Introduction to Data Mining

Part 3: Data Processing

Spring 2015

Ming Li

Department of Computer Science and Technology
Nanjing University
The real world data

• Data in the real world is dirty
  – *incomplete*
    • lacking attribute value, lacking certain attributes
    • of interest, containing only aggregate data
  – *noisy*
    • containing errors or outliers
  – *inconsistent*
    • containing (semantic) discrepancies
Why incomplete?

- **mainly from data collection**
  - attributes of interest may not always be available
    - *e.g., age of a female customer*
  - relevant data may have been considered unimportant at the time of entry
    - *e.g., height of a customer*
  - malfunctions of data entry equipment
  - inconsistent data may have been removed
  - ......
Why noisy?

- mainly from data collection
  - data collection instruments may not be accurate  
    *e.g., measuring wind speed with stopwatch*
  - human or computer errors occurring at data entry
  - errors in data transmission  *e.g., network error*
  - resulting from inconsistencies
  - ......
Why inconsistent?

- mainly from data integration
  - an attribute may have different names in different databases  
    *e.g., customer, cust, and cust_id*
  - a value of an attribute may have different measures in different databases  *e.g., 1.75m and 175cm*
  - ......

Data integration may also encounter redundancy
some attributes may be derived or inferred from others
Why data preprocessing?

No quality data, no quality mining results

• How about mining those data directly?
  – Although most mining routines have some ability of dealing with incomplete or noisy data, they are not always robust
  – Most mining routines cannot deal with inconsistency
  – Even without incompleteness, noise, and inconsistency, the size of the data may be too large to mine

Data preprocessing usually occupies about 70% workload in a data mining task
Major tasks of data preprocessing (I)

- Data cleaning （数据清洗）
  filling in missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies
Major tasks of data preprocessing (II)

• Data integration
  (数据集成)
  integrating multiple databases, data cubes, files, etc.

• Data transformation
  (数据转换)
  normalization and aggregation
  
  $$(-2, 32, 100, 59, 48) \rightarrow (-0.02, 0.32, 1.00, 0.59, 0.48)$$
Major tasks of data preprocessing (III)

- Data reduction (数据约简)

obtains reduced representation in volume which could produce the same or similar analytical results
Data cleaning

Data cleaning attempts to do:

- filling in missing values
- smoothing out noisy data
- correcting inconsistent data
Filling in missing data (I)

- Ignore the tuple
  - poor when the percentage of missing values per attribute
  - varies considerably

- Fill in the missing value manually
  - tedious
  - may be infeasible

- Use a global constant  
  *e.g., “unknown”*

  the mining procedure may incorrectly regard the filled value as an interesting concept
Filling in missing data (II)

• Use the attribute mean
  use the mean of the attribute across all the samples to fill in the missing value

• Use the attribute mean of the same class
  use the mean of the attribute across the samples belonging to the same class to fill in the missing value

• Use the most probable value
  model the attribute with other available attributes
  smarter strategy, which uses the most information contained in the available data
Smoothing out noisy data (I)

What is noise?

Noise is a *random error* or *variance* in a measured variable

Property:
- *Noise is much less than normal data*
- *Noise is much different from the normal data*

**Key:** Exploiting the Local and Global Consistency of Data
Smoothing out noisy data (II)

- **Binning:**
  - Smooth a sorted data value by consulting its “neighborhood”

  **partitioning**

  **equi-depth** bins:
  - bin 1: 4, 8, 15
  - bin 2: 21, 21, 24
  - bin 3: 25, 28, 34
  - equi-depth: each bin contains roughly **same number** of values

  **equi-width** bins:
  - bin 1: 4, 8  [4, 13)
  - bin 2: 15, 21, 21  [14, 23)
  - bin 3: 24, 25, 28, 34  [24, 34]
  - equi-width: each bin contains roughly **same range** of values

  **smoothing**

  **by bin means:**
  - 9, 9, 9, 22, 22, 22, 29, 29, 29
  - each value in a bin is replaced by the mean value of the bin

  **by bin medians:**
  - 8, 8, 8, 21, 21, 21, 28, 28, 28
  - each value in a bin is replaced by the median value of the bin

  **by bin boundaries:**
  - 4, 4, 15, 21, 21, 24, 25, 25, 34
  - each value in a bin is replaced by the closet boundary value of the bin
Smoothing out noisy data (III)

• Clustering
  organize similar values into groups or clusters
  values falling out of the set of clusters may be regarded as outliers
Smoothing out noisy data (IV)

• Regression

fit the data to a mathematical function

values deviate from the expected values for some distance may be regarded as outliers

\[ y = x + 1 \]
Smoothing out noisy data (V)

- Data editing
  - Exploiting the *local* “smoothness”
    - *Use the value derived from the neighborhood of a data point*
    - *Exploiting the local smoothness based on cut edges over a data graph* [e.g., Li & Zhou, PAKDD’05]
  - Exploiting *global* consistency with a noise tolerant model
    - *Use a neural network ensemble filter the noise* [e.g., Jiang & Zhou, ISNN’04]
Correcting inconsistent data

- Some data inconsistencies may be corrected manually using external references
  
  *e.g., errors made at data entry may be corrected by examining a paper trace*

- Knowledge engineering tools may also be used to detect the violation of known data constraints

In general, correcting inconsistent data requires user intervention
Data integration (I)

Combine data from multiple sources into a coherent data store

• Schema integration
  
  – *entity identification problem*: How can entities from multiple data sources be “matched up”? *e.g., family name vs. surname*
  
  – *semantic heterogeneity problem*: for the same real world entity, due to differences in representation, scaling, or encoding, attribute values from different sources may be different *e.g., 175cm vs. 1.75m*

*Metadata can be used to help schema integration!*
Data integration (II)

• Handling redundancy

Redundancy often occurs in the integration of multiple databases
• the same attribute may have different names in different databases
• an attribute may be redundant if it can be derived from other attributes
• some tuples may be duplicated

Some redundancies can be detected by correlation analysis

The correlation between attributes A and B can be measured by

\[ \text{cor} = \frac{\text{Cov}(A, B)}{\sigma_A \sigma_B} = \frac{\sum (A - \bar{A})(B - \bar{B})}{N \sigma_A \sigma_B} \]

• cor = 0 A and B are uncorrelated  
  e.g., A: rain B: good TV program
  note: uncorrelated is not equal to independent
• cor > 0 A and B are positively correlated  
  e.g., A: rain B: cloudy
• cor < 0 A and B are negatively correlated  
  e.g., A: rain B: sunshine

The larger the value of cor, the more each attribute implies the other
Data transformation

• **Normalization**
  scale the attribute values to a small specified range

• **Smoothing**
  remove noise from the data

• **Aggregation**
  summarize the data

• **Generalization**
  climb concept hierarchy

• **Attribute construction**
  construct new attributes
Normalization (I)

- **min-max normalization**

\[ v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new}_A - \text{new}_A) + \text{new}_A \]

comes from

\[ \frac{v - \min_A}{\max_A - \min_A} = \frac{v' - \text{new}_A}{\text{new}_A - \text{new}_A} \]

- preserves the relationship among the original data values
- may encounter “out of boundary” error
- linear
Normalization (II)

• **z-score normalization (zero-mean normalization)**

\[ v' = \frac{v - \mu}{\sigma_v} \]

- particularly useful when the actual min and max of the attribute value are unknown, or when there are outliers dominating the min-max normalization
- change the original data

• **decimal scaling normalization**

\[ v' = \frac{v}{10^j} \]

where \( j \) is the smallest integer such that \( \max(|v'|) < 1 \)  

- very simple
- change the original data
Data reduction

Obtain a reduced representation of the data set that is much smaller in volume, yet produce the same (or almost same) analytical results.

Data reduction strategies:

• data cube aggregation
• feature selection
• dimensionality reduction
• numerosity reduction
• discretization
• concept hierarchy generalization analytical results
Feature selection

Remove irrelevant attributes from the data

- excluding relevant attributes is harmful to the mining process
- including irrelevant attributes may confuse the mining process
- including redundant attributes may slow down the mining process

*(but sometimes including redundant attributes may be helpful)*

**Attribute Subset Selection**

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 the attributes

exhaustive search is usually prohibitively expensive, heuristic methods are commonly used

the attributes are typically selected using relevance measures such as information gain
Attribute subset selection: Greedy approach (I)

• Step-wise forward selection
  – select the best of the original attributes
  – select the best of the remaining original attributes
  – ……

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

initial selected set: \{\}

step 1: \{A1\}

step 2: \{A1, A4\}

step 3: \{A1, A4, A6\}
Attribute subset selection: Greedy approach (II)

• Step-wise backward elimination
  – eliminate the worst of the original attributes
  – eliminate the worst of the remaining original attributes
  – ……

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

initial selected set: \{A1, A2, A3, A4, A5, A6\}

step 1: \{A1, A3, A4, A5, A6\}

step 2: \{A1, A4, A5, A6\}

step 3: \{A1, A4, A6\}
Attribute subset selection: Greedy approach (III)

- Combined forward selection and backward elimination
  - select the best of the original attributes and eliminate the worst of the remaining original attributes
  - ……

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

initial selected set: \{A1, A2, A3, A4, A5, A6\}

step 1: \{A1, A3, A4, A5, A6\}

step 2: \{A1, A4, A5, A6\}

step 3: \{A1, A4, A6\}
Attribute subset selection: Greedy approach (IV)

- Decision tree induction
  - train a decision tree
  - remove the attributes that have not appeared in the tree

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

selected set: \{A1, A4, A6\}
Attribute subset selection: Beyond Greedy

- **Genetic Algorithm**

  use genetic algorithm to evolve the best combination of attributes

  - represent models as chromosomes (binary string)
  - evolve a population of such chromosomes by selectively pairing (according to their fitness defined by a score function) and mutate chromosomes to create off-springs

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

  **Best chromosome code**
  \[1, 0, 0, 1, 0, 1\]

  **selected set:** \{A1, A4, A6\}
The general idea behind attribute relevance analysis is to compute some measure which is used to quantify the relevance of an attribute with respect to a given class.

Popular measures include:

- information gain
- gain ratio
- gini index
- uncertainty
- correlation coefficients
- ....
Information gain measure

$S$: training set
$S_i$: training instances of class $C_i (i = 1, \ldots, m)$
$a_j$: values of attribute $A (j = 1, \ldots, v)$

the information needed to correctly classify the training set is

$$I(S_1, S_2, \ldots, S_m) = - \sum_{i=1}^{m} \frac{S_i}{S} \log_2 \frac{S_i}{S}$$

suppose attribute $A$ is selected to partition the training set into the subsets $\{S^A_1, S^A_2, \ldots, S^A_v\}$, then the entropy of the subsets, i.e., the information needed to classify all the instances in those subsets is

$$Ent(A) = \sum_{i=1}^{v} \frac{S^A_i}{S} \left( - \sum_{j=1}^{m} \frac{S^A_{ij}}{S^A_i} \log_2 \frac{S^A_{ij}}{S^A_i} \right)$$

where $S^A_{ij}$ is the instances of class $C_j$ contained in $S^A_i$.

then the information gain of selecting $A$ is

$$Gain(A) = I(S_1, S_2, \ldots, S_m) - Ent(A)$$

the bigger the information gain, the more relevant the attribute $A$
Example of information gain (I)

Target class: Graduate students ($\sum = 120$)

<table>
<thead>
<tr>
<th>gender</th>
<th>Major</th>
<th>birth_country</th>
<th>age_range</th>
<th>gpa</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Science</td>
<td>Canada</td>
<td>20-25</td>
<td>Very_good</td>
<td>16</td>
</tr>
<tr>
<td>F</td>
<td>Science</td>
<td>Foreign</td>
<td>25-30</td>
<td>Excellent</td>
<td>22</td>
</tr>
<tr>
<td>M</td>
<td>Engineering</td>
<td>Foreign</td>
<td>25-30</td>
<td>Excellent</td>
<td>18</td>
</tr>
<tr>
<td>F</td>
<td>Science</td>
<td>Foreign</td>
<td>25-30</td>
<td>Excellent</td>
<td>25</td>
</tr>
<tr>
<td>M</td>
<td>Science</td>
<td>Canada</td>
<td>20-25</td>
<td>Excellent</td>
<td>21</td>
</tr>
<tr>
<td>F</td>
<td>Engineering</td>
<td>Canada</td>
<td>20-25</td>
<td>Excellent</td>
<td>18</td>
</tr>
</tbody>
</table>

Contrasting class: undergraduate students ($\sum = 130$)

<table>
<thead>
<tr>
<th>gender</th>
<th>major</th>
<th>birth_country</th>
<th>age_range</th>
<th>gpa</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Science</td>
<td>Foreign</td>
<td>&lt;20</td>
<td>Very_good</td>
<td>18</td>
</tr>
<tr>
<td>F</td>
<td>Business</td>
<td>Canada</td>
<td>&lt;20</td>
<td>Fair</td>
<td>20</td>
</tr>
<tr>
<td>M</td>
<td>Business</td>
<td>Canada</td>
<td>&lt;20</td>
<td>Fair</td>
<td>22</td>
</tr>
<tr>
<td>F</td>
<td>Science</td>
<td>Canada</td>
<td>20-25</td>
<td>Fair</td>
<td>24</td>
</tr>
<tr>
<td>M</td>
<td>Engineering</td>
<td>Foreign</td>
<td>20-25</td>
<td>Very_good</td>
<td>22</td>
</tr>
<tr>
<td>F</td>
<td>Engineering</td>
<td>Canada</td>
<td>&lt;20</td>
<td>Excellent</td>
<td>24</td>
</tr>
</tbody>
</table>
Example of information gain (II)

the information needed to correctly classify the training set is

\[ I(S_1, S_2) = I(120, 130) = - \frac{120}{250} \log_2 \frac{120}{250} - \frac{130}{250} \log_2 \frac{130}{250} = 0.9988 \]

suppose attribute *major* is selected to partition the training set

for *major* = "Science":

\[ S_{11} = 84, S_{12} = 42 \quad I(S_{11}, S_{12}) = I(84, 42) = - \frac{84}{126} \log_2 \frac{84}{126} - \frac{42}{126} \log_2 \frac{42}{126} = 0.9183 \]

for *major* = "Engineering":

\[ S_{21} = 36, S_{22} = 46 \quad I(S_{21}, S_{22}) = I(36, 46) = - \frac{36}{82} \log_2 \frac{36}{82} - \frac{46}{82} \log_2 \frac{46}{82} = 0.9892 \]

for *major* = "Business":

\[ S_{31} = 0, S_{32} = 42 \quad I(S_{31}, S_{32}) = 0 \]

then the entropy of *major* is

\[ E(major) = \frac{126}{250} I(S_{11}, S_{12}) + \frac{82}{250} I(S_{21}, S_{22}) + \frac{42}{250} I(S_{31}, S_{32}) = 0.7873 \]
then the information gain of major is

\[ \text{Gain}(\text{major}) = I(S_1, S_2) - E(\text{major}) = 0.2115 \]

we can also get the information gain of other attributes:

\[ \text{Gain}(\text{gender}) = 0.0003 \]
\[ \text{Gain}(\text{birth\_country}) = 0.0407 \]
\[ \text{Gain}(\text{gpa}) = 0.4490 \]
\[ \text{Gain}(\text{age\_range}) = 0.5971 \]

now suppose we use an attribute relevance threshold of 0.1: 

_gender and birth\_country are removed as weakly relevant attributes _major, gpa, and age\_range are kept as strong relevant attributes
Categorization of FS approaches

A popular categorization of feature selection (attribute selection) approaches:

• wrapper approach
  the learner itself is used to determine the attribute subset
  popular scheme: Using the learner to evaluate the goodness of different attribute subsets

• filter approach
  the learner itself is not involved in the feature selection process

Usually, wrapper approach leads to better accuracy but requires more computational cost
Dimensionality reduction

Reduce the dimensionality of the data,-reserving characteristics that are important to the concerned task

*Try to identify some “hidden” aspects that determines the characteristics of the data*

**Generally by using some mathematical projection**

- **Linear methods (using linear projection)**
  
e.g., PCA, LDA

- **Nonlinear methods (using nonlinear projection)**
  
e.g., KPCA, KLDA
  *manifold learning* methods: ISOMAP, LLE
Principal component analysis (PCA)

What *hidden aspect* that control the data?

The spreading tendency
Principal component analysis (PCA)

**Principal component analysis seeks a space of lower dimensionality, in which the variance of the projected data are maximized.**

- try to model how data are spread by maximizing the variance of the projected data
- find $d$ orthogonal vectors that represent the “spread tendency” of the data ($d << D$)
- the $d$ vectors will be used as the bases of the new space
Principal component analysis (PCA)

- **Data representations**
  \[
  X = [x_1, \ldots, x_N], \quad x_n \in \mathbb{R}^D, \quad n = 1, \ldots, N
  \]
  where
  \[
  \sum_{n=1}^{N} x_n = 0 \quad \text{zero-mean}
  \]

- **Projection of a data point**
  \[
  y = w^T x = \sum_{j=1}^{d} w(j)x(j)
  \]

- **Projection of the data set**
  \[
  y^T = w^T X
  \]
Principal component analysis (PCA)

- **Variance in projected space**
  \[
  \sigma_w^2 = \frac{1}{n} y^\top y \\
  = \frac{1}{n} (w^\top X) (w^\top X)^\top \\
  = \frac{1}{n} w^\top XX^\top w \\
  = w^\top Sw
  \]

- **Maximization**
  \[
  \max_w w^\top Sw \\
  \text{s.t. } w^\top w = 1
  \]

- **Solution:**
  1. by introducing Lagrange multiplier
     \[
     Sw = \lambda w
     \]
  2. Solve the equation by **Eigen decomposition**
Principal component analysis (PCA)

• **Find** $d$ orthogonal principal components

  It can be shown that the $d$ orthogonal principle components correspond to the $d$ eigen vectors with $d$ largest eigen values

**Intuitions:**

– *Eigen value is the variance of the data projected to the corresponding eigen vector*
– *Eigen vectors are essentially orthogonal*

• **Determine “$d$”**

  – **What to discard**: the eigen vectors with small eigen values
  – Limit the loss of information within an acceptable interval (usually 5%)
Principal component analysis (PCA)

- **Conducting PCA**

  1. Shifting data set to have zero-means
  2. Compute the covariance matrix $S$
  3. Conduct eigen decomposition on $S$, and rank the eigen vectors according to their eigen values
  4. Determine the number of dimensions $d$ of the new space
  5. Select the first $d$ eigen vectors with $d$ largest eigen values as the basis of the new space
Applications of PCA

- Reduce data for storing and visualization
  - Data are projected into 2D or 3D subspace using PCA in order to visualize the projected data for domain experts

- For subsequent data mining algorithms
  E.g. face recognition

Eigen face
How to reduce such data?

Problem: Data do NOT spread *linearly*

PCA that finds the linear spreading tendency of data will fail on such data

Swiss Roll
Isometric feature mapping (ISOMAP)

- It seeks a low dimensional embedding of data such that the distance in the new space roughly equals the geodesic distances constructed through neighborhoods of instances.

**Key step:** Find the shortest path between any two points on a neighborhood graph.

---

**Fig. 3.** The “Swiss roll” data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (A) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (B) The neighborhood graph $G$ constructed in step one of Isomap (with $K = 7$ and $N = 1000$ data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in $G$. (C) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).
Wavelet transformation

• Discrete Wavelet Transform (DWT): linear signal processing technique

• 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

pyramid algorithm

• length, $L$, must be an integer power of 2 (padding with 0s when necessary)
• each transformation involves two functions: smoothing and difference
• applies to pairs of data $(x_{2i}, x_{2i+1})$, resulting in two sets of data of length $L/2$
• applies two functions recursively, until the resulting data sets obtained are of length 2
• a selection of values are designated the wavelet coefficients
Numerosity reduction

Reduce the data volume by choosing alternative, “smaller” form of data representation

- **parametric**
  - use a mathematical model to fit the data, so that only the parameters of the data need to be stored

- **non-parametric**
  - histogram, clustering, sampling, etc.
Regression models

• **linear regression**
  
  data are modeled to fit a straight line
  
  \[ Y = \alpha + \beta X \]

• **multiple regression**
  
  a response variable \( Y \) is modeled as a linear function of multiple features
  
  \[ Y = \alpha + \beta X_1 + \chi X_2 + \delta X_3 + \ldots \]
Histograms (I)

Partition the data distribution into disjoint buckets

- **equi-depth**
  
each bucket contains roughly same *frequency* of values

- **equi-width**
  
each bucket contains roughly same *range* of values

- **v-optimal**
  
  the histogram with the least variance
  
histogram variance is the weighted sum of the variance of the original values that each bucket contains

- **maxdiff**
  
bucket boundary is established between each pair for pairs having the \( \beta - 1 \) largest differences
  
  \( \beta \) determines the number of buckets
**Histograms (II)**

an example: partition \{1, 5, 8, 10\} into 3 buckets

- **equi-depth**
  - \{1\} \{5\} \{8, 10\}
  - or \{1\} \{5, 8\} \{10\}
  - or \{1, 5\} \{8\} \{10\}

- **equi-width**
  - \{1\} \{5\} \{8, 10\}
  - in fact: [1, 4), [4, 7), [7, 10]

- **maxdiff**
  - \{1\} \{5\} \{8, 10\}

\[ v = 0 + 0 + 2 \times 2 = 4 \]
\[ v = 0 + 4.5 \times 2 + 0 = 9 \]
\[ v = 8 \times 2 + 0 + 0 = 16 \]

**v-optimal**

\{1\} \{5\} \{8, 10\}

**beacuse**

in general, v-optimal and maxdiff histograms are more accurate

histograms for single attributes can be generalized to multidimensional histograms

**v-optimal uses weighted variance**

\[ \text{variance} = \frac{1}{n-1} \sum_{i} (x_i - \bar{x})^2 \]
Clustering

Partition the objects into clusters, so that objects within a cluster are similar to each other while dissimilar to objects in other clusters

The quality of a cluster may be measured by:

- **Diameter**: the maximum distance between any two objects in the cluster.
- **Centroid distance**: the average distance of each cluster object from the cluster centroid.
Use a much smaller random sample to represent a large data set

- random sampling without replacement

- random sampling with replacement (bootstrap sampling)
Sampling (II)

- cluster sampling

- stratified sampling (分层采样)
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.

Types of attributes:

- Categorical (nominal)
  - e.g., {“apple”, “banana”, “strawberry”} \{1, 2, 3\}
- Continuous (numerical)
  - e.g., \[0.1, 0.9\]

or

- Unordered
  - e.g., {“apple”, “banana”, “strawberry”}
- Ordered
  - e.g., \[0.1, 0.9\] \{1, 2, 3\}
**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 required for further partitioning $S_1$ and $S_2$ is

\[
I(S,T) = \frac{|S_1|}{S} \text{Ent}(S_1) + \frac{S_2}{S} \text{Ent}(S_2)
\]

- the boundary that minimizes the information $I(S,T)$ over all possible boundaries is selected as a binary discretization for $S$

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

\[\text{e.g. } \text{Ent}(S) - I(S,T) > \delta\]
Motivation:

to better understand the data: central tendency, dispersion

dispersion: the degree to which numeric data tend to spread

measures for central tendency:  measures for dispersion:

• mean
• median
• mode
• midrange
• ......
Measuring the central tendency

- **Mean**
  
  \[ x = \frac{1}{n} \sum_{i=1}^{n} x_i \]

  weighted arithmetic mean (algebraic)

  \[ x = \frac{\sum_{i=1}^{n} w_ix_i}{\sum_{i=1}^{n} w_i} \]

- **Median**
  
  middle value if odd number of values, or average of the middle two values otherwise (holistic)

- **Mode**
  
  value that occurs most frequently in the data set (holistic)

  unimodal, bimodal, trimodal, multimodal, no mode

- **Midrange**
  
  the average of the min and max values (algebraic)
Measuring the dispersion

- **Quartiles**
  
  Quartiles (holistic) $Q_1(25^{th} \text{ percentile}), Q_3(75^{th} \text{ percentile})$
  
  Inter-quartile range (holistic): $IQR = Q_3 - Q_1$
  
  Five number summary (holistic): min, $Q_1$, median, $Q_3$, max

- **Variance**

  $$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 = \frac{1}{n-1} \left[ \sum_{i=1}^{n} x_i^2 - \frac{1}{n} \left( \sum_{i=1}^{n} x_i \right)^2 \right] \quad \text{(algebraic)}$$

- **Standard deviation**

  the square root of $s^2$ (algebraic)
Graph displays of descriptive summaries – for single attribute (I)

- **Box plots**
  - graphically depicting groups of numerical data through their five-number summaries
  
  **Five number summary:**
  
  (Min, Q1, Median, Q3, Max)
  
  - **Functionality:** identifying possible outliers
Graph displays of descriptive summaries – for single attribute (II)

- **Histogram**
  - The number of different values of a nominal attribute
  - The number of values of an numerical attribute lie in consecutive intervals

Random fluctuations and alternative choices for ends may affect the diagram if the data set is small

bar chart is a special case of histogram, where each rectangle is of uniform width
Graph displays of descriptive summaries – for single attribute (III)

- Quartile plot

For data $x_i$ sorted in increasing order, $f_i$ indicates that approximately $100 \times f_i\%$ of the data are smaller than or equal to the value $x_i$. 
Graph displays of descriptive summaries – for pair of attributes (I)

- Scatterplot

  It is a bimodality plot, where each pair of values belonging to the same instance is treated as a 2-d coordinate

  **Functionality**: revealing certain the correlation of the two variables

  Might not be useful especially for large data set (with long-tailed distribution)

  The bimodality (0-1) representation ignores the frequency of certain coordinate appears.
Graph displays of descriptive summaries – for pair of attributes (II)

- Loess (local regression) curve

Loess curve is added a smooth curve to a scatter plot in order to provide better perception of the pattern of dependence.

**Functionality:** Providing better perception of the pattern of dependence.
Graph displays of descriptive summaries – for pair of attributes (III)

- Q-Q plot (Quantile-Quantile plot)

  the quantiles of one univariate distribution against the corresponding quantiles of another one
Graph displays of descriptive summaries – for pair of attributes (IV)

- **Contour plot**

  It plots a 2-d density contour with respect to the concerned two variables, where the density is estimated from the observed data points.

  **Functionality:** revealing the correlation of two variables in terms of the joint distribution
Graph displays of descriptive summaries – for multiple attributes (I)

- Scatterplot matrix (pseudo-multivariate)

  Aligning scatterplots for every pair of attributes

  **Functionality**: Revealing certain correlation of any two attributes

  *A pseudo-multivariate tool: since it is multiple bivariate solutions*
Graph displays of descriptive summaries – for multiple attributes (II)

- **Trellis plot**
  - Fixing a particular pair of attributes that is to be displayed
  - produces a series of scatterplots conditioned on levels of one or more other attributes

  **Functionality:** Revealing certain correlation of any two attributes with consideration of other attributes’ values

  - **Considers multivariate to some extent**
  - **Any type of graph can be used besides scatter plots**
Graph displays of descriptive summaries – for multiple attributes (III)

- Icons plots

It represents each instance as a multidimensional symbol

**Functionality:** Providing a selection of instances by revealing its multivariate correlation

- More difficult to read.
- Failing to apply to large data set
- Revealing individual characteristics instead of global distributional information
Graph displays of descriptive summaries – for multiple attributes (IV)

• Icons plots
Graph displays of descriptive summaries – for multiple attributes (V)

- Parallel coordinates plot

  It represents each instance as a piecewise linear plot connecting the measured values for that instance.

  **Functionality:** Providing a selection of instances by revealing its multivariate correlation.

  - More difficult to read.
  - Failing to apply to large data set.
  - Revealing individual characteristics instead of global distributional information.
Let’s move to Part 4