

COMP3420: Advanced Databases and Data Mining

Data linkage and geocode matching

Lecture outline

- Why data integration and data linkage?
- Data and schema integration
- Deduplication and handling redundant data
- Data linkage / database matching
 - Linkage process
 - Linkage techniques and challenges
 - Data cleaning and standardisation
 - Indexing / blocking
 - Classification
- Geocode matching
- Data linkage research at the ANU

Why data integration and data linkage?

- Increasingly, data mining, processing, and management 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)

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 *data/record linkage* or *data matching*
- 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)

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.)
 - Important for longitudinal (over time) studies, business mailing lists, 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

Applications for data linkage

- Health sector: Link various data collections to enrich data for studies that are not possible otherwise
- National security: Terrorism or crime watch lists, identify records of criminals who provide wrong identities
- Identity fraud: Find identities in several databases that do not 'look right' (stolen or fabricated identities)
- Census: Link records across time to reconstruct families and households (allows many social studies, as well as 'genealogical epidemiology')
- Measuring research impact: Match bibliographic databases with researcher's citation counts

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.

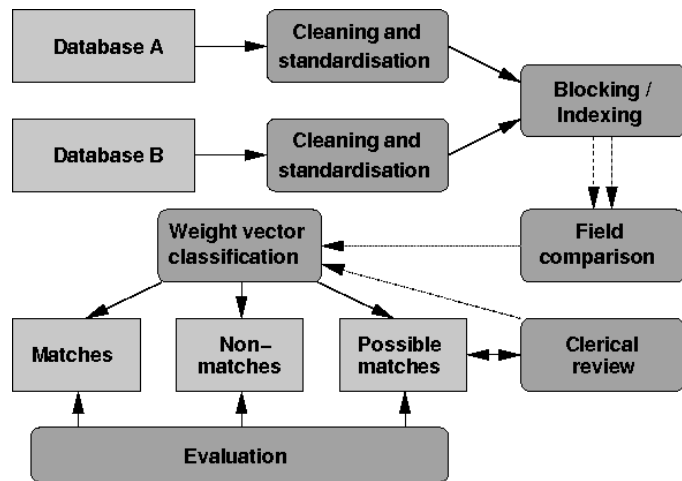
- Modern approaches

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

Data linkage challenges

- Often no *unique entity identifiers* (keys) are available
- Real world data is dirty (typographical errors and variations, missing and out-of-date values, different coding schemes, etc.)
- Scalability to very large databases
 - Naive comparison of all record pairs is $O(n \times n)$
 - Some form of blocking, indexing or filtering is required (more later)
- Privacy and confidentiality (because personal information, like names and addresses, are commonly required for matching)
- No training data in many matching applications
 - No record pairs with known true match status
 - Possible to manually prepare training data (but, how accurate will manual classification be?)

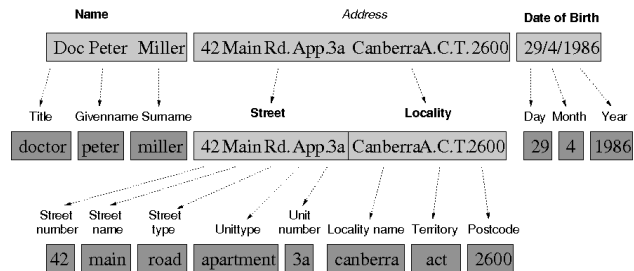
Data linkage process



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



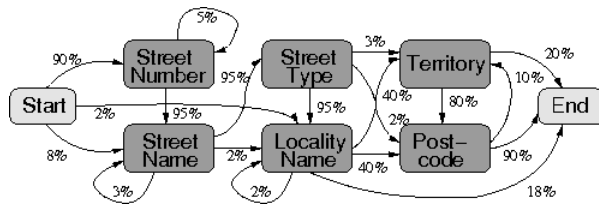
- Clean input
 - Remove unwanted characters and words
 - Expand abbreviations and correct misspellings
- Segment name/address into well defined output fields
 - Verify if address (or parts of it) exists

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

Traditional blocking

- Traditional blocking works by only comparing record pairs that have the same value for a blocking variable
(for example, only compare records which have the same *postcode* value)
- Problems with traditional blocking
 - An erroneous value in a blocking variable results in a record being inserted into the wrong block (several passes with different blocking variables can solve this)
 - Values of blocking variable should be uniformly distributed (as the most frequent values determine the size of the largest blocks)
 - Example: Frequency of 'Smith' in NSW: 25,425

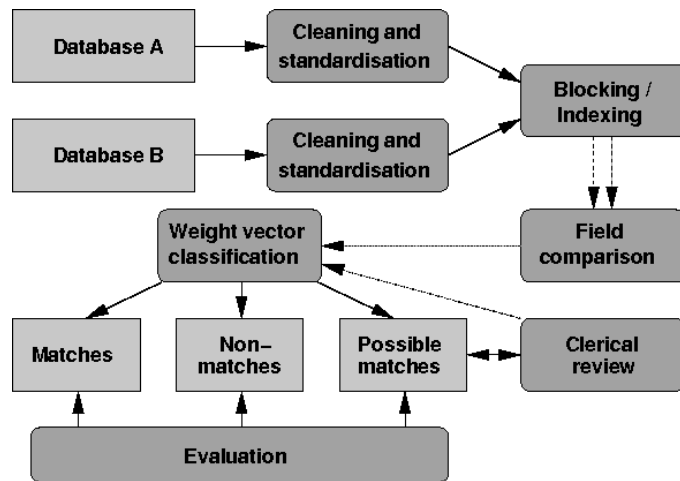
Why blocking / indexing?

- The number of record pair comparisons equals the product of the sizes of the two data sets
(for example, linking two data sets with 1 and 5 million records will result in $1,000,000 \times 5,000,000 = 5 \times 10^{12}$ record pairs)
- Performance bottleneck in a data linkage system is usually the (expensive) comparison of field values between record pairs (similarity measures / field comparison functions)
- Blocking / indexing / filtering techniques are used to reduce the large amount of comparisons
- Aim of blocking: Cheaply remove candidate record pairs which are obviously not matches

Improved blocking

- Recent research methods
 - Sorted neighbourhood approach (sliding window over sorted blocking variable)
 - Fuzzy blocking using n-grams (for example: *bigrams*: 'peter' -> ['pe','et','te','er'], 'pete' -> ['pe','et','te'])
 - Overlapping canopy clustering (where records are inserted into several clusters)
- Post-blocking filtering (like length differences or n-grams count differences)
- US Census Bureau: *BigMatch*
(pre-process 'smaller' data set so its values can be directly accessed in main memory; with all blocking passes in one go)

Data linkage process



Probabilistic data or record linkage

- Computer assisted data linkage goes back as far as the 1950s
 - Based on ad-hoc heuristic methods
- Basic ideas of probabilistic linkage were introduced by *Newcombe and Kennedy* (1962)
- Theoretical foundation by *Fellegi and Sunter* (1969)
 - Compare common attributes from record pairs and calculate similarity values (also called *matching weights*)
 - Using matching weights based on frequency ratios
 - Summation of matching weights is used to classify a pair of records as *match*, *possible match* or *non-match*
 - Manual clerical review required to determine the match status of the possible matches

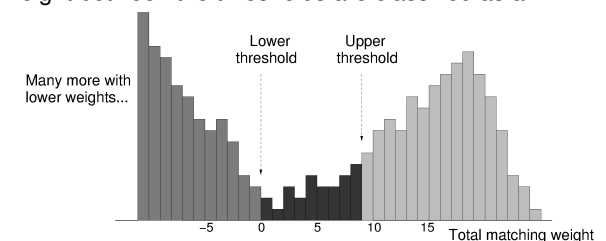
Record pair comparison example

- Attributes are compared using various comparison functions (like exact or approximate string, numeric, date, age, etc.)

Record A: [Dr, Peter, Paul, Miller]
 Record B: [Mr, John, , Miller]
 [0.2, -3.2, 0.0, 2.4] -0.6

Record pair classification

- The final *matching weight* is the sum of the attribute comparison weights (similarity values)
 - Record pairs with a weight above an upper threshold are classified as a *match*
 - Record pairs with a weight below a lower threshold are classified as a *non-match*
 - Record pairs with a weight between the thresholds are classified as a *possible match*



Linkage example: Month-of-birth weight calc

- Assume two databases that have a 3 % error in the month-of-birth attribute
 - Probability that two matched records (that represent the same person) have the same month is 97 % (*M agreement*)
 - Probability that two matched records (that represent the same person) do not have the same month is 3 % (*M disagreement*)
 - Probability that two un-matched records (randomly picked) have the same month is $1/12 = 8.3\%$ (*U agreement*)
 - Probability that two un-matched records (randomly picked) do not have the same month is $11/12 = 91.7\%$ (*U disagreement*)
- Agreement weight (M_{ag} / U_{ag}): $\log_2(0.97 / 0.083) = 3.54$
- Disagreement weight (M_{disag} / U_{disag}): $\log_2(0.03 / 0.917) = -4.92$

Improved record pair classification

- Summing of weights results in loss of information
(like same name but different address, or different address but same name)
- View record pair classification as a *multidimensional binary classification* problem
(use weight vector to classify record pairs a matches or non-matches, but no possible matches)
- Many machine learning techniques can be used
 - Supervised: Decision trees, neural networks, learnable string comparisons, active learning, etc.
 - Un-supervised: Various clustering algorithms
- Major issue: Lack of training data

Value specific frequencies

- Example: Surnames
 - Assume the frequency of surname *Smith* is higher than *Dijkstra* (NSW Whitepages: 25,425 *Smith*, only 3 *Dijkstra*)
 - Intuitively, two records with surname values *Dijkstra* are more likely to correspond to the same person than two records with surname value *Smith*
- The matching weights need to be adjusted
 - Difficulty: How to get value specific frequencies that are characteristic for a given database
 - Earlier linkages done on same or similar data, or maybe a small linkage done on a sample
 - Information from external data sets (e.g. *Australian Whitepages*)

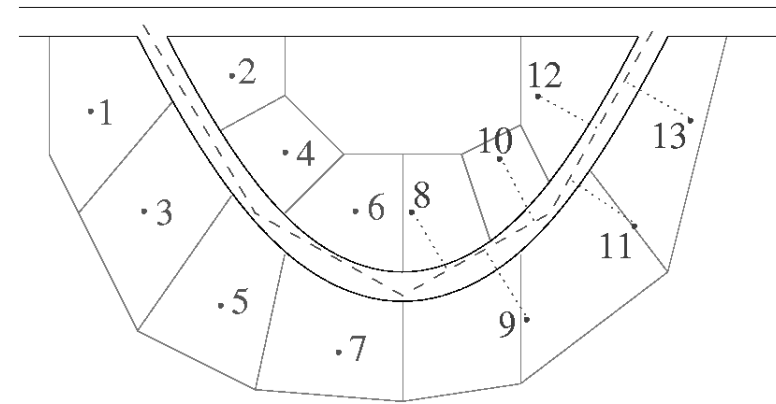
Classification challenges

- In many cases there is no training data available
 - Possible to use results of earlier linkage projects?
 - Or from clerical review process?
- How confident can we be about correct manual classification of possible links?
- Often there is no *gold standard* available (no data sets with true known linkage status)
- No large test data set collection available
(like in information retrieval or machine learning)

Geocode matching

- Match addresses against *geocoded* reference data (addresses and their geographic locations: latitudes and longitudes)
- Useful for spatial data analysis / mining and for loading data into geographical information systems
- Matching accuracy is critical for good geocoding (as is accurate geocoded address data)
- Australia has a *Geocoded National Address File (G-NAF)* since early 2004 (all Australian property addresses and their locations)
- Commercial geocoding systems mainly work on *street centreline* data

Geocoding example (1)



Geocoding example (2)



Data linkage research at the ANU

- We have been working in data linkage since 2002 (see: <http://datamining.anu.edu.au/linkage.html>)
- Research project in collaboration with the New South Wales Department of Health, Sydney
- We have developed an open source data linkage system called *Febri* (Freely Extensible Biomedical Record Linkage)
- Only data cleaning, deduplication and record linkage system in the world that is free and has a graphical user interface
- Interested in this area? We have projects at various levels (implementation, honours, PhD/MPhil)