Data Mining and Data Warehousing

Data Preprocessing

(based on notes by Jiawei Han and Micheline Kamber)

Agenda

- Why data preprocessing?
- Data cleaning
- Data integration and transformation
- Data reduction
- Summary

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  - noisy: containing errors or outliers
  - inconsistent: containing discrepancies in codes or names
- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data
- A multi-dimensional measure of data quality:
  - A well-accepted multi-dimensional view:
    - accuracy, completeness, consistency, timeliness, believability, value added, interpretability, accessibility
  - Broad categories:
    - intrinsic, contextual, representational, and accessibility.

Why Data Preprocessing?

- Data have quality if they satisfy the requirements of the intended use.
- 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?

Why Data Preprocessing?

Accuracy: correct or wrong, accurate or not

• There are many possible reasons for inaccurate data.

– Human or computer errors occurring at data entry.

– Users may purposely submit incorrect data values for mandatory fields when they do not wish to submit personal information such as choosing the default value "January 1" displayed for birthday.

– Incorrect data may also result from inconsistencies in naming conventions or data codes, or inconsistent formats for input fields (e.g., date).

Why Data Preprocessing?

Completeness: not recorded, unavailable, …

• Attributes of interest may not always be available

• Data may not be included simply because they were not considered important at the time of entry.

• Relevant data may not be recorded due to a misunderstanding.

• Missing data, tuples with missing values for some attributes, may need to be inferred

Consistency: some modified but some not, dangling, …

• Containing discrepancies in the department codes used to categorize items.

• Inconsistencies in data codes, or inconsistent formats for input fields (e.g., date).

Why Data Preprocessing?

Timeliness: timely update?

- Is the data is timely updated?

Believability: how trustable the data are correct?

- How much the data are trusted by users?
- The past errors can effect the trustability of the data.

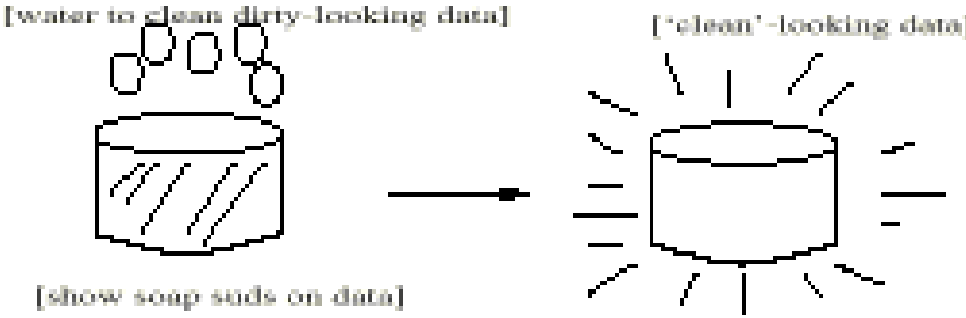
Interpretability: how easily the data can be understood?

Major Tasks in Data Preprocessing

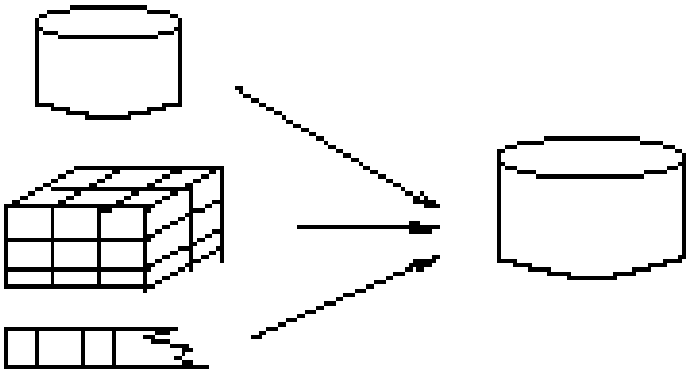
- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, files, or notes
- Data transformation
  - Normalization (scaling to a specific range)
  - Aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
  - Data discretization: with particular importance, especially for numerical data
  - Data aggregation, dimensionality reduction, data compression,generalization

Forms of data preprocessing

Data Cleaning



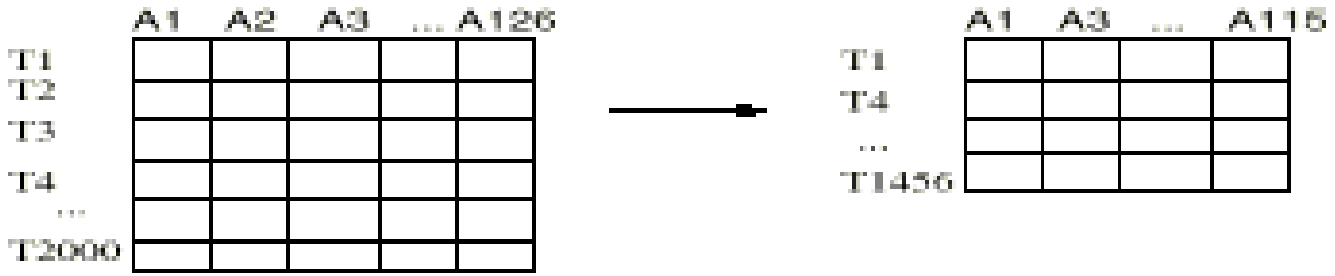
Data Integration



Data Transformation

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

Data Reduction



Agenda

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Summary

Data Cleaning

- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data

Missing Data

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

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (assuming the task is classification—not effective in certain cases)
- Fill in the missing value manually: tedious + infeasible?
- Use a global constant to fill in the missing value: e.g., "unknown", a new class?!
- Use the attribute mean to fill in the missing value
- Use the attribute mean for all samples of the same class to fill in the missing value: smarter
- Use the most probable value to fill in the missing value: inference-based such as regression, Bayesian formula, decision tree

Noisy Data

- Q: What is noise?
- A: Random error in a measured variable.
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which requires data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

How to Handle Noisy Data?

- Binning method:
  - first sort data and partition into (equi-depth) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
  - used also for discretization
- Clustering
  - detect and remove outliers
- Semi-automated method: combined computer and human inspection
  - detect suspicious values and check manually
- Regression
  - smooth by fitting the data into regression functions

Simple Discretization Methods: Binning

- Equal-width (distance) partitioning:
  - It 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:
  - It divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky.

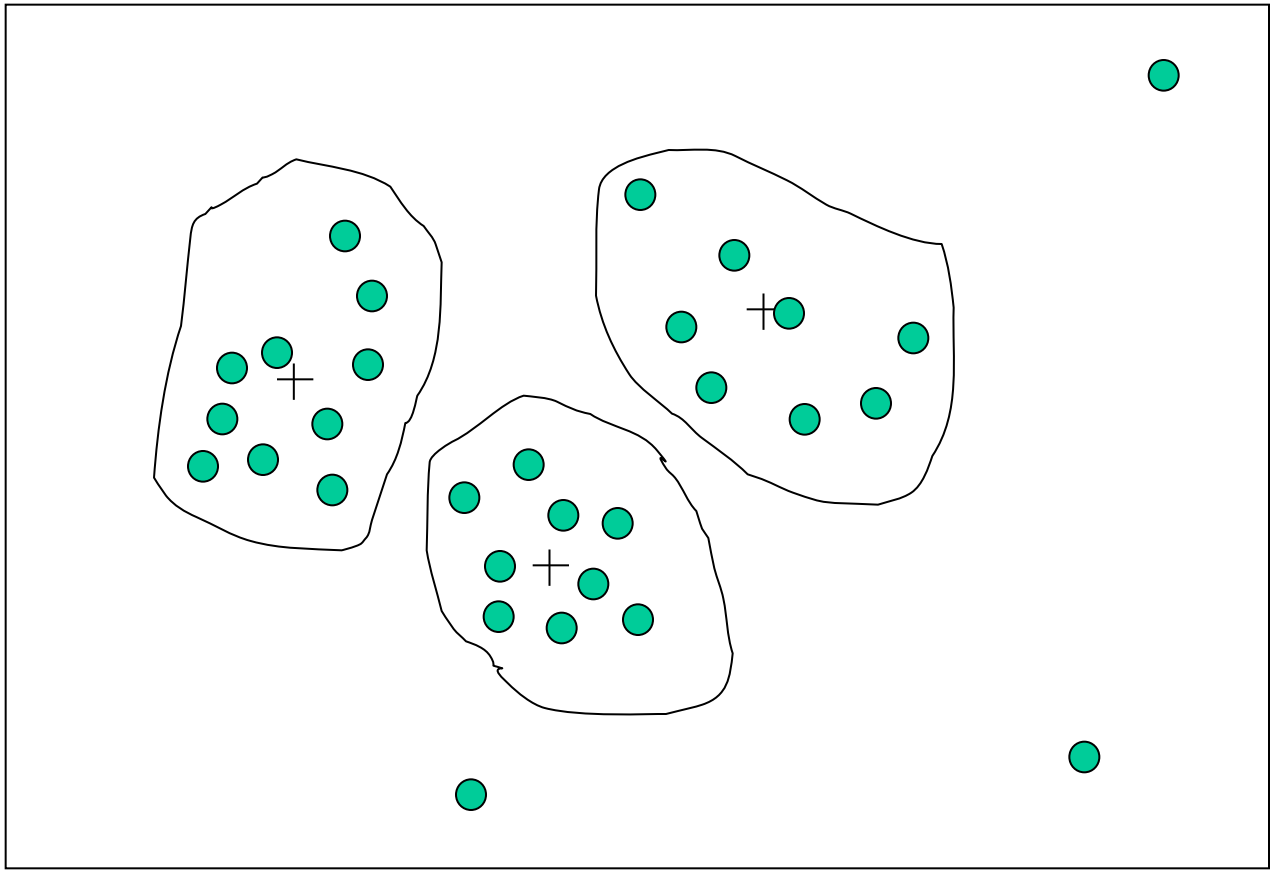
Binning Methods for Data Smoothing

* Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
* Partition into (equi-depth) bins:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Smoothing by bin means:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Smoothing by bin boundaries:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

Binning Methods for Data Smoothing

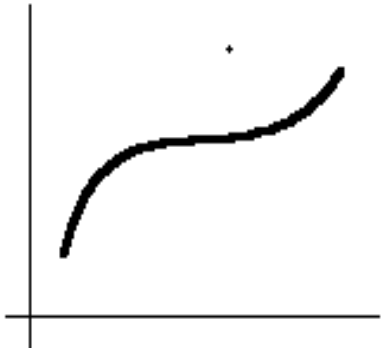
8 16, 9, 15, 21, 21, 24, 30, 26, 27, 30, 34

Cluster Analysis

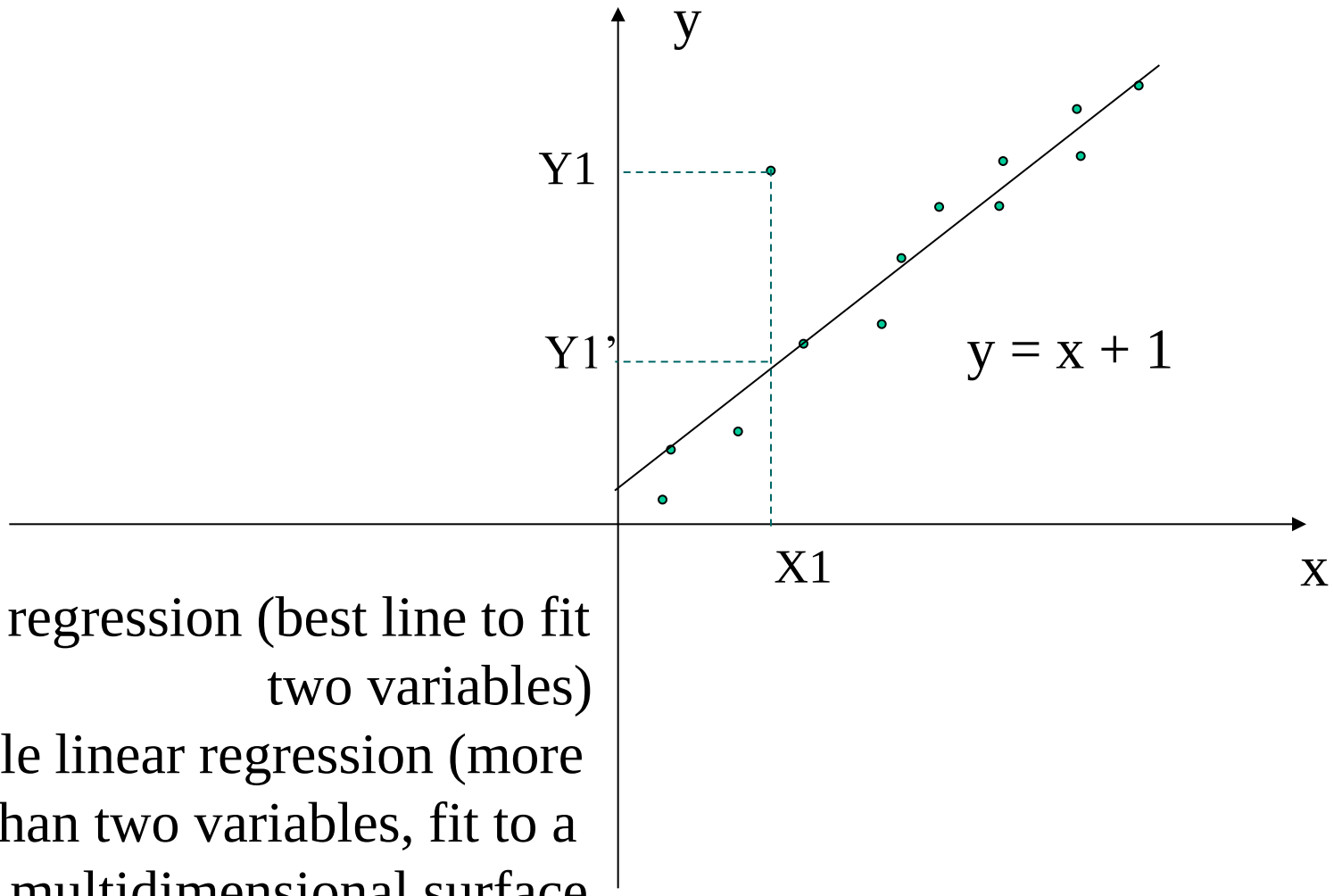


Outlier Removal

- Data points inconsistent with the majority of data
- Different outliers
  - Valid: CEO's salary,
  - Noisy: One's age = 200, widely deviated points
- Removal methods
  - Clustering
  - Curve-fitting
  - Hypothesis-testing with a given model



Regression



- Linear regression (best line to fit two variables)
- Multiple linear regression (more than two variables, fit to a multidimensional surface

How to Handle Inconsistent Data?

- Manual correction using external references
- Semi-automatic using various tools
  - To detect violation of known functional dependencies and data constraints
  - To correct redundant data

Agenda

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Summary

Data Integration

- Data integration:
  - combines data from multiple sources into a coherent store
- Schema integration
  - integrate metadata from different sources
  - Entity identification problem: identify real world entities from multiple data sources, e.g., A.cust-id ≡ B.cust-#
- 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, different currency

Handling Redundant Data in Data Integration

- Redundant data occur often when integrating multiple DBs
  - The same attribute may have different names in different databases
  - One attribute may be a "derived" attribute in another table, e.g., annual revenue
- Redundant data may be able to be detected by correlational analysis

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

- Careful integration can help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Covariance:Example

- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5, 10), (4, 11), (7, 14).
- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?

– $E(A) = (2 + 3 + 5 + 4 + 7)/ 5 = 21/5 = 4.2$

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

– $Cov(A,B) = (2\times5+3\times8+5\times10+4\times11+7\times14)/5 - 4.2 \times 9.6 = 4.88$

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

Correlation Analysis (for Numeric Data)

- **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 one of the attributes tends to be above its expected value when the other attribute is below its expected value,
- **Independence**: $Cov_{A,B} = 0$
  - If A and B are independent, $Cov_{A,B} = 0$.
  - But the converse is not true:
- Some pairs of random variables may have a covariance of 0 but they are not independent.

– Covariance indicates linear relationship (not non-linear relationship)

- Only under some additional assumptions (the data follow multivariate normal distributions) does a covariance of 0 imply independence.

Correlation Test (for Nominal Data)

Chi-Square Test

- For **nominal data**, a *correlation relationship* between two attributes, A and B, can be discovered by a **χ^2 (chi-square) test**.
- Suppose A has c distinct values, $a_1 \ldots a_c$. B has r distinct values, $b_1 \ldots b_r$.

χ^2 (chi-square) Test:

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

where o_{ij} is the ***observed frequency*** (i.e., actual count) of the **joint event (A_i, B_j)** and e_{ij} is the ***expected frequency*** of (A_i, B_j), which can be computed as

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

where **n** is the number of data tuples, count(A=a_i) is the number of tuples having value a_i for A, and count(B = b_j) is the number of tuples having value b_j for B.

- The larger the χ^2 value, the more likely the variables are related.
  - The cells that contribute the most to the χ^2 value are those whose actual count is very different from the expected count.

Chi-Square Calculation: An Example

- **Contingency Table** for two attributes **LikeScienceFiction** and **PlayChess**

	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

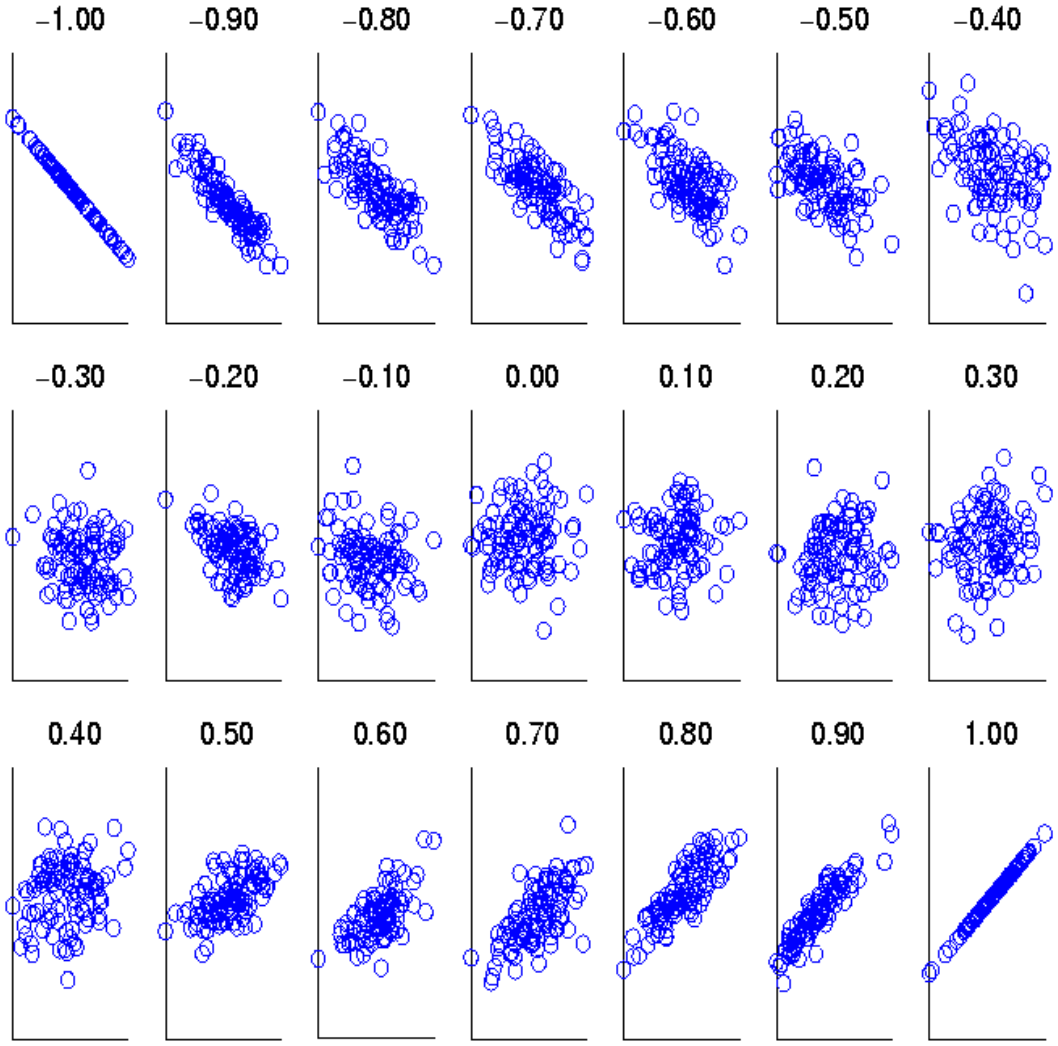
- Numbers in cells are ***observed frequencies*** (*numbers in parenthesis are* ***expected counts*** *calculated based on the data distribution in the two categories*).

$$e_{LSF,PC} = \text{count(LSF)*count(PC)} / n = 300*450 / 1500 = 90$$

- **χ^2 (chi-square) calculation**

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

Visually Evaluating Correlation



Scatter plots showing the similarity from –1 to 1.

Data Transformation

- In **data transformation**, the data are transformed or consolidated into forms appropriate for mining.

- In 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.

Some of data transformation strategies:

- **Normalization**: The attribute data are scaled so as to fall within a small specified range.
- **Discretization**: A numeric attribute is replaced by a categorical attribute.
- Other data transformation strategies

– **Smoothing**: Remove noise from data. Smoothing is also a data cleaning method.

– **Attribute/feature construction**: New attributes constructed from the given ones. Attribute construction is also a data reduction medhod.

– **Aggregation**: Summarization, data cube construction. Aggregation is also a data reduction method.

Data Transformation: Normalization

Particularly useful for classification (NNs, distance measurements, nn classification, etc)

- min-max normalization

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

- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- normalization by decimal scaling

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

Min-Max Normalization

- **Min-max normalization** performs a linear transformation on the original data.
- Suppose that $\mathbf{min_A}$ and $\mathbf{max_A}$ are minimum and maximum values of an attribute A.
- **Min-max normalization** maps a value, v_i of an attribute A to v_i' in the range $[\mathbf{new\_min_A}, \mathbf{new\_max_A}]$ by computing:

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

- Min-max normalization preserves the relationships among the original data values.
- We can standardize the range of all the numerical attributes to [0,1] by applying ***min-max normalization*** with *newmin=0* and *newmax=1* to all the numeric attributes.

Min-Max Normalization: Example

- Suppose that the range of the attribute *income* is \$12,000 to \$98,000. We want to normalize *income* to range [0.0, 1.0].
- Then \$73,000 is mapped to

$$\text{newvalue}(73000) = \frac{73000 - 12000}{98000 - 12000}(1.0 - 0.0) + 0 = 0.716$$

- Suppose that the range of the attribute *income* is \$12,000 to \$98,000. We want to normalize *income* to range[1.0, 5.0].
- Then \$73,000 is mapped to

$$\text{newvalue}(73000) = \frac{73000 - 12000}{98000 - 12000}(5.0 - 1.0) + 1.0 = 3.864$$

Z-score Normalization

- In **z-score normalization** (or ***zero-mean normalization***), the values for an attribute A are normalized based on the *mean* and *standard deviation* of A.
- A value v_i of attribute A is normalized to v_i' by computing

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

where and $\bar{A}$ and σ_A are the mean and standard deviation of attribute A.

- **z-score normalization** is useful when the actual minimum and maximum of an attribute are unknown.
- Suppose that the mean and standard deviation of the values for the attribute *income* are \$54,000 and \$16,000. With z-score normalization a value of \$73,600 for *income:*

$$\text{newvalue}(73600) = \frac{73600 - 54000}{16000} = 1.225$$

Normalization by Decimal Scaling

- **Normalization by decimal scaling** normalizes by moving the decimal point of values of attribute A.
- The number of decimal points moved depends on the maximum absolute value of A.
- A value $\boldsymbol{v_i}$ of attribute A is normalized to $\boldsymbol{v_i'}$ by computing

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

where $\boldsymbol{j}$ is the smallest integer such that $max(|v_i'|) < 1$.

Example:

- Suppose that the recorded values of A range from -986 to 917.
- The maximum absolute value of A is 986.
- To normalize by decimal scaling, we therefore divide each value by 1000 so that -986 normalizes to -0.986 and 917 normalizes to 0.917.

Discretization

Discretization: To transform a numeric (continuous) attribute into a categorical attribute.

- Some data mining algorithms require that data be in the form of categorical attributes.
- In discretization:
  - The range of a continuous attribute is divided into intervals.
  - Then, interval labels can be used to replace actual data values to obtain a categorical attribute.

Simple Discretization Example: *income* attribute is discretized into a categorical attribute.

- Target categories (low, medium, high).
- Calculate average *income:* AVG.
  - If income> 2* AVG, new_income_value = “high”.
  - If income < 0.5* AVG, new_income_value = “low”.
  - Otherwise, new_income_value = “medium”.

Discretization Methods

A basic distinction between discretization methods for classification is whether class information is used (**supervised**) or not (**unsupervised**).

Unsupervised Discretization: If class information is not used, then relatively simple approaches are common.

– Binning

– Clustering analysis

Supervised Discretization:

– Classification (e.g., decision tree analysis)

– Correlation (e.g., chi-sqauare) analysis

Discretization by Binning: Example equal-width approach

- Suppose a group of 12 values of *price* attribute has been sorted as follows:

price	5	10	11	13	15	35	50	55	72	89	204	215

equal-width partitioning: The width of each interval is (215-5)/3 = 70.

- Partition them into ***three bins***

bin1	5, 10, 11, 13, 15, 35, 50, 55, 72
bin2	89
bin3	204, 215

- Replace each value with its bin label to discretize.

price	5	10	11	13	15	35	50	55	72	89	204	215
categorical attr.	1	1	1	1	1	1	1	1	1	2	3	3

Discretization by Binning: Example equal-frequency approach

- Suppose a group of 12 values of *price* attribute has been sorted as follows:

price	5	10	11	13	15	35	50	55	72	89	204	215

equal-frequency partitioning:

- Partition them into ***three bins:*** each interval contains 4 values

bin1	5, 10, 11, 13
bin2	15, 35, 50, 55
bin3	72, 89, 204, 215

- Replace each value with its bin label to discretize.

price	5	10	11	13	15	35	50	55	72	89	204	215
categorical attr.	1	1	1	1	2	2	2	2	3	3	3	3

Discretization by Clustering

- A **clustering** algorithm can be applied to discretize a numeric attribute.

– The values of the attribute are partitioned into clusters by a clustering algorithm.

– Each value in a cluster is replaced by the label of that cluster to discretize.

- Clustering takes the distribution and closeness of attribute values into consideration, and therefore is able to produce high-quality discretization results.

Discretization by Clustering: Example

Simple clustering: ***partition data from biggest gaps***.

- Example: partition data along 2 biggest gaps into three bins.

bin1	5, 10, 11, 13, 15
bin2	35, 50, 55, 72, 89
bin3	204, 215

- Replace each value with its bin label to discretize.

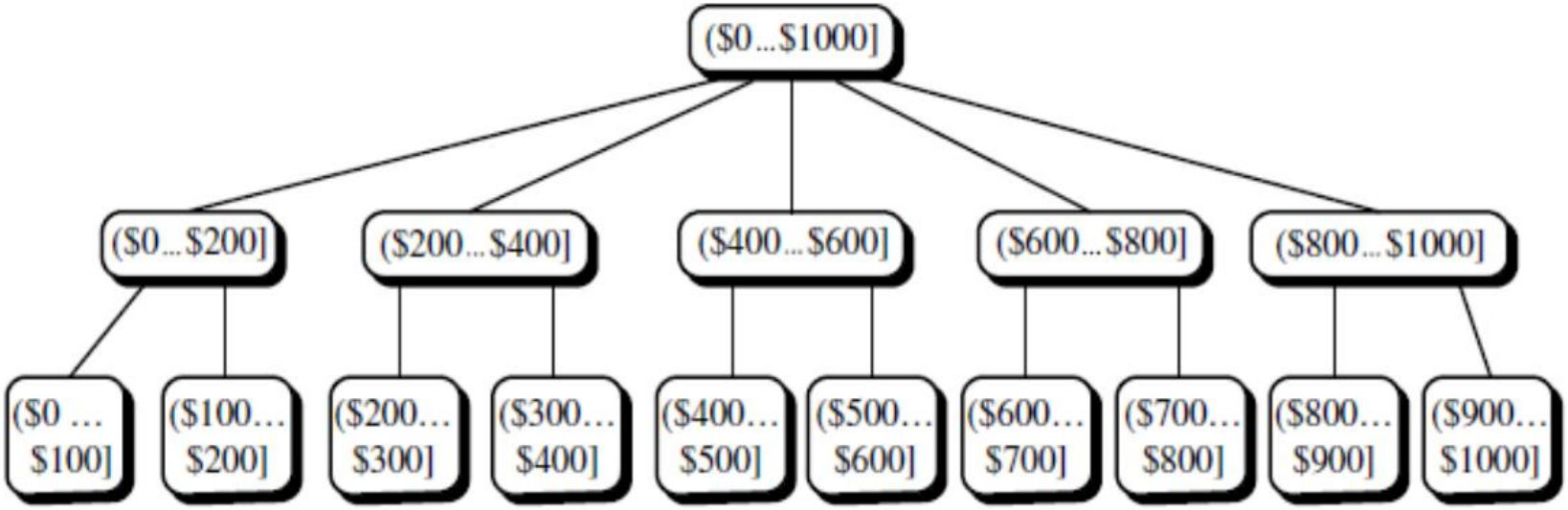
price	5	10	11	13	15	35	50	55	72	89	204	215
categorical attr.	1	1	1	1	1	2	2	2	2	2	3	3

Concept Hierarchy

- A **concept hierarchy** defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.
- Many concept hierarchies are implicit within the database schema.
- Concept hierarchies may be provided manually by system users or may be automatically generated based on statistical analysis of the data distribution.
- A concept hierarchy that is a total or partial order among attributes.
- Concept hierarchies may also be defined by discretizing or grouping values for a given dimension.

Concept Hierarchy

- A concept hierarchy for the attribute *price*

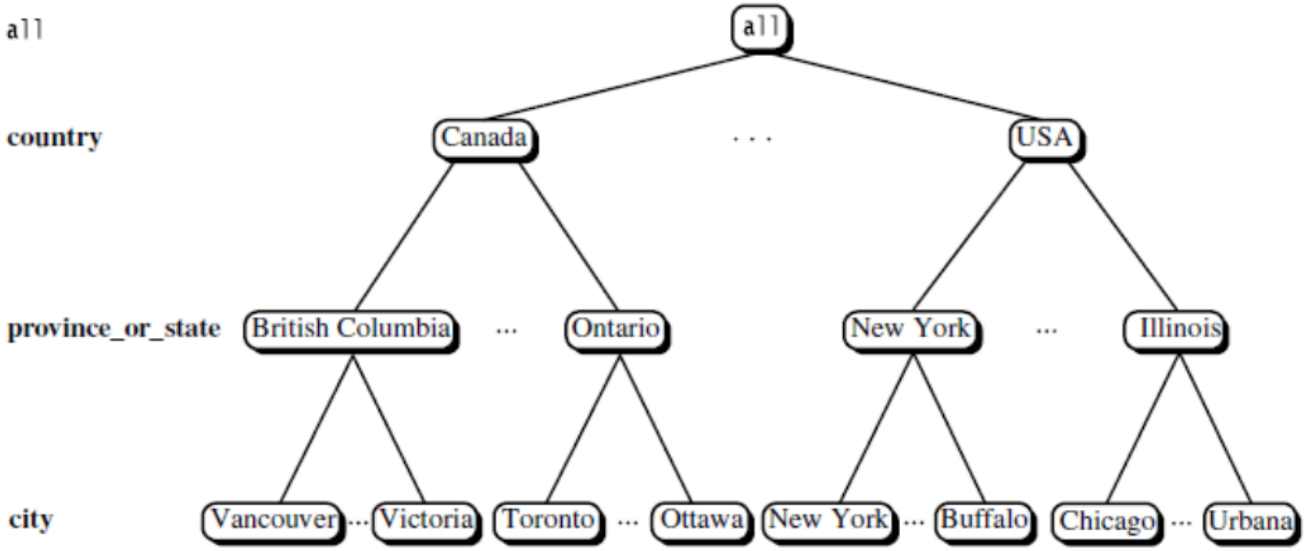


- A concept hierarchy for a numeric attribute can be constructed automatically or it can be created by the user.
- A concept hierarchy can be used for discretization.

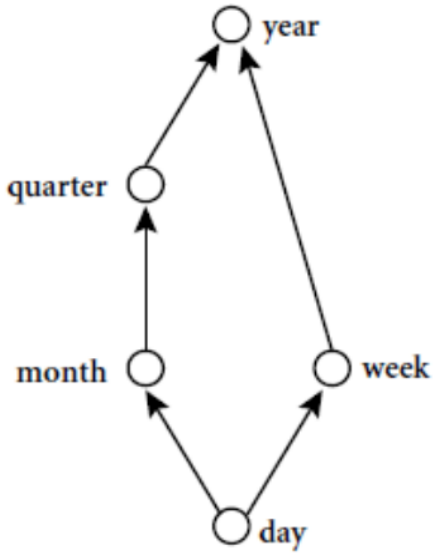
Concept Hierarchy

- Concept hierarchies for nominal attributes can also be created.

location attribute



time attribute



Agenda

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Summary

Data Reduction

- 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

- Data reduction strategies
  - –Dimensionality reduction: e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection, feature creation
  - — Numerosity reduction:
    - Data cube aggregation
    - Sampling
    - Clustering, …

Data Reduction: Dimensionality Reduction

•Curse of dimensionality

– When dimensionality increases, data becomes increasingly sparse.

– 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.

Data Reduction: Dimensionality Reduction

- Dimensionality reduction techniques

– Wavelet transforms

– Principal Component Analysis

– Supervised and nonlinear techniques (e.g., feature selection)

Attribute Subset Selection

Data sets for analysis may contain hundreds of attributes, many of which may be **irrelevant** to the mining task or **redundant**.

• **Redundant Attributes** duplicate much or all of the information contained in one or more other attributes.

– price of a product and the sales tax paid contain much of the same information.

• **Irrelevant Attributes** contain almost no useful information for the data mining task.

– students' IDs are irrelevant to predict students' grade.

Attribute Subset Selection

Attribute Subset Selection reduces the data set size by removing irrelevant or redundant attribute.

– The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.

– Attribute subset selection reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand.

Attribute Subset Selection

Attribute Subset Selection Techniques:

• Brute-force approach:

– Try all possible feature subsets as input to data mining algorithm.

• Embedded approaches:

– Feature selection occurs naturally as part of the data mining algorithm.

• Filter approaches:

– Features are selected before data mining algorithm is run.

Heuristic Search in Attribute Subset Selection

• How can we find a ‘good’ subset of the original attributes?

• There are 2^n possible attribute combinations of n attributes

• Typical heuristic attribute selection methods:

– Best single attribute under the attribute independence assumption: choose by significance tests (eg. information gain measure)

– Best step-wise feature selection:

  • The best single-attribute is picked first

  • Then next best attribute condition to the first, ...

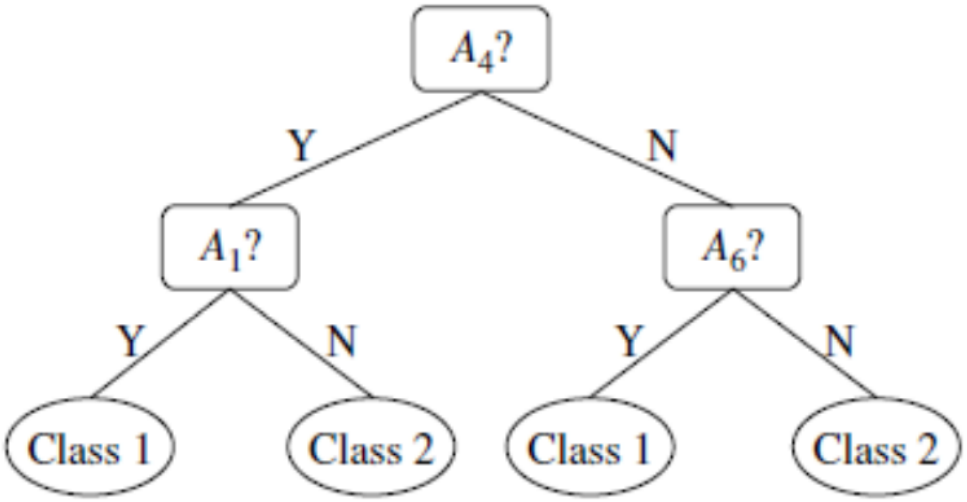
– Step-wise attribute elimination:

  • Repeatedly eliminate the worst attribute

– Best combined attribute selection and elimination

– Decision tree induction

Heuristic Search in Attribute Subset Selection

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: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  => Reduced attribute set: $\{A_1, A_4, A_6\}$

Attribute Creation (Feature Generation)

• Create new attributes that can capture the important information in a data set much more efficiently than the original attributes

Three general methodologies:

• Attribute Extraction

– The creation of a new set of attributes from the original data is known as attribute extraction.

– Image processing: Extracting high-level attributes from low-level attributes (pixels)

• Mapping Data to New Space

– Wavelet transform, Principal Component Analysis (PCA), …

Attribute Creation (Feature Generation)

• Attribute Construction

– combining attributes

– new attributes are constructed from given attributes in order to help improve accuracy and understanding of structure in high-dimensional data.

• For example, add the attribute area based on the attributes height and width.

Wavelet Transformation

Discrete wavelet transform is a dimension reduction technique.

• The **discrete wavelet transform (DWT)** is a linear signal processing technique that, when applied to a n-dimensional data vector X, transforms it to a numerically different n-dimensional vector, X', of **wavelet coefficie**nts.

• **Compressed approximation**: store only a small fraction of the strongest of the wavelet coefficients.

– all wavelet coefficients larger than some user-specified threshold can be retained.

– All other coefficients are set to 0.

– The resulting data representation is therefore very sparse, so that operations that can take advantage of data sparsity are computationally very fast if performed in wavelet space.

Wavelet Transformation

- Wavelets: A math tool for space-efficient hierarchical decomposition of functions
- 8-dimensional data vector S = [2, 2, 0, 2, 3, 5, 4, 4] can be transformed to 8-dimensional wavelet coefficient vector $S_{\wedge} = [2^{3}/_{4}, -1^{1}/_{4}, {}^{1}/_{2}, 0, 0, -1, -1, 0]$
- Compression: many small detail coefficients can be replaced by 0's, and only the significant coefficients are retained

Resolution	Averages	Detail Coefficients
8	[2, 2, 0, 2, 3, 5, 4, 4]	
4	[2, 1, 4, 4]	[0, −1, −1, 0]
2	$[1\frac{1}{2}, 4]$	$[\frac{1}{2}, 0]$
1	$[2\frac{3}{4}]$	$[-1\frac{1}{4}]$

Principal Component Analysis (PCA)

Suppose that the data to be reduced consist of tuples described by n dimensions.

• Principal components analysis (PCA) searches for k n-dimensional orthogonal vectors that can best be used to represent data, $k \leq n$.

• The original data are thus projected onto a much smaller space, resulting in dimensionality reduction.

• Unlike attribute subset selection, which reduces the attribute set size by retaining a subset of the initial set of attributes, PCA "combines" the essence of attributes by creating an alternative, smaller set of variables.

• The initial data can then be projected onto this smaller set.

• PCA often reveals relationships that were not previously suspected and thereby allows interpretations that would not ordinarily result.

Principal Component Analysis (PCA)

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

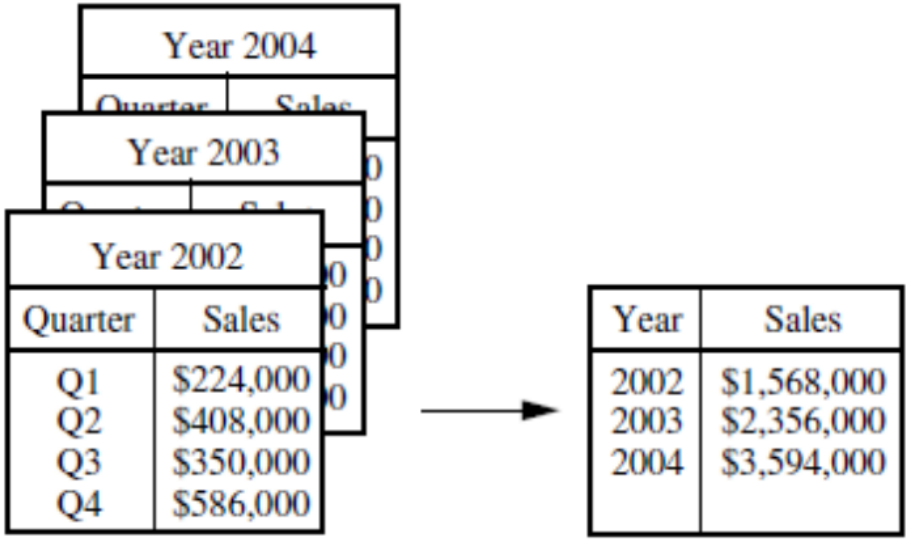
– The principal components are sorted in order of decreasing "significance" or strength

– 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)

$$A = \begin{bmatrix} 2 & 4 & 3 & 5 \\ 1 & 3 & 2 & 4 \\ 5 & 7 & 6 & 8 \\ 3 & 5 & 4 & 6 \\ 4 & 6 & 5 & 7 \end{bmatrix}$$

Numerosity Reduction
Data Cube Aggregation

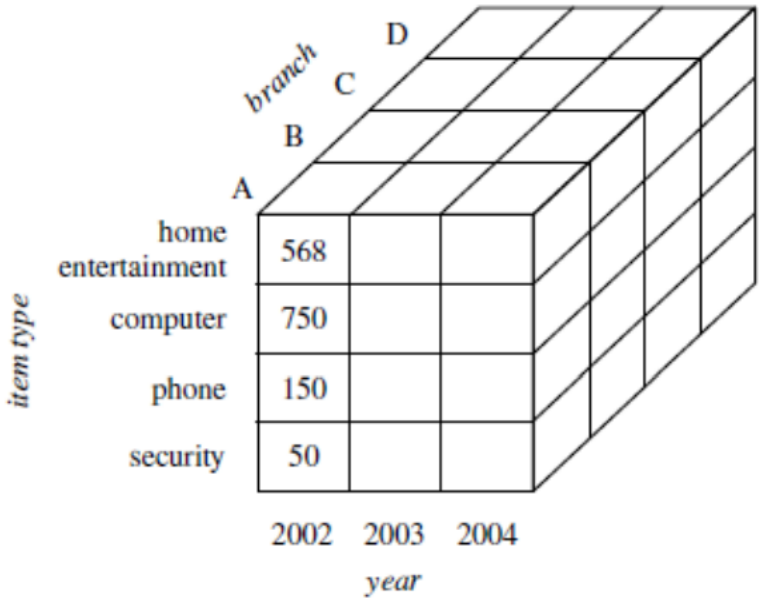
- If the data has sales per quarters, and we are interested in annual sales
- the data can be aggregated so that the resulting data summarize the total sales per year instead of per quarter.
- The resulting data set is smaller in volume, without loss of information necessary for the analysis task.



sales per quarter are aggregated to provide the annual sales.

Numerosity Reduction
Data Cube Aggregation

- **Data cubes** store multidimensional aggregated information.
- Data cubes provide fast access to precomputed, summarized data, thereby benefiting on-line analytical processing as well as data mining.
- The following data cube for multidimensional analysis of sales data with respect to annual sales per item type.
  - Each cell holds an aggregate data value, corresponding to the data point in multidimensional space.



Numerosity Reduction: Sampling

• **Sampling** is the main technique employed for data selection.

– It is often used for both the preliminary investigation of the data and the final data analysis.

• Statisticians sample because obtaining the entire set of data of interest is too expensive or time consuming.

• **Sampling is used in data mining** because processing the entire set of data of interest is too expensive or time consuming.

• The **key principle for effective sampling** is the following:

– using a sample will work almost as well as using the entire data sets, if the sample is representative

– A sample is representative if it has approximately the same property (of interest) as the original set of data

Numerosity Reduction: Sampling

Simple Random Sampling

– There is an equal probability of selecting any particular item

Sampling without replacement

• As each item is selected, it is removed from the population

Sampling with replacement

• Objects are not removed from the population as they are selected for the sample.

Numerosity Reduction: Sampling

Stratified Sampling

– Split the data into several partitions; then draw random samples from each partition.

• In the simplest version, equal numbers of objects are drawn from each group even though the groups are of different sizes.

• In another variation, the number of objects drawn from each group is proportional to the size of that group.

Numerosity Reduction: Sampling Example

- Simple Random Sampling — Sample Size=7
  - Sampling without replacement (SRS_WOR)
  - Sampling with replacement (SRS_WR)
- Stratified sampling (STRAT)

	age
1	youth
2	youth
3	youth
4	youth
5	middle-aged
6	middle-aged
7	middle-aged
8	middle-aged
9	middle-aged
10	middle-aged
11	middle-aged
12	middle-aged
13	senior
14	senior

SRS_WOR

	age
1	youth
2	youth
3	youth
5	middle-aged
8	middle-aged
10	middle-aged
12	middle-aged

SRS_WR

	age
1	youth
4	youth
6	middle-aged
6	middle-aged
9	middle-aged
10	middle-aged
14	senior

STRAT

	age
1	youth
4	youth
6	middle-aged
7	middle-aged
9	middle-aged
11	middle-aged
13	senior

Data Preprocessing 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 transformation and data discretization**
  - Normalization
  - Concept hierarchy generation
- **Data reduction**
  - Dimensionality reduction
  - Numerosity reduction