

COMP8400: Algorithms and Techniques for Data Mining

Data integration and data linkage

Why data integration and data linkage?

- Increasingly, data mining projects require data from more than one data source
- Data is often distributed (different databases or data warehouses)
 - For example an epidemiological study that needs information about hospital admissions and car accidents
- Geographically distributed data or historical data
 - For example, integrate historical data into a new data warehouse
- Enrich data with additional (external) data (to improve data mining accuracy)

Lecture outline

- Why data integration and data linkage?
- Data and schema integration
- Correlation analysis
- Handling redundant data
- Data linkage / matching
 - Linkage process
 - Linkage techniques
- Data cleaning and standardisation

Data integration

- Data integration
 - Combines data from multiple sources into a coherent form
 - Schema integration (for example, $A.cust-id \Leftrightarrow B.cust-no$)
 - Integrate Metadata from different sources
- Entity resolution (identification) problem
 - Identify real world entities from multiple data sources (for example, *Bill Clinton = William Clinton*, or *Mr Obama = the president*)
 - Also called *record linkage* or *data matching* (more later)
- Detecting and resolving data value conflicts
 - For the same real world entity, attribute values from different sources can be different
 - Possible reasons: different representations, different codings, different scales (for example metric vs. British units)

Handling redundancy in data integration

- Redundant data often occurs when multiple databases are integrated
 - 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, for example *annual revenue*
- Redundant attributes may be able to be detected by *correlation analysis*
- Careful (manual) 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*):

$$r_{A,B} = \frac{\sum (a_i - \bar{A})(b_i - \bar{B})}{N \sigma_A \sigma_B} = \frac{\sum (a_i b_i) - N \bar{A} \bar{B}}{N \sigma_A \sigma_B}$$

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 $\sum (a_i b_i)$ is the sum of the AB cross-product. Note that $-1 < r_{A,B} < +1$

- 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)

- χ^2 (chi-square) test

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

- The larger the χ^2 value, the more likely the variables are related
- The cells that contribute the most to the χ^2 value are those whose actual count is very different from the expected count
- Correlation does not imply causality
 - Number of hospitals and number of car-thefts in a city are correlated
 - Both are causally linked to a third variable: population size

Example Chi-square calculation

	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

χ^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

Schema integration

- Imagine two database tables

PID	Name	DOB
1234	Mayer	01/01/75
4791	Simmons	21-10-1969

PID	Surname	Age
1234	Meyer	32
4791	Simonds	38

- Integration issues

- The same attribute may have different names
- An attribute may be derived from another
- Attributes might be redundant
- There can be duplicate records (under different keys)

- Conflicts have to be detected and resolved

- Integration is made easier if unique entity keys are available in all the data sets (or tables) to be linked

Handling redundant data (1)

- Use correlation analysis

- Then decide which attributes to use and which not to use
- Possible to merge values from attributes (for example if some have missing values)

- Deduplication (also called *internal* data linkage)

- More than one record representing the same real world entity (for example *customers, patients, businesses*, etc.)
- If no unique entity keys are available (but even with unique keys a problem!)
- If no consistency checks are performed and enforced
- Analysis of values in attributes to find duplicates

Handling redundant data (2)

- Process redundant and inconsistent data

- Easy if values are the same
- Delete one of the values / records
- Calculate average values (only for numerical attributes)
- Take majority values (if more than two duplicates and some values are the same)
- Take most recent value (for changing data, like names and addresses)
- Use external data to find correct values
- Apply rule based system to determine which values to use

Data linkage / matching (1)

- Task of linking together records from one or more data sources that represent the same entity

- If there are no unique entity keys in data, the available attributes have to be used

- Often personal information (like names, addresses, dates of birth, etc.)
- Privacy and confidentiality becomes an issue (*more later in course*)

- Application areas

- Health (epidemiology)
- Census, taxation, immigration, social welfare
- Business mailing lists, collaborative e-Commerce
- Crime, fraud and terror detection (US: TIA, MATRIX)

Data linkage / matching (2)

- Different parts of the linked records are of interest

- Personal information (crime, fraud and terror detection, mailing lists)
- Non-personal information (epidemiology, census, most data mining)

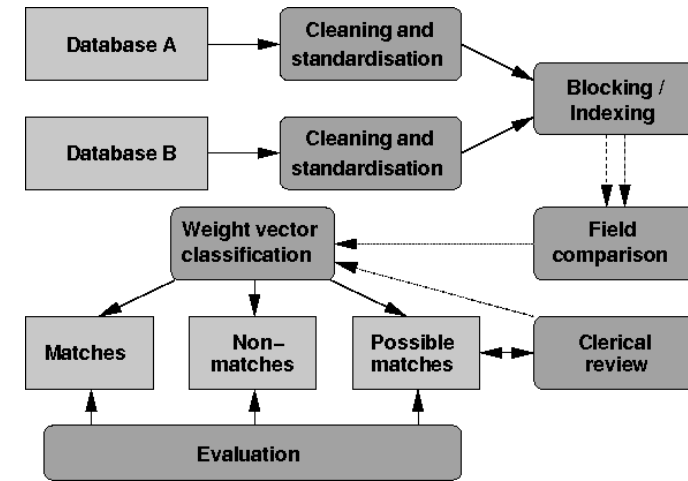
For example:

Age	Disease	Name
55	Cancer	John Miller
32	Diabetes	Joe Meyer
67	Cancer	Lucy Smith

Name	DoBirth	DoDeath
J. Miller	04/08/47	12/12/02
J. Meier	11/09/69	26/02/01
L. Smith	01/01/34	08/09/01

Disease	DoBirth	DoDeath	Gender
Cancer	04/08/47	12/12/02	M
Diabetes	11/09/69	26/02/01	M
Cancer	01/01/34	08/09/01	F

Data linkage process



Data linkage techniques

- Deterministic linkage

- Exact linkage (if a unique identifier of high quality is available: precise, robust, stable over time)
- Examples: *Medicare*, *ABN* or *Tax file number* (??)
- Rules based linkage (complex to build and maintain)

- Probabilistic linkage

- Use available (personal) information for linkage (which can be missing, wrong, coded differently, out-of-date, etc.)
- Examples: *names*, *addresses*, *dates of birth*, etc.

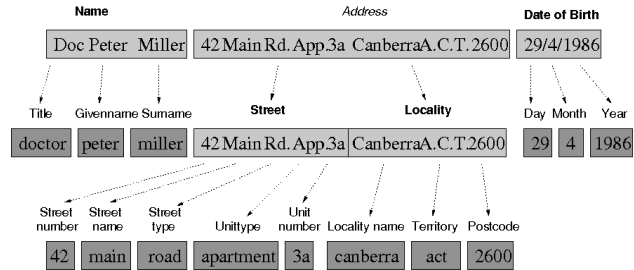
- Modern approaches

- Based on machine learning, data mining, artificial intelligence or information retrieval techniques

Why data cleaning and standardisation?

- Real world data is often dirty
 - Typographical and other errors
 - Different coding schemes
 - Missing values
 - Data changing over time
- Name and addresses are especially prone to data entry errors
 - Scanned, hand-written, over telephone, hand-typed
 - Same person often provides her/his details differently
 - Different correct spelling variations for proper names (for example *Gail* and *Gayle*, or *Dixon* and *Dickson*)

Cleaning and standardisation tasks



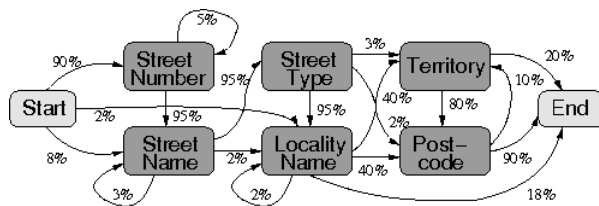
- Clean input
 - Remove unwanted characters and words
 - Expand abbreviations and correct misspellings
- Segment name/address into well defined output fields
 - Verify if address (or parts of it) exists

Cleaning and standardisation approaches

- Traditionally: Rules based
 - Manually developed parsing and transformation rules
 - Time consuming and complex to develop and maintain
- Recently: Probabilistic methods
 - Mainly based on hidden Markov models (HMMs)
 - More flexible and robust with regard to new unseen data
 - Drawback: Training data needed for most methods

HMMs are widely used in natural language processing and speech recognition, as well as for text segmentation and information extraction.

What is a hidden Markov model?



- A HMM is a probabilistic finite state machine
 - Made of a set of states and transition probabilities between these states
 - In each state an observation symbol is emitted with a certain probability
 - In our approach, the states correspond to the (address) output fields