Parallel Techniques for High-Performance Record Linkage (Data Matching)

Peter Christen

In collaboration with: Tim Churches, Markus Hegland, Kim Lim,
Ole M. Nielsen, Stephen Roberts and Justin Zhu

Data Mining Group, Australian National University
Epidemiology and Surveillance Branch, NSW Department of Health

Contact: **peter.christen@anu.edu.au**
Project web page: **http://datamining.anu.edu.au/linkage.html**

Outline

- Record linkage in a nutshell
  - Applications
  - Privacy and ethics
  - Techniques
- Our project: Open source record linkage
  - Aim, approach and status
  - *Python* prototype
  - Parallelisation
  - Using machine learning and data mining

Record Linkage – A Definition

Record linkage is the task of linking together information from one or more data sources that represent the same entity.

- Record linkage is also called...
  - data linkage, record matching or database matching
  - data cleaning, data scrubbing or data standardisation
  - ETL (extraction, transformation and loading), or the merge/purge problem
- Data standardisation is an important first step (well defined, clean and standardised fields needed)

Applications

- Two applications
  - De-duplicate a data set
  - Link two data sets (merge into a master data set)
- Widespread use of record linkage
  - Epidemiology (patient oriented and longitudinal studies)
  - Census statistics
  - Business mailing lists (cleaning and updating)
  - Crime and fraud detection

Record linkage is often an initial step in epidemiological studies and in data analysis / data mining projects.

Privacy and Ethics

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

Record Linkage Techniques

- Deterministic or exact linkage
  - A **unique identifier** is needed, which is of high quality (precise, robust, stable over time, highly available)
  - Examples: *Medicare number* or *Tax file number* (are they *really* unique, stable, trustworthy?)
- Probabilistic linkage
  - Apply linkage using available (personal) information (can be missing, wrong, coded differently, outdated, etc.)
  - Examples: *name*, *address*, *date of birth*, etc.
- Other techniques
  Rule-based, fuzzy approach, information retrieval, etc.

Probabilistic Record Linkage

- Computer assisted record linkage goes back as far as the 1950s (based on ad-hoc heuristic methods)
- Basic ideas of probabilistic linkage were introduced by *Newcombe & Kennedy* (1962)
- Theoretical foundation by *Fellegi & Sunter* (1969)
  - Using matching weights based on frequency ratios (global or value specific ratios)
  - Compute matching weights for all fields used in linkage
  - Summation of matching weights is used to designate a pair of records as *link*, *possible-link* or *non-link*

Linkage Example: Month of Birth

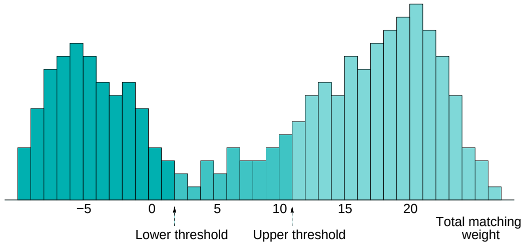
- Assume two data sets with a 3% error in field *month of birth*
- Probability that two linked records (that represent the same person) have the same month is 97% *(L agreement)*
- Probability that two linked records do not have the same month is 3% *(L disagreement)*
- Probability that two (randomly picked) unlinked records have the same month is $1/12 = 8.3\%$ *(U agreement)*
- Probability that two unlinked records do not have the same month is $11/12 = 91.7\%$ *(U disagreement)*
- Agreement weight (L_{ag}/U_{ag}): $\log_2(0.97/0.083) = 3.54$
  Disagreement weight (L_{di}/U_{di}): $\log_2(0.03/0.917) = -4.92$

Value Specific Frequencies

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

Final Linkage Decision

- 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*



Our Record Linkage Project

- Why this project?
  - Commercial software is expensive
  - It often doesn’t provide the flexibility a user wants
  - No use of high-performance and parallel computing
  - No use of machine learning and data mining techniques (specially for the manual *clerical review* process)
- Project aim
  - Develop improved techniques for probabilistic record linkage
  - Implement open source prototype software

A Collaborative Research Project

- ANU Data Mining Group
  - Department of Computer Science
  - Mathematical Sciences Institute
  - Australian Partnership for Advanced Computing (APAC)
- New South Wales Department of Health
  - Epidemiology and Surveillance Branch

Equally funded by ANU and NSW Health under an ANU Industry Collaboration Scheme (AICS)

Project Aim and Description

- The aim is to facilitate (epidemiological) research with free and improved tools for record linkage
  1. Implement prototype software for data standardisation and record linkage ⇒ **Feedback from (test) users**
  2. Develop high-performance and parallel computing techniques ⇒ **Faster linkage of larger data sets**
  3. Explore machine learning and data mining techniques for record linkage ⇒ **Better linkage quality**

Use of open source software tools to develop open source record linkage software

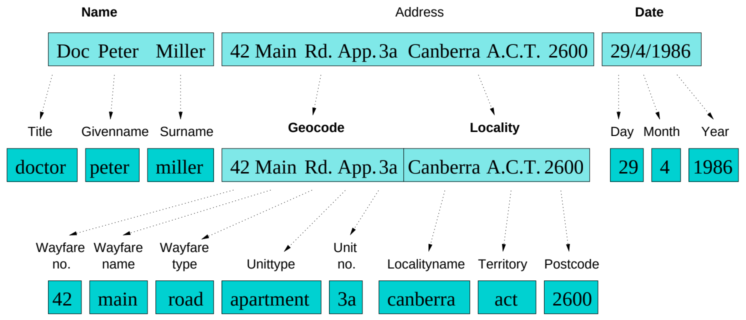
Open Source Software Tools

- Scripting language *Python*
  - Easy and rapid prototype software development
  - Provides lists and dictionaries (great for lookup-tables)
  - Can handle large data sets stable and efficiently
  - Many external modules, easy to extend
  - Available from **www.python.org** *(Unix, Windows, Mac)*
- Parallel libraries *MPI* and *OpenMP*
  - Widespread use in high-performance computing (quasi standards) ⇒ Portability and availability
  - Parallel *Python* extensions: *PyRO* (Remote Objects) and *PyPar* (MPI; Ole Nielsen, MSI/APAC)

Target Computing Platforms

- Workstation or PC cluster
  - Commodity PCs connected via local area network
  - Widespread availability, no extra costs
  - Use as virtual parallel computer (over night / weekends)
- Multiprocessor (SMP) servers
  - Example: *Sun Enterprise, HP Superdome*
  - 4 – 30 CPUs, Gigabytes of memory, Terabytes of disk
- High-performance super-cluster
  - Example: *APAC National Facility (Compaq Alphaserver)*
  - >100 CPUs, Gigabytes of memory, mass data storage

Standardisation



- Two different approaches
  - Rules based (traditional) [e.g. *AutoStan* software]
  - Probabilistic *Hidden Markov Models (HMM)* [new] (standardised training data needed)

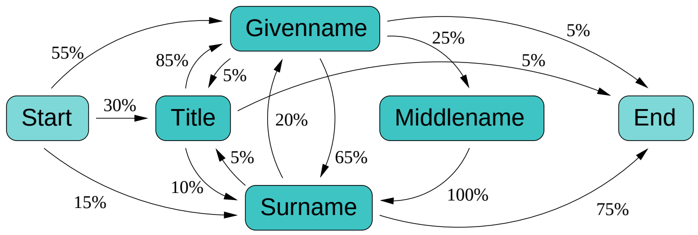
Standardisation Approach

- Four different components of a record: *Name*, *Geocode*, *Locality* and *Date*
- Assign words and numbers into 20 output fields
- Five processing steps...
  1. Clean (make lowercase, remove unwanted characters and replace certain characters with others)
  2. Split into a list of words, numbers and separators
  3. Correct misspellings and expand abbreviations using lookup-tables (*Python* dictionaries)
  4. Label list elements with tags (using lookup-tables)

Standardisation Approach (cont.)

- Example:
  - Input: *"Peter Miller-Meyer PhD", "42 Miller St (3A)"*
  - Cleaned and tagged:
    – Name word list: *[’peter’, ’miller’, ’-’, ’meyer’, ’doctor’]*
    – Name tag list: *[’GM/SN’, ’SN’, ’HY’, ’SN’, ’TI’]*
    – Geocode word list: *[’42’, ’miller’, ’street’, ’|’, ’3a’, ’|’]*
    – Geocode tag list: *[’NU’, ’UN’, ’WT’, ’VB’, ’AN’, ’VB’]*
- Last processing step
  5. Assign list elements to output fields
     (a) Rule based approach (complex programs)
     (b) Use *HMM* (one per component, structure learned from training data)

Name Hidden Markov Model Example



- Input to HMM are *tag lists*
  Word list *[’Doctor’, ’Peter’, ’Paul’, ’Miller’]* results in tag lists:
  [’TI’, ’GM’, ’GM’, ’SN’]
  [’TI’, ’SN’, ’GM’, ’SN’]
- Use *Viterbi* algorithm to compute most likely output sequence
  (for each tag list we get a probability, take the highest)

Linkage Approach

- Implement probabilistic linkage techniques *(Fellegi & Sunter)*
- Use frequency tables for surnames, suburbs, etc.
  (Sources: *Australian Whitepages*, *Australia Post*, etc.)
- Different phonetic name encodings
  (Soundex, NYSIIS, Double-Metaphone)
- Different approximate string comparators
  (Edit-Distance, Bigram, Winkler, Jaro-Winkler)
- Various blocking techniques
  (use name encodings)

Phonetic Name Encoding

- Bringing together spellings variations of the same name
- Examples:

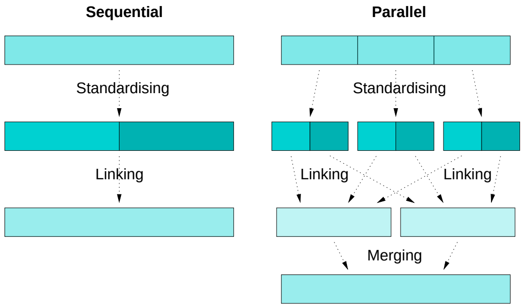
Name	Soundex	NYSIIS	Double-Metaphone
peter	p360	pata	ptr
christen	c623	chra	krst
nielsen	n425	nals	nlsn
markus	m622	marc	mrks
hegland	h245	hagl	hkln
stephen	s315	staf	stfn
steve	s310	staf	stf

Approximate String Comparison

- Account for partial agreement between strings
- Return score between *0.0* (totally different) and *1.0* (exactly the same)
- Examples:

String_1	String_2	Winkler	Bigram	Edit-Dist
tanya	tonya	0.880	0.500	0.800
dwayne	duane	0.840	0.222	0.667
sean	susan	0.805	0.286	0.600
jon	john	0.933	0.400	0.750
itman	smith	0.567	0.250	0.000
1st	ist	0.778	0.500	0.667
peter	ole	0.511	0.000	0.200

Parallelisation Approach



- Each record can be standardised independently
- Linkage is done using *blocking*
(each block can be processed independently)

Data Mining Approach

- *Data mining* and *machine learning* techniques to learn data characteristics
  - *Clustering* (as alternative for blocking?)
  - *Predictive modelling*
  - *Decision trees* and *rules* (for matches / non-matches?)
- *Training data* needed to build model
(pairs of known matches and known non-matches)

ANU Data Mining group has several years of experience in predictive modelling, handling of health data sets, data processing, etc.

Project Plan and Status

- Project started in January 2002 (officially March)
- Project plan
  - Standardisation *(January - July)*
  - Probabilistic Linkage *(August - September)*
  - Parallelisation *(September - October)*
  - Data Mining and Machine Learning *(November - March)*
- Free *Python* prototype software is available online
  - Routines for *Name*, *Address* and *Date* standardisation
  - Routines for *approximate string comparison* and *name encodings* (Soundex, NYSIIS, Metaphone, etc.)

Data Sets, Privacy and Ethics

- Real-world data sets are needed to develop and test record linkage software
- Data sets provided by *NSW Department of Health* under confidentiality agreement and ethics approval
- In epidemiological research, personal information is only needed for linkage, not for analysis, so it is removed after linkage is done
  (as opposed to *big brother* type of linkage for crime and fraud detection, anti-terrorism, business lists, etc.)

Outlook

- A new approach to probabilistic record linkage
  - Free open source software
  - High-performance and parallel computing
  - Data mining and machine learning techniques
- Future extension of this project likely
  - ARC Linkage grant for 2003
- Further collaborations are welcome
- Free prototype software available online:

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