Data Preprocessing

- Data Quality and Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Transformation and Data Discretization
- Data Reduction

- **Data Quality and Major Tasks in Data Preprocessing**
- Data Cleaning
- Data Integration
- Data Transformation and Data Discretization
- Data Reduction

Data Preprocessing

- Today’s real-world databases are highly susceptible to noisy, missing, and inconsistent data due to their typically huge size and their likely origin from multiple, heterogenous sources.
- Low-quality data will lead to low-quality mining results.
- “How can the data be preprocessed in order to help improve the quality of the data and, consequently, of the mining results?
- How can the data be preprocessed so as to improve the efficiency and ease of the mining process?”
- Data preprocessing techniques, when applied before mining, can substantially improve the overall quality of the patterns mined and/or the time required for the actual mining.

Data Quality

- What kinds of data quality problems?
- How can we detect problems with the data?
- What can we do about these problems?

- Examples of data quality problems:
  - Noise and outliers
    - Noise: random error or variance in a measured variable
    - Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set
  - Missing values
  - Duplicate data

Data Quality
Missing Values and Duplicate Data

- Reasons for missing values
  - Information is not collected
    (e.g., people decline to give their age and weight)
  - Attributes may not be applicable to all cases
    (e.g., annual income is not applicable to children)

- Handling missing values
  - Eliminate Data Objects
  - Estimate Missing Values
  - Ignore the Missing Value During Analysis
  - Replace with all possible values (weighted by their probabilities)

- Data set may include data objects that are **duplicates**, or almost duplicates of one another
  - Major issue when merging data from heterogenous sources

Data Quality: Why Preprocess the Data?

- **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?

Data Quality: Why Preprocess the Data?

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).

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.

Data Quality: Why Preprocess the Data?

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*).

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.

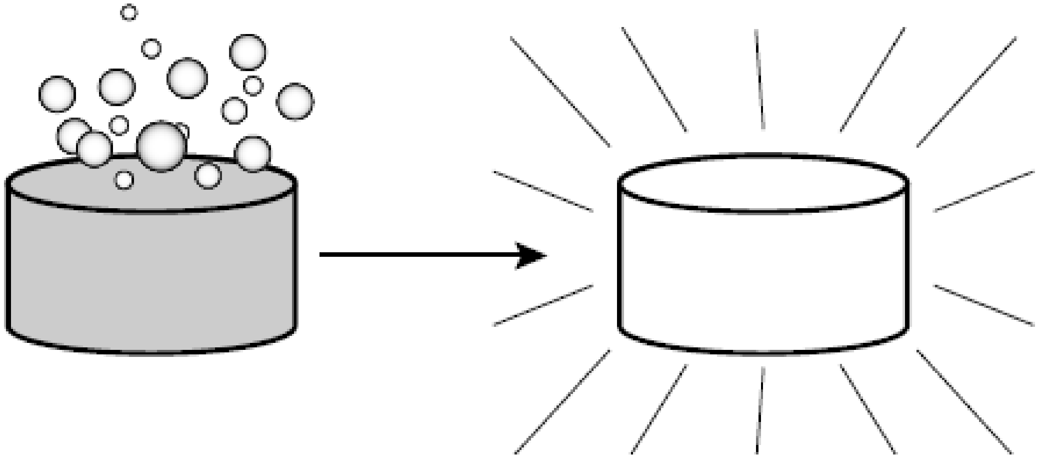
Interpretability: how easily the data can be understood?

Major Tasks in Data Preprocessing

- **Data cleaning** can be applied to remove noise and correct inconsistencies in the data.
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration** merges data from multiple sources into a coherent data store, such as a data warehouse.
  - Integration of multiple databases, data cubes, or files
- **Data reduction** can reduce the data size by aggregating, eliminating redundant features, or clustering.
  - Dimensionality reduction, Numerosity reduction, Data compression
- **Data transformations and Data Discretization**, such as normalization, may be applied.
  - For example, normalization may improve the accuracy and efficiency of mining algorithms involving distance measurements.
  - Concept hierarchy generation

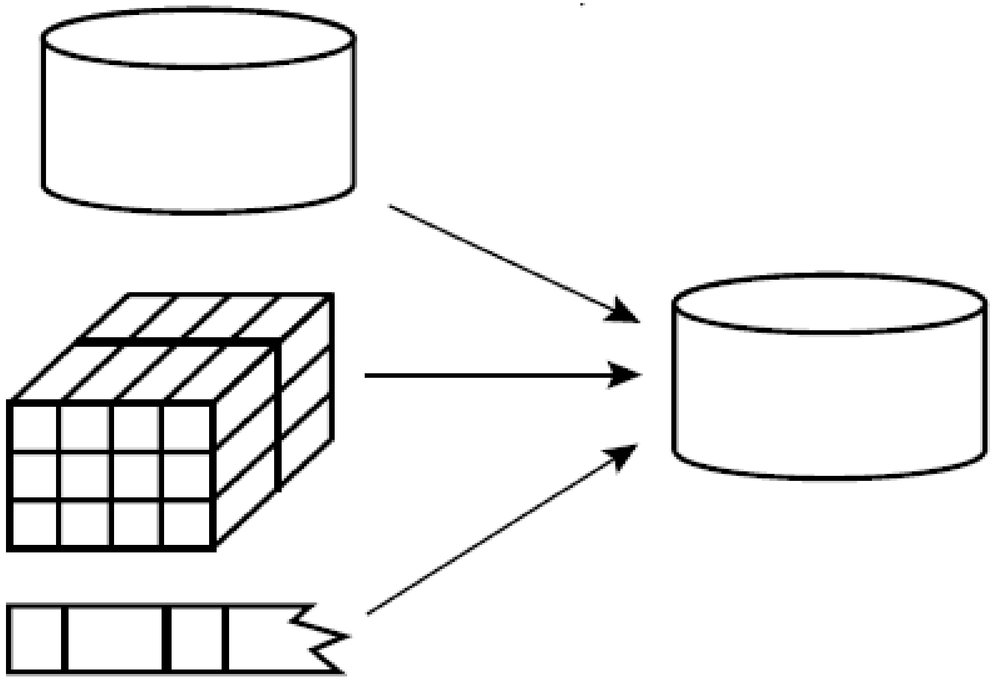
Major Tasks in Data Preprocessing
Data Cleaning

- **Data cleaning** routines work to “clean” the data by filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies.
  - If users believe the data are dirty, they are unlikely to trust the results of any data mining that has been applied to it.
  - Dirty data can cause confusion for the mining procedure, resulting in unreliable output



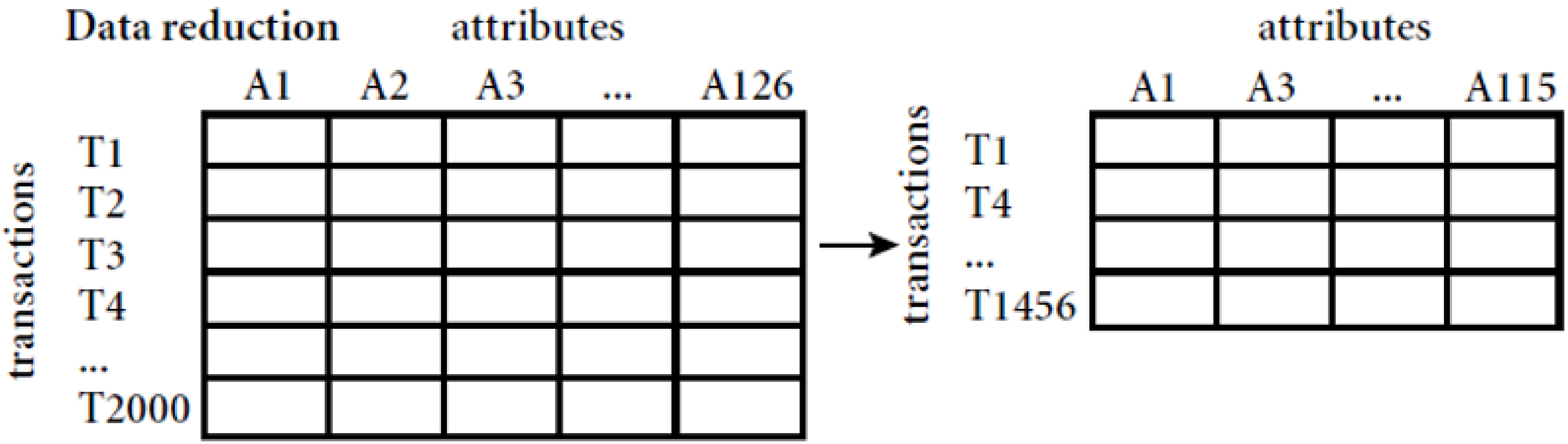
Major Tasks in Data Preprocessing
Data Integration

- **Data integration** merges data from multiple sources into a coherent data store, such as a data warehouse.



Major Tasks in Data Preprocessing
Data Reduction

- **Data reduction** obtains a reduced representation of the data set that is much smaller in volume, yet produces the same (or almost the same) analytical results.
- Data reduction strategies include **dimensionality reduction** and **numerosity reduction**.



Major Tasks in Data Preprocessing
Data transformations and Data Discretization

- The data are transformed or consolidated so that the resulting mining process may be more efficient, and the patterns found may be easier to understand.

- **Data discretization** is a form of **data transformation**.
  - Data discretization transforms numeric data by mapping values to interval or concept labels.

- Data Transformation: Normalization

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

- Data Quality and Major Tasks in Data Preprocessing
- **Data Cleaning**
- Data Integration
- Data Transformation and Data Discretization
- Data Reduction

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.
    - a popular strategy.
    - In comparison to the other methods, it uses the most information from the present data to predict missing values.

Noisy Data and How to Handle Noisy Data?

- **Noise**: random error or variance in a measured variable
- Outliers may represent noise.
- Given a numeric attribute such as, say, *price*, how can we "**smooth**" out the data to remove the noise?

Data Smoothing Techniques:

- **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)

Binning Methods for Data Smoothing

- **Binning methods** smooth a sorted data by distributing them into bins (buckets).

Smoothing by bin means:

- Each value in a bin is replaced by the mean value of the bin.

Smoothing by bin medians:

- Each bin value is replaced by the bin median.

Smoothing by bin boundaries:

- The minimum and maximum values in a given bin are identified as the bin boundaries.
- Each bin value is then replaced by the closest boundary value.

Binning Methods for Data Smoothing: Example

- Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34
- **Partition into (equal-frequency) bins**:
  - Bin 1: 4, 8, 15
  - Bin 2: 21, 21, 24
  - Bin 3: 25, 28, 34
- **Smoothing by bin means**:
  - Bin 1: 9, 9, 9
  - Bin 2: 22, 22, 22
  - Bin 3: 29, 29, 29
- **Smoothing by bin medians**:
  - Bin 1: 8, 8, 8
  - Bin 2: 21, 21, 21
  - Bin 3: 28, 28, 28
- **Smoothing by bin boundaries**:
  - Bin 1: 4, 4, 15
  - Bin 2: 21, 21, 24
  - Bin 3: 25, 25, 34

Data Smoothing

- *Many methods for data smoothing* are also methods for *data reduction involving discretization*.
  - For example, the binning techniques reduce the number of distinct values per attribute.
    - This acts as a form of data reduction for logic-based data mining methods, such as decision tree induction, which repeatedly make value comparisons on sorted data.

- Concept hierarchies are a form of data discretization that can also be used for data smoothing.
  - A concept hierarchy for price, for example, may map real price values into inexpensive, moderately priced, and expensive, thereby reducing the number of data values to be handled by the mining process.

Data Cleaning as a Process

- **Data discrepancy detection**
  - Use metadata (e.g., domain, range, dependency, distribution)
  - Check uniqueness rule, consecutive rule and null rule
  - For example, values that are more than two standard deviations away from the mean for a given attribute may be flagged as potential outliers.
  - 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 is a data cleaning tool)

- Data Quality and Major Tasks in Data Preprocessing
- Data Cleaning
- **Data Integration**
- Data Transformation and Data Discretization
- Data Reduction

Data Integration

- **Data integration**:
  - Combines data from multiple sources into a coherent source.
  - Careful integration can help reduce and avoid redundancies and inconsistencies.
- Schema integration:
  - Integrate metadata from different sources
  - e.g., A.cust-id ≡ B.cust-#
- 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

- **Redundancy** is another important issue in data integration.
- An attribute may be **redundant** if it can be “derived” from another attribute or set of attributes.
- Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set after data integration.
- Redundant data occur often after 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.
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality.
- Redundant attributes may be able to be detected by **correlation analysis**.
  - χ^2 **(*chi-square*)** test for nominal attributes.
  - **correlation coefficient** and **covariance** for numeric attributes

Correlation Analysis (for Numeric Data)
Correlation Coefficient

- For numeric attributes, we can evaluate the *correlation between two attributes*, A and B, by computing the **correlation coefficient.**

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

where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B.

- Note that **$-1 \le r_{A,B} \le 1$**
- If **$r_{A,B} > 0$**: A and B are **positively correlated** (A's values increase as B's).
  - The higher value implies a stronger correlation.
- **$r_{A,B} = 0$: independent**;
- **$r_{AB} < 0$: negatively correlated**

Correlation Analysis (for Numeric Data)
Covariance

- The **mean values** of A and B, are also known as the **expected values** on A and B.

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

- The **covariance** between A and B is defined as:

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

- **Covariance** is similar to **correlation coefficient**:

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

- It can also be shown that:

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

  – This equation simplifies the calculation of Cov(A,B).

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)
Covariance

- **Positive covariance**: If $\text{Cov}_{A,B} > 0$, then A and B both tend to be larger than their expected values

- **Negative covariance**: If $\text{Cov}_{A,B} < 0$ then if one of the attributes tends to be above its expected value when the other attribute is below its expected value,

- **Independence**: $\text{Cov}_{A,B} = 0$
  - If A and B are independent, $\text{Cov}_{A,B} = 0$.
  - But the converse is not true:
    - Some pairs of random variables may have a covariance of 0 but they are not independent.
      - Covarianve indicates linear relationship (not non-linear relationship)
    - Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence.

Correlation Test (for Nominal Data)
χ^2 (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 $\boldsymbol{o_{ij}}$ is the ***observed frequency*** (i.e., actual count) of the **joint event ($\boldsymbol{A_i,B_j}$)** and $\boldsymbol{e_{ij}}$ is the ***expected frequency*** of **($\boldsymbol{A_i,B_j}$)**, which can be computed as

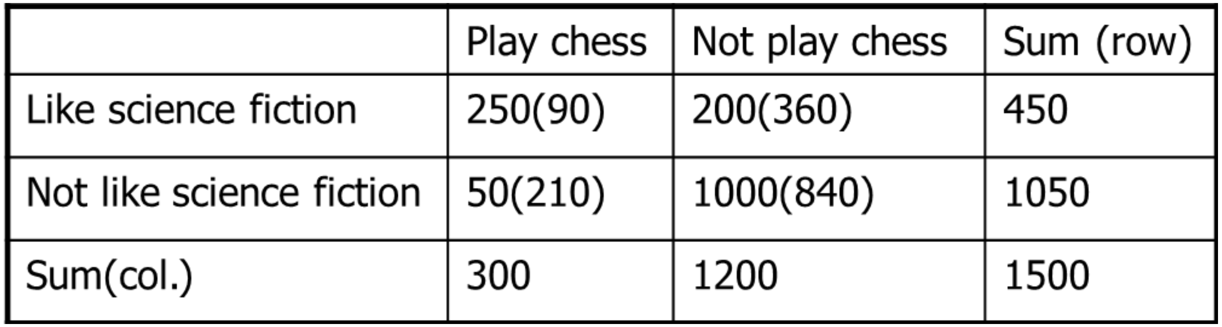
$$e_{ij} = \frac{count(A = a_i) \times count(B = b_j)}{n}$$

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*).

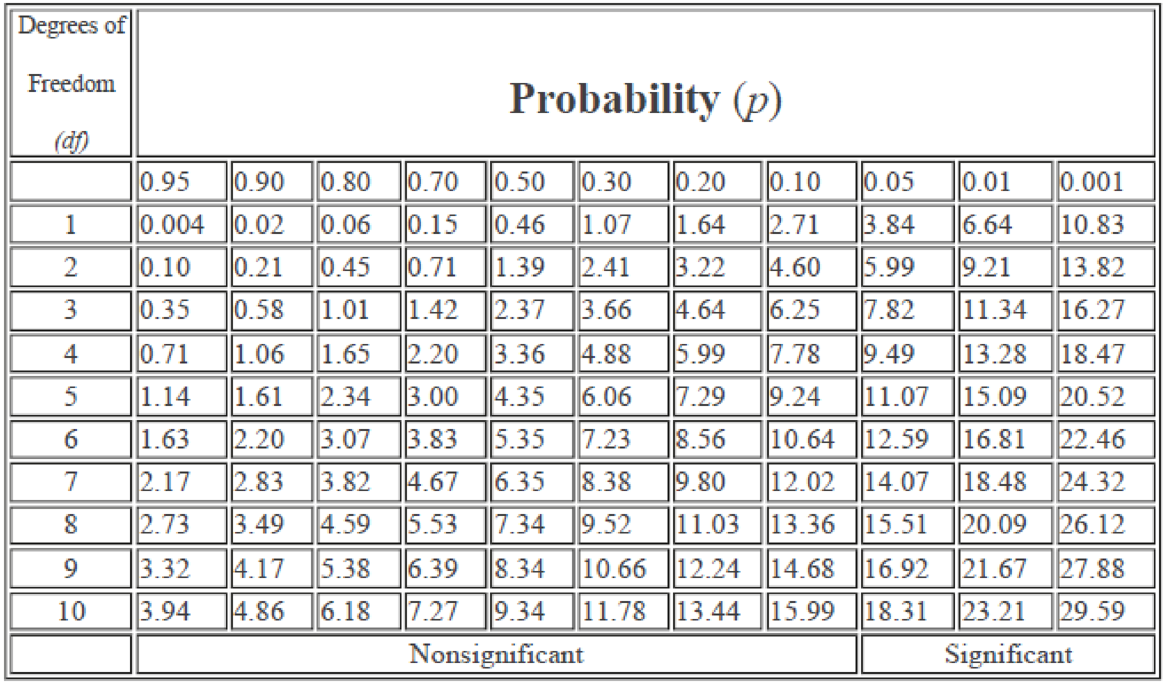
  $e_{LSF,PC}$ = count(LSF)*count(PC) / n = 300*450 / 1500 = 90

- **χ^2 (chi-square) calculation**

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

Chi-Square Calculation: An Example

- For this 2x2 table, the degrees of freedom are (2-1)(2-1) = 1.
  - There are two possible values for *LikeScienceFiction* attribute and two possible values for *PlayChess* attribute.
- For 1 degree of freedom, the χ^2 value needed to reject the hypothesis at the 0.001 significance level is 10.83 (from table of upper percentage points of χ^2 distribution).



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

- Since our computed value 507.93 is above 10.83, we can reject the hypothesis that *LikeScienceFiction* and *PlayChess* are independent and conclude that the two attributes are (strongly) correlated for the given group of people.

- Data Quality and Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- **Data Transformation and Data Discretization**
- Data Reduction

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.*

Normalization

- An attribute is **normalized** by scaling its values so that they fall within a small specified range.
- A larger range of an attribute gives a greater effect (weight) to that attribute.
  - This means that an attribute with a larger range can have greater weight at data minining tasks than an attribute with a smaller range.
- Normalizing the data attempts to give all attributes an equal weight.
  - Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering.

Some Normalization Methods:

- **Min-max normalization**
- **Z-score normalization**
- **Normalization by decimal scaling**

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, $\boldsymbol{v_i}$ of an attribute A to $\boldsymbol{v_i'}$ 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 $\boldsymbol{v_i}$ of attribute A is normalized to $\boldsymbol{v_i'}$ by computing

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

where and $\bar{\mathbf{A}}$ and $\boldsymbol{\sigma_A}$ 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**).

- Some of discretization methods are as follows:

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., χ^2) analysis

Discretization by Binning

- Attribute values can be discretized by applying **equal-width** or **equal-frequency binning**.
- Binning aproaches sorts the atribute values first, then partition them into the bins.
  - **equal width approach** divides the range of the attribute into a user-specified number of intervals each having the same width.
  - **equal frequency (equal depth) approach** tries to put the same number of objects into each interval.
- After bins are determined, all values are replaced by **bin labels** to discretize that attribute.
  - Instead of bin labels, values may be replaced by bin means (or medians).

- Binning does not use class information and is therefore an unsupervised discretization technique.

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.
  - Later, we will talk different clustering algorithms (such as k-means).

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

Discretization by Classification

- Techniques used for a **classification** algorithm such as decision tree can be applied to discretization.
- *Decision tree approaches to discretization* are **supervised**, that is, they make use of class label information.
- These techniques employ a top-down splitting approach for attribute values:
  - Class distribution information is used in the calculation and determination of split-points.
  - The main idea is to select split-points so that a given resulting partition contains as many tuples of the same class as possible.
    - **Entropy** is the most commonly used measure for this purpose.

- Later, we will talk about classification algorithms.

Discretization by Correlation Analysis

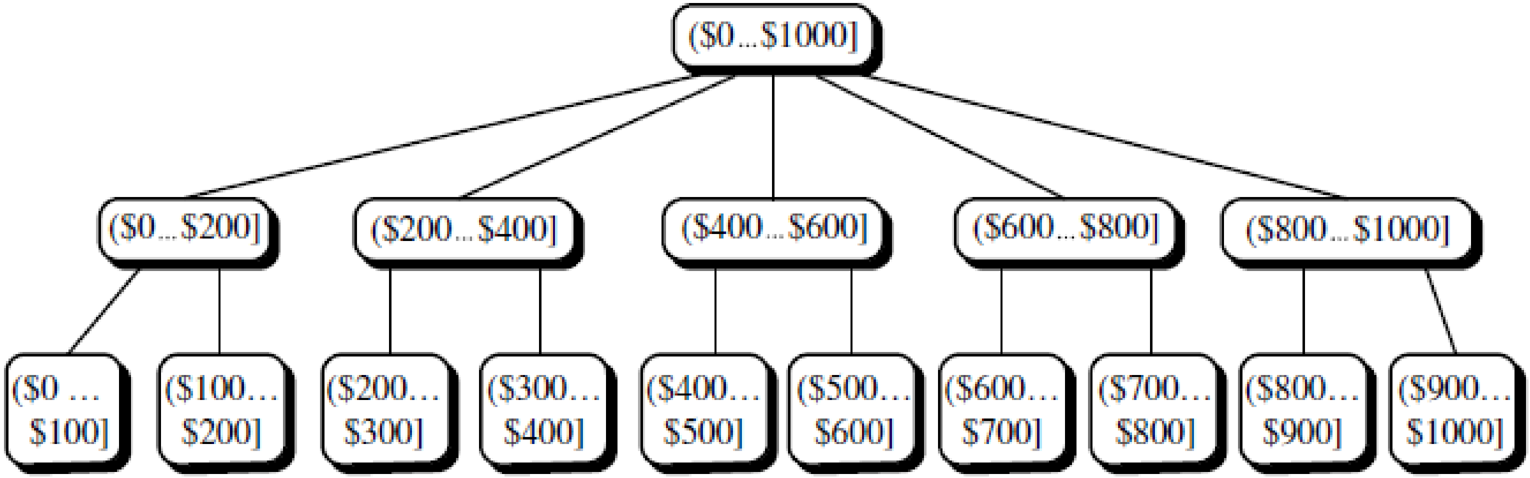
- Measures of correlation can be used for discretization.
- *ChiMerge* is a χ^2 – based discretization method.
  - ChiMerge employs a bottom-up approach.
  - ChiMerge finds the best neighboring intervals and then merge them to form larger intervals, recursively.
  - ChiMerge is supervised since it uses class information.
- ChiMerge proceeds as follows:
  - Initially, each distinct value of a numeric attribute is considered to be one interval,
  - χ^2 tests are performed for every pair of adjacent intervals.
  - Adjacent intervals with the least χ^2 values merged together, because low χ^2 values for a pair indicate similar class distributions.
  - This merging process proceeds recursively until a predefined stopping criterion is met.

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*

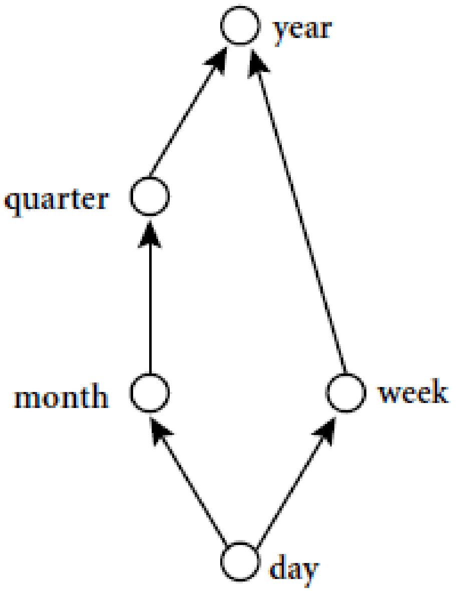


- A concept hierarchy for a numeric attribute can be constructed automatically or it can be created by the user.
- A concept hierarchy can be used for discretization.

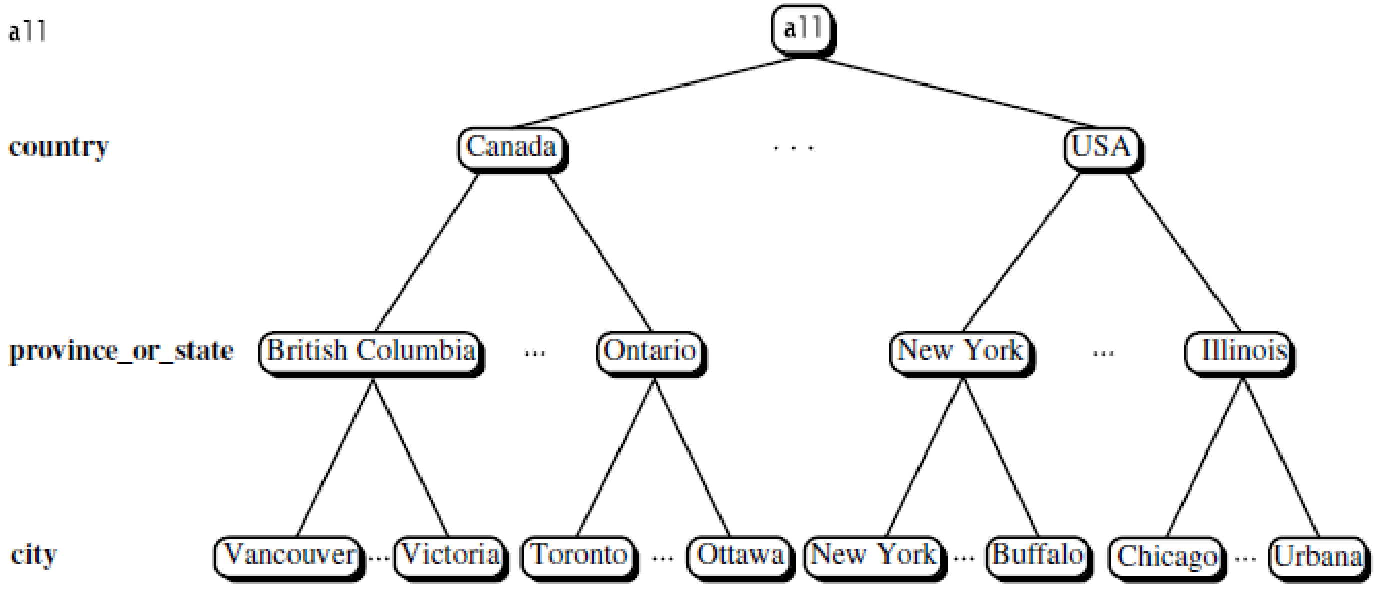
Concept Hierarchy

- Concept hierarchies for nominal attributes can also be created.

time attribute



location attribute



- Data Quality and Major Tasks in Data Preprocessing
- Data Cleaning
- Data Integration
- Data Transformation and Data Discretization
- **Data Reduction**

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 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.
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful.
  - The possible combinations of subspaces will grow exponentially.
- **Dimensionality reduction**
  - Avoid the curse of dimensionality.
  - Help eliminate irrelevant features and reduce noise.
  - Reduce time and space required in data mining.
  - Allow easier visualization.
- **Dimensionality reduction techniques**
  - Wavelet transforms
  - Principal Component Analysis
  - Supervised and nonlinear techniques (e.g., feature selection)

Dimensionality Reduction
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** 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.

Dimensionality Reduction
Attribute Subset Selection

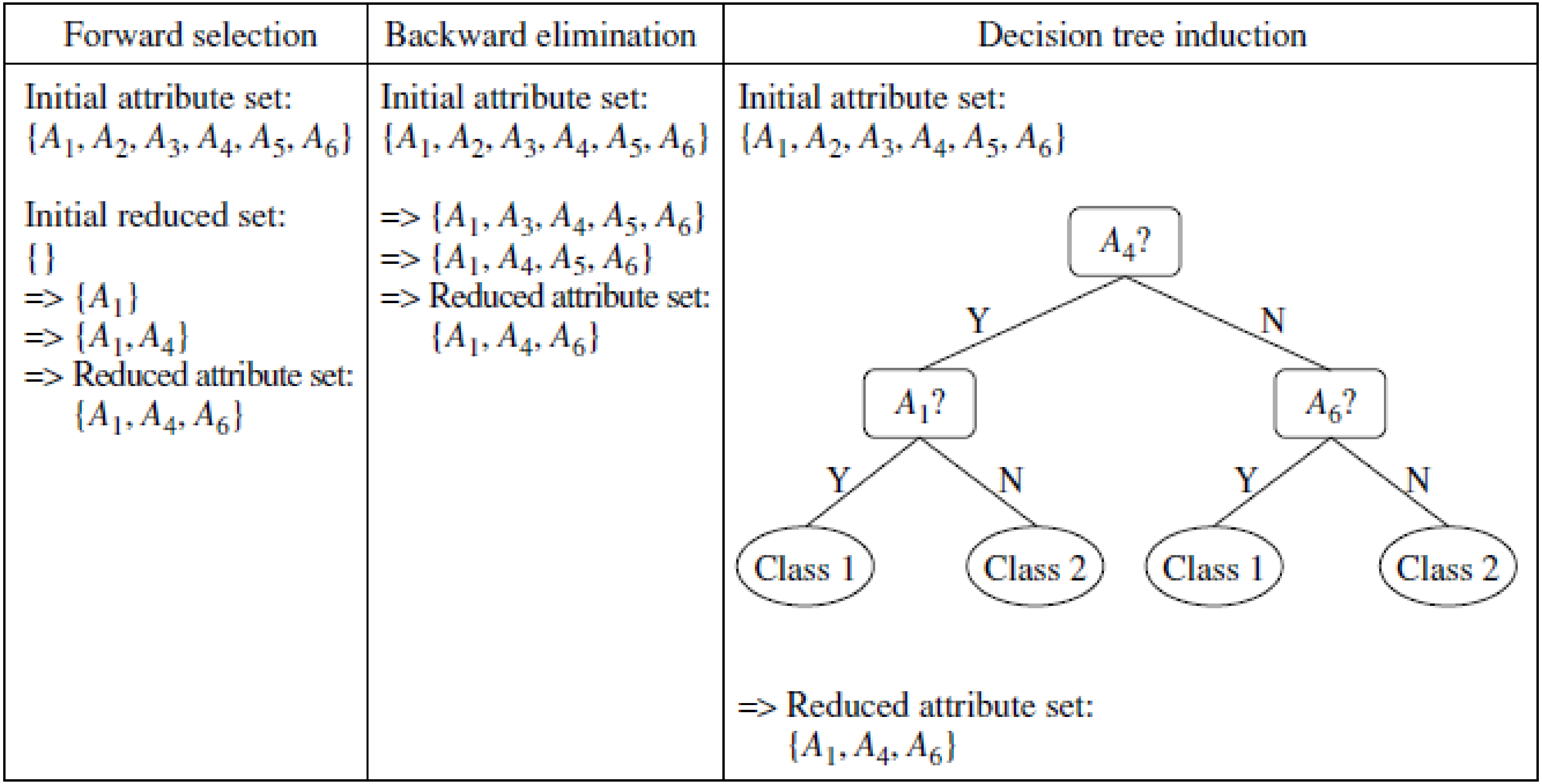
Attribute Subset Selection Techniques:

- 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.

Dimensionality Reduction
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

Dimensionality Reduction
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: $\{A_1, A_4, A_6\}$

A4?
Y
N
A1?
A6?
Y
N
Y
N
Class 1
Class 2
Class 1
Class 2

Dimensionality Reduction
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)

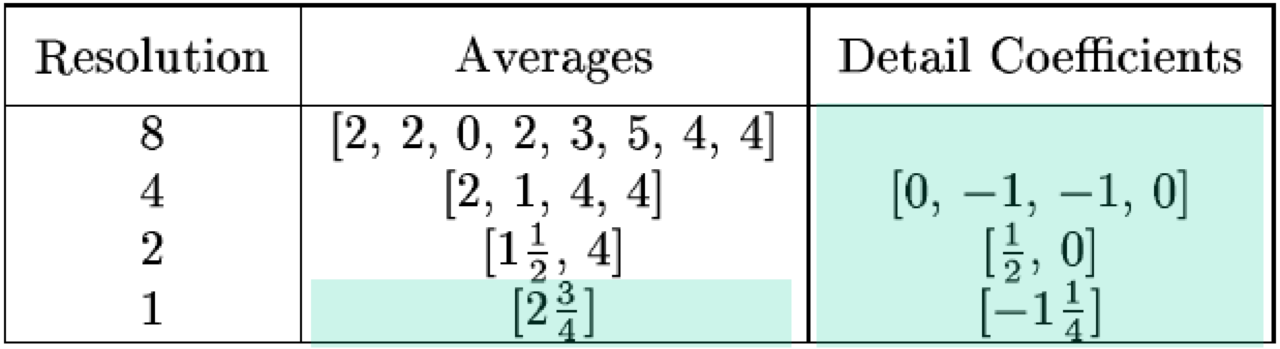
- **Mapping Data to New Space**
  - Wavelet transform, Principal Component Analysis (PCA), …
- **Attribute Construction**
  - combining attributes
  - new attributes are constructed from given attributes in order to help improve accuracy and understanding of structure in high-dimensional data.
    - For example, add the attribute area based on the attributes height and width.

Dimensionality Reduction
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 coefficients**.
- 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

Dimensionality Reduction 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	$[2\frac{3}{4}]$	$[-1\frac{1}{4}]$

Dimensionality Reduction
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 ≤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.

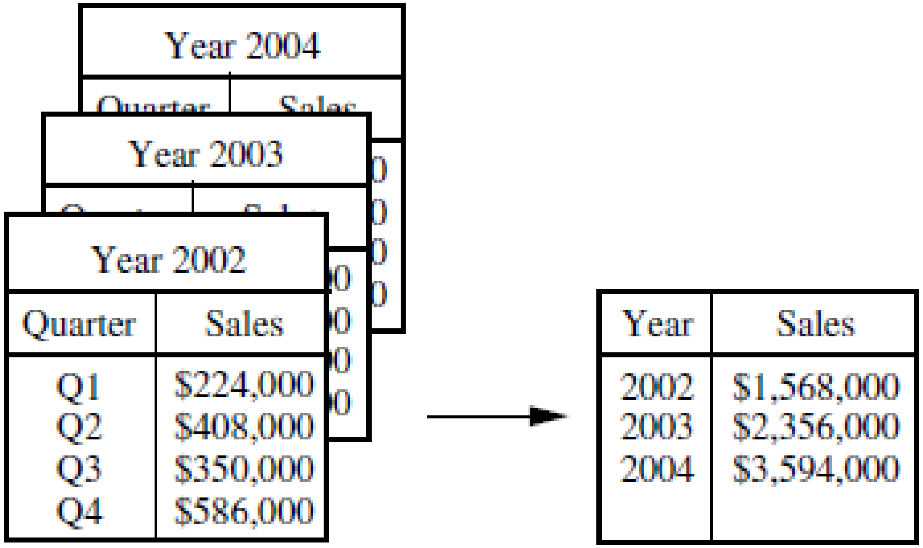
Dimensionality Reduction
Principal Component Analysis (PCA)

Principal Component Analysis Steps: Given N data vectors from n-dimensions, find k ≤ 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)

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.



Year 2004

Year 2003

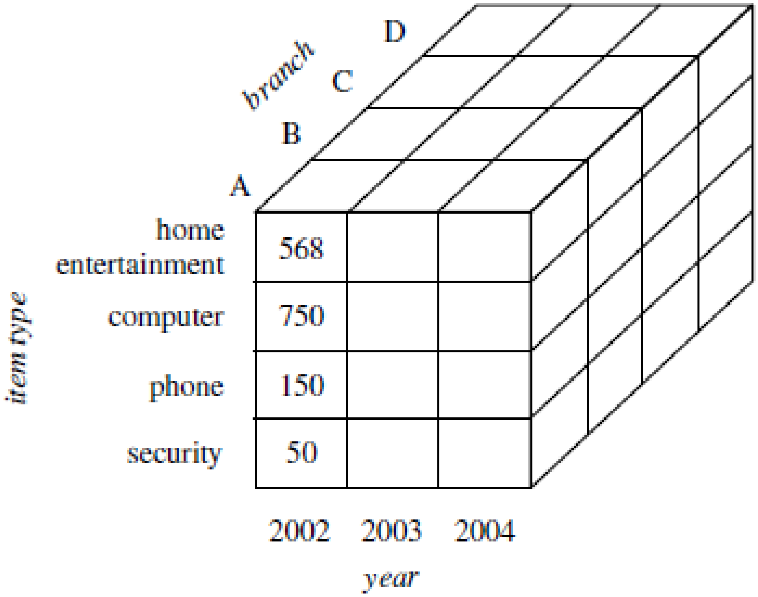
Year 2002	
Quarter	Sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

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

sales 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.

Stratified Sampling

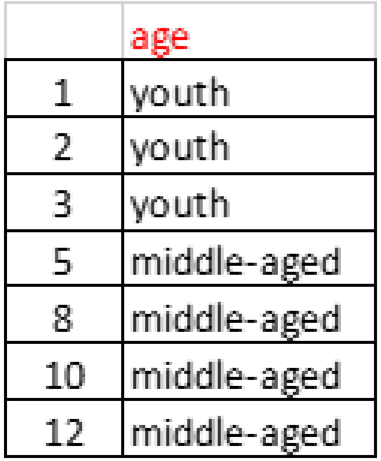
- Split the data into several partitions; then draw random samples from each partition.
  - In the simplest version, equal numbers of objects are drawn from each group even though the groups are of different sizes.
  - In an other variation, the number of objects drawn from each group is proportional 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)

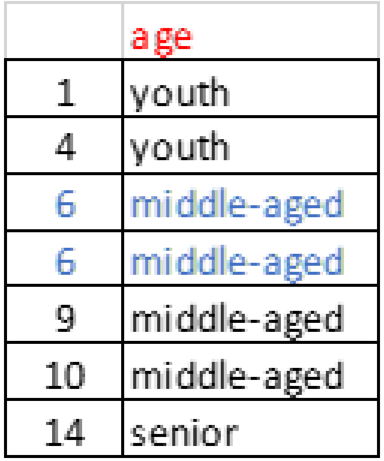
	age
1	youth
2	youth
3	youth
4	youth
5	middle-aged
6	middle-aged
7	middle-aged
8	middle-aged
9	middle-aged
10	middle-aged
11	middle-aged
12	middle-aged
13	senior
14	senior

SRS_WOR



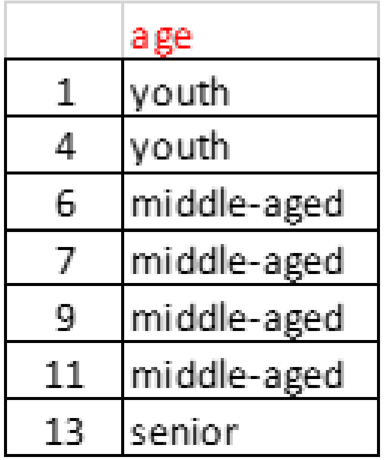
	age
1	youth
2	youth
3	youth
5	middle-aged
8	middle-aged
10	middle-aged
12	middle-aged

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