Probabilistic Deduplication, Record Linkage and Geocoding

Peter Christen

Data Mining Group, Australian National University
in collaboration with
Centre for Epidemiology and Research, New South Wales Department of Health

Contact: **peter.christen@anu.edu.au**

Project web page: **http://datamining.anu.edu.au/linkage.html**

Funded by the ANU, the NSW Department of Health, the Australian Research Council (ARC), and the Australian Partnership for Advanced Computing (APAC)

Outline

- Background and illustrative example
- Record linkage
- Applications, privacy and ethics
- Our project and our tools
- Data cleaning and standardisation
- Probabilistic data standardisation and HMMs
- Blocking / indexing
- Record pair classification
- Geocoding
- Outlook

Background

- Many organisations and businesses collect large amounts of data
  (employ *data mining* to find hidden patterns, rules, etc.)
- Databases may contain duplicate records
  (for example, customers in a mailing list receive advertising mail twice)
- Sometimes one is interested in linking databases
  (for example, study effects of car accidents and injury types)

The *NSW Department of Health* approached us in 2002. They were interested in improving **record linkage** techniques and algorithms.

Illustrative example

- A research project is interested in car accidents and the types of injuries they cause
- Hospital database: *name*, *address*, *date of birth*, admission date, length of stay, injury code, costs
- Car insurance database: *name*, *address*, *date of birth*, car type and colour, date and type of accident, claim
- No unique identifier available
  We have to use common attributes to link records (*name*, *address*, and *date of birth*)

Record linkage

- Record linkage is the task of linking together information from one or more data sources representing the same entity
- Record linkage is also called *database matching*, *data integration*, *data scrubbing*, or *ETL (extraction, transformation and loading)*
- Three records, which represent the same person?

1.	*Dr Smith, Peter; 42 Miller Street 2602 O'Connor*
2.	*Pete Smith; 42 Miller St 2600 Canberra A.C.T.*
3.	*P. Smithers, 24 Mill Street 2600 Canberra ACT*

Record linkage techniques

- Deterministic or exact linkage
  - A *unique identifier* is needed, which is of high quality (precise, robust, stable over time, highly available)
  - For example *Medicare*, *ABN* or *Tax file* number (are they *really* unique, stable, trustworthy?)
- Probabilistic linkage (*Fellegi & Sunter*, 1969)
  - Apply linkage using available (personal) information
  - Examples: *names*, *addresses*, *dates of birth*
- Other techniques
  (rule-based, fuzzy approach, information retrieval, AI)

Applications and usage

- Applications of record linkage
  - Remove duplicates in a data set (internal linkage)
  - Merge new records into a larger master data set
  - Create patient or customer oriented statistics
  - Compile data for longitudinal (over time) studies
  - Clean data sets for data analysis and mining projects
- Widespread use of record linkage
  - Census statistics
  - Business mailing lists
  - Health and biomedical research (epidemiology)
  - Fraud and crime detection

Privacy and ethics

- For some applications, personal information is not of interest and is removed from the linked data set
  (for example epidemiology, census statistics, data mining)
- In other areas, the linked information is the aim
  (for example business mailing lists, crime and fraud detection, data surveillance)
- Personal privacy and ethics is most important
  - *Privacy Act*, 1988
  - *National Statement on Ethical Conduct in Research Involving Humans*, 1999

Why this project?

- Commercial software for record linkage is often expensive and cumbersome to use
- Project aims
  - Allow linkage of larger data sets (high-performance and parallel computing techniques)
  - Reduce the amount of human resources needed (improve linkage quality by using machine learning)
  - Reduce costs (free open source software)
- Software for data cleaning and standardisation, deduplication, record linkage, and geocoding

  Febrl – Freely extensible biomedical record linkage

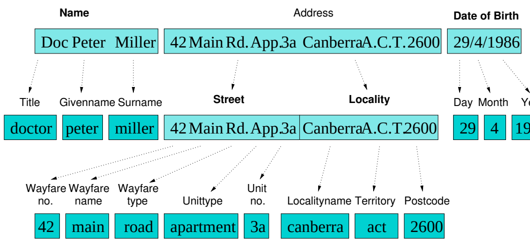
Open source software tools

- Scripting language *Python* **www.python.org**
  - Easy and rapid prototype software development
  - Object-oriented and cross-platform *(Unix, Win, Mac)*
  - Can handle large data sets stable and efficiently
  - Many external modules, easy to extend
  - Large user community
- Parallel libraries *MPI* and *OpenMP*
  - Widespread use in high-performance computing (quasi standards) ⇒ Portability and availability
  - Parallel *Python* extensions: *PyRO* and *PyPar*

Data cleaning and standardisation (1)

- Real world data is often *dirty*
  - Missing values, inconsistencies
  - Typographical and other errors
  - Different coding schemes / formats
  - Out-of-date data
- Names and addresses are especially prone to data entry errors
- Cleaned and standardised data is needed for
  - Loading into databases and data warehouses
  - Data mining and other data analysis studies
  - Record linkage and data integration

Data cleaning and standardisation (2)

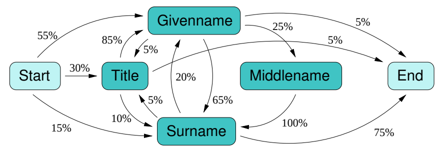


- Remove unwanted characters and words
- Expand abbreviations and correct misspellings
- Segment data into well defined *output fields*

Probabilistic data cleaning and standardisation

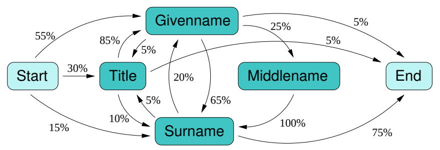
- Three step approach
  1. Cleaning
     – Based on look-up tables and correction lists
     – Remove unwanted characters and words
     – Correct various misspellings and abbreviations
  2. Tagging
     – Split input into a list of words, numbers and separators
     – Assign one or more tags to each element of this list (using look-up tables and some hard-coded rules)
  3. Segmenting
     – Use either rules or a *hidden Markov model (HMM)* to assign list elements to *output fields*

Hidden Markov model (HMM)



- 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 distribution
  - In our approach, the observation symbols are *tags* and the states correspond to the *output fields*

HMM segmentation example



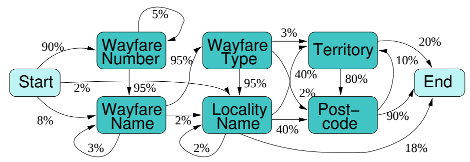
- Input word and tag list

```
['dr',  'peter',  'paul',  'miller']
['TI',  'GM/SN',  'GM',     'SN'   ]
```

- Two example paths through the HMM

```
1: Start -> Title (TI) -> Givenname (GM) -> Middlename (GM) ->
          Surname (SN) -> End
2: Start -> Title (TI) -> Surname (SN) -> Givenname (GM) ->
          Surname (SN) -> End
```


Address HMM standardisation example



1. Raw input: *'73 Miller St, NORTH SYDENY 2060'*
   Cleaned into: *'73 miller street north sydney 2060'*
2. Word and tag lists:

```
['73', 'miller', 'street', 'north_sydney', '2060']
['NU', 'UN',     'WT',     'LN',           'PC'  ]
```

3. Example path through HMM

```
Start -> Wayfare Number (NU) -> Wayfare Name (UN) -> Wayfare
    Type (WT) -> Locality Name (LN) -> Postcode (PC) -> End
```

Address standardisation results

- Various NSW Health data sets
  - *HMM1* trained on 1,450 Death Certificate records
  - *HMM2* contains *HMM1* plus 1,000 Midwifes Data Collection training records
  - *HMM3* is *HMM2* plus 60 unusual training records
- *AutoStan* rules (for ISC) developed over years

	HMM/Method			
Test Data Set (1,000 records each)	HMM 1	HMM 2	HMM 3	Auto Stan
Death Certificates	95.7%	96.8%	97.6%	91.5%
Inpatient Statistics Collection	95.7%	95.9%	97.4%	95.3%

Blocking / indexing

- Number of possible links equals the product of the sizes of the two data sets to be linked
  (two databases with 1,000,000 and 5,000,000 records will result in 1,000,000 $\times$ 5,000,000 = 5^{12} = 5 Trillion record pair comparisons)
- Performance bottleneck is the (expensive) comparison of field values (similarity measures) between record pairs
- Blocking / indexing techniques are used to reduce the large amount of record comparisons
  (for example, only compare records which have the same *postcode* value)

Field comparison functions in Febrl

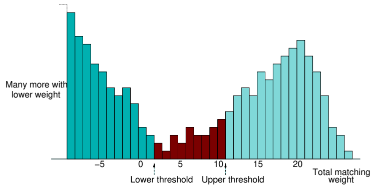
- Exact string
- Truncated string (only consider beginning of strings)
- Approximate string (using *Winkler*, *Edit dist*, *Bigram* etc.)
- Encoded string (using *Soundex*, *NYSIIS*, etc.)
- Keying difference (allow a number of different characters)
- Numeric percentage (allowing percentage tolerance)
- Numeric absolute (allow absolute tolerance)
- Date (allow day tolerance)
- Age (allow percentage tolerance)
- Time (allow minute tolerance)
- Distance (allow kilometre tolerance)

Record pair classification

- For each record pair compared a vector containing *matching weights* is calculated
  Example:
  Record A: `['dr', 'peter', 'paul', 'miller']`
  Record B: `['mr', 'john', '', 'miller']`
  Matching weights: `[0.2, -3.2, 0.0, 2.4 ]`
- Matching weights are used to classify record pairs as *links*, *non-links*, or *possible links*
- *Fellegi & Sunter* classifier simply sums all the weights, then uses two thresholds to classify
- Improved classifiers are possible
  (for example using machine learning techniques)

Final linkage decision (F & S)

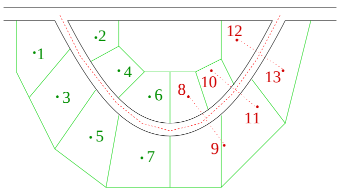
- The final weight is the sum of weights of all fields
  - Record pairs with a weight above an *upper threshold* are designated as a *link*
  - Record pairs with a weight below a *lower threshold* are designated as a *non-link*
  - Record pairs with a weight between are *possible link*



Geocoding

- The process of matching addresses with geographic locations (longitude and latitude)
- It is estimated that 80% to 90% of governmental and business data contain address information
  (US Federal Geographic Data Committee)
- Geocoding tasks
  - Pre-process the geocoded reference data (cleaning, standardisation and indexing)
  - Clean and standardise the user addresses
  - (Approximate) matching of user addresses with the reference data

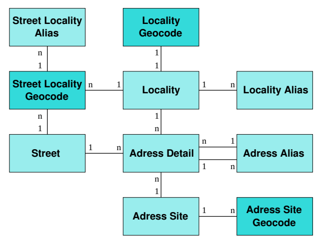
Geocoding techniques



- Street centreline based (many commercial systems)
- Property parcel centre based (our approach)
- A recent study found substantial differences
  (specially in rural areas)
  Cayo and Talbot; Int. Journal of Health Geographics, 2003

Geocoded national address file

- G-NAF: Available since early 2004
  (PSMA, `http://www.g-naf.com.au/`)
- Source data from 13 organisations
  (around 32 million source records)
- Processed into 22 normalised database tables



Febrl geocoding match engine

- Uses cleaned and standardised user address(es) and G-NAF inverted index data
- Fuzzy rule based approach
  1. Find street match set (street name, type and number)
  2. Find postcode and locality match set (with *no*, then *direct*, then *indirect* neighbour levels)
  3. Intersect postcode and locality sets with street match set (if no match increase neighbour level and go back to 2.)
  4. Refine with unit, property, and building match sets
  5. Retrieve corresponding location (or locations)
  6. Return location and match status (address, street or locality level match; none, one or many matches)

Geocoding examples

- Red dots: *Febrl* geocoding (G-NAF based)
- Blue dots: Street centreline based geocoding

Outlook

- Several research areas
  - Improving probabilistic data standardisation
  - New and improved blocking / indexing methods
  - Apply machine learning techniques for record pair classification
  - Improve performances (scalability and parallelism)
- Project web page

  http://datamining.anu.edu.au/linkage.html

We always have student projects available...
(for summer students, honours and Masters/PhDs).
If you are interested please contact me.

Contributions / Acknowledgements

- Dr Tim Churches (New South Wales Health Department, Centre for Epidemiology and Research)
- Dr Markus Hegland (ANU Mathematical Sciences Institute)
- Dr Lee Taylor (New South Wales Health Department, Centre for Epidemiology and Research)
- Ms Kim Lim (New South Wales Health Department, Centre for Epidemiology and Research)
- Mr Alan Willmore (New South Wales Health Department, Centre for Epidemiology and Research)
- Mr Karl Goiser (ANU Computer Science PhD student)
- Mr Puthick Hok (ANU Computer Science honours student, 2004)
- Mr Justin Zhu (ANU Computer Science honours student, 2002)
- Mr David Horgan (ANU Computer Science summer student, 2003/2004)