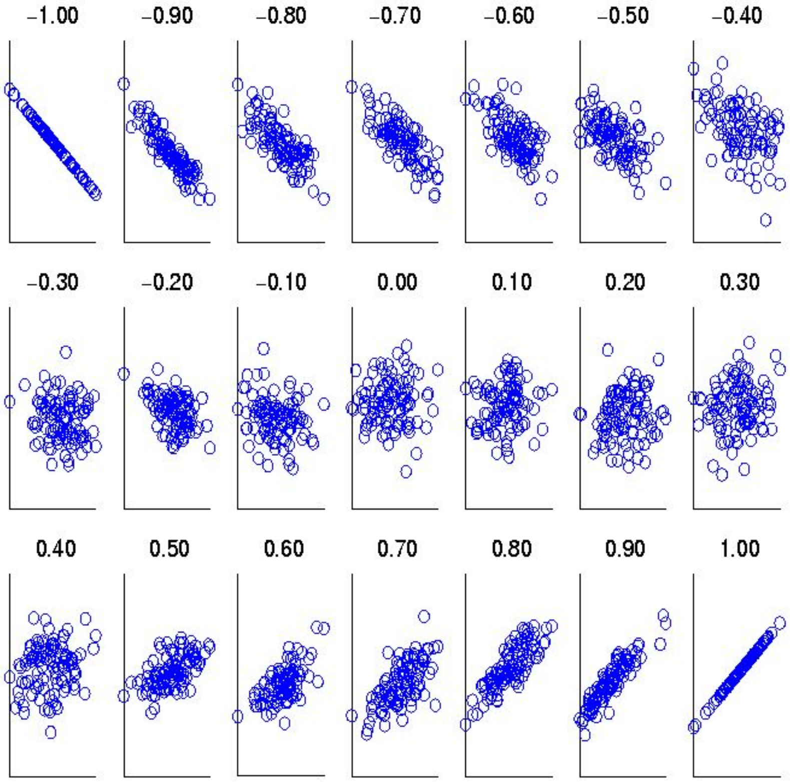
- Correlation coefficient (also called Pearson's product moment coefficient)

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

where n is the number of tuples, $\overline{A}$ and $\overline{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product.

- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent; $r_{AB} < 0$: negatively correlated

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

$$a'_k = (a_k - mean(A)) / std(A)$$

$$b'_k = (b_k - mean(B)) / std(B)$$

$$correlation(A, B) = A' \bullet B'$$

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, σ_A and σ_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:

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

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 simplifi

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

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

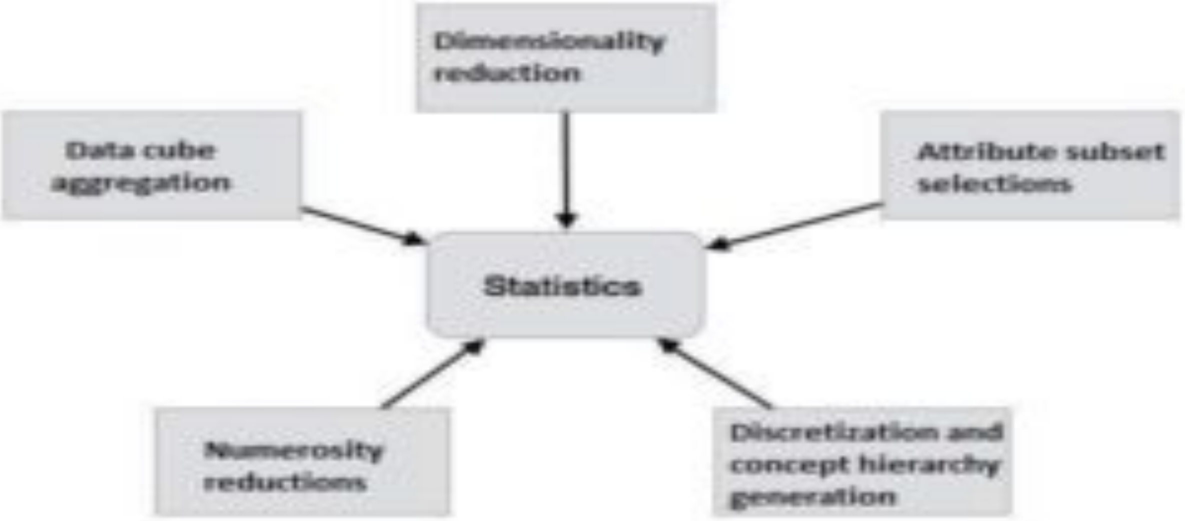
E(B) = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6

Cov(A,B) = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4

Thus, A and B rise together since Cov(A, B) > 0.

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.



-Dimensionality reduction

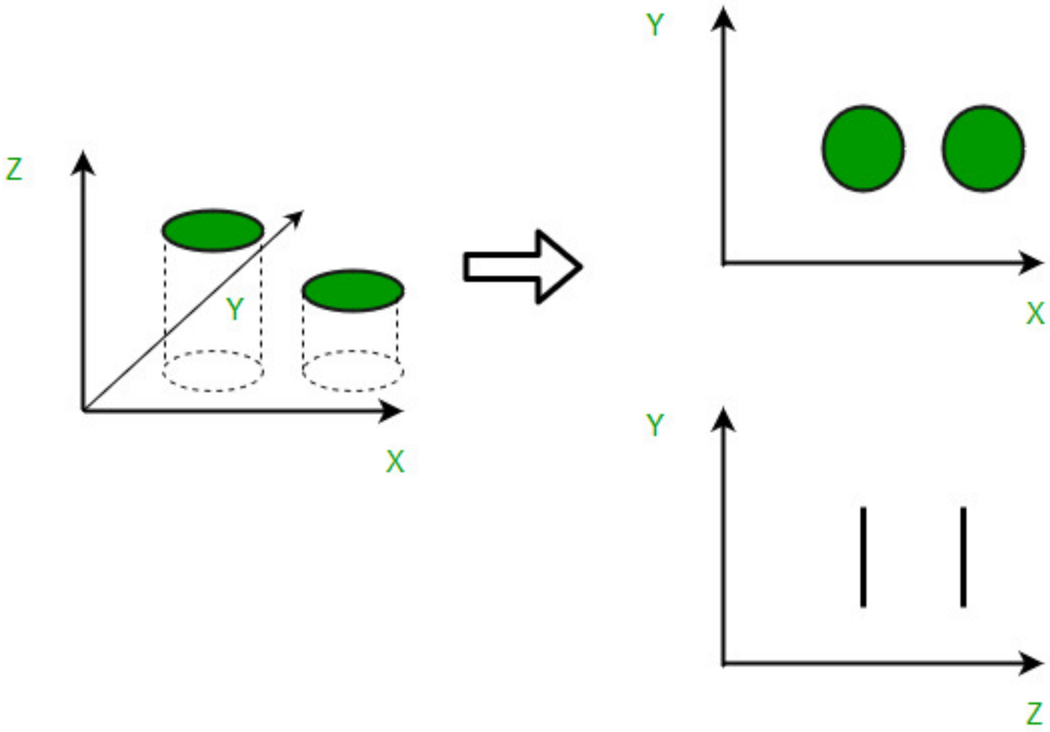
- In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. T
- hese factors are basically variables called features.
- The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant.
- This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables
- . It can be divided into feature selection and feature extraction.

- Data reduction strategies

  , e.g., remove unimportant attributes

  - Wavelet transforms(uses filters, Remove outliers, performed clustering)
  - Principal Components Analysis (PCA)-(PCA is useful when there is data on a large number of variables, and (possibly) there is some redundancy in those variables.
  - In this case, redundancy means that some of the variables are correlated with one another.
  - And because of this redundancy, PCA can be used to reduce the observed variables into a smaller number of principal components that will account for most of the variance in the observed variables.)

Dimensionality Reduction



Components of Dimensionality Reduction

Feature selection: In this, we try to find a subset of the original set of variables, or features, to get a smaller subset which can be used to model the problem. It usually involves three ways:

- Filter
- Wrapper
- Embedded

Feature extraction: This reduces the data in a high dimensional space to a lower dimension space, i.e. a space with lesser no. of dimensions.

Advantages of Dimensionality Reduction

- It helps in data compression, and hence reduced storage space.
- It reduces computation time.
- It also helps remove redundant features, if any.

Disadvantages of Dimensionality Reduction

- It may lead to some amount of data loss.
- PCA tends to find linear correlations between variables, which is sometimes undesirable.
- PCA fails in cases where mean and covariance are not enough to define datasets.
- We may not know how many principal components to keep- in practice, some thumb rules are applied.

Data Reduction Strategies

  - 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

Attribute Subset Selection

Another way to reduce dimensionality of data

Redundant attributes

- Duplicate much or all of the information contained in one or more other attributes
- E.g., purchase price of a product and the amount of sales tax paid

Irrelevant attributes

- Contain no information that is useful for the data mining task at hand
- E.g., students' ID is often irrelevant to the task of predicting students' GPA

Heuristic Search in Attribute Selection

There are *2^d* possible attribute combinations of *d* attributes

- Typical heuristic attribute selection methods:
  - Best single attribute under the attribute independence assumption: choose by significance tests
  - Best step-wise feature selection:
    - The best single-attribute is picked first
    - Then next best attribute condition to the first, ...
  - Step-wise attribute elimination:
    - Repeatedly eliminate the worst attribute
  - Best combined attribute selection and elimination
  - Optimal branch and bound:
    - Use attribute elimination and backtracking

Attribute Creation (Feature Generation)

Create new attributes (features) that can capture the important information in a data set more effectively than the original ones

Three general methodologies

- Attribute extraction
  - Domain-specific
- Mapping data to new space (see: data reduction)
  - E.g., Fourier transformation, wavelet transformation, manifold approaches (not covered)
- Attribute construction
  - Combining features (see: discriminative frequent patterns in Chapter 7)
  - Data discretization

Data Reduction 2: Numerosity Reduction

Reduce data volume by choosing alternative, *smaller forms* of data representation

Parametric methods (e.g., regression)

- Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
- Ex.: Log-linear models—obtain value at a point in *m*-D space as the product on appropriate marginal subspaces

Non-parametric methods

- Do not assume models
- Major families: histograms, clustering, sampling, …

Parametric Data Reduction: Regression and Log-Linear Models

Linear regression(one independent variable and one dependent variable)

Data modeled to fit a straight line

Often uses the least-square method to fit the line

Multiple regression(Multiple Independent variable one dependent variable)

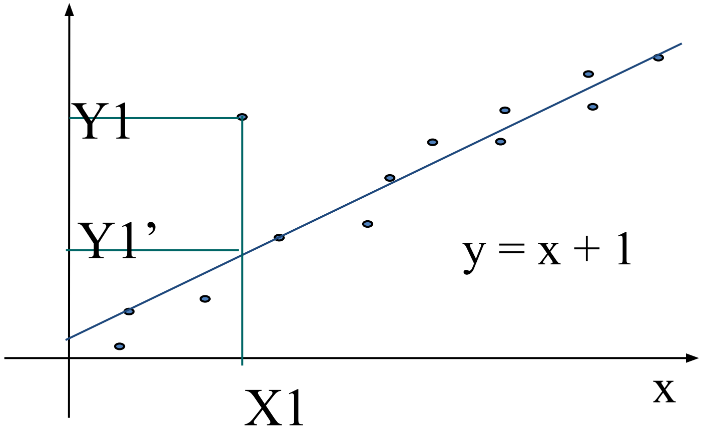
Allows a response variable Y to be modeled as a linear function of multidimensional feature vector

Log-linear model

- Approximates discrete multidimensional probability distributions
- Logistic regression is used to find the probability of event=Success and event=Failure. We should use logistic regression when the dependent variable is binary (0/ 1, True/ False, Yes/ No) in nature.

Regression Analysis

- Regression analysis: A collective name for techniques for the modeling and analysis of numerical data consisting of values of a ***dependent variable*** (also called ***response variable*** or *measurement*) and of one or more *independent variables* (aka. ***explanatory variables*** or ***predictors***)
- The parameters are estimated so as to give a "**best fit**" of the data
- Most commonly the best fit is evaluated by using the ***least squares method***, but other criteria have also been used



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

Regress Analysis and Log-Linear Models

Linear regression: $Y = w X + b$

Two regression coefficients, *w* and *b,* specify the line and are to be estimated by using the data at hand

Using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.

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

Non-Parametric Methods

Histograms:
Histogram is the data representation in terms of frequency. It uses binning to approximate data distribution and is a popular form of data reduction.

Clustering:
Clustering divides the data into groups/clusters. This technique partitions the whole data into different clusters. In data reduction, the cluster representation of the data are used to replace the actual data. It also helps to detect outliers in data.

Sampling:
Sampling can be used for data reduction because it allows a large data set to be represented by a much smaller random data sample (or subset).

Data Cube Aggregation:
Data cube aggregation involves moving the data from detailed level to a fewer number of dimensions. The resulting data set is smaller in volume, without loss of information necessary for the analysis task.

Lecture 10

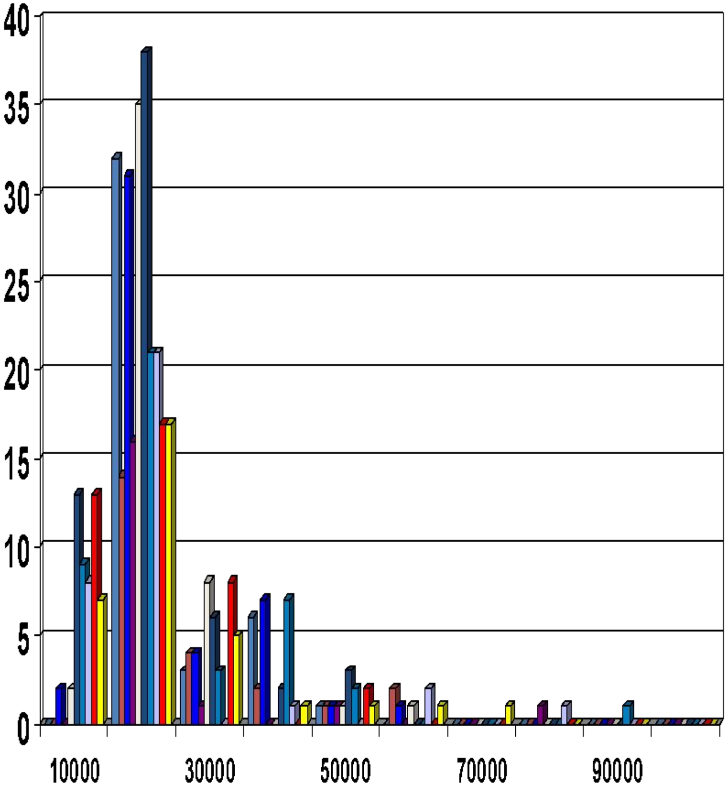
Histograms, Clustering and Sampling, Data Transformation

Non Parametric reduction: Histogram Analysis

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

Partitioning rules:

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



Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms
- Cluster analysis will be studied in depth in Chapter 10

Sampling

- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a representative subset of the data
    - Simple random sampling may have very poor performance in the presence of skew
    - Develop adaptive sampling methods, e.g., stratified sampling:
- Note: Sampling may not reduce database I/Os (page at a time)

Types of Sampling

Simple random sampling

- There is an equal probability of selecting any particular item

Sampling without replacement

Once an object is selected, it is removed from the population

Sampling with replacement

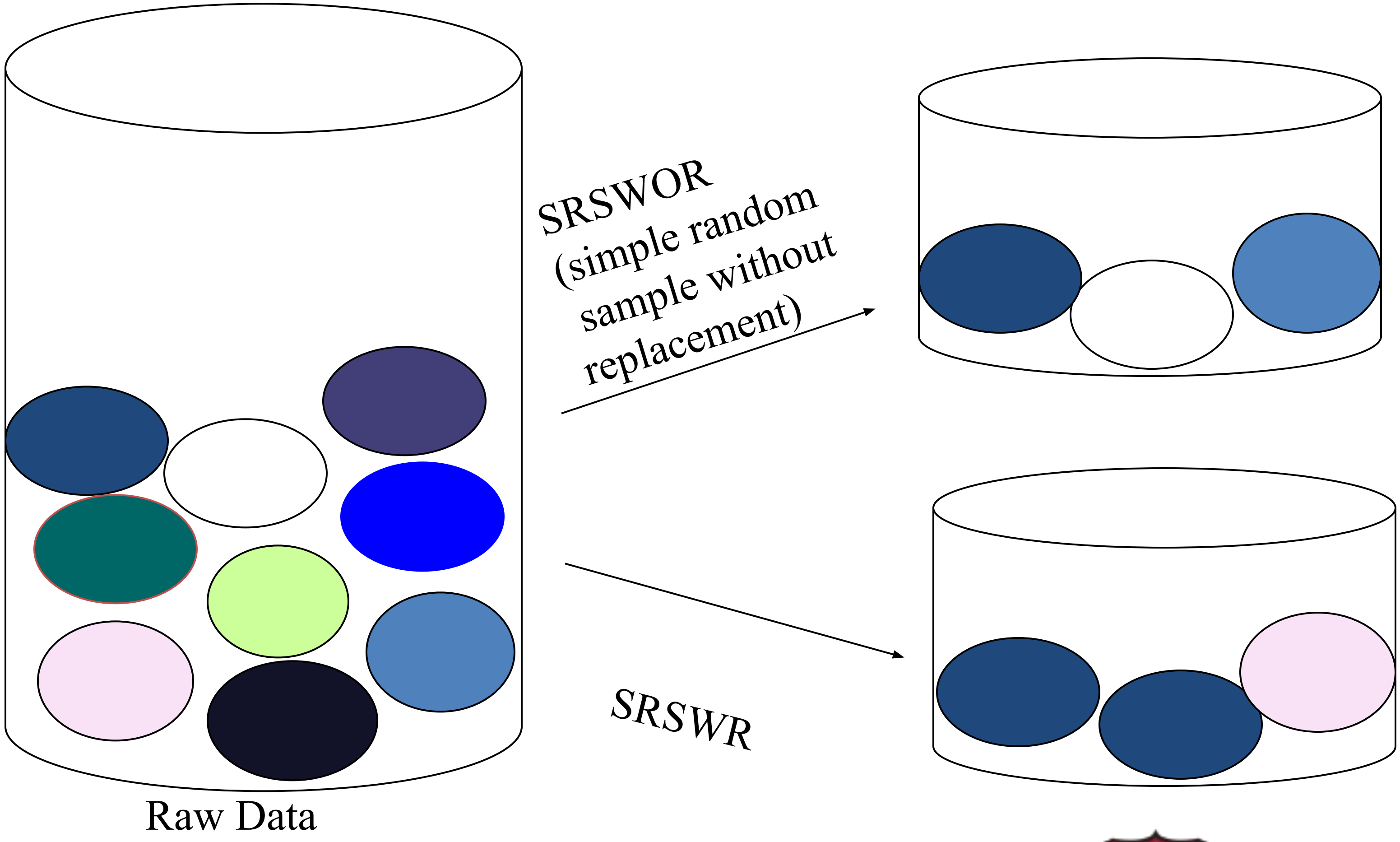
A selected object is not removed from the population

Stratified sampling:

Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)

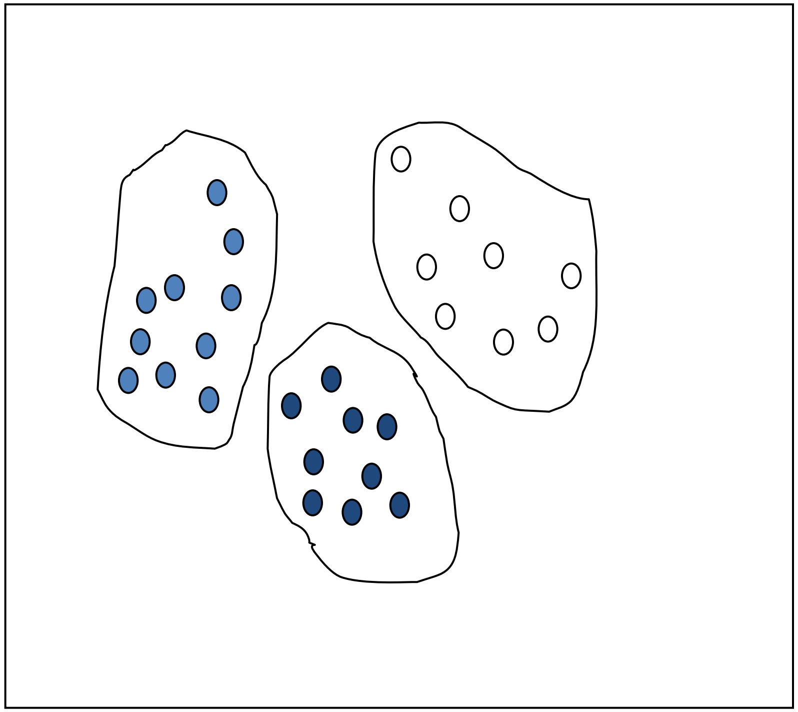
Used in conjunction with skewed data

Sampling: With or without Replacement

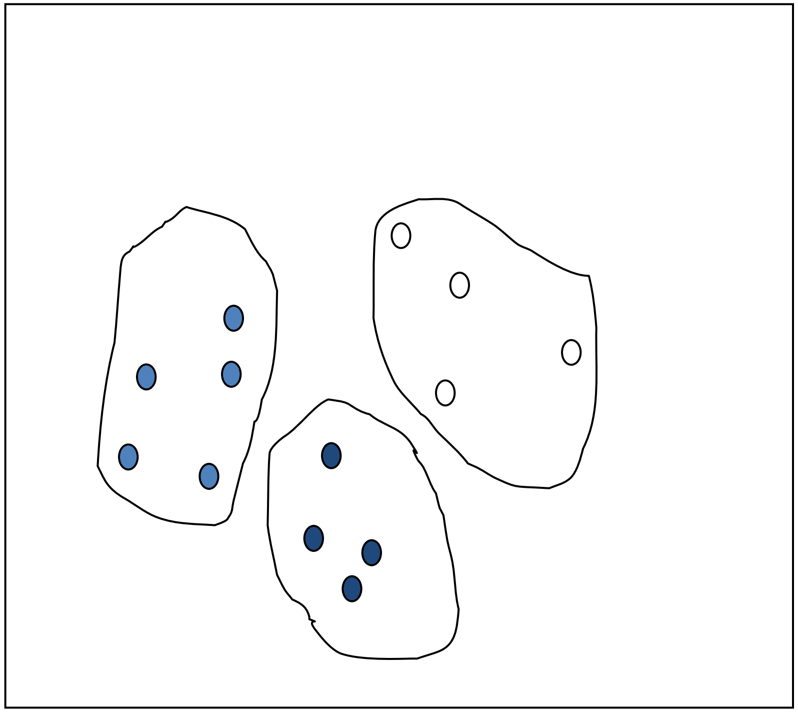


Sampling: Cluster or Stratified Sampling

Raw Data



Cluster/Stratified Sample



Data Cube Aggregation

The lowest level of a data cube (base cuboid)

- The aggregated data for an individual entity of interest
- E.g., a customer in a phone calling data warehouse

Multiple levels of aggregation in data cubes

- Further reduce the size of data to deal with

Reference appropriate levels

- Use the smallest representation which is enough to solve the task

Queries regarding aggregated information should be answered using data cube, when possible

Data Reduction 3: Data Compression

String compression

- There are extensive theories and well-tuned algorithms
- Typically lossless, but only limited manipulation is possible without expansion

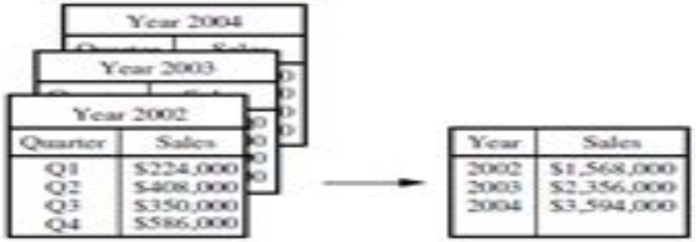
Audio/video compression

- Typically lossy compression, with progressive refinement
- Sometimes small fragments of signal can be reconstructed without reconstructing the whole

Time sequence is not audio

- Typically short and vary slowly with time

Dimensionality and numerosity reduction may also be considered as forms of data compression

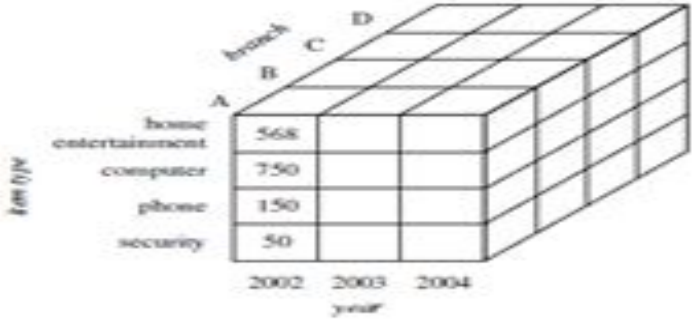


Year 2002	
Quarter	Sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

Year 2004
Year 2003

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

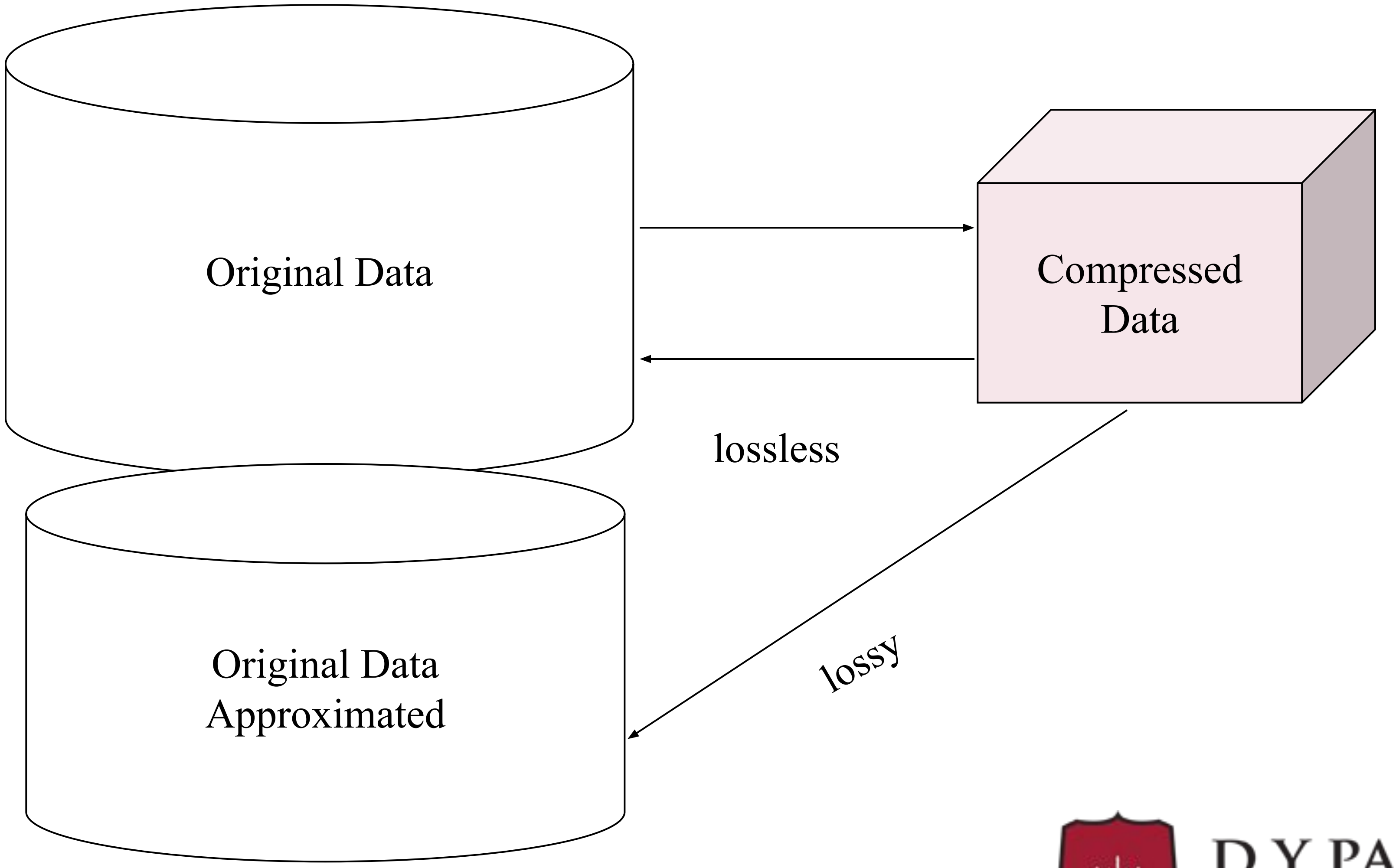
Figure 2.13 Sales data for a given branch of *AllElectronics* for the years 2002 to 2004. On the left, the sales are shown per quarter. On the right, the data are aggregated to provide the annual sales.



item type	2002	2003	2004
home entertainment	568		
computer	750		
phone	150		
security	50		

Figure 2.14 A data cube for sales at *AllElectronics*.

Data Compression



Data Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values such 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

Lecture 11

Normalization, Binning, Histogram analysis and concept hierarchy generation

Normalization

- **Min-max normalization**: to [new_min$_A$, new_max$_A$]

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

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

- **Z-score normalization** (μ: mean, σ: standard deviation):

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

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

- **Normalization by decimal scaling**

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

Discretization

- Data discretization converts a large number of data values into smaller once, so that data evaluation and data management becomes very easy.
- Three types of attributes
  - Nominal—values from an unordered set, e.g., color, profession
  - Ordinal—values from an ordered set, e.g., military or academic rank
  - Numeric—real numbers, e.g., integer or real numbers
- Discretization: Divide the range of a continuous attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Reduce data size by discretization
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification

Data Discretization Methods

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

Simple Discretization: Binning

- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky

Binning Methods for Data Smoothing

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

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

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

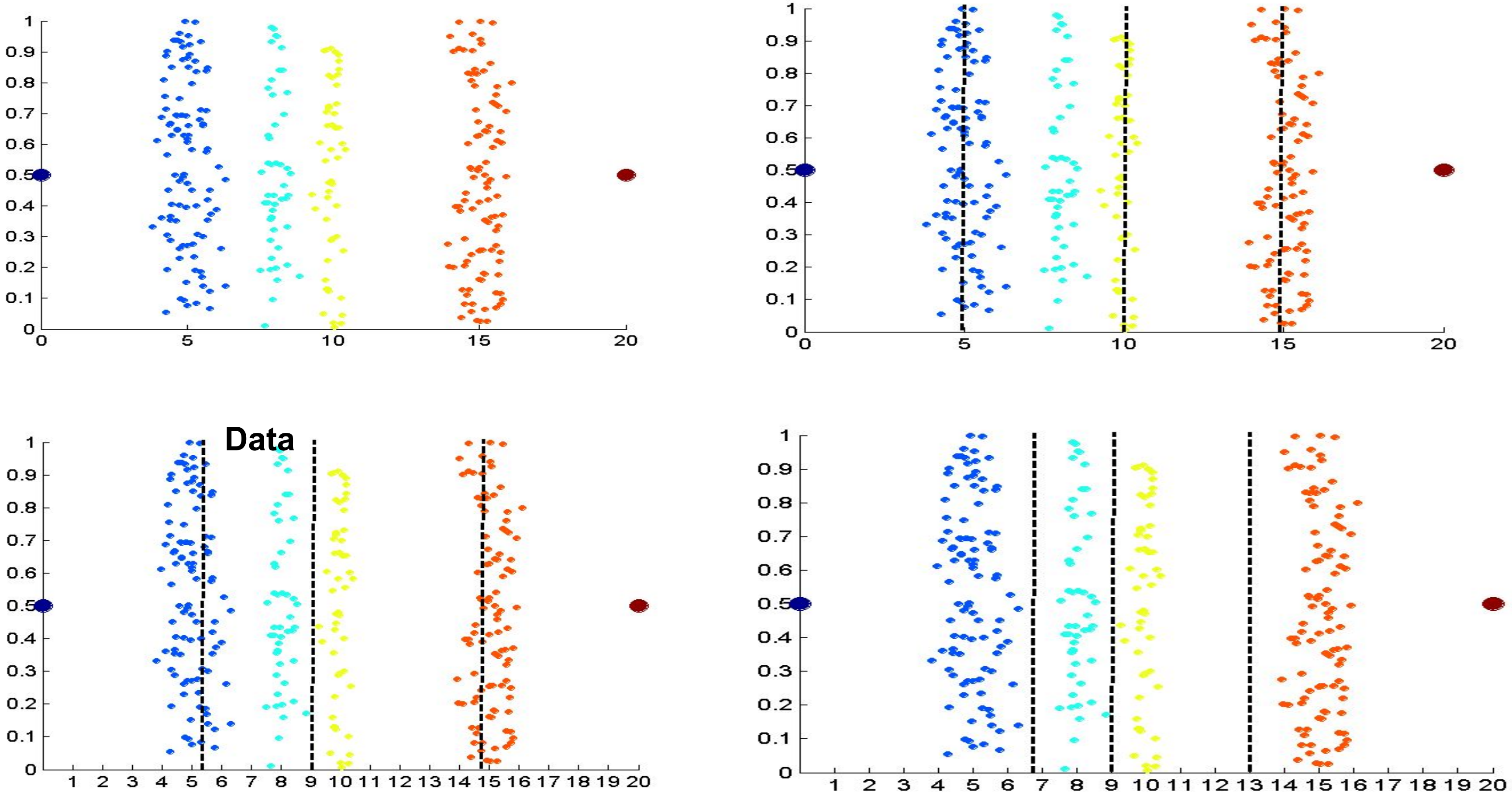
* Smoothing by **bin means**:

- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

Smoothing by **bin boundaries**(Minimum and maximum values are identified and replace the values with closer value of boundary)

- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

Discretization Without Using Class Labels (Binning vs. Clustering)



Data

Equal frequency (binning)

K-means clustering leads to better results

Discretization by Classification & Correlation Analysis

- Classification (e.g., decision tree analysis)
  - Supervised: Given class labels, e.g., cancerous vs. benign
  - Using *entropy* to determine split point (discretization point)
  - Top-down, recursive split
  - Details to be covered in Chapter 7
- Correlation analysis (e.g., Chi-merge: χ^2-based discretization)
  - Supervised: use class information
  - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
  - Merge performed recursively, until a predefined stopping condition

Concept Hierarchy Generation

- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate 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 (such as numeric values for *age*) by higher level concepts (such as *youth, adult*, or *senior*)
- Concept hierarchies can be explicitly specified by domain experts and/or data warehouse designers
- Concept hierarchy can be automatically formed for both numeric and nominal data. For numeric data, use discretization methods shown.

Concept Hierarchy Generation for Nominal Data

Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts

street < *city* < *state* < *country*

Specification of a hierarchy for a set of values by explicit data grouping

{Urbana, Champaign, Chicago} < Illinois

Specification of only a partial set of attributes

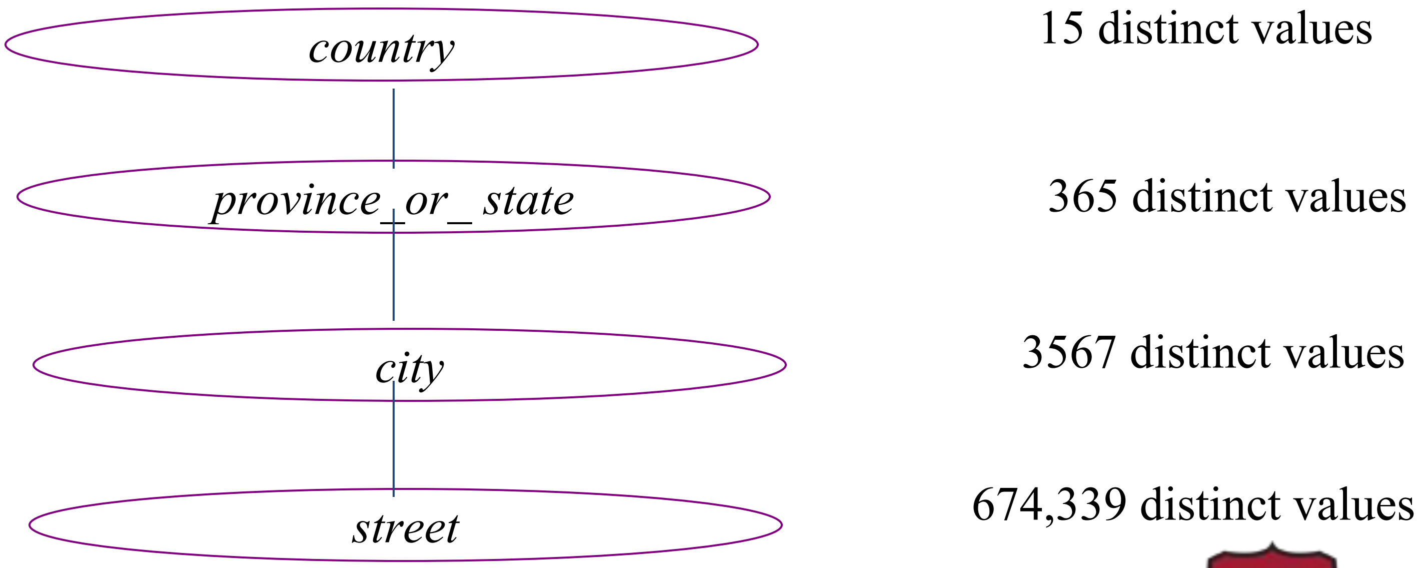
E.g., only *street* < *city*, not others

Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values

E.g., for a set of attributes: {*street, city, state, country*}

Automatic Concept Hierarchy Generation

- Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy
  - Exceptions, e.g., weekday, month, quarter, year



country — 15 distinct values

province_or_ state — 365 distinct values

city — 3567 distinct values

street — 674,339 distinct values

Thank You