

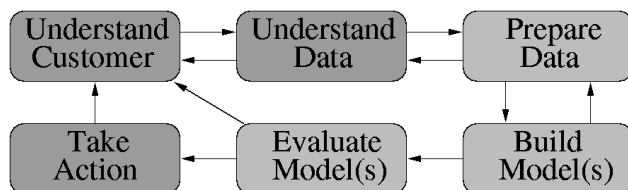
COMP8400: Algorithms and Techniques for Data Mining

Data pre-processing

Lecture outline

- The data mining / KDD process
- Why data pre-processing
 - Forms of data pre-processing
- Root conditions of data quality problems
- Data quality measures
- Data pre-processing
 - Data cleaning
 - Data transformation
 - Attribute / feature construction
 - Data reduction and discretisation
 - Data parsing and standardisation
 - Data integration, matching and linkage (*next week*)

The data mining / KDD process

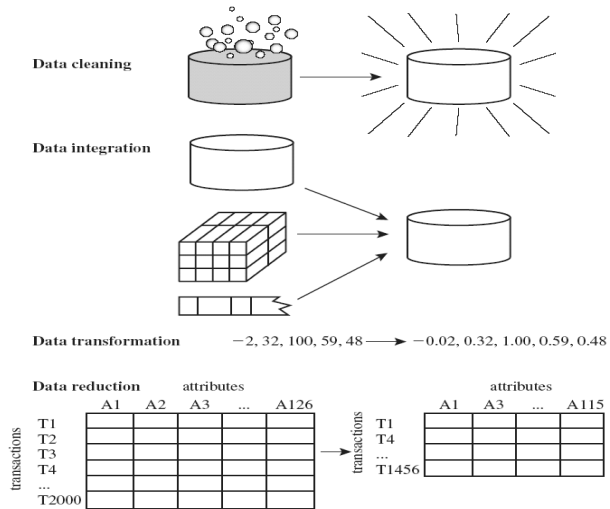


- Understand customer: 10 – 20 %
- Understand data: 10 – 30 %
- Prepare data: 20 – 60 %
- Build model(s): 10 – 20 % (data mining)
- Evaluate model(s): 10 – 20 %
- Take action: 10 – 20 %

Why data pre-processing

- Real world data is dirty
 - Incomplete data (missing attributes, missing attribute values, only aggregated data, etc.)
 - Inconsistent data (different coding, different naming, impossible values, or out-of-date values)
 - Noisy data (containing errors, variations, outliers, not accurate values)
- For quality data mining results, quality data is needed
 - *Garbage-in, garbage-out*
- Pre-processing is an important step for successful data mining
 - Data extraction, cleaning, transformation and integration are the majority of work required when building a data warehouse

Forms of data pre-processing



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

Root conditions of data quality problems

- Multiple data sources (*next week*)
- Subjective judgment in data production
- Limited computing resources
- Security/accessibility trade-off
- Coded data across disciplines
- Complex data representations
- Volume of data
- Input rules too restrictive or bypassed
- Changing data needs
- Distributed heterogeneous systems

Data quality measures

- Accuracy
- Completeness
- Consistency
- Timeliness
- Believability
- Interpretability
- Accessibility

Data pre-processing tasks

- Data cleaning
 - Fill-in missing values, smooth noisy data, identify/remove outliers, resolve inconsistencies
- Data transformation
 - Normalise and/or aggregate data
- Data reduction and discretisation
 - Reduce volume of data, but still produce same or similar analytical result, discretisation in particular for numerical data
- Data integration, matching and linking
 - *Next week*

Data cleaning

- Data cleaning tasks
 - Fill-in (impute) missing values
 - Detect and correct inconsistent data
 - Identify outliers / smooth noisy data
- Missing data may be due to
 - Attributes not considered important
 - Misunderstanding at data entry
 - Inconsistencies with other data and thus deleted
 - Equipment malfunction (for example EFTPOS down, so only cash transactions)
- Missing data may need to be inferred (data imputation)

Data cleaning – Inconsistent data / outliers

- Why inconsistent data?
 - Due to data entry errors or data integration (different formats, codes, etc.)
 - Important to have data entry verification (check both format and values of data entered), most of the time only format is checked
 - Correct with help of external reference data (look-up tables, e.g. *Sydney, NSW, 7000* -> *Sydney, NSW, 2000*) or rules (e.g. *male / 0* -> *M*, *female / 1* -> *F*)
- Identify outliers and noisy data
 - Noise: Random error or variance in a measurement
 - Incorrect attribute values (faulty data collection, data entry problems, data transmission problems, data conversion errors, inconsistent naming, technology limitations, bugs, for example buffer overflow or attribute length limits)
 - Handle noisy data through binning, clustering, regression, manual inspection
 - Don't remove or modify outliers for outlier detection!

Data cleaning – Missing values

- How to handle missing data?
 - Ignore the records that contain missing values
 - Fill in missing value manually (often unfeasible)
 - Fill in with a global constant (e.g. *unknown* or *n/a*). Not recommended as a data mining algorithm might see this as a normal value!
 - Fill in with attribute mean or median
 - Fill in with class mean or median (classes need to be known)
 - Fill in with most likely value (using regression, decision trees, most similar records, etc.)
 - Use other attributes to predict value (e.g. if a *postcode* is missing use *suburb* value and external look-up table – if one-to-one relationship)
 - Data editing/imputation (rules based)

Data transformation

- Consolidate data into forms suitable for data mining
- Smooth data (remove noise)
- Aggregate data (summarisation, data cube construction)
- Generalise data (replace data with higher level concepts, e.g. *address details* -> *city* -> *state* -> *country*)
- Normalise data (scale to within a specified range)
 - Min-max (for example into [0..1] interval, or 0%..100%)
 - Z-score or zero-mean (based on mean and standard deviation of an attribute)
 - Decimal scaling (move decimal point for all values)
- Important to save normalisation parameters in meta-data repository

Attribute / feature construction

- Sometimes it is helpful or necessary to construct new attributes or *features*
 - Based on existing attributes in data
 - Helpful for understanding
 - For example: Create attribute *volume* based on attributes *height*, *depth* and *width* (for example in a post or parcel database)
- Construction is based on mathematical or logical operations
- Attribute / feature construction can help to discover missing information about the relationships between the original data attributes

Data reduction and discretisation (1)

- Databases or data warehouses often contain Terabytes of data, resulting in (very) long run times for data mining algorithms
- High-dimensionality often prohibits the use of algorithms on the original data (curse of dimensionality)
- Data reduction techniques
 - Data cube aggregation
 - Dimensionality reduction
 - Data compression
 - Numerosity reduction
 - Discretisation and concept hierarchy generation

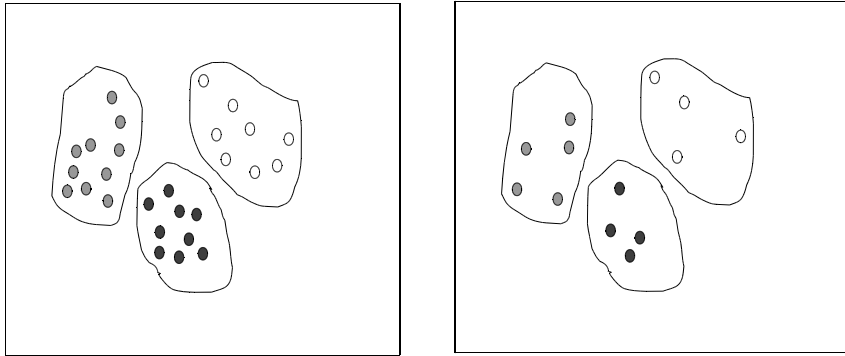
Data reduction and discretisation (2)

- Data cube aggregation (roll-up)
 - Data warehouses often have data stored at different levels of granularity (e.g. *day*, *week*, *month*, *quarter*, *year*)
 - Use the smallest representation that is enough to solve the problem
- Dimensionality reduction
 - Select a (minimum) sub-set of the available attributes (with similar probability distribution of classes compared to the original data)
 - For d attributes there are 2^d sub-sets of attributes!
 - Find correlated, redundant or derived attributes (e.g. *age* and *date of birth*)
 - Step-wise forward selection (find and select most useful attribute) or backward elimination (find and eliminate least useful attribute)
 - Use decision tree induction to find minimum attribute sub-set necessary

Data reduction and discretisation (3)

- Data compression
 - Data encoding or transformation
 - Lossless or lossy encoding
 - Examples: String compression (e.g. ZIP/GZIP, only allow limited manipulation of data), wavelet transformation, discrete Fourier transformation, media compression (e.g. JPEG, MPEG)
- Numerosity reduction
 - Parametric methods (e.g. regression and log-linear models)
 - Non-parametric methods (histograms / binning, clustering, sampling)
 - Can be computationally expensive
 - Principal component analysis (find best dimensions/attributes to represent numerical data)

Example of cluster or stratified sampling

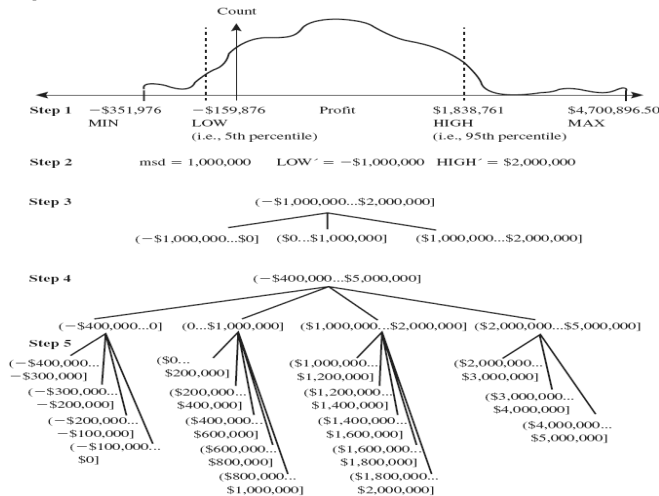


Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

Data reduction and discretisation (4)

- Discretisation and concept hierarchy generation
 - Reduce the number of values for a continuous attribute by dividing the range into intervals
 - Concept hierarchies for numerical attributes can be constructed automatically
 - Binning (sort data and partition into bins, then replace each value with mean, median or boundaries of the bin)
 - Histogram analysis (equi-width, equi-depth, etc.)
 - Clustering
 - Entropy based discretisation
 - Segmentation by natural partitioning (partition into 3, 4, or 5 relatively uniform intervals)

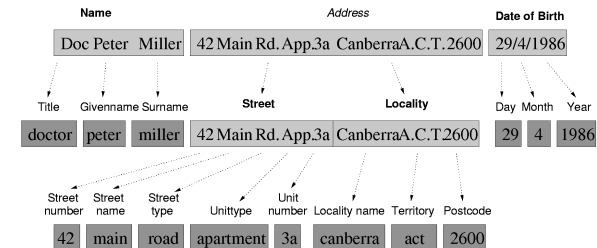
Example of 3-4-5 rule



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

Data parsing and standardisation

- Parse free format data into specific, well defined attributes
- Standardise using rules and look-up tables (correction and replacement tables), or probabilistically (using e.g. *hidden Markov models*)
- Important for data matching and linkage (for names, addresses, etc.)



What now... things to do

- Finalise lab/tutorial times and venues (will be announced on COMP8400 Web site and forum)
- Get and read first lab sheet (available on COMP8400 Web site very soon)
- Read assignment 1 draft (available next week)
- Read chapters 2 and 3 (Data Warehousing) in text book
 - Chapter 4 if you are interested in data warehouse implementations (not part of assessable material)