Data Mining:
Concepts and Techniques

— Chapter 2 —

Jiawei Han
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj

Chapter 2: Data Preprocessing

- Why preprocess the data?
- Descriptive data summarization
- Data cleaning
- Data integration
- Data reduction
- Data Transformation and Discretization

Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., occupation=" "
  - noisy: containing errors or outliers
    - e.g., Salary="-10"
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age="42" Birthday="03/07/1997"
    - e.g., Was rating "1,2,3", now rating "A, B, C"
    - e.g., discrepancy between duplicate records

Why Is Data Dirty?

- Incomplete data may come from
  - Different considerations between the collection and analysis time
  - Human/hardware/software problems (*disguised missing data*)
- Noisy data (incorrect values) may come from
  - Faulty data collection instruments
  - Human or computer error at data entry (buffer size)
  - Errors in data transmission
- Inconsistent data may come from
  - Different data sources
  - Name conventions
  - Functional dependency violation (e.g., modify some linked data)
- Duplicate records also need data cleaning

Why Is Data Preprocessing Important?

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
  - Data warehouse needs consistent integration of quality data (integrity constraints)
  - Data ***extraction***, ***cleaning***, and ***transformation*** comprises the majority of the work of building a data warehouse

How can the data be preprocessed…?

1. *“How can the data be preprocessed in order to help* ***improve the quality of the data*** *and, consequently, of* ***the mining results****?*
2. *“How can the data be preprocessed so as to* ***improve the efficiency and ease of the mining process****?”*

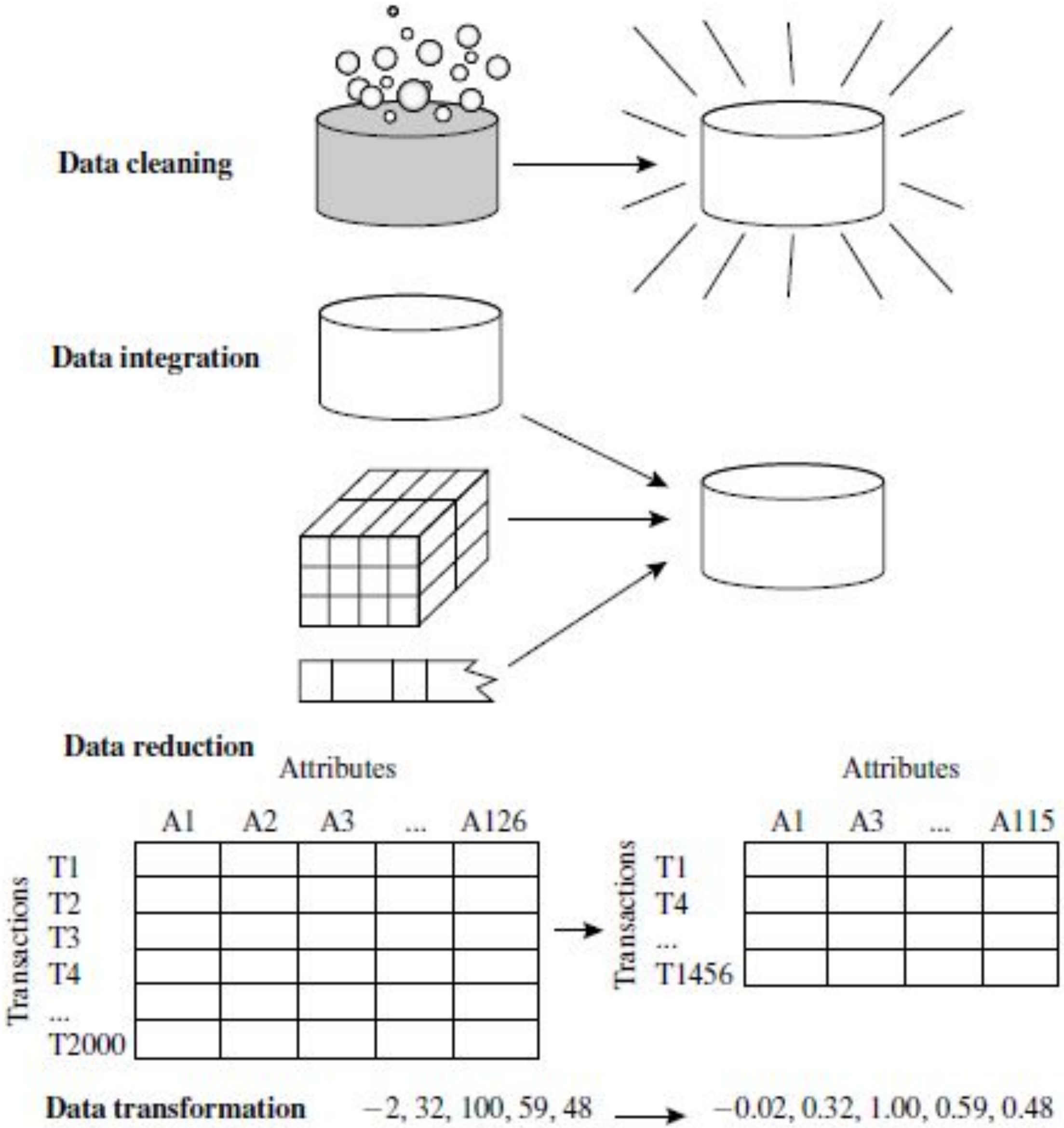
Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
  - **Accuracy** (no noise)
  - **Completeness** (no missing values)
  - **Consistency** (no inconsistency)
  - **Timeliness** (delayed update)
  - **Believability** (how much the data are trusted by users)
  - **Interpretability** (how easy the data to understand)

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 transformation**
  - Normalization and aggregation
- **Data reduction**
  - Obtains reduced representation in volume but produces the same or similar analytical results
- **Data discretization**
  - Part of data reduction but with particular importance, especially for numerical data

Forms of Data Preprocessing



Chapter 2: Data Preprocessing

- Why preprocess the data?
- Descriptive data summarization
- Data cleaning
- Data integration
- Data reduction
- Data Transformation and Data Discretization
- Summary

Data Cleaning

- Importance
  - "Data cleaning is one of the three biggest problems in data warehousing"—Ralph Kimball
  - "Data cleaning is the number one problem in data warehousing"—DCI survey

Data cleaning tasks

1. Fill in missing values
2. Identify outliers and smooth out noisy data
3. Correct inconsistent data
4. Resolve redundancy caused by data integration

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 (calculated)**

How to Handle Missing Data?

- **Ignore the tuple**: usually done when class label is missing (such data could have been useful)
- **Fill in** the missing value **manually**: tedious + infeasible?
- **Fill in** it **automatically** with
  - a **global constant** : e.g., “unknown”, a new class?! (not foolproof)
  - 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, decision tree, regression

bias the data:
the filled-in value may not be correct

Noisy Data

- **Noise**: random error or variance in a measured variable

- Incorrect attribute values may due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention

How to Handle Noisy Data?

- Binning (consulting “neighborhood”)
  - first sort data and partition into bins
  - then **smooth** by bin *means, median, 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

- **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

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

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:

Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:

Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

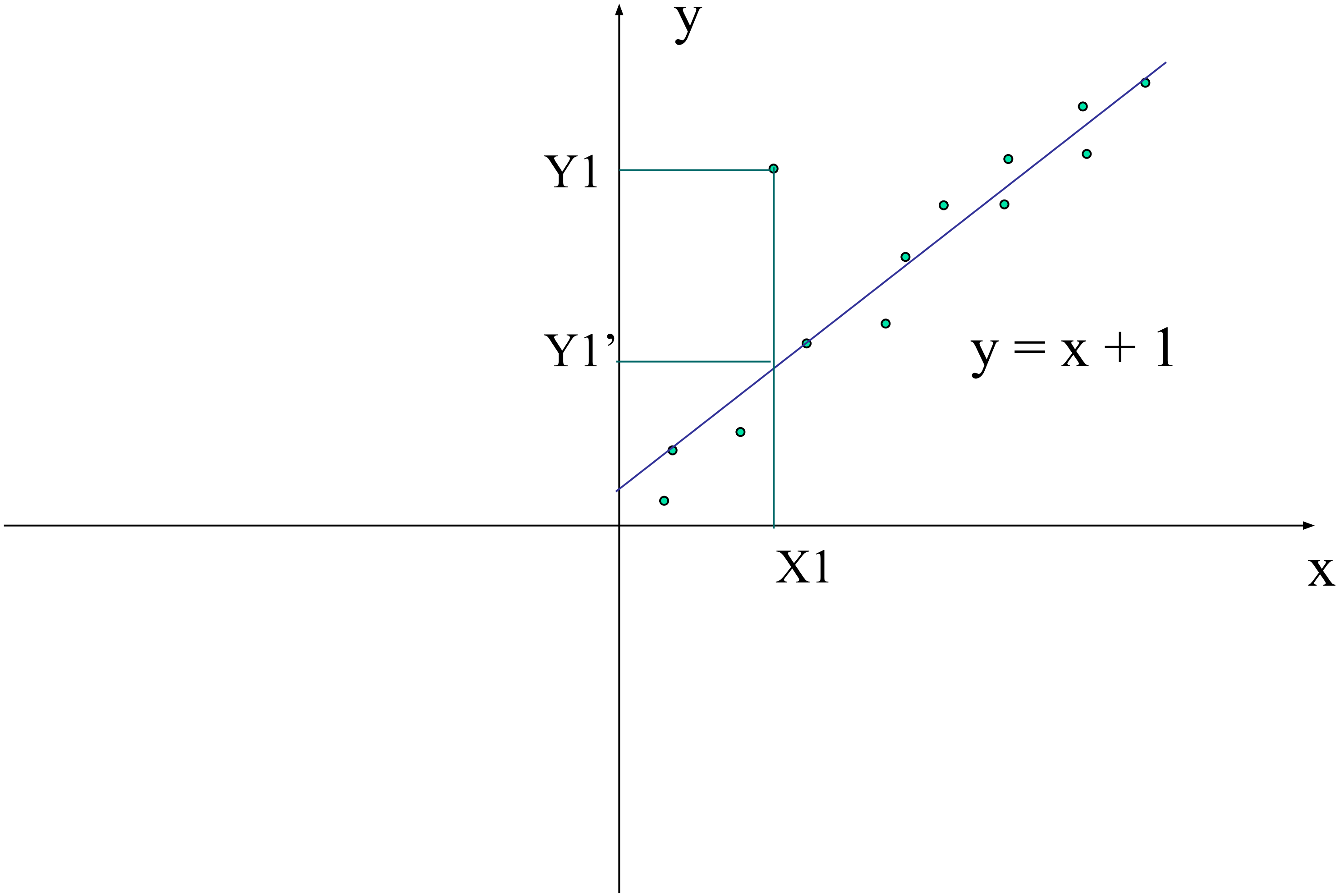
Regression

- Conforms data values to a ***function***

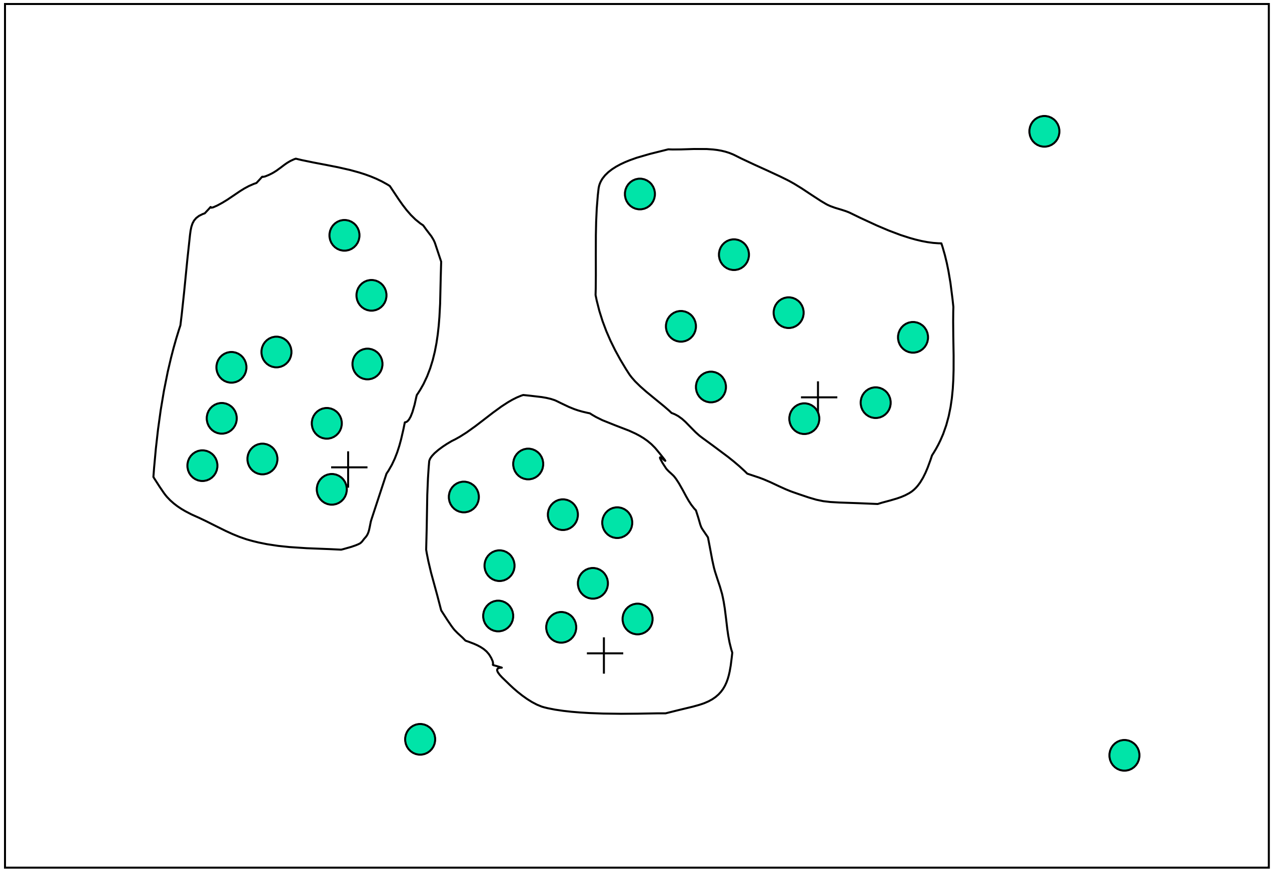
- ***Linear regression*** finds the "best" line to fit two attributes (variables) so that one attribute can be used to predict the other.

- ***Multiple linear regression*** - more than two attributes are involved and the data are fit to a multidimensional *surface*.

Regression



Cluster Analysis



More on smoothing…

- Data smoothing methods are also used for *data discretization* and *data reduction*

- For example, binning reduces the number of distinct values per attribute.
  - Concept hierarchies are a form of data discretization:
    - ***price***, real values ->
    {*inexpensive, moderately priced, expensive*}

Data Cleaning as a Process

- *"But data cleaning is a big job.*
- *What about data cleaning as a process?*
- *How exactly does one proceed in tackling this task?*
- *Are there any tools out there to help?"*

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 GUI

Chapter 2: Data Preprocessing

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

Data Integration

- Combines data from multiple sources into a coherent store
- Careful integration can reduce and avoid *redundancies* and *inconsistencies*
- Is about:
  - How can we match schema and objects from different sources?
  - Are any attributes correlated?
  - Tuple duplication
  - Detection and resolution of data value conflicts

Data Integration

- Schema integration: e.g., A.**cust-id** ≡ B.**cust-#**
  - Integrate metadata from different sources
- Entity identification (schema matching) 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 (*temperature*)
  - Possible reasons: different representations, different scales, e.g., metric vs. British units (*meter* <-> *foot*)

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*

Correlation Analysis (Numerical Data)

- Correlation coefficient (Pearson's product moment coef.)

$$r_{A,B} = \frac{\sum (A - \bar{A})(B - \bar{B})}{(n-1)\sigma_A \sigma_B} = \frac{\sum (AB) - n\bar{A}\bar{B}}{(n-1)\sigma_A \sigma_B}$$

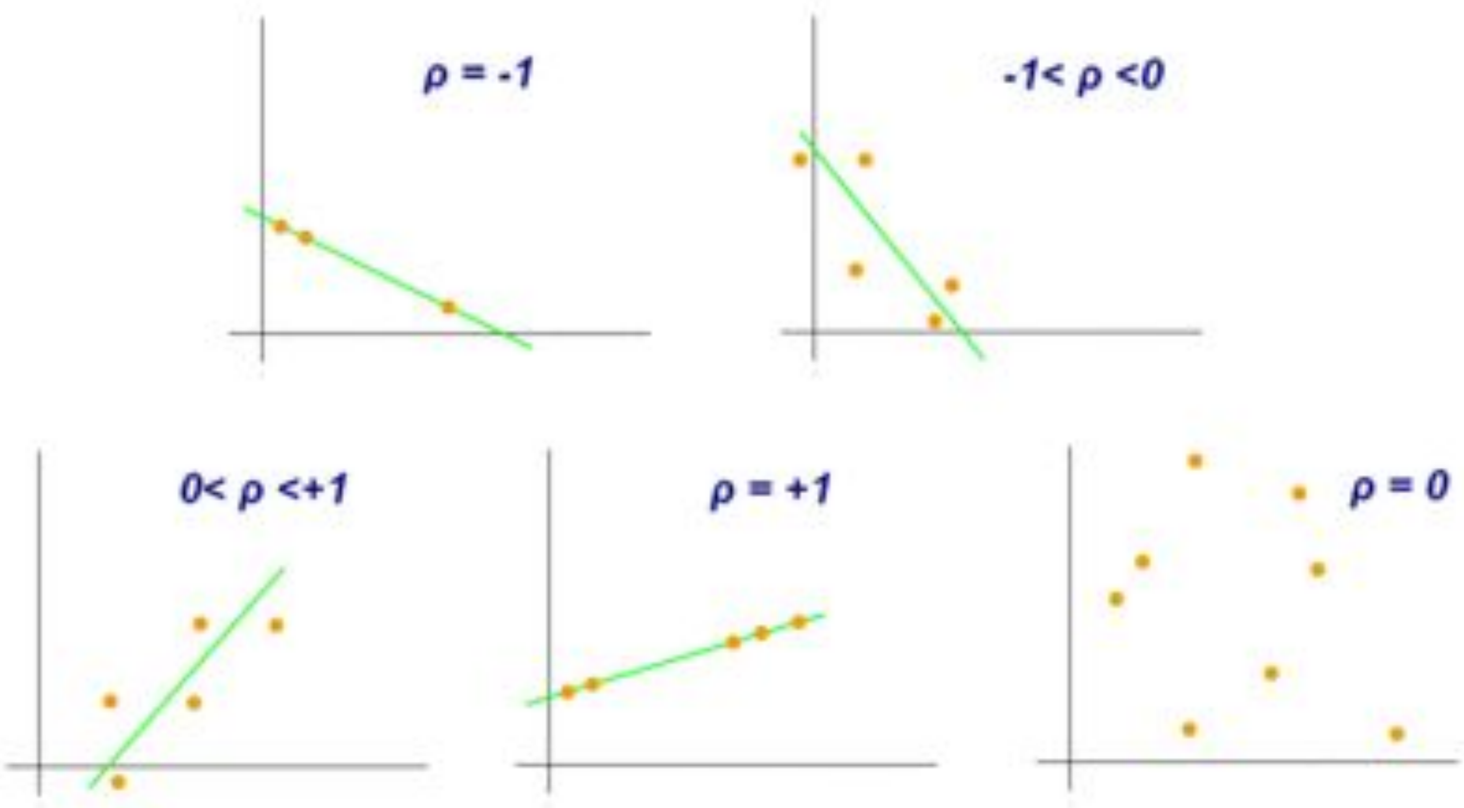
where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(AB)$ is the sum of the AB cross-product.

- 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*;
- $r_{A,B} < 0$: *negatively correlated*

Pearson's coefficient

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

Examples with different values of correlation coefficient (p)



Covariance of Numeric Data

- *correlation* and *covariance* are two similar measures for assessing how much two attributes change together
- Consider two numeric attributes A and B, and a set of n observations $\{(a_1,b_1), \ldots , (a_n,b_n)\}$.
- The mean values of A and B, are also known as the **expected values** on A and B

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

correlation and covariance

- Pearson's correlation

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

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

- If *A* and *B* are *independent* (do not correlate), then $E(A \cdot B) = E(A) \cdot E(B)$
- Therefore, $Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B} = E(A) \cdot E(B) - \bar{A}\bar{B} = 0$

Example

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

Stock Prices for *AllElectronics* and *HighTech*

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

$$E(AllElectronics) = \frac{6+5+4+3+2}{5} = \frac{20}{5} = \$4$$

$$E(HighTech) = \frac{20+10+14+5+5}{5} = \frac{54}{5} = \$10.80$$

$$Cov(AllElectroncis, HighTech) = \frac{6\times 20 + 5\times 10 + 4\times 14 + 3\times 5 + 2\times 5}{5} - 4\times 10.80$$
$$= 50.2 - 43.2 = 7.$$

Correlation Analysis (Categorical 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

Chapter 2: Data Preprocessing

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

Data Reduction Strategies

- Why data reduction?
  - A database/data warehouse may store terabytes of data
  - Complex data mining may take a very **long time** to run on the complete data set
- Data reduction
  - Obtain a **reduced representation** of the data set that is much smaller in volume but yet **produce the same analytical results**

Data reduction strategies

- Dimensionality reduction —remove unimportant attributes
  - Wavelet transforms
  - Principal Component Analysis
  - Feature Selection

- Data Compression

- Numerosity reduction —fit data into models
  - Parametric: parameters instead of data
  - Non-parametric: sampling, histograms, cube aggregation, etc.

Attribute Subset Selection

Feature selection (i.e., attribute subset selection):

- Select a *minimum set of features* such that the **probability distribution** of different classes given the values for those features is as close as possible to the original distribution given the values of all features

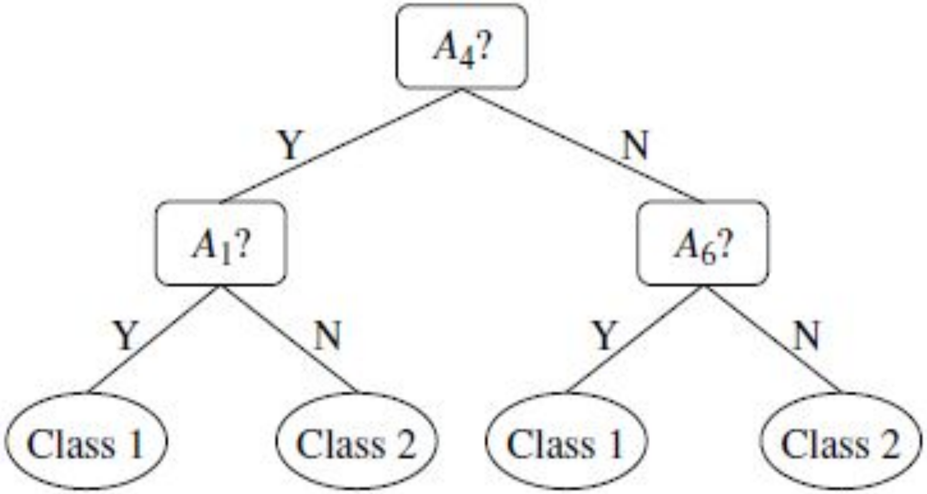
Feature Selection

- Heuristic methods (due to exponential # of choices):
  - Step-wise forward **selection**
  - Step-wise backward **elimination**
  - Combining forward selection and backward elimination
  - Decision-tree induction

- Use tests of **statistical significance** (assume that the attributes are independent of one another), **information gain**, etc.

Heuristic methods

- Greedy approach

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

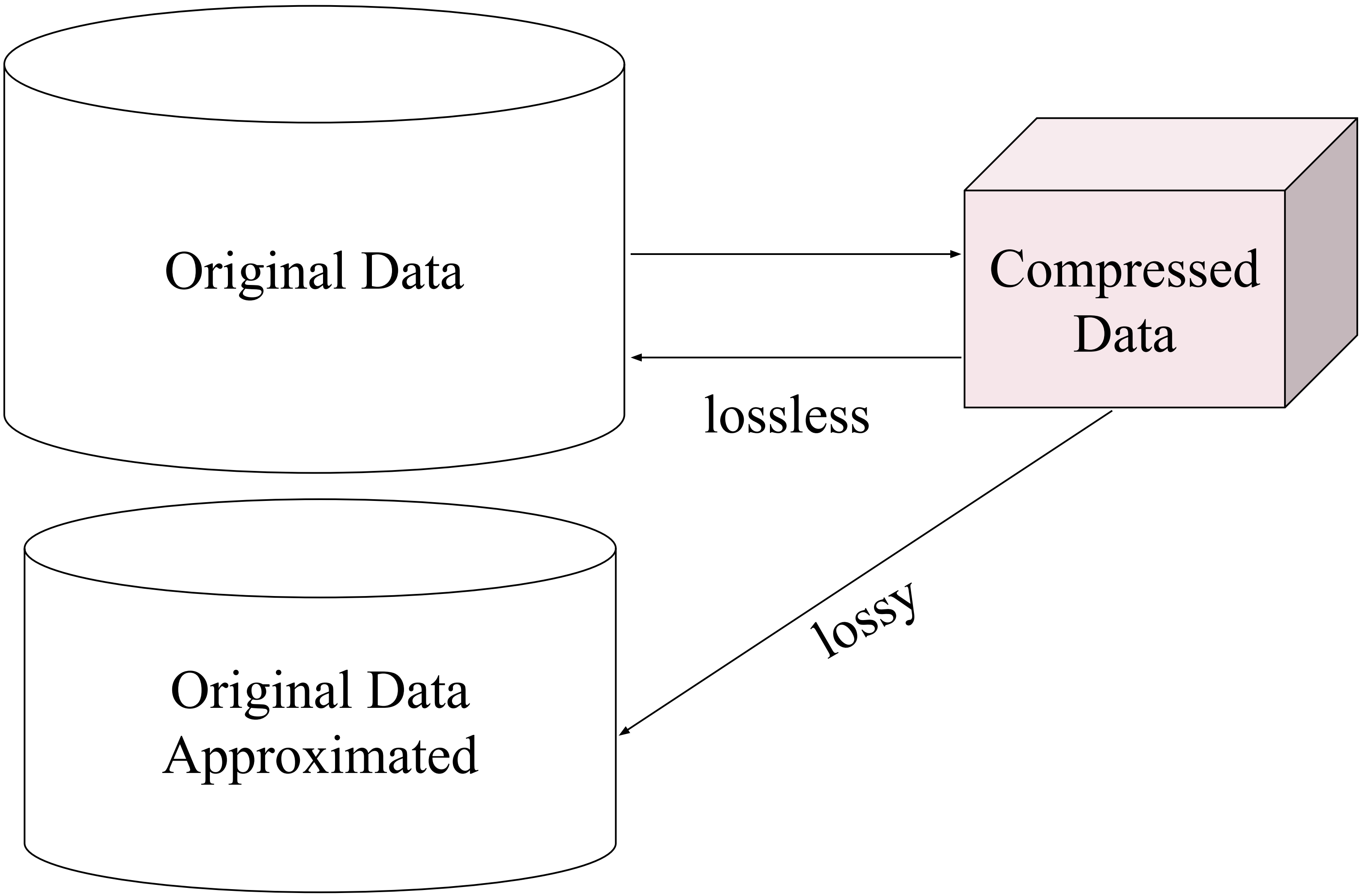
Heuristic Feature Selection Methods

- Best step-wise feature **selection**:
  - The best single-feature is picked first
  - Then next best feature condition to the first, ...
- Step-wise feature **elimination**:
  - Repeatedly eliminate the worst feature
- Best **combined** feature selection and elimination:
  - Repeatedly, select the best attribute and removed the worst from among the remaining attributes
- **Tree induction**:
  - Keep all attributes appearing in the tree (class prediction)

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

Data Compression



Numerosity Reduction

- Reduce data volume by choosing alternative, **smaller forms of data representation**
- Parametric methods
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Example: 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

Data Reduction Method: Regression and Log-Linear Models

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

Regress Analysis and Log-Linear Models

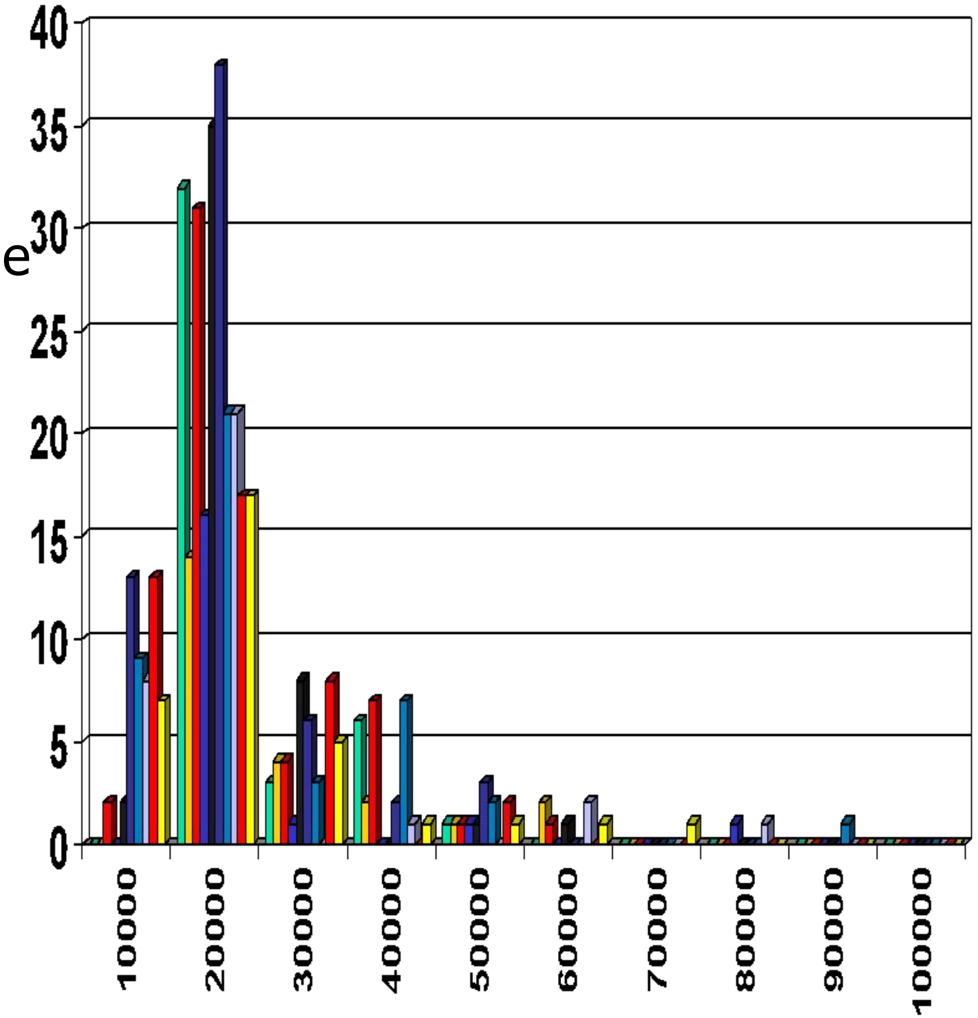
- Linear regression: $Y = w X + b$
  - Two regression coefficients, w and b, specify the line and are to be estimated by using the data at hand
  - Using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.
- Multiple regression: $Y = b_0 + b_1 X1 + b_2 X2$
  - Many nonlinear functions can be transformed into the above
- Log-linear models:
  - The multi-way table of joint probabilities is approximated by a product of lower-order tables
  - Probability: $p(a, b, c, d) = \alpha ab\ \beta ac\ \chi ad\ \delta bcd$

Data Reduction Method: Histograms

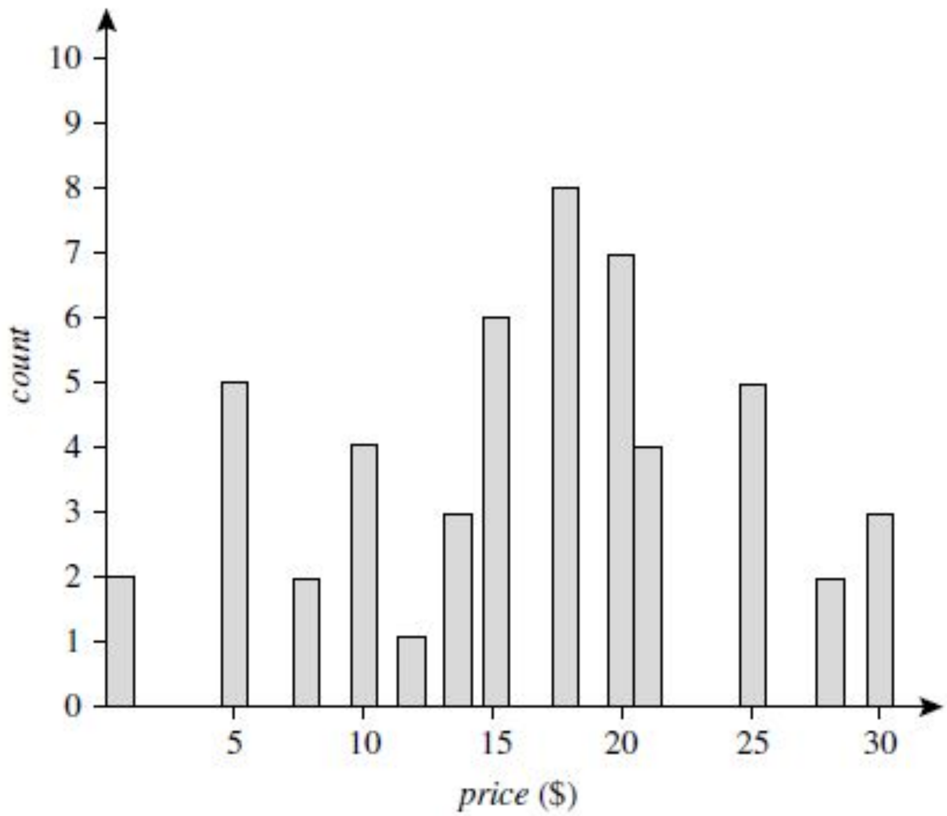
- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
  - Equal-width: equal bucket range
  - Equal-frequency/depth
  - V-optimal: with the least *histogram variance*

$$W = \sum_{j=1}^{J} n_j V_j \,,$$

  - MaxDiff, etc.

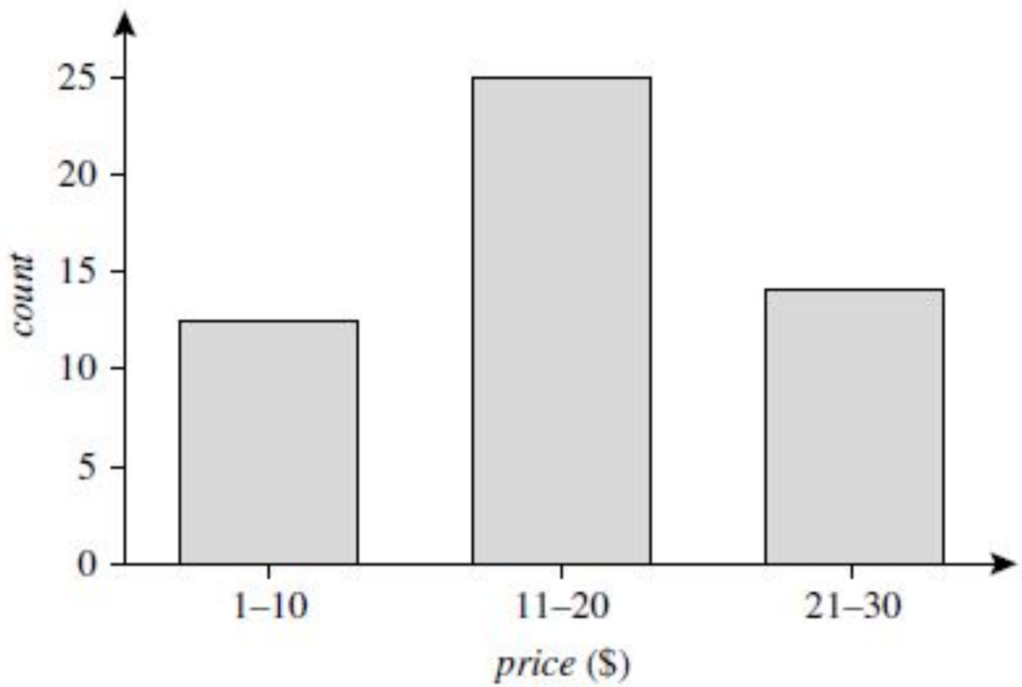


Histograms. The following data are a list of *AllElectronics* prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.



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

Histograms. The following data are a list of *AllElectronics* prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.



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

Data Reduction Method: Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., **centroid** and **diameter/centroid distance**)
- 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

Data Reduction Method: 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

- Choose a **representative** subset of the data
  - Simple random sampling may have very poor performance in the presence of skew

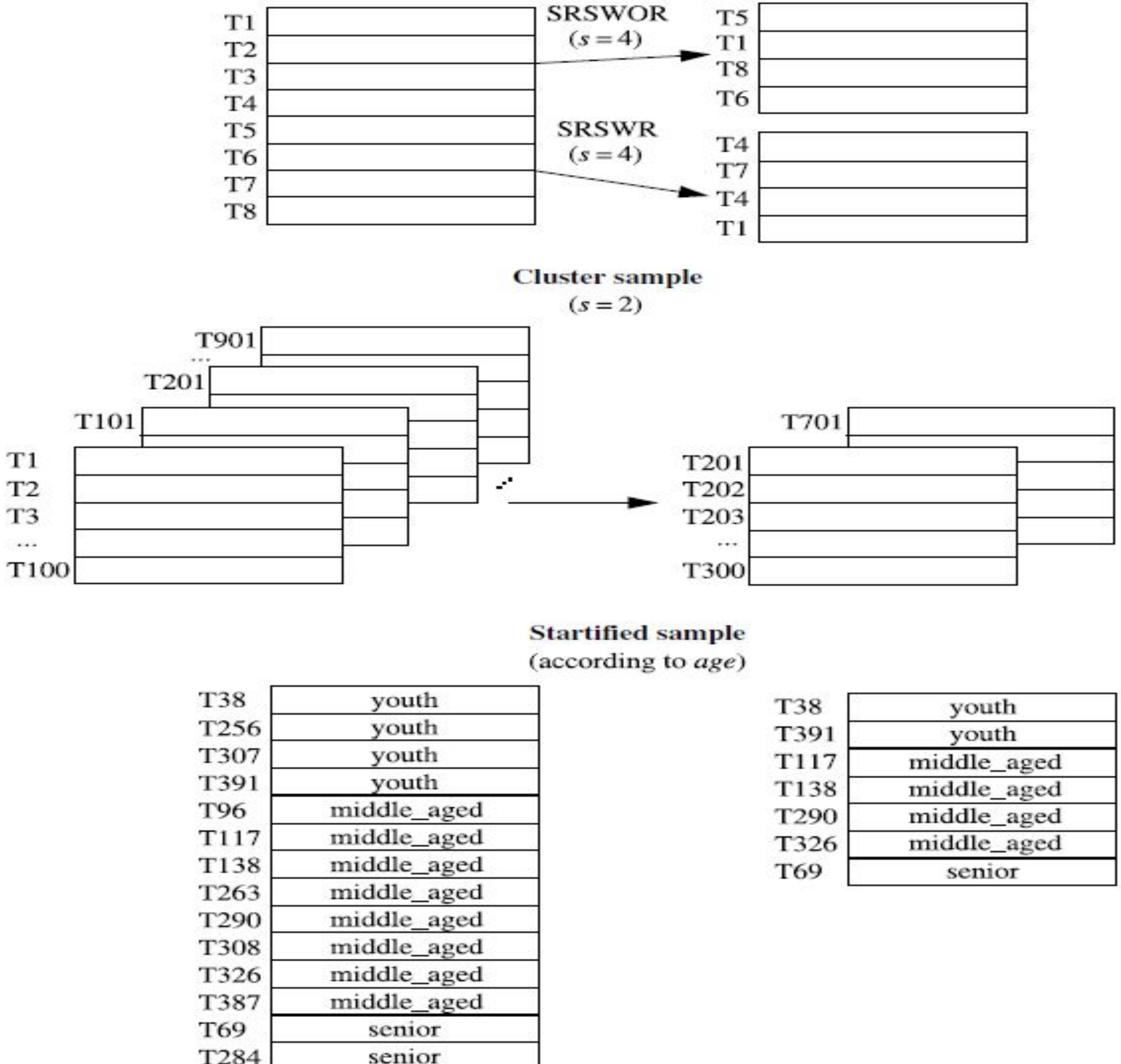
Smart Sampling techniques

- **Simple random sample without replacement (SRSWOR) of size** *s*: drawing *s* of the *N* tuples from *D* ($s < N$), where the probability of drawing any tuple in *D* is $1/N$, that is, all tuples are equally likely to be sampled.

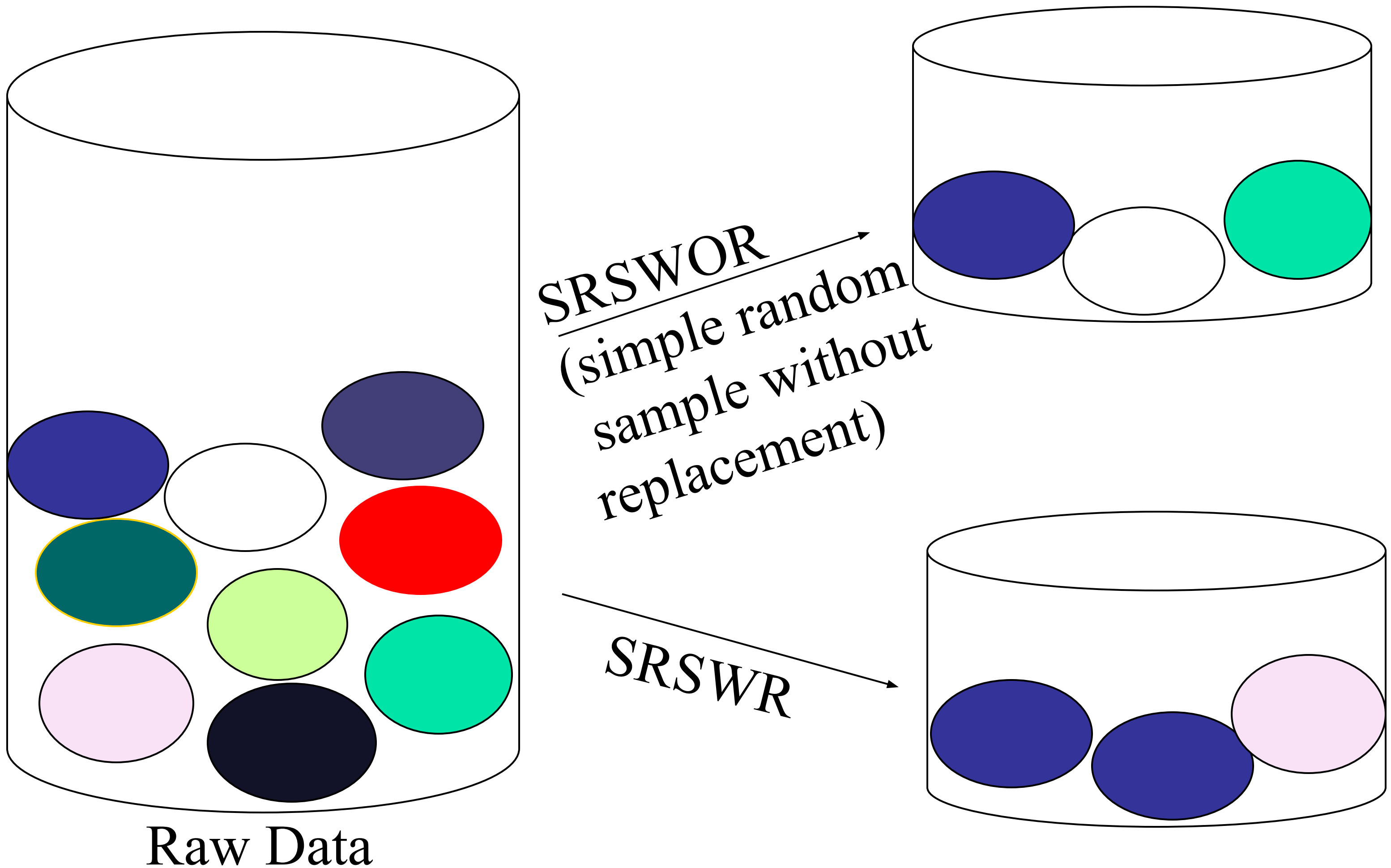
- **Simple random sample with replacement (SRSWR) of size** *s*: similar to SRSWOR, except that each time a tuple is drawn from *D*, it is recorded and then *replaced*. That is, after a tuple is drawn, it is placed back in *D* so that it may be drawn again.

- **Cluster sample**: If the tuples in *D* are grouped into *M* mutually disjoint "clusters," then an SRS of *s* clusters can be obtained, where $s < M$. For example, each page can be considered a cluster. Creates a sample of clusters.

- **Stratified sample**: If *D* is divided into mutually disjoint parts called *strata,* a stratified sample of *D* is generated by obtaining an SRS at each stratum. This helps ensure a **representative** sample. Even small groups are represented!

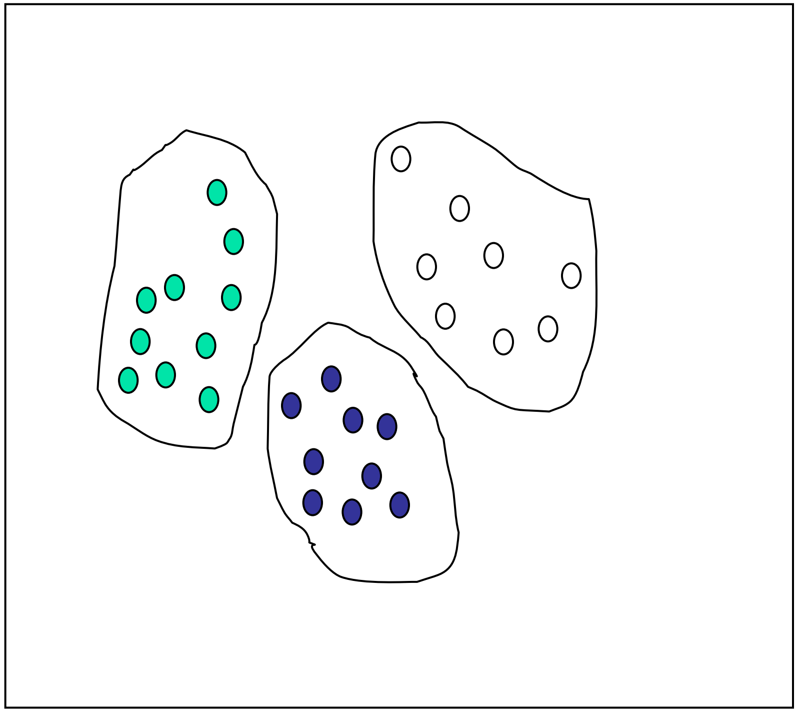

T1
T2
T3
T4
T5
T6
T7
T8
SRSWOR
(s = 4)
T5
T1
T8
T6
SRSWR
(s = 4)
T4
T7
T4
T1
Cluster sample
(s = 2)
T901
...
T201
T101
T1
T2
T3
...
T100
T701
T201
T202
T203
...
T300
Startified sample
(according to age)
T38 youth
T256 youth
T307 youth
T391 youth
T96 middle_aged
T117 middle_aged
T138 middle_aged
T263 middle_aged
T290 middle_aged
T308 middle_aged
T326 middle_aged
T387 middle_aged
T69 senior
T284 senior
T38 youth
T391 youth
T117 middle_aged
T138 middle_aged
T290 middle_aged
T326 middle_aged
T69 senior

Sampling: with or without Replacement

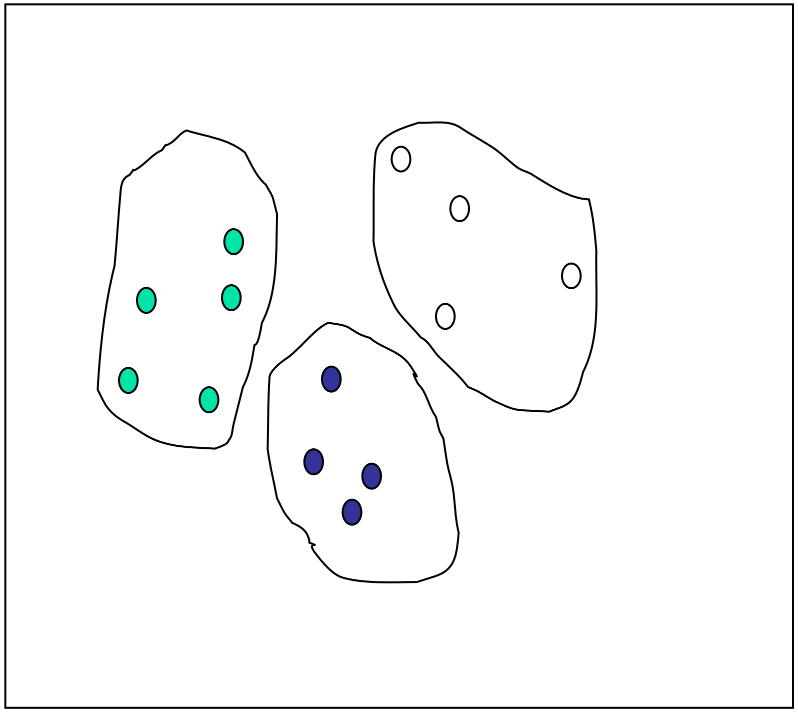


Sampling: Cluster or Stratified Sampling

Raw Data



Cluster/Stratified Sample



Data Cube Aggregation

- Data cubes store multidimensional aggregated information
  - Each cell holds an aggregate data value, corresponding to the data point in multidimensional space
  - *E.g., annual sales per item type for each AllElectronics branch*
- *Concept hierarchies* for each attribute, allowing the analysis of data at multiple abstraction levels
  - *E.g., a hierarchy for branch could allow branches to be grouped into regions, based on their address*
- Provide fast access to **precomputed**, summarized data
  - benefiting online analytical processing

Example: sales at *AllElectronics*

- Sales data for a given branch of *AllElectronics* for 2008 - 2010

Year 2010

Year 2009

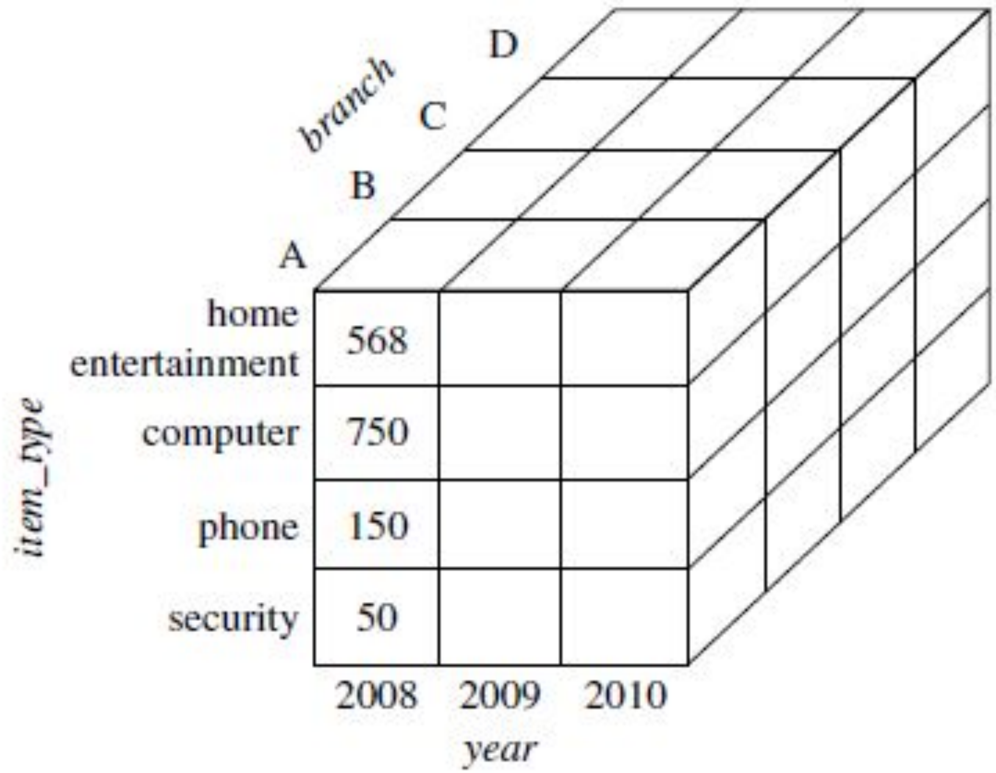
Year 2008	
Quarter	Sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

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

sales per quarter

sales are aggregated to provide the annual sales

Example: A data cube for sales at *AllElectronics*



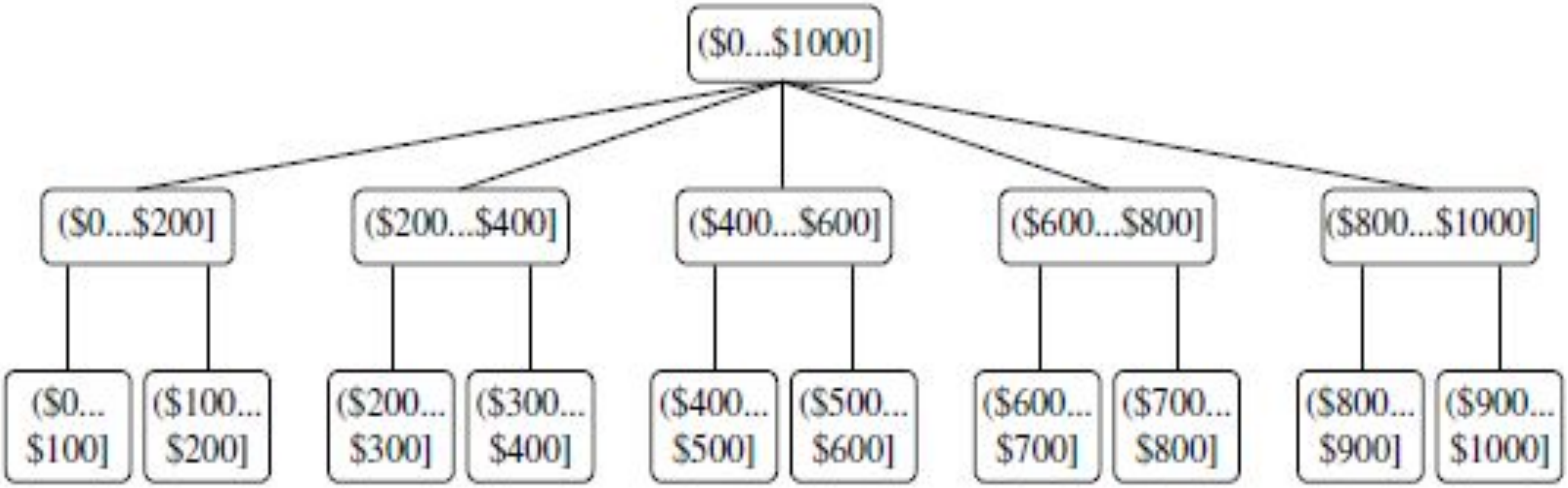
Chapter 2: Data Preprocessing

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

Data Transformation Strategies

1. **Smoothing** (noise removing by binning, regression, etc.) **V**
2. **Attribute/feature construction** (new attributes are constructed and added) **V**
3. **Aggregation** (data cubes) **V**
4. **Normalization** (fit data to a smaller range)
5. **Discretization** (replace raw values by interval or conceptual labels)
6. **Concept hierarchy generation for nominal data** (street -> city or country)

A concept hierarchy for the attribute *price*



Data Transformation: Normalization

- Attempts to give all attributes an equal weight
- Useful for classification algorithms involving neural networks or
- Distance measurements such as nearest-neighbor classification and clustering

Data Transformation: 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 (next lesson)

- 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
  - **Continuous** — real numbers, e.g., integer or real numbers
- Discretization:
  - Divide the range of a continuous attribute into intervals
  - Some classification algorithms only accept categorical attributes.
  - Reduce data size by discretization
  - Prepare for further analysis

Chapter 2: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Chapter 2: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Summary

- Data preparation or preprocessing is a big issue for both data warehousing and data mining
- Descriptive data summarization is need for quality data preprocessing
- Data preparation includes
  - Data **cleaning** and data **integration**
  - Data **reduction** and **feature selection**
  - **Discretization** (details in the next lesson)
- A lot a methods have been developed but data preprocessing still an active area of research

Bayesian Classification: Why?

- A statistical classifier: performs *probabilistic prediction, i.e.,* predicts class membership probabilities
- Foundation: Based on Bayes' Theorem.
- Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayes' Theorem: Basics

- **Bayes' Theorem**: $$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H)/P(\mathbf{X})$$
  - Let **X** be a data sample ("***evidence***"): class label is unknown
  - Let H be a ***hypothesis*** that X belongs to class C
  - Classification is to determine P(H|**X**), (i.e., ***posteriori probability):*** the probability that the hypothesis holds given the observed data sample **X**
  - P(H) (***prior probability***): the initial probability
    - E.g., **X** will buy computer, regardless of age, income, …
  - P(**X**): probability that sample data is observed
  - P(**X**|H) (***likelihood***): the probability of observing the sample **X**, given that the hypothesis holds
    - E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income

Prediction Based on Bayes' Theorem

- Given training data **X**, ***posteriori probability*** *of a hypothesis* H, P(H|**X**), follows the Bayes' theorem

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$

- Informally, this can be viewed as

  posteriori = ***likelihood*** x ***prior***/***evidence***

- Predicts **X** belongs to C_i iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes
- **Practical difficulty**: It requires initial knowledge of many probabilities, involving significant computational cost

Classification Is to Derive the Maximum Posteriori

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector **X** = (x_1, x_2, …, x_n)
- Suppose there are *m* classes C_1, C_2, …, C_m.
- Classification is to derive the maximum posteriori, i.e., the **maximal *P(C$_i$|X)***
- This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since P(X) is constant for all classes, only

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

needs to be maximized

Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution
- If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i, D}|$ (# of tuples of C_i in D)
- If A_k is continous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

and $P(x_k|C_i)$ is

$$P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

age	income	student	credit rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayes Classifier: An Example

age	income	student	credit_rating	_comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- **P(C_i)**: P(buys_computer = “yes”) = 9/14 = 0.643
  P(buys_computer = “no”) = 5/14= 0.357
- Compute P(X|C_i) for each class
  P(age = “<=30” | buys_computer = “yes”) = 2/9 = **0.222**
  P(age = “<= 30” | buys_computer = “no”) = 3/5 = 0.6
  P(income = “medium” | buys_computer = “yes”) = 4/9 = **0.444**
  P(income = “medium” | buys_computer = “no”) = 2/5 = 0.4
  P(student = “yes” | buys_computer = “yes) = 6/9 = **0.667**
  P(student = “yes” | buys_computer = “no”) = 1/5 = 0.2
  P(credit_rating = “fair” | buys_computer = “yes”) = 6/9 = **0.667**
  P(credit_rating = “fair” | buys_computer = “no”) = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

P(X|C_i) : P(X|buys_computer = “yes”) = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = “no”) = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

P(X|C_i)*P(C_i) : P(X|buys_computer = “yes”) * P(buys_computer = “yes”) = **0.028**
P(X|buys_computer = “no”) * P(buys_computer = “no”) = 0.007

Therefore, X belongs to class (“buys_computer = yes”)

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Naïve Bayes Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.
      Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier
- How to deal with these dependencies? Bayesian Belief Networks (Chapter 9)

References

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Communications of ACM, 42:73-78, 1999
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley & Sons, 2003
- T. Dasu, T. Johnson, S. Muthukrishnan, V. Shkapenyuk. Mining Database Structure; Or, How to Build a Data Quality Browser. SIGMOD'02.
- H.V. Jagadish et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), December 1997
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999
- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Bulletin of the Technical Committee on Data Engineering. Vol.23, No.4*
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'2001
- T. Redman. Data Quality: Management and Technology. Bantam Books, 1992
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. Communications of ACM, 39:86-95, 1996
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995