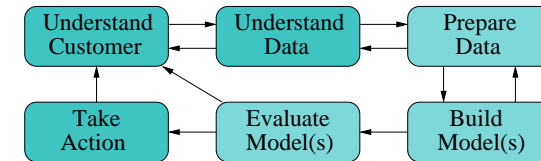


## Lecture 3: Data Mining Preprocessing

- The data mining / KDD process
- Why data preprocessing?
- Data quality measures
- Data preprocessing tasks
  - Data cleaning
  - Data transformation
  - Attribute / feature construction
  - Data reduction and discretisation
  - Data parsing and standardisation
  - Data integration and linkage (→ Lecture 4)

## The data mining / KDD process



- Understanding customer: 10% to 20%
- Understanding data: 20% to 30%
- Prepare data: 40% to 70%
- Build model(s): 10% to 20% (data mining)
- Evaluate model(s): 10% to 20%
- Take action: 10% to 20%

(Follows: *Cross Industry Standard Process for Data Mining*, <http://www.crisp-dm.org/>)

## Why data preprocessing?

- 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-range values)
  - Noisy data  
(data containing errors, outliers, not accurate values)
- For quality mining results, quality data is needed
- Preprocessing is an important step for successful data mining

## Data quality measures

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

## Data preprocessing tasks

- Data cleaning  
(fill in missing values, smooth noisy data, identify/remove outliers, resolve inconsistencies)
- Data transformation  
(normalisation and aggregation)
- Data reduction and discretisation  
(reduce volume of data, but still produce same or similar analytical result, discretisation in particular for numerical data)
- Data integration and linkage (→ Lecture 4)  
(linkage / matching / integration of multiple data sources, deduplication, geocoding)

## Data cleaning (1)

- Data cleaning tasks
  - Fill in (impute) missing values
  - Detect and correct inconsistent data
  - Identify outliers / smooth noisy data
- Missing data may be due to
  - attribute not considered important
  - misunderstanding at data entry
  - inconsistent with other data and thus deleted
  - equipment malfunction (e.g. EFTPOS down → use cash)
- Missing data may need to be inferred

## Data cleaning (2)

- How to handle missing data?
  - Ignore the record
  - Fill in missing value manually (often unfeasible)
  - Fill in with a global constant (e.g. *unknown*, or *n/a*)  
Not recommended (data mining algorithm will 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)

## Data cleaning (3)

- Inconsistent data
  - Due to data entry errors, data integration (different formats, codes, etc.)
  - Important to have data entry verification (check both format and values of data entered)
  - Correct with help of external reference data (look-up tables, e.g. *sydney*, *nsw*, *7000*) 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, e.g. buffer overflow or field size limits)
  - Handle noisy data through binning, clustering, regression, manual inspection

## Data transformation

- Consolidate data into forms suitable for data mining
  - Smoothing (remove noise)
  - Aggregation (summarisation, data cube construction)
  - Generalisation (replace data with higher level concepts, e.g. *address details* → *city*)
  - Normalisation (scale to within a specified range)
    - \* Min-max (e.g. into 0...1 interval)
    - \* 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*
  - Helpful for understanding and accuracy
  - For example: Create attribute *volume* based on attributes *height*, *depth* and *width*
- Construction is based on mathematical or logical operations
- Attribute construction can help to discover missing information about the relationships between 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 (roll-up)
  - 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*)
  - 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)
  - Find correlated, redundant or derived attributes (e.g. *age* and *date of birth*)
  - Step-wise forward selection (find and select best attribute) or backward elimination (find and eliminate worst attribute)
  - Use decision tree induction to find minimum attribute sub-set necessary

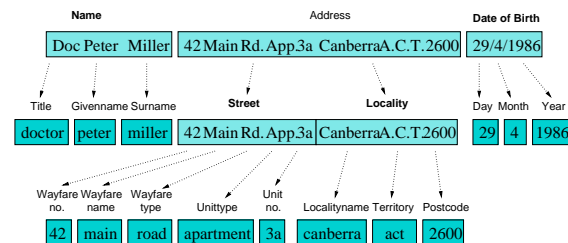
## Data reduction and discretisation (3)

- Data compression
  - Data encoding or transformation
  - Lossless or lossy encoding
  - Examples: String compression (e.g. ZIP, only allow limited manipulation of data), wavelet transformation, discrete Fourier transformation, principal component analysis
- Numerosity reduction
  - Parametric methods (e.g. regression and log-linear models) (can be computationally expensive)
  - Non-parametric methods (histograms / binning, clustering, sampling)

## 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 (smoothing, distributing values 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)

## 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 (hidden Markov models)
- Important for data linkage (based on names, addresses, etc.)