Recent Developments in Data Linkage Technologies

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)

THE AUSTRALIAN NATIONAL UNIVERSITY

Outline

- Short introduction to data linkage techniques
- Probabilistic data cleaning and standardisation
- Modern blocking approaches
- Improved classification techniques
- Privacy preserving data linkage
- Measures for data linkage quality and complexity
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)
- Outlook

Recent interest in data linkage

- Traditionally, data linkage has been used in statistics and epidemiology
- In recent years, increased interest from computer science community
  - A lot of data is being collected by many organisations
  - Increased computing power and storage capacities
  - Data warehousing and data integration
  - Data mining of large data collections
  - E-Commerce and Web applications (for example `http://froogle.google.com` for online comparison of consumer products)
  - Geocoding and spatial data analysis

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 (*Fellegi & Sunter*, 1969)
  Use available (personal) information for linkage (which can be missing, wrong, coded differently, out-of-date, etc.)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, or information retrieval techniques (more later...)

Probabilistic data 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 & Kennedy* (1962)
- Theoretical foundation by *Fellegi & Sunter* (1969)
  - Compare common record attributes (or fields)
  - Compute matching weights based on frequency ratios (global or value specific ratios) and error estimates
  - Sum of the matching weights is used to classify a pair of records as *match*, *non-match*, or *possible match*
  - Problems: Estimating errors and threshold values, assumption of independence, and manual *clerical review*

Weight calculation: Month of birth

- Assume two data sets with a *3%* error in field *month of birth*
- Probability that two matched records (representing the same person) have the same month value is *97% (L agreement)*
- Probability that two matched records do not have the same month value is *3% (L disagreement)*
- Probability that two (randomly picked) un-matched records have the same month value is *1/12 = 8.3% (U agreement)*
- Probability that two un-matched records do not have the same month value 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}): $\log_2(0.03 / 0.917) = -4.92$

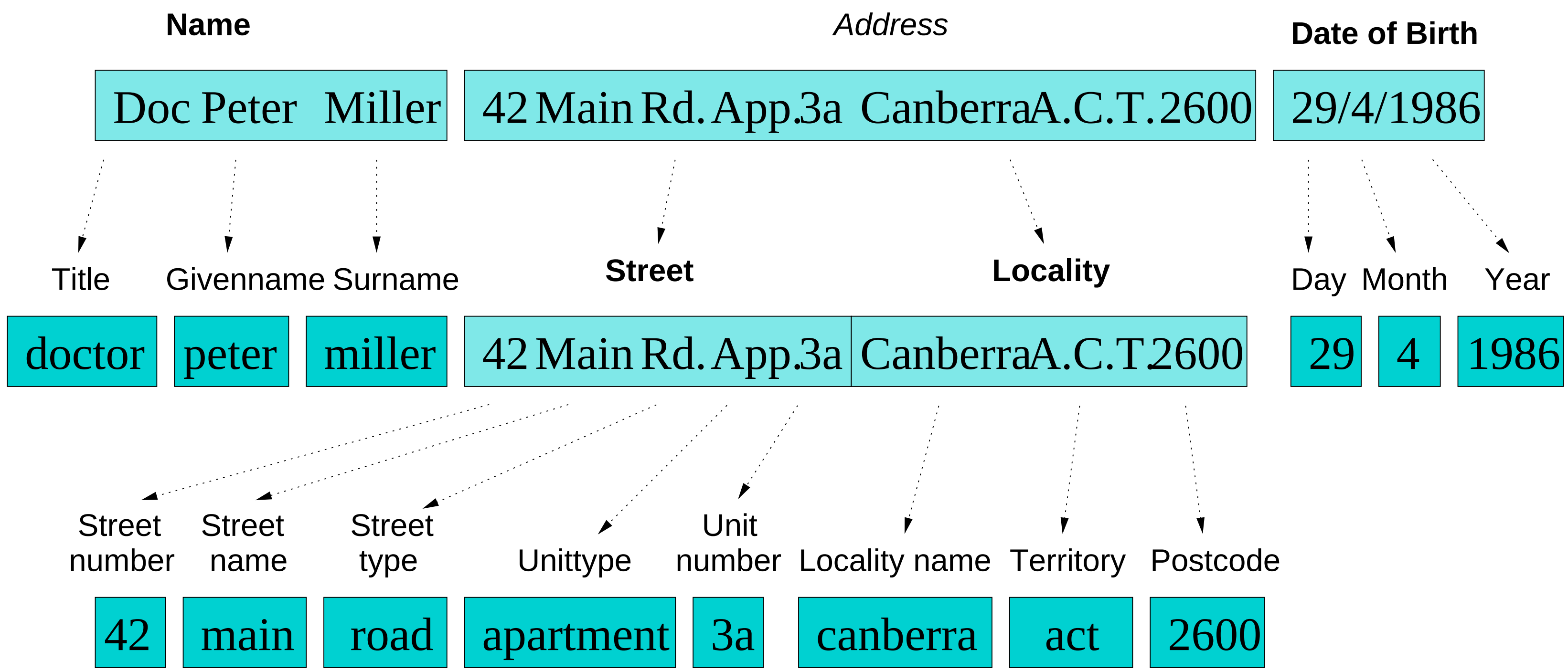
Outline

- Short introduction to data linkage techniques
- Probabilistic data cleaning and standardisation
- Modern blocking approaches
- Improved classification techniques
- Privacy preserving data linkage
- Measures for data linkage quality and complexity
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)
- Outlook

Data cleaning and standardisation

- 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 (phone, hand-written, scanned)
- Cleaned and standardised data is needed for
  - Loading into databases and data warehouses
  - Data mining and other data analysis studies
  - Data linkage and deduplication

Cleaning and standardisation steps



1. Remove unwanted characters and words
2. Expand abbreviations and correct misspellings
3. Segment data into well defined *output fields*

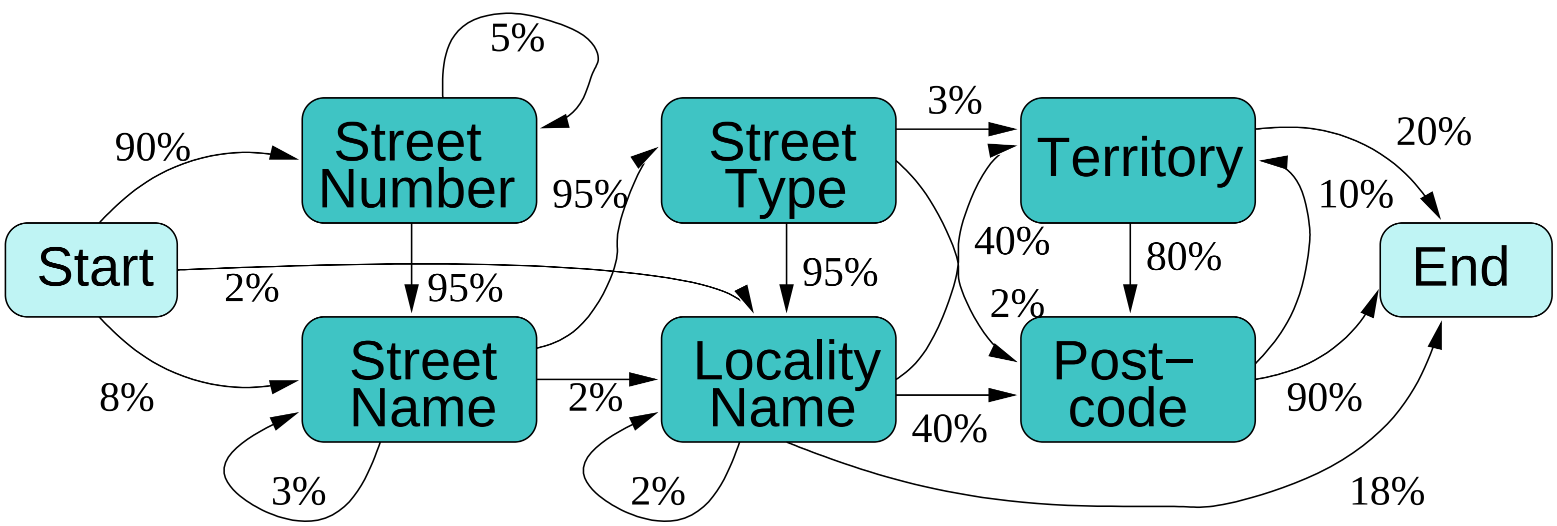
THE AUSTRALIAN NATIONAL UNIVERSITY

Traditional approach: Rule based

- User develops *rules* that cover as many variations in the input data as possible
  - Complex (hundreds or even thousands of rules needed)
  - Human expertise essential (rule system and domain)
  - Time consuming (try and refine)
  - Data dependent (update or modify rules for new data)
- Example: *AutoStan*
  - Re-entrant regular expression based
  - Rule files as developed by *NSW Health* over years:
    – 8,395 text lines for localities
    – 3,149 text lines for streets

New approaches: Probabilistic

- Mainly based on hidden Markov models (HMM) and related techniques
  - Probabilistic model used to *segment* input data (step 3)
  - Mainly useful for addresses (more structure than names)
  - Drawback: Model needs to be *trained*



Supervised techniques

- Probabilistic models need to be trained
  - With *supervised* approaches, manually prepared *training data* is needed
  - Generating training data is easier than creating rules
  - *Bootstrapping* approach can facilitate the training process
  - *Active learning* approach can help selecting good training examples
- Alternative: Use large, complete, and clean databases to train a model automatically
  - Based on *attribute recognition model (ARM)*

Un-supervised techniques

- In Australia, we can use *G-NAF*
  - G-NAF: Geocoded National Address File
  - Several million complete, correct and segmented address records ($\sim$4.5 million for NSW)
  - 26 address attributes (level, flat, street, building, locality, postcode, and state)
  - Type and length of values characterise attributes
    Examples: 3-letter value in 89% corresponds to a state, 4-letter value is in 77% a street type, etc.
- Current research project
  (ANU computer science honours: Combine HMM with ARM for automated address standardisation)

Outline

- Short introduction to data linkage techniques
- Probabilistic data cleaning and standardisation
- Modern blocking approaches
- Improved classification techniques
- Privacy preserving data linkage
- Measures for data linkage quality and complexity
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)
- Outlook

THE AUSTRALIAN NATIONAL UNIVERSITY

Why blocking?

- Number of record pair comparisons equals the product of the sizes of the two data sets
  (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 or 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

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*

Improved blocking approaches

- Recent research methods
  - Sorted neighbourhood approach
    (sliding window over sorted blocking variable)
  - Fuzzy blocking using *n*-grams (e.g. *bigrams*)
    (*'peter'* $\rightarrow$ *['pe','et','te','er']*, *'pete'* $\rightarrow$ *['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; with all blocking passes in one go)

Outline

- Short introduction to data linkage techniques
- Probabilistic data cleaning and standardisation
- Modern blocking approaches
- Improved classification techniques
- Privacy preserving data linkage
- Measures for data linkage quality and complexity
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)
- Outlook

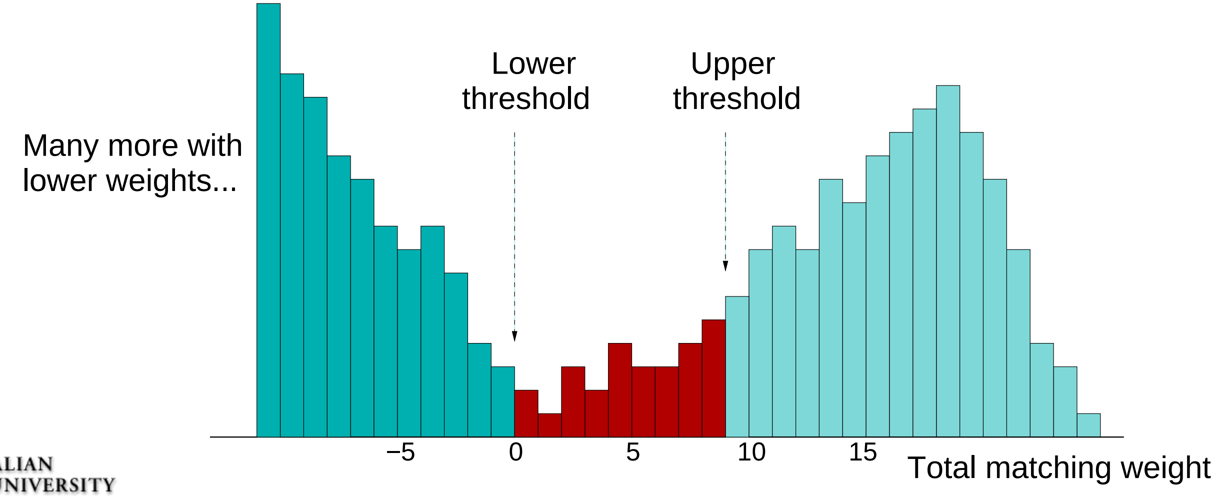
THE AUSTRALIAN NATIONAL UNIVERSITY

Fellegi and Sunter classification

- For each compared record pair a vector containing *matching weights* is calculated

  Record A: `['dr', 'peter', 'paul', 'miller']`
  Record B: `['mr', 'john', '', 'miller']`
  Matching weights: `[0.2, -3.2, 0.0, 2.4 ]`

- *Fellegi & Sunter* approach sums all weights
  (then uses two thresholds to classify record pairs as *non-matches*, *possible matches*, or *matches*)



THE AUSTRALIAN NATIONAL UNIVERSITY

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 *multi-dimensional 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

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*)
- Recent small repository: *RIDDLE*

  `http://www.cs.utexas.edu/users/ml/riddle/`

  (Repository of Information on Duplicate Detection, Record Linkage, and Identity Uncertainty)

Outline

- Short introduction to data linkage techniques
- Probabilistic data cleaning and standardisation
- Modern blocking approaches
- Improved classification techniques
- Privacy preserving data linkage
- Measures for data linkage quality and complexity
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)
- Outlook

Privacy and confidentiality issues

- Traditionally, data linkage requires that *identified data* is being given to the person or institution doing the linkage
- Privacy of individuals in data sets is invaded
  - Consent of individuals involved is needed
  - Alternatively, seek approval from ethics committees

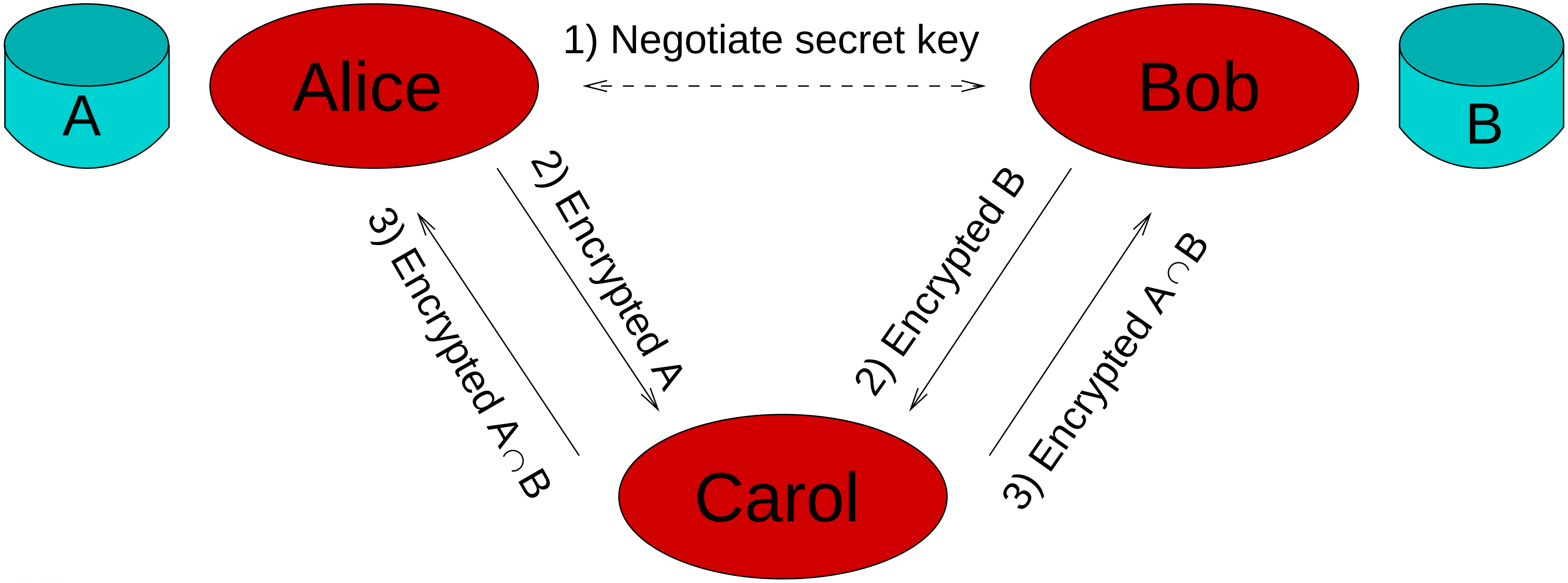
Invasion of privacy could be avoided (or mitigated) if some method were available to determine which records in two data sets match, without revealing any identifying information.

Privacy preserving approach

- *Alice* has a database **A** she wants to link with *Bob* (without revealing the actual values in **A**)
- *Bob* has a database **B** he wants to link with *Alice* (without revealing the actual values in **B**)
- Easy if only *exact matches* are considered
  - Encrypt data using one-way hashing (like *SHA*)
  - Example: *'tim'* $\longrightarrow$ *'51ddc7d3a611eeba6ca770'*
- More complicated if values contain errors or typographical variations
  (even a single character difference between two strings will result in very different hash encodings)

Third party linkage protocol

- *Alice* and *Bob* negotiate a shared secret key
  (for example a 160 bit long SHA hash code)
- A third party (*Carol*) performs the actual linkage
- Only encrypted data is transmitted



Privacy preserving research

- Pioneered by French researchers in mid-to-late 1990s (for situations where de-identified data needs to be centralised and linked for follow-up studies)
- *Blindfolded record linkage*
  [Churches and Christen, 2004] (allow approximate linkage of strings with typographical errors based on *n*-gram techniques)
- *Privacy-preserving data linkage protocols*
  [O'Keefe et.al., 2004] (several protocols with improved security and less information leakage)
- *Blocking aware private record linkage*
  [Al-Lawati et.al., 2005] (approximate linkage based on tokens and TF-IDF, and three blocking approaches)

Outline

- Short introduction to data linkage techniques
- Probabilistic data cleaning and standardisation
- Modern blocking approaches
- Improved classification techniques
- Privacy preserving data linkage
- Measures for data linkage quality and complexity
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)
- Outlook

Measuring data linkage quality

- Classifying record pairs results in four outcomes
  1. True matches classified as matches *(True Pos)*
  2. True matches classified as non-matches *(False Neg)*
  3. True non-matches classified as matches *(False Pos)*
  4. True non-matches classified as non-matches *(True Neg)*
- Various quality measures
  - Accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$
  - Precision (or positive predictor value): $\frac{TP}{TP+FP}$
  - Recall (or sensitivity): $\frac{TP}{TP+FN}$
  - Specificity (or true negative rate): $\frac{TN}{TN+FP}$

Measuring quality issues

- Big question: *What to count?*
  - Actually compared record pairs (after blocking)?
  - All possible record pairs (full comparison space)?
  - Matched and non-matched *entities*?
- When counting record pairs, the number of TN will be increased quadratically
  (but not the numbers of TP, FN and FP)
  - Quality measures which include the number of TN can produce deceptive accuracy results
- Blocking also affects quality measures
  (aim of blocking is to remove as many TN and FP as possible, without removing any TP and FN)

Measuring data linkage complexity

- Recently proposed measures on blocking performance
  - Reduction ratio: $1 - \frac{N_b}{|\mathbf{A}| \times |\mathbf{B}|}$
    (with $N_b \leq |\mathbf{A}| \times |\mathbf{B}|$ being the number of record pairs produced by a blocking algorithm)
  - Pairs completeness: $\frac{N_m}{|M|}$
    (with N_m being the number of correctly classified true matched record pairs (TP) in the blocked comparison space, and $|M|$ total number of true matches)
- There is a trade-off between the reduction ratio and pairs completeness
  (similar to the precision-recall trade-off)

Outline

- Short introduction to data linkage techniques
- Probabilistic data cleaning and standardisation
- Modern blocking approaches
- Improved classification techniques
- Privacy preserving data linkage
- Measures for data linkage quality and complexity
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)
- Outlook

Our project: Febrl

- Aims at developing new and improved techniques for parallel large scale data linkage
- Main research areas
  - Probabilistic techniques for automated data cleaning and standardisation (mainly on addresses)
  - New and improved blocking and indexing techniques
  - Improved record pair classification using (un-supervised) machine learning techniques (reduce clerical review)
  - Improved performance (scalability and parallelism)
- Project Web page:

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

THE AUSTRALIAN NATIONAL UNIVERSITY

Febrl prototype software

- An experimental platform for new and improved data linkage algorithms
- Modules for data cleaning and standardisation, data linkage, deduplication, geocoding, and data set generation
- Open source **https://sourceforge.net/projects/febrl/**
- Implemented in *Python* **http://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

Outlook

- Recent interest in data linkage from the computer science community
  - Data mining and data warehousing
  - E-Commerce and Web applications
- Main improvements
  - More automated data standardisation and linkage
  - More accurate linkage
  - Higher performance (linking larger data sets)
  - Early research in privacy preserving data linkage
- For more information see project Web page (publications, talks, software, further links)

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)
- Mr Alan Willmore (New South Wales Health Department, Centre for Epidemiology and Research)
- Ms Kim Lim (New South Wales Health Department, Centre for Epidemiology and Research)
- Mr Karl Goiser (ANU Computer Science PhD student)
- Mr Daniel Belacic (ANU Computer Science honours student, 2005)
- 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)