Data Linkage Techniques: Past, Present and Future

Peter Christen

Department of Computer Science, The Australian National University

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

Project Web site: **http://datamining.anu.edu.au/linkage.html**

Funded by the Australian National University, the NSW Department of Health, the Australian Research Council (ARC) under Linkage Project 0453463, and the Australian Partnership for Advanced Computing (APAC)

Outline

- What is data linkage?
  - Applications and challenges
- The past
  - A short history of data linkage
- The present
  - Computer science based approaches: *Learning to link*
- The future
  - Scalability, automation, and privacy and confidentiality
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)

What is data (or record) linkage?

- The process of linking and aggregating records from one or more data sources representing the same entity (patient, customer, business name, etc.)
- Also called *data matching*, *data integration*, *data scrubbing*, *ETL (extraction, transformation and loading)*, *object identification*, *merge-purge*, etc.
- Challenging if no unique entity identifiers available
  E.g., which of these records represent the same person?

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

Recent interest in data linkage

- Traditionally, data linkage has been used in health (epidemiology) and statistics (census)
- In recent years, increased interest from businesses and governments
  - A lot of data is being collected by many organisations
  - Increased computing power and storage capacities
  - Data warehousing and distributed databases
  - Data mining of large data collections
  - E-Commerce and Web applications (for example online product comparisons: `http://froogle.com`)
  - Geocoding and spatial data analysis

Applications and usage

- Applications of data 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
  - Geocode matching (with reference address data)
- Widespread use of data linkage
  - Immigration, taxation, social security, census
  - Fraud, crime and terrorism intelligence
  - Business mailing lists, exchange of customer data
  - Social, health and biomedical research

Challenge 1: Dirty data

- Real world data is often *dirty*
  - Missing values, inconsistencies
  - Typographical errors and other variations
  - Different coding schemes / formats
  - Out-of-date data
- Names and addresses are especially prone to data entry errors (over 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

Challenge 2: Scalability

- Data collections with tens or even hundreds of millions of records are not uncommon
- Number of possible record pairs to compare equals the product of the sizes of the two data sets (linking two data sets with *1,000,000* records each will result in $10^6 \times 10^6 = 10^{12}$ record pairs)
- Performance bottleneck in a data linkage system is usually the (expensive) comparison of attribute (field) values between record pairs
- Blocking / indexing / filtering techniques are used to reduce the large amount of comparisons
- Linkage process should be automatic

Challenge 3: Privacy and confidentiality

- General public is worried about their information being linked and shared between organisations
  - Good: research, health, statistics, crime and fraud detection (taxation, social security, etc.)
  - Scary: intelligence, surveillance, commercial data mining (not much information from businesses, no regulation)
  - Bad: identify fraud, re-identification
- Traditionally, *identified data* has to be given to the person or organisation performing the linkage
  - Privacy of individuals in data sets is invaded
  - Consent of individuals involved is needed
  - Alternatively, seek approval from ethics committees

Outline: The past

- What is data linkage?
  - Applications and challenges
- The past
  - A short history of data linkage
- The present
  - Computer science based approaches: *Learning to link*
- The future
  - Scalability, automation, and privacy and confidentiality
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)

Traditional data linkage techniques

- Computer assisted data linkage goes back as far as the 1950s (based on ad-hoc heuristic methods)
- Deterministic linkage
  - Exact linkage, if a *unique identifier* of high quality is available (has to be precise, robust, stable over time)
    Examples: *Medicare*, *ABN* or *Tax file* number
    (are they *really* unique, stable, trustworthy?)
  - Rules based linkage (complex to build and maintain)
- Probabilistic linkage
  - Apply linkage using available (personal) information (like *names*, *addresses*, *dates of birth*, etc)

Probabilistic data linkage

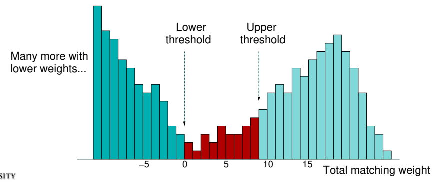
- Basic ideas of probabilistic linkage were introduced by *Newcombe & Kennedy* (1962)
- Theoretical foundation by *Fellegi & Sunter* (1969)
  - No unique entity identifiers available
  - Compare common record attributes (or fields)
  - Compute matching weights based on frequency ratios (global or value specific) 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*
  - Still the basis of many linkage systems

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*)



Traditional blocking

- Traditional blocking works by only comparing record pairs that have the same value for a *blocking variable* (for example, only compare records that 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*

Outline: The present

- What is data linkage?
  - Applications and challenges
- The past
  - A short history of data linkage
- The present
  - Computer science based approaches: *Learning to link*
- The future
  - Scalability, automation, and privacy and confidentiality
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)

Improved classification

- Summing of matching weights results in loss of information (e.g. two record pairs: same name but different address ⇔ different address but same name)
- View record pair classification as a *multi-dimensional binary classification* problem
  (use matching weight vectors to classify record pairs into *matches* or *non-matches*, but **no** *possible matches*)
- Different machine learning techniques can be used
  - Supervised: *Manually prepared* training data needed (record pairs and their match status), almost like manual clerical review *before* the linkage
  - Un-supervised: Find (local) structure in the data (similar record pairs) without 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 test data set collection available
  (like in information retrieval or data mining)
  - Recent small repository: *RIDDLE*
    `http://www.cs.utexas.edu/users/ml/riddle/`
    (Repository of Information on Duplicate Detection, Record Linkage, and Identity Uncertainty)

Classification research (1)

- Information retrieval based
  - Represent records as document vectors
  - Calculate distance between vectors (*tf-idf* weights)
- Database research approaches
  - Extend SQL language (fuzzy join operator)
  - Implement linkage algorithms using SQL statements
- Supervised machine learning techniques
  - Learn string distance measures (edit-distance costs for character insert, delete, substitute)
  - Decision trees, genetic programming, association rules, expert systems, etc.

Classification research (2)

- Semi-supervised techniques
  - Aim is to reduce manual training effort
  - Active learning (select a record pair for manual classification that a set of classifiers disagree on the most)
  - Semi-supervised clustering (provide some manually classified record pairs)
- Un-supervised techniques
  - Clustering techniques (k-means, farthest first, etc.)
  - Hierarchical graphical models (probabilistic approaches)
- Main critic point: Often only confidential or small test data sets (like bibliographic citations, restaurant names, etc.)

Blocking research

- Sorted neighbourhood approach
  (sliding window over sorted blocking variable)
- Fuzzy blocking using *q*-grams (e.g. *bigrams*)
  (*'peter'* → *['pe','et','te','er']*, *'pete'* → *['pe','et','te']*)
- Overlapping *canopy* clustering
  (cheaply insert records into several clusters)
- Post-blocking filtering
  (like length differences or *q*-grams count differences)
- Supervised learning for blocking
  (minimise removal of true matches by the blocking process)
- US Census Bureau: *BigMatch*
  (pre-process 'smaller' data set so its record values can be accessed directly; with all blocking passes in one go)

Outline: The future

- What is data linkage?
  - Applications and challenges
- The past
  - A short history of data linkage
- The present
  - Computer science based approaches: *Learning to link*
- The future
  - Scalability, automation, and privacy and confidentiality
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)

The main future challenges

- Scalability
  New computational techniques are required to allow large scale linking of massive data collections on modern parallel and distributed computing platforms.
- Automation
  Decision models are needed that will reduce or even eliminate the manual clerical review (or preparation of training data) while keeping a high linkage quality.
- Privacy and confidentiality
  Techniques are required that will allow the linking of large scale data collections between organisations without revealing any personal or confidential information.
- Public acceptance

Scalability / Computational issues

- Aim is to be able to link very large data sets with tens to hundreds of millions of records
  - Large parallel machines needed (super-computers)
  - Alternatively, use networked PCs / workstations (run linkage jobs in background, over night or weekends)
- Linkage between organisations
  - Parallel and/or distributed computing platforms (clusters, computational grids, Web services)
  - Fault tolerance (networks, computing nodes), (dynamic) load balancing, heterogeneous platforms (standards, transformations, meta data)
  - Security, access, interfaces, charging policies, etc.

Privacy and confidentiality issues

- Traditional data linkage requires that *identified data* is given to the person or organisation performing the linkage (names, address, dates of birth, etc.)
  - Approval from ethics committees is required as it is unfeasible to get consent from large number of individuals
  - Complete trust in linkage organisation, their staff, and computing and networking systems

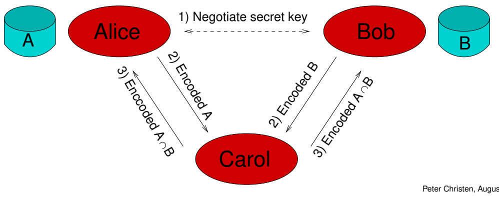
Invasion of privacy could be avoided (or mitigated) if some method were available to determine which records in two data sets match, without revealing any identifying information.

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
  - Encode data using one-way hashing (like *SHA*)
  - Example: *'peter'* → *'51ddc7d3a611eeba6ca770'*
- More complicated if values contain typographical errors or other variations
  (even a single character difference between two strings will result in very different hash encodings)

Three-party linkage protocol

- *Alice* and *Bob* negotiate a shared secret key
- They encode their data (using this key) and send it to a third party (*Carol*) that performs the linkage
- Results are sent back to *Alice* and *Bob*
- All transmitted data is encrypted using a public key infrastructure (PKI)



Privacy preserving research

- Pioneered by French researchers in 1990s
  [Quantin et al., 1998] (for situations where de-identified data needs to be centralised and linked for follow-up studies)
- *Blindfolded record linkage*
  [Churches and Christen, 2004] (allows approximate linkage of strings with typographical errors based on *q*-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: Our project: Febrl

- What is data linkage?
  - Applications and challenges
- The past
  - A short history of data linkage
- The present
  - Computer science based approaches: *Learning to link*
- The future
  - Scalability, automation, and privacy and confidentiality
- Our project: *Febrl*
  (Freely extensible biomedical record linkage)

The Febrl project

- 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 site:

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

Febrl prototype software

- An experimental platform for new and improved data linkage algorithms
- Modules for data cleaning and standardisation, data linkage, deduplication, geocoding, and generation of synthetic data sets
- 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

What can you do..?!

- Commercial data linkage consultancies and software are expensive (~$10,000 to many $100,000)
- Support local research and training
  - Develop local knowledge and expertise, rather than relying upon overseas software vendors
  - Training of PhD, Masters and honours students
- Australian Research Council (ARC) *Linkage* projects
  - Partially funded by industry / government organisation
  - Develop techniques and methods specific to industry
  - Smallest contributions ~ $15,000 plus in-kind (per annum over 3-4 years)

Outlook

- Recent interest in data linkage from governments and businesses
  - Data mining and data warehousing
  - E-Commerce and Web applications
  - Census, crime/fraud detection, intelligence/surveillance
- Main future challenges
  - Automated and accurate linkage
  - Higher performance (linking very large data sets)
  - Secure and privacy-preserving data linkage
- For more information see our project Web site
  (publications, talks, software, Web resources / 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)