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

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

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.

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

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

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 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
**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

---

**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

---

**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

---

**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 &rarr; Title (TI) &rarr; Givenname (GM) &rarr; Middlename (GM) &rarr; Surname (SN) &rarr; End
  2. Start &rarr; Title (TI) &rarr; Surname (SN) &rarr; Givenname (GM) &rarr; Surname (SN) &rarr; End

---

**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 (2)**

- **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

---

**Address HMM standardisation example**

1. Raw input: `’73 Miller St, NORTH SYDNEY 2060’`
   Cleaned into: `’73 miller street north sydney 2060’`
2. Word and tag lists:
  - `’73’, ’miller’, ’street’, ’north_sydney’, ’2060’`
  - `’SN’, ’GM’, ’WT’, ’LN’, ’PC’`
3. Example path through HMM
   - Start &rarr; Wayfare Number (NU) &rarr; Wayfare Name (UN) &rarr; Wayfare Type (WT) &rarr; Locality Name (LN) &rarr; Postcode (PC) &rarr; End
Address standardisation results

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

<table>
<thead>
<tr>
<th>Test Data Set</th>
<th>HMM1</th>
<th>HMM2</th>
<th>HMM3</th>
<th>AutoStan</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,000 records each)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>Stan</td>
</tr>
<tr>
<td>Death Certificates</td>
<td>95.7%</td>
<td>96.8%</td>
<td>97.6%</td>
<td>91.5%</td>
</tr>
<tr>
<td>Inpatient Statistics Collection</td>
<td>95.7%</td>
<td>95.9%</td>
<td>97.4%</td>
<td>95.3%</td>
</tr>
</tbody>
</table>

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, NYSSIS, 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)

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

Geocoded national address file

- Source data from 13 organisations (around 32 million source records)
- Processed into 22 normalised database tables
**Febri 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)

---

**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

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)