Reminder – “shopping basket”

• Database consists of sets of items bought together
• Describe the data
• Characterise the “typical” purchase patterns
  – Frequent subsets of items
  – Association rules -> correlations
• Be clever in how to count
• Lattices, borders, RAM/disk, sets, indexes...
Motivation for apriori

• RAM can hold a fraction of disk
Algorithms

- Theoretical characteristics – $O()$ notations
  - linear time
  - $n \log n$
  - on paper vs in practice

- Machine learning
  - Quality of predictions
  - time, etc.
Need

• Scan DB only a limited # of times
• Collect necessary information, generate new candidates
• In consecutive scans, check/verify
Data preprocessing
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
  - “Not applicable” data value when collected
  - Different considerations between the time when the data was collected and when it is analyzed.
  - Human/hardware/software problems

- Noisy data (incorrect values) may come from
  - Faulty data collection instruments
  - Human or computer error at data entry
  - Errors in data transmission

- Inconsistent data may come from
  - Different data sources
  - 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
- Data extraction, cleaning, and transformation comprises the majority of the work of building a data warehouse

- Garbage in, garbage out
Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility

- Broad categories:
  - Intrinsic, contextual, representational, and accessibility
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

Data Cleaning
- [water to clean dirty-looking data]
- [‘clean’-looking data]
- [show soap suds on data]

Data Integration

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

Data Reduction

T2
T3
T4
...
T2000

A1 A2 A3 ... A126

T1
T4
...
T1456

A1 A3 ... A115
Chapter 2: Data Preprocessing

- Why preprocess the data?
- Descriptive data summarization
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary
<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.51</td>
<td>9.36</td>
<td>5.93</td>
<td>2.09</td>
<td>4.67</td>
<td>7.38</td>
<td>8.83</td>
<td>2.35</td>
<td>4.90</td>
<td>4.00</td>
</tr>
<tr>
<td>7.25</td>
<td>6.93</td>
<td>8.22</td>
<td>2.15</td>
<td>4.04</td>
<td>5.00</td>
<td>6.15</td>
<td>5.39</td>
<td>0.09</td>
<td>6.39</td>
</tr>
<tr>
<td>3.78</td>
<td>1.37</td>
<td>4.44</td>
<td>8.45</td>
<td>8.97</td>
<td>2.39</td>
<td>6.08</td>
<td>4.04</td>
<td>3.39</td>
<td>0.87</td>
</tr>
<tr>
<td>9.61</td>
<td>5.33</td>
<td>6.46</td>
<td>4.08</td>
<td>8.16</td>
<td>6.41</td>
<td>7.64</td>
<td>5.41</td>
<td>6.95</td>
<td>8.30</td>
</tr>
<tr>
<td>0.61</td>
<td>9.48</td>
<td>9.87</td>
<td>1.05</td>
<td>5.50</td>
<td>4.18</td>
<td>9.86</td>
<td>1.45</td>
<td>1.48</td>
<td>5.95</td>
</tr>
<tr>
<td>0.85</td>
<td>9.16</td>
<td>3.10</td>
<td>3.45</td>
<td>5.58</td>
<td>0.99</td>
<td>7.52</td>
<td>1.26</td>
<td>8.64</td>
<td>0.39</td>
</tr>
<tr>
<td>9.32</td>
<td>7.17</td>
<td>8.96</td>
<td>9.27</td>
<td>1.53</td>
<td>6.08</td>
<td>8.97</td>
<td>0.03</td>
<td>8.75</td>
<td>8.85</td>
</tr>
<tr>
<td>0.58</td>
<td>8.33</td>
<td>8.90</td>
<td>6.71</td>
<td>9.53</td>
<td>1.37</td>
<td>5.01</td>
<td>0.58</td>
<td>0.45</td>
<td>2.26</td>
</tr>
<tr>
<td>7.44</td>
<td>9.90</td>
<td>4.57</td>
<td>8.24</td>
<td>0.02</td>
<td>7.37</td>
<td>1.00</td>
<td>9.48</td>
<td>0.50</td>
<td>4.50</td>
</tr>
<tr>
<td>0.21</td>
<td>5.62</td>
<td>7.07</td>
<td>7.81</td>
<td>5.33</td>
<td>9.14</td>
<td>7.68</td>
<td>6.95</td>
<td>2.85</td>
<td>8.39</td>
</tr>
</tbody>
</table>
Characterise data

```sql
use Big.University_DB
mine characteristics as "Science_Students"
in relevance to
name, gender, major, birth_date, residence, phone#, gpa
from student
where status in graduate
```
Mining Data Descriptive Characteristics

- **Motivation**
  - To better understand the data: central tendency, variation and spread

- **Data dispersion characteristics**
  - median, max, min, quantiles, outliers, variance, etc.

- **Numerical dimensions correspond to sorted intervals**
  - Data dispersion: analyzed with multiple granularities of precision
  - Boxplot or quantile analysis on sorted intervals

- **Dispersion analysis on computed measures**
  - Folding measures into numerical dimensions
  - Boxplot or quantile analysis on the transformed cube
Measuring the Central Tendency

- **Mean** (algebraic measure) (sample vs. population):
  \[ \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \quad \mu = \frac{\sum x}{N} \]
  - Weighted arithmetic mean:
  \[ \bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \]
  - Trimmed mean: chopping extreme values

- **Median**: A holistic measure
  - Middle value if odd number of values, or average of the middle two values otherwise
  - Estimated by interpolation (for grouped data):
    \[ \text{median} = L_1 + \left( \frac{n/2 - (\sum f)l}{f_{\text{median}}} \right)c \]

- **Mode**
  - Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal
  - Empirical formula:
    \[ \text{mean} - \text{mode} = 3 \times (\text{mean} - \text{median}) \]
• Histograms and Probability Density Functions
• Probability Density Functions
  – Total area under curve integrates to 1
• Frequency Histograms
  – Y-axis is counts
  – Simple interpretation
  – Can’t be directly related to probabilities or density functions
• Relative Frequency Histograms
  – Divide counts by total number of observations
  – Y-axis is relative frequency
  – Can be interpreted as probabilities for each range
  – Can’t be directly related to density function
    • Bar heights sum to 1 but won’t integrate to 1 unless bar width = 1
• Density Histograms
  – Divide counts by (total number of observations X bar width)
  – Y-axis is density values
  – Bar height X bar width gives probability for each range
  – Can be directly related to density function
    • Bar areas sum to 1

http://www.geog.ucsb.edu/~joel/g210_w07/lecture_notes/lect04/oh07_04_1.html
histograms

• equal sub-intervals, known as `bins`

• break points

• bin width
The data are (the log of) wing spans of aircraft built in from 1956 - 1984.

Histogram with breaks at n.0 and n.5
binwidth=0.5

The data are (the log of) wing spans of aircraft built in from 1956 - 1984.
Histogram with breaks at n.25 and n.75
binwidth=0.5
Histogram vs kernel density

- properties of histograms with these two examples:
  - they are not smooth
  - depend on end points of bins
  - depend on width of bins

- We can alleviate the first two problems by using kernel density estimators.
- To remove the dependence on the end points of the bins, we centre each of the blocks at each data point rather than fixing the end points of the blocks.
'Histogram' with blocks centred over data points

block of width 1 and height 1/12 (the dotted boxes) as they are 12 data points, and then add them up
• Blocks - it is still discontinuous as we have used a discontinuous kernel as our building block
• If we use a smooth kernel for our building block, then we will have a smooth density estimate.
• It's important to choose the most appropriate bandwidth as a value that is too small or too large is not useful.

• If we use a normal (Gaussian) kernel with bandwidth or standard deviation of 0.1 (which has area 1/12 under the each curve) then the kernel density estimate is said to undersmoothered as the bandwidth is too small in the figure below.

• It appears that there are 4 modes in this density - some of these are surely artifices of the data.
Oversmoothed

Log span

Probability density function
• Choose optimal bandwidth
  – Need to estimate it

• AMISE = Asymptotic Mean Integrated Squared Error
• optimal bandwidth = arg min AMISE
• The optimal value of the bandwidth for our dataset is about 0.25.
• From the optimally smoothed kernel density estimate, there are two modes. As these are the log of aircraft wing span, it means that there were a group of smaller, lighter planes built, and these are clustered around 2.5 (which is about 12 m).
• Whereas the larger planes, maybe using jet engines as these used on a commercial scale from about the 1960s, are grouped around 3.5 (about 33 m).
• The properties of kernel density estimators are, as compared to histograms:
  – smooth
  – no end points
  – depend on bandwidth
Kernel Density estimation

• Gentle introduction

• Tutorial
1-Dimensional Distributions

Click to add points

Bandwidth (width of kernel)
- Set BW: 0.5
- Automatic BW

Bandwidth selector: Local NN BW 1.0
Bandwidth factor: 1.0
Number of neighbours: 4

Kernel type: Uniform

Estimated distribution

P(X)

Entropy of distribution = 6.11
1-Dimensional Distributions
MDL Histogram Density Estimation

Petri Kontkanen, Petri Myllymäki
Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
University of Helsinki and Helsinki University of Technology
P.O.Box 68 (Department of Computer Science)
FIN-00014 University of Helsinki, Finland
{Firstname}.{Lastname}@hiit.fi

Abstract

We regard histogram density estimation as a model selection problem. Our approach is based on the information-theoretic minimum description length (MDL) principle, which can be applied for tasks such as data clustering, density estimation, image denoising and model selection in general. MDL-based model selection is formalized via the normalized maximum likelihood (NML) distribution, which has several desirable optimality properties. We show how this frame-
only on finding the optimal bin count. These regular histograms are, however, often problematic. It has been argued (Rissanen, Speed, & Yu, 1992) that regular histograms are only good for describing roughly uniform data. If the data distribution is strongly non-
uniform, the bin count must necessarily be high if one wants to capture the details of the high density portion of the data. This in turn means that an unnecessary large amount of bins is wasted in the low density region.

To avoid the problems of regular histograms one must allow the bins to be of variable width. For these irregular histograms, it is necessary to find the optimal set
Figure 2: The Gaussian finite mixture densities \( gm6 \) and \( gm8 \) and the NML-optimal histograms with sample size 10000.
R – example (due K. Tretjakov)

d = c(1,2,2,2,1,2,2,2,3,2,3,4,5,4,3,2,3,4,5,6,7);

kernelsmooth <- function(data, sigma, x) {
  result = 0;
  for (d in data) {
    result = result + exp(-(x-d)^2/2/sigma^2);
  }
  result/sqrt(2*pi)/sigma;
}

x = seq(min(d), max(d), by=0.1);
y = sapply(x, function(x) { kernelsmooth(d, 1, x ) });
hist(d);
lines(x,y);
• R tutorial
  – http://cran.r-project.org/doc/manuals/R-intro.html
  – http://www.google.com/search?q=R+tutorial
More links on R and kernel density

- http://sekhon.berkeley.edu/stats/html/density.html
Symmetric vs. Skewed Data

- Median, mean and mode of symmetric, positively and negatively skewed data
Measuring the Dispersion of Data

- Quartiles, outliers and boxplots
  - **Quartiles**: Q₁ (25th percentile), Q₃ (75th percentile)
  - **Inter-quartile range**: IQR = Q₃ – Q₁

- **Five number summary**: min, Q₁, M, Q₃, max
  - **Boxplot**: ends of the box are the quartiles, median is marked, whiskers, and plot outlier individually
  - **Outlier**: usually, a value higher/lower than 1.5 x IQR

- Variance and standard deviation (sample: s, population: σ)
  - **Variance**: (algebraic, scalable computation)
    \[ s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 = \frac{1}{n-1} \left[ \sum_{i=1}^{n} x_i^2 - \frac{1}{n} \left( \sum_{i=1}^{n} x_i \right)^2 \right] \]
    \[ \sigma^2 = \frac{1}{N} \sum_{i=1}^{n} (x_i - \mu)^2 = \frac{1}{N} \sum_{i=1}^{n} x_i^2 - \mu^2 \]
  - **Standard deviation** s (or σ) is the square root of variance s² (or σ²)
Properties of Normal Distribution Curve

- The normal (distribution) curve
  - From $\mu - \sigma$ to $\mu + \sigma$: contains about 68% of the measurements ($\mu$: mean, $\sigma$: standard deviation)
  - From $\mu - 2\sigma$ to $\mu + 2\sigma$: contains about 95% of it
  - From $\mu - 3\sigma$ to $\mu + 3\sigma$: contains about 99.7% of it
Boxplot Analysis

- **Five-number summary** of a distribution:
  - Minimum, Q1, M, Q3, Maximum

- **Boxplot**
  - Data is represented with a box
  - The ends of the box are at the first and third quartiles, i.e., the height of the box is IRQ
  - The median is marked by a line within the box
  - Whiskers: two lines outside the box extend to Minimum and Maximum
Box Plots

  - [http://informationandvisualization.de/blog/box-plot](http://informationandvisualization.de/blog/box-plot)
Jaak Vilo and other authors UT: Data Mining 2009 52
Visualization of Data Dispersion: Boxplot Analysis
A Box Plot can show the difference in variance between replicates
Variance regularization can remove the bias
Quantile Plot

- Displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences)
- Plots quantile information
  - For a data $x_i$, data sorted in increasing order, $f_i$ indicates that approximately 100 $f_i\%$ of the data are below or equal to the value $x_i$
Kemmeren et al. (Mol. Cell, 2002)

- Randomized expression data
- Yeast 2-hybrid studies
- Known (literature) PPI

MPK1 YLR350w
SNF4 YCL046W
SNF7 YGR122W.
Quantile-Quantile (Q-Q) Plot

- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- Allows the user to view whether there is a shift in going from one distribution to another
Scatter plot

- Provides a first look at bivariate data to see clusters of points, outliers, etc.
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane.
Positively and Negatively Correlated Data
Not Correlated Data
Loess Curve

- Adds a smooth curve to a scatter plot in order to provide better perception of the pattern of dependence
- Loess curve is fitted by setting two parameters: a smoothing parameter, and the degree of the polynomials that are fitted by the regression
Elements of microarray statistics

Reference

Test

\[ M = \log_2 R - \log_2 G \]
\[ = \log_2 (R/G) \]

\[ A = \frac{1}{2} (\log_2 R + \log_2 G) \]

MA – plot

\[ M = \log_2 R - \log_2 G \]
\[ A = \frac{1}{2} (\log_2 R + \log_2 G) \]
Normalisation can be used to transform data.
Before and After Normalization

- Two histograms showing data distribution before and after normalization.
- Scatter plots comparing the intensity of self-self hybridization.
- Statistical analyses indicating improvements in data alignment after normalization.
Graphic Displays of Basic Statistical Descriptions

- Histogram: (shown before)
- Boxplot: (covered before)
- Quantile plot: each value $x_i$ is paired with $f_i$ indicating that approximately 100 $f_i\%$ of data are $\leq x_i$
- Quantile-quantile (q-q) plot: graphs the quantiles of one univariant distribution against the corresponding quantiles of another
- Scatter plot: each pair of values is a pair of coordinates and plotted as points in the plane
- Loess (local regression) curve: add a smooth curve to a scatter plot to provide better perception of the pattern of dependence
Chapter 2: Data Preprocessing

- Why preprocess the data?
- Descriptive data summarization
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- 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
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data
  - 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.
How to Handle Missing Data?

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

- Noise: random error or variance in a measured variable
- Incorrect attribute values may 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**
  - 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)
Simple Discretization Methods: 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 the same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky
Binning Methods for Data Smoothing

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

\[ y = x + 1 \]
Cluster Analysis
Data Cleaning as a Process

- Data discrepancy detection
  - Use metadata (e.g., domain, range, dependency, distribution)
  - Check field overloading
  - Check uniqueness rule, consecutive rule and null rule
  - Use commercial tools
    - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
    - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)

- Data migration and integration
  - Data migration tools: allow transformations to be specified
  - ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface

- Integration of the two processes
  - Iterative and interactive (e.g., Potter’s Wheels)
Chapter 2: Data Preprocessing

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

- **Data integration:**
  - Combines data from multiple sources into a coherent store

- **Schema integration:** e.g., A.cust-id ≡ B.cust-#
  - Integrate metadata from different sources

- **Entity identification problem:**
  - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton

- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units
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*
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality
Correlation Analysis (Numerical Data)

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

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

where \( n \) is the number of tuples, \( \bar{A} \) and \( \bar{B} \) are the respective means of \( A \) and \( B \), \( \sigma_A \) and \( \sigma_B \) 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
Correlation Analysis (Categorical Data)

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

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

- The larger the $\chi^2$ value, the more likely the variables are related

- The cells that contribute the most to the $\chi^2$ value are those whose actual count is very different from the expected count

- Correlation does not imply causality
  - # of hospitals and # of car-theft in a city are correlated
  - Both are causally linked to the third variable: population
Chi-Square Calculation: An Example

<table>
<thead>
<tr>
<th></th>
<th>Play chess</th>
<th>Not play chess</th>
<th>Sum (row)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Like science fiction</td>
<td>250(90)</td>
<td>200(360)</td>
<td>450</td>
</tr>
<tr>
<td>Not like science fiction</td>
<td>50(210)</td>
<td>1000(840)</td>
<td>1050</td>
</tr>
<tr>
<td>Sum(col.)</td>
<td>300</td>
<td>1200</td>
<td>1500</td>
</tr>
</tbody>
</table>

- \( \chi^2 \) (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
Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute/feature construction
  - New attributes constructed from the given ones
Data Transformation: Normalization

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

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

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

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

  - Ex. Let $\mu = 54,000$, $\sigma = 16,000$. Then

$$\frac{73,600 - 54,000}{16,000} = 1.225$$

- Normalization by decimal scaling

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

  Where $j$ is the smallest integer such that $\text{Max}(|v'|) < 1$
Chapter 2: Data Preprocessing

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

- Why data reduction?
  - A database/data warehouse may store terabytes of data
  - Complex data analysis/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 (or almost the same) analytical results
- Data reduction strategies
  - Data cube aggregation:
  - Dimensionality reduction — e.g., remove unimportant attributes
  - Data Compression
  - Numerosity reduction — e.g., fit data into models
  - Discretization and concept hierarchy generation
Data Cube Aggregation

- The lowest level of a data cube (base cuboid)
  - The aggregated data for an individual entity of interest
  - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible
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
- reduce # of patterns in the patterns, easier to understand

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

Initial attribute set:
\{A1, A2, A3, A4, A5, A6\}

Reduced attribute set:  \{A1, A4, A6\}
Heuristic Feature Selection Methods

- There are $2^d$ possible sub-features of $d$ features
- Several heuristic feature selection methods:
  - Best single features under the feature independence assumption: choose by significance tests
  - 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
  - Optimal branch and bound:
    - Use feature elimination and backtracking
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

Original Data

Compressed Data

lossless

Original Data Approximated

lossy

September 17, 2009

Data Mining: Concepts and Techniques

93
Dimensionality Reduction: Wavelet Transformation

- Discrete wavelet transform (DWT): linear signal processing, multi-resolutional analysis
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space
- Method:
  - Length, $L$, must be an integer power of 2 (padding with 0’s, when necessary)
  - Each transform has 2 functions: smoothing, difference
  - Applies to pairs of data, resulting in two set of data of length $L/2$
  - Applies two functions recursively, until reaches the desired length
Dimensionality Reduction: Principal Component Analysis (PCA)

- Given $N$ data vectors from $n$-dimensions, find $k \leq n$ orthogonal vectors (principal components) that can be best used to represent data.

Steps

- Normalize input data: Each attribute falls within the same range.
- Compute $k$ orthonormal (unit) vectors, i.e., principal components.
- Each input data (vector) is a linear combination of the $k$ principal component vectors.
- The principal components are sorted in order of decreasing “significance” or strength.
- Since the components are sorted, the size of the data can be reduced by eliminating the weak components, i.e., those with low variance. (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data.

- Works for numeric data only.
- Used when the number of dimensions is large.

Works for numeric data only
Principal Component Analysis
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 (1): 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**: \( Y = b_0 + b_1 X_1 + b_2 X_2 \)
  - 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 (2): Histograms

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

- Partitioning rules:
  - Equal-width: equal bucket range.
  - Equal-frequency (or equal-depth).
  - V-optimal: with the least histogram variance (weighted sum of the original values that each bucket represents).
  - MaxDiff: set bucket boundary between each pair for pairs having the β−1 largest differences.
Data Reduction Method (3): Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms
- Cluster analysis will be studied in depth in Chapter 7
Data Reduction Method (4): Sampling

- Sampling: obtaining a small sample $s$ to represent the whole data set $N$
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Used in conjunction with skewed data
- Note: Sampling may not reduce database I/Os (page at a time)
Sampling: with or without Replacement

Raw Data

SRSWOR (simple random sample without replacement)

SRSWR
Sampling: Cluster or Stratified Sampling

Raw Data

Cluster/Stratified Sample
Chapter 2: Data Preprocessing

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

Three types of attributes:

- Nominal — values from an unordered set, e.g., color, profession
- Ordinal — values from an ordered set, e.g., military or academic rank
- 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
Discretization and Concept Hierarchy

- Discretization
  - Reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute

- Concept hierarchy formation
  - Recursively reduce the data by collecting and replacing low level concepts (such as numeric values for age) by higher level concepts (such as young, middle-aged, or senior)
Discretization and Concept Hierarchy Generation for Numeric Data

- Typical methods: All the methods can be applied recursively
  - Binning (covered above)
    - Top-down split, unsupervised,
  - Histogram analysis (covered above)
    - Top-down split, unsupervised
  - Clustering analysis (covered above)
    - Either top-down split or bottom-up merge, unsupervised
  - Entropy-based discretization: supervised, top-down split
  - Interval merging by $\chi^2$ Analysis: unsupervised, bottom-up merge
  - Segmentation by natural partitioning: top-down split, unsupervised
Entropy-Based Discretization

- Given a set of samples $S$, if $S$ is partitioned into two intervals $S_1$ and $S_2$ using boundary $T$, the information gain after partitioning is
  \[ I(S, T) = \frac{|S_1|}{|S|} \text{Entropy} (S_1) + \frac{|S_2|}{|S|} \text{Entropy} (S_2) \]

- Entropy is calculated based on class distribution of the samples in the set. Given $m$ classes, the entropy of $S_1$ is
  \[ \text{Entropy} (S_1) = - \sum_{i=1}^{m} p_i \log_2 (p_i) \]
  where $p_i$ is the probability of class $i$ in $S_1$

- The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization

- The process is recursively applied to partitions obtained until some stopping criterion is met

- Such a boundary may reduce data size and improve classification accuracy
Interval Merge by $\chi^2$ Analysis

- Merging-based (bottom-up) vs. splitting-based methods
- Merge: Find the best neighboring intervals and merge them to form larger intervals recursively
- ChiMerge [Kerber AAAI 1992, See also Liu et al. DMKD 2002]
  - Initially, each distinct value of a numerical attr. $A$ is considered to be one interval
  - $\chi^2$ tests are performed for every pair of adjacent intervals
  - Adjacent intervals with the least $\chi^2$ values are merged together, since low $\chi^2$ values for a pair indicate similar class distributions
  - This merge process proceeds recursively until a predefined stopping criterion is met (such as significance level, max-interval, max inconsistency, etc.)
Segmentation by Natural Partitioning

- A simply 3-4-5 rule can be used to segment numeric data into relatively uniform, “natural” intervals.
  - If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit, partition the range into 3 equi-width intervals
  - If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into 4 intervals
  - If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into 5 intervals
Example of 3-4-5 Rule

Step 1:
- $351 - $159 profit $1,838 $4,700

Step 2:
- Min $1,000 Low (i.e, 5%-tile $1,000 High $2,000

Step 3:
- (-$1,000 - $2,000)
- (-$1,000 - 0)
- (0 - $1,000)
- ($1,000 - $2,000)

Step 4:
- (-$400 - $5,000)
  - (-$400 - 0)
    - (-$400 - $300)
    - (-$300 - $200)
    - (-$200 - $100)
    - (-$100 - 0)
  - (0 - $1,000)
    - ($600 - $800)
    - ($800 - $1,000)
  - ($1,000 - $2,000)
    - ($1,600 - $1,800)
    - ($1,800 - $2,000)
  - ($2,000 - $5,000)
    - ($2,000 - $3,000)
    - ($3,000 - $4,000)
    - ($4,000 - $5,000)
Example

-351,976.00 .. 4,700,896.50

MIN=-351,976.00
MAX=4,700,896.50
LOW = 5th percentile -159,876
HIGH = 95th percentile 1,838,761
msd = 1,000,000 (most significant digit)
LOW = -1,000,000 (round down) HIGH = 2,000,000 (round up)

3 value ranges
1. (-1,000,000 .. 0]
2. (0 .. 1,000,000]
3. (1,000,000 .. 2,000,000]

Adjust with real MIN and MAX
1. (-400,000 .. 0]
2. (0 .. 1,000,000]
3. (1,000,000 .. 2,000,000]
4. (2,000,000 .. 5,000,000]
Recursive …

1.1. (-400,000 .. -300,000 ]
1.2. (-300,000 .. -200,000 ]
1.3. (-200,000 .. -100,000 ]
1.4. (-100,000 .. 0 ]

2.1. (0 .. 200,000 ]
2.2. (200,000 .. 400,000 ]
2.3. (400,000 .. 600,000 ]
2.4. (600,000 .. 800,000 ]
2.5. (800,000 .. 1,000,000 ]

3.1. (1,000,000 .. 1,200,000 ]
3.2. (1,200,000 .. 1,400,000 ]
3.3. (1,400,000 .. 1,600,000 ]
3.4. (1,600,000 .. 1,800,000 ]
3.5. (1,800,000 .. 2,000,000 ]

4.1. (2,000,000 .. 3,000,000 ]
4.2. (3,000,000 .. 4,000,000 ]
4.3. (4,000,000 .. 5,000,000 ]
Concept Hierarchy Generation for Categorical Data

- Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts
  - street < city < state < country
- Specification of a hierarchy for a set of values by explicit data grouping
  - \{Urbana, Champaign, Chicago\} < Illinois
- Specification of only a partial set of attributes
  - E.g., only street < city, not others
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values
  - E.g., for a set of attributes: \{street, city, state, country\}
Automatic Concept Hierarchy Generation

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

```
country                      | 15 distinct values
province or state            | 365 distinct values
city                         | 3567 distinct values
street                       | 674,339 distinct values
```
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
- A lot of methods have been developed but data preprocessing still an active area of research.
References

- 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