Recent Developments and Research Challenges in Data Linkage

Peter Christen

Research School of Computer Science,

ANU College of Engineering and Computer Science,

The Australian National University

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

This work was partially funded by the Australian Research Council (ARC) under Discovery Projects DP130101801 and DP160101934; and the ARC, Veda, and Funnelback Pty. Ltd., under Linkage Project LP100200079.

Outline

- A short introduction to data linkage
- Challenges of data linkage
- Techniques for scalable real-time data linkage
- Advanced classification techniques for data linkage (active learning)
- Privacy aspects in data linkage
- Research directions

What is data linkage?

- The process of linking records that represent the same entity in one or more databases
  (patients, customers, businesses, consumer products, publications, etc.)
- Also known as *record linkage*, *data matching*, *entity resolution*, *duplicate detection*, etc.
- Major challenge is that unique *entity identifiers* are not available in the databases to be linked
  (or if available, they are not consistent or change over time)

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 Rd 2600 Canberra ACT*

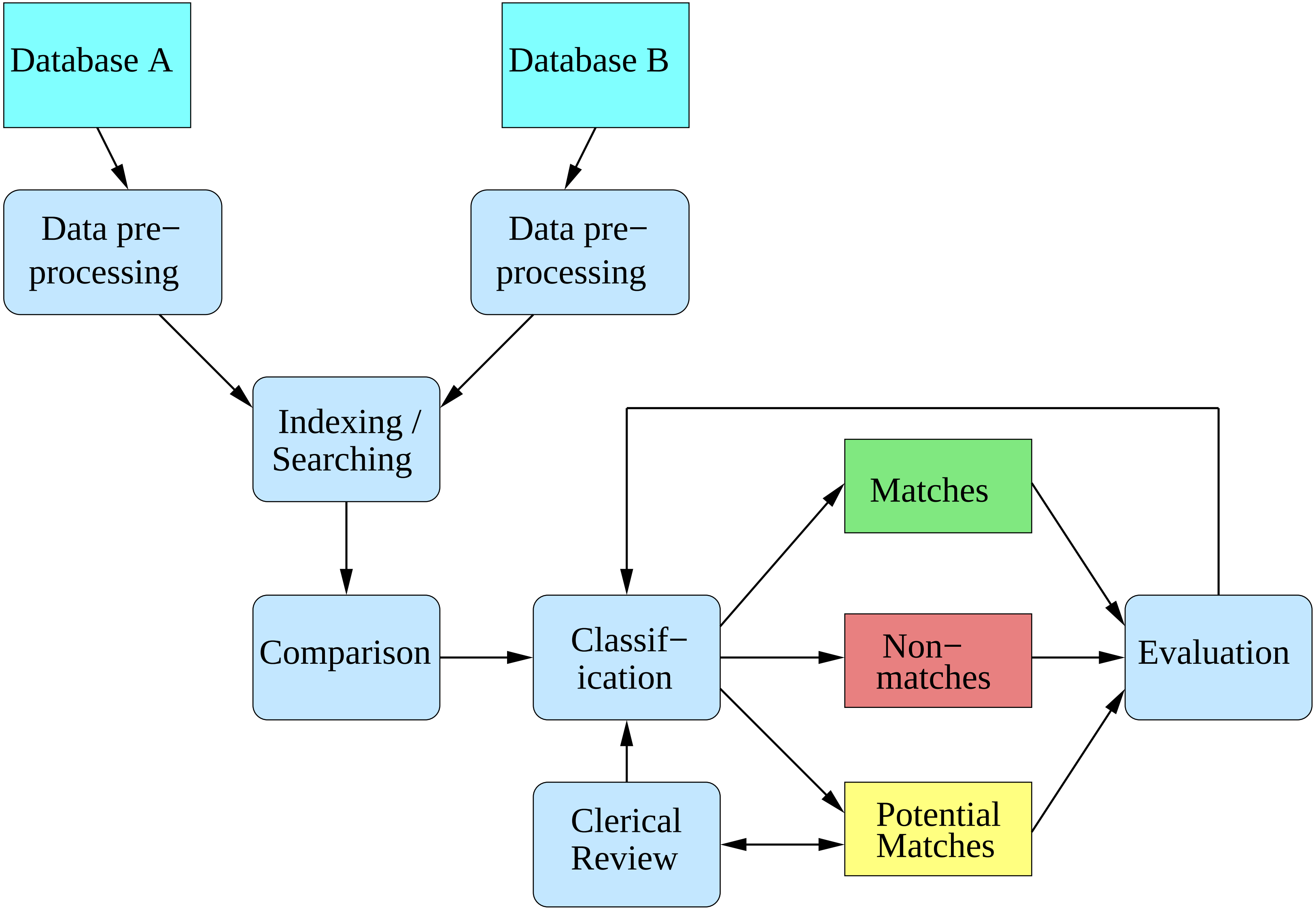
Applications of data linkage

- Remove duplicates in one data set (deduplication)
- Merge new records into a larger master data set
- Create patient or customer oriented statistics (for example for longitudinal studies)
- Clean and enrich data for analysis and mining
- 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
  - Health and social science research

Recent interest in data linkage

- Traditionally, data linkage has been used in statistics (census) and health (epidemiology)
  - First computer based techniques developed in 1960s
- In recent years, increased interest from businesses and governments
  - Massive amounts of data are being collected, and computing power and storage capacities are increasing
  - Often data from different sources need to be integrated
  - Need for data sharing between organisations
  - Data mining (analysis) of large data collections
  - E-Commerce and Web services (comparison shopping)
  - Spatial data analysis and online map applications

The data linkage process



Data linkage techniques

- Deterministic matching
  - Rule based matching (complex to build and maintain)
- Probabilistic record linkage (*Fellegi and Sunter*, 1969)
  - Use available attributes for linking (often personal information, like names, addresses, dates of birth, etc.)
  - Calculate match weights for attributes
- "Computer science" approaches
  - Based on machine learning, data mining, database, or information retrieval techniques
  - Supervised classification: Requires training data (true matches)
  - Unsupervised: Clustering, collective, and graph based

Major data linkage challenges

- No unique entity identifiers available
- Real world data are dirty

  (typographical errors and variations, missing and out-of-date values, different coding schemes, etc.)
- Scalability
  - Naïve comparison of all record pairs is quadratic
  - Remove likely non-matches as efficiently as possible
- No training data in many linkage applications
  - No record pairs with known true match status
- Privacy and confidentiality

  (because personal information, like names and addresses, is commonly required for linking)

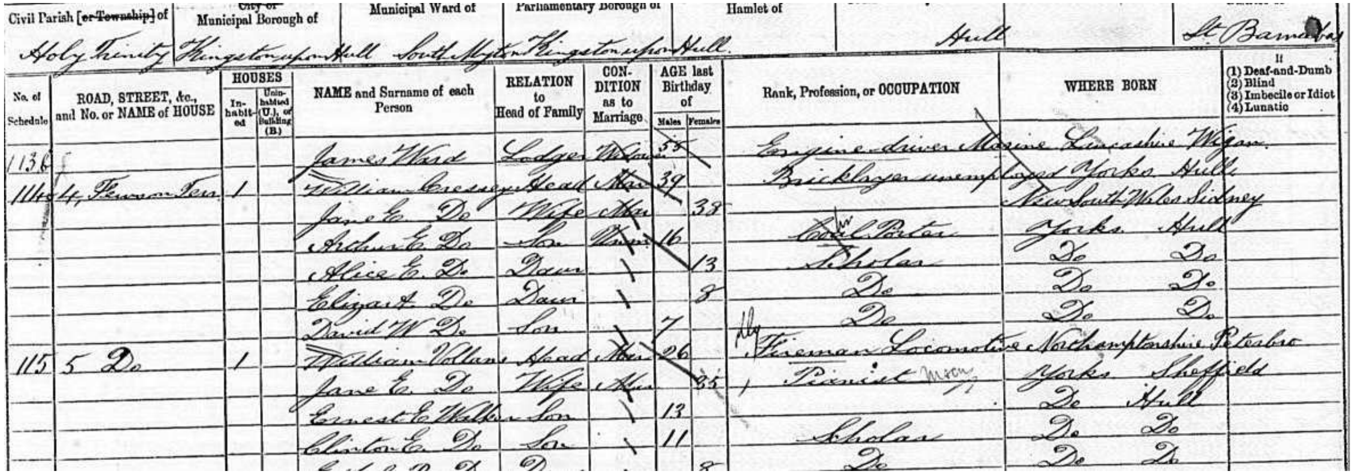
Challenges for linking Big Data

- Size (*volume*) and complexity (*variety*) of data
  - Possibly hundreds of millions of records about entities
  - From many different sources
  - Containing more complex types and more detailed data (free-format text or multimedia)
- Dynamic nature of Big Data (*velocity*)
  - Streams of data (unpredictable rate and volume)
  - (Near) real-time linking and analysis are required
- Trustworthiness of (external) data (*veracity*)
- Diverse requirements on linked data
- Privacy and confidentiality

Challenges for linking social science data

- Increasing use of large databases in social science research
- Aim is to create ‘*social genomes*’ for individuals by linking population databases
  (*Population Informatics*, Kum et al. IEEE Computer, 2013)
- Knowing how individuals and families change over time allows for a diverse range of studies
  (fertility, employment, education, health, crime, etc.)
- Different challenges for historical data compared to present-day data, but some are common
  - Database sizes (computational aspects)
  - Accurate match classification (data quality)

Challenges for historical data



Civil Parish [~~or Township~~] of	City or Municipal Borough of	Municipal Ward of	Parliamentary Borough of	Hamlet of		
Holy Trinity	Kingston upon Hull	South Myton	Kingston upon Hull		Hull	St Barnabas

No. of Schedule	ROAD, STREET, &c., and No. or NAME of HOUSE	HOUSES		NAME and Surname of each Person	RELATION to Head of Family	CONDITION as to Marriage	AGE last Birthday of		Rank, Profession, or OCCUPATION	WHERE BORN	If (1) Deaf-and-Dumb (2) Blind (3) Imbecile or Idiot (4) Lunatic
		Inhabited	Uninhabited (U.), or Building (B.)				Males	Females			
113				James Ward	Lodger	Widr	55		Engine driver Marine	Lancashire Wigan	
114	4, Fernon Terr	1		William Cressey	Head	Mar	39		Bricklayer unemployed	Yorks Hull	
				Jane E. Do	Wife	Mar		38		New South Wales Sidney	
				Arthur E. Do	Son	Unm	16		Coal Porter	Yorks Hull	
				Alice E. Do	Daur			13	Scholar	Do Do	
				Eliza A. Do	Daur			8	Do	Do Do	
				David W. Do	Son		7		Do	Do Do	
115	5 Do	1		William Vollans	Head	Mar	26		Fireman Locomotive	Northamptonshire Peterbro	
				Jane E. Do	Wife	Mar		35	Pianist	Yorks Sheffield	
				Ernest E Walker	Son		13			Do Hull	
				Clinton E. Do	Son		11		Scholar	Do Do	
				[illegible]	D[illegible]			8	Do	Do Do	

- Low literacy (recording errors and unknown exact values), no address or occupation standards
- Large percentage of a population had one of just a few common names (‘John’ or ‘Mary’)
- Households and families change over time
- Immigration and emigration, birth and death
- Scanning, OCR, and transcription errors

Challenges for present-day data

- Privacy is of concern as data are about people alive today (when data are linked between organisations)
- Linked data allow analysis not possible on single databases (potentially revealing sensitive information)
- This makes access to suitable data sources that are required for a linkage project challenging
  - Assume a researcher is interested in analysing the effects of car accidents upon the health system
  - She needs access to data from hospitals, doctors, car and health insurers, from the police, etc.
  - All identifying data have to be given to the researcher, or alternatively a trusted data linkage unit

Scalable real-time data linkage, and Advanced classification

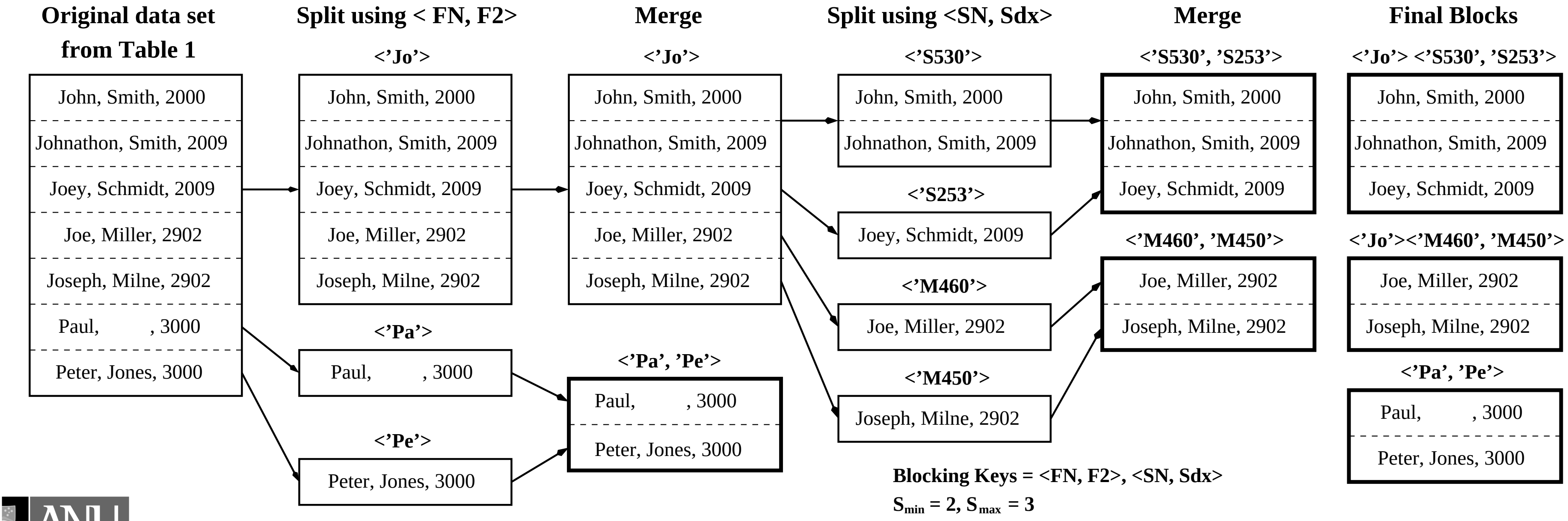
Techniques for scalable data linkage

- Number of all record pair comparisons equals the product of the sizes of the two databases
  - But the number of true matches is generally less than the number of records in the smaller of the two databases (assuming no duplicate records)
- Performance bottleneck in data linkage is usually the detailed comparison of attribute values (using approximate (string) comparison functions)
- Aim of indexing / blocking: Cheaply remove record pairs that are obviously not matches
- Traditional blocking only compares record pairs with the same value in a *blocking key* (for example, only compare records with the same *postcode*)

Controlling block sizes (1)

(Fisher et al., SIGKDD, 2015)

- Many blocking techniques generate blocks of different sizes (depending upon data characteristics)
  - Having blocks within a certain range is important for real-time and privacy-preserving record linkage, and with certain machine learning algorithms
- We employ an iterative split-merge approach



Original data set from Table 1

John, Smith, 2000
Johnathon, Smith, 2009
Joey, Schmidt, 2009
Joe, Miller, 2902
Joseph, Milne, 2902
Paul, , 3000
Peter, Jones, 3000

Split using < FN, F2>

<'Jo'>
John, Smith, 2000
Johnathon, Smith, 2009
Joey, Schmidt, 2009
Joe, Miller, 2902
Joseph, Milne, 2902

<'Pa'>
Paul, , 3000

<'Pe'>
Peter, Jones, 3000

Merge

<'Jo'>
John, Smith, 2000
Johnathon, Smith, 2009
Joey, Schmidt, 2009
Joe, Miller, 2902
Joseph, Milne, 2902

<'Pa', 'Pe'>
Paul, , 3000
Peter, Jones, 3000

Split using <SN, Sdx>

<'S530'>
John, Smith, 2000
Johnathon, Smith, 2009

<'S253'>
Joey, Schmidt, 2009

<'M460'>
Joe, Miller, 2902

<'M450'>
Joseph, Milne, 2902

Merge

<'S530', 'S253'>
John, Smith, 2000
Johnathon, Smith, 2009
Joey, Schmidt, 2009

<'M460', 'M450'>
Joe, Miller, 2902
Joseph, Milne, 2902

Final Blocks

<'Jo'> <'S530', 'S253'>
John, Smith, 2000
Johnathon, Smith, 2009
Joey, Schmidt, 2009

<'Jo'><'M460', 'M450'>
Joe, Miller, 2902
Joseph, Milne, 2902

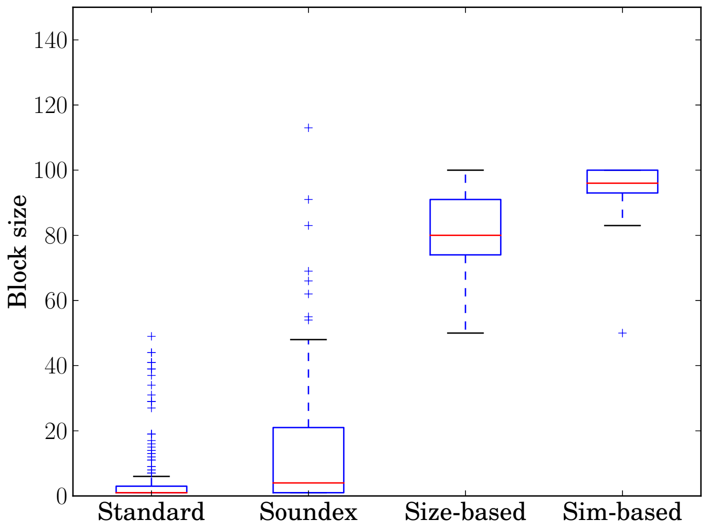
<'Pa', 'Pe'>
Paul, , 3000
Peter, Jones, 3000

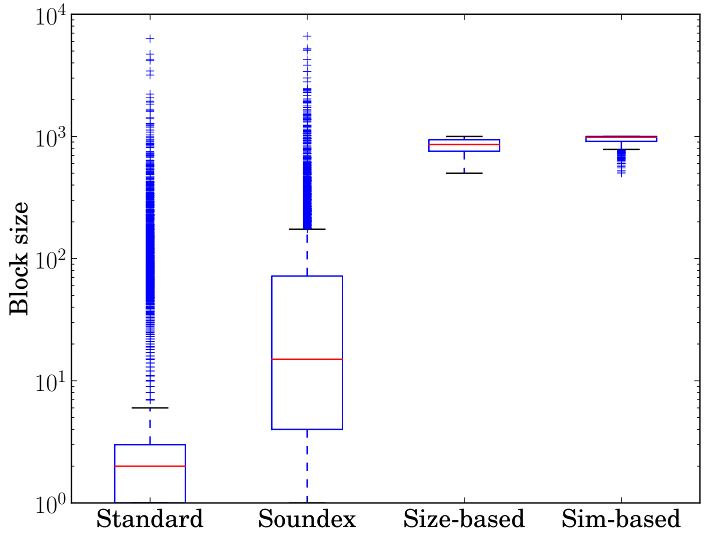
Blocking Keys = <FN, F2>, <SN, Sdx>
S_{min} = 2, S_{max} = 3

Controlling block sizes (2)

- Split data set into an initial set of blocks using a first blocking key (such as first two letters of first name)
- Merge blocks that are smaller than s_{min}
- Split blocks that are larger than s_{max} using a second blocking key (such as *Soundex* of surname)
- Continue to split and merge until all blocks have a size between s_{min} and s_{max}
- Two approaches that order blocks differently during the merge step (similarity and size based)
- Using a penalty function allows even greater control over the merging step
  (trade-off between merge similarities and block sizes for blocks larger than s_{max})

Experimental results (1)

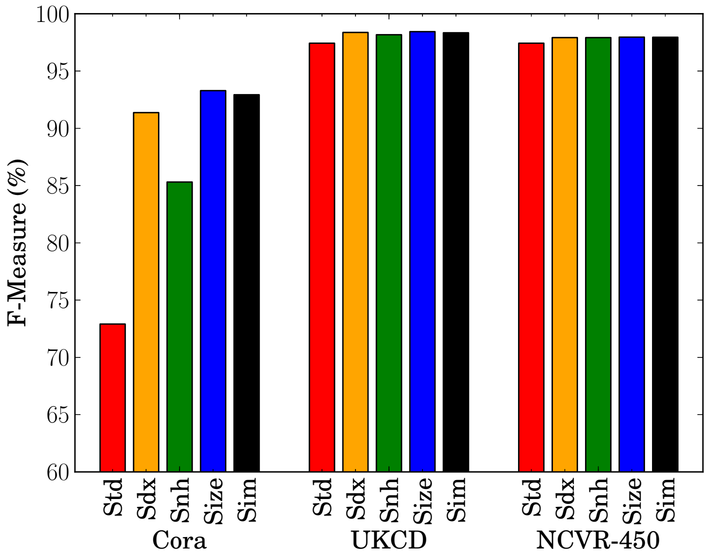


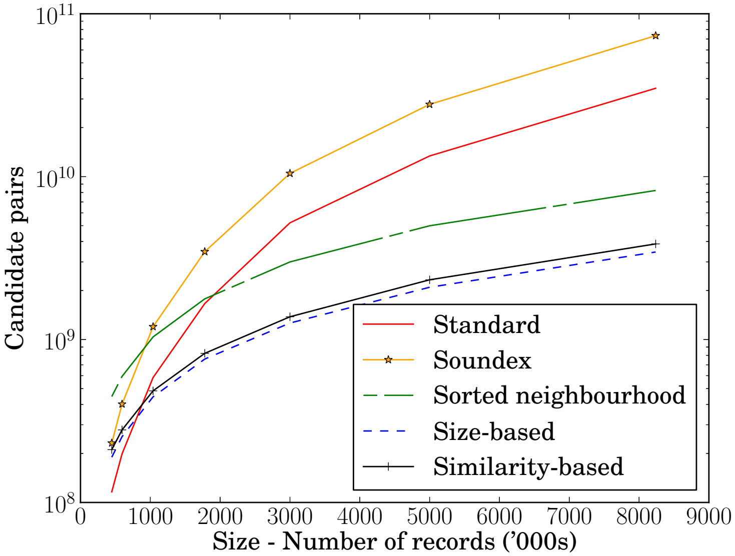


- Left: Cora (bibliographic data) with blocking key: $\langle$*title*,*exact*$\rangle$ and $\langle$*author*,*exact*$\rangle$

  Right: North Carolina voter registration data with blocking key: $\langle$*surname*,*first two*$\rangle$ and $\langle$*first name*,*first two*$\rangle$
- Baselines: Standard blocking (without and with *Soundex* encoded values)

Experimental results (2)





- Increase in linkage quality for Cora data set
- Significantly improved performance due to reduction in the number of record pairs compared
- Controlling the maximum size of blocks ensures the total number of candidate pairs increases linearly with data set size

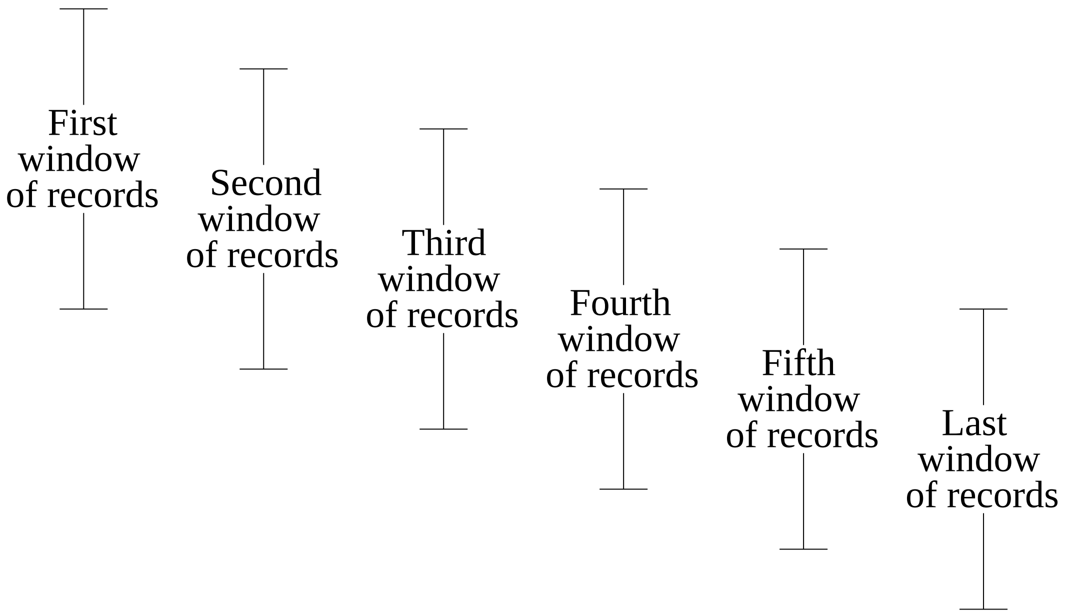
Dynamic indexing for real-time linkage (1) *(Ramadan et al., ACM JDIQ, 2015)*

- Most indexing / blocking techniques assume static databases, and they work offline using batch linkage of full databases
- Real-time linkage requires matching of query records with entity records in a (large) database
- Once matched, query records are added to the database (a dynamic index data structure is required)
- Static sorted neighbourhood indexing
  - Sliding window over sorted databases
  - Use several passes with different sorting criteria
  - Window size can be fixed or adaptive (based on similarities between records)

Dynamic indexing for real-time linkage (2)

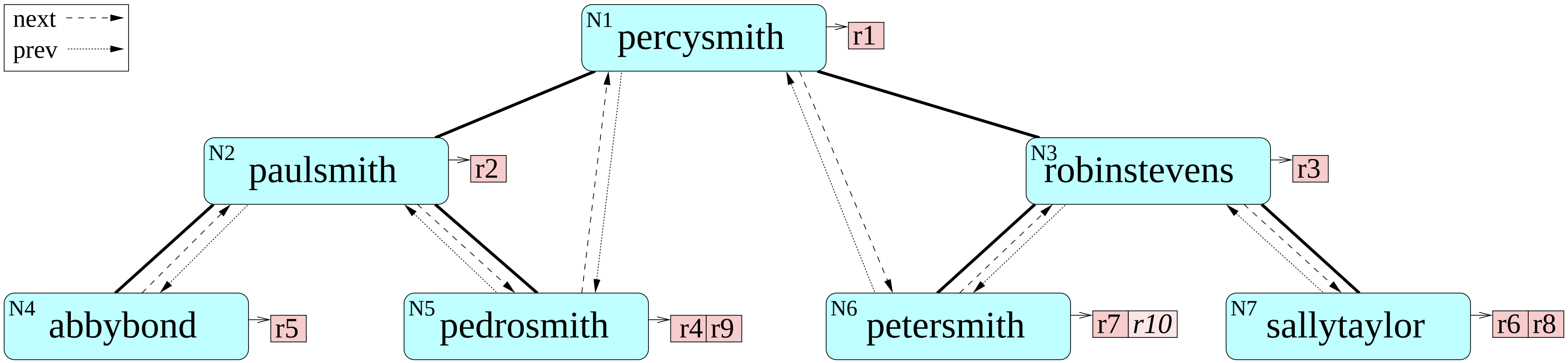
- As an example, a database sorted based on the concatenation of first and last names (*sorting key*):

abbybond	r5
paulsmith	r2
pedrosmith	r4
pedrosmith	r9
percysmith	r1
petersmith	r7
petersmith	r10
robinstevens	r3
sallytaylor	r6
sallytaylor	r8



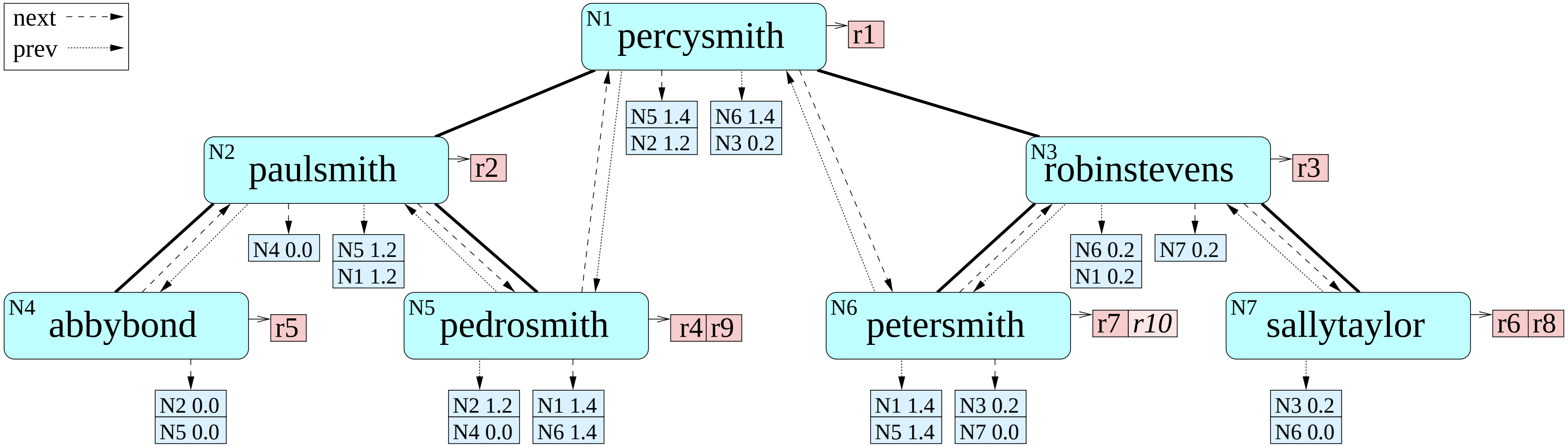
- To make this a dynamic approach, we developed tree-based index data structures

Dynamic sorted neighbourhood indexing (1)



- Based on a braided AVL tree, which is balanced and sorted alphabetically (using the sorting key)
- Each node contains a list of record identifiers that have the same sorting key
- A query record is searched in the tree, and its window are the records in neighbouring nodes
  - Window size can be either static or dynamic
  - If a sorting key is not in the tree it is inserted

Dynamic sorted neighbourhood indexing (2)



next

prev

N1 percysmith r1

N5 1.4
N2 1.2

N6 1.4
N3 0.2

N2 paulsmith r2

N4 0.0

N5 1.2
N1 1.2

N3 robinstevens r3

N6 0.2
N1 0.2

N7 0.2

N4 abbybond r5

N2 0.0
N5 0.0

N5 pedrosmith r4 r9

N2 1.2
N4 0.0

N1 1.4
N6 1.4

N6 petersmith r7 *r10*

N1 1.4
N5 1.4

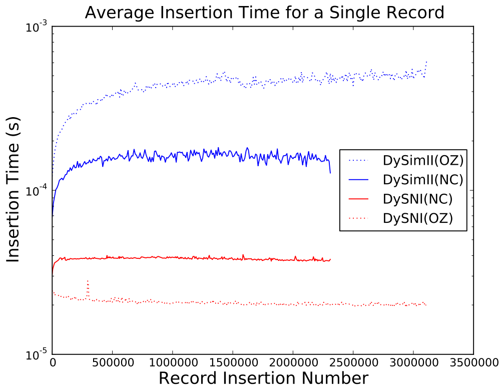
N3 0.2
N7 0.0

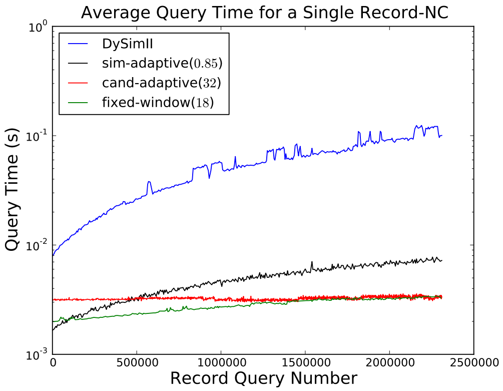
N7 sallytaylor r6 r8

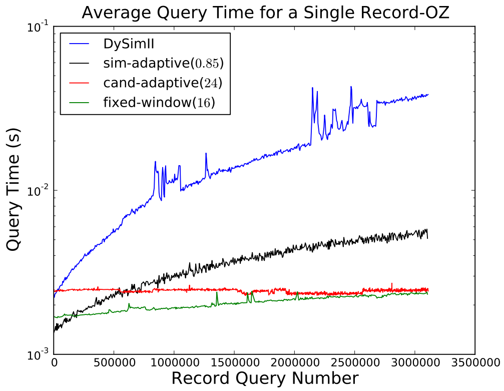
N3 0.2
N6 0.0

- To reduce matching time further, we pre-calculate similarities between keys
  - Attribute-wise approximate string similarities
  - For up-to a fixed window size, or until similarity drops below a threshold (for adaptive window sizes)
- At query time, we only need to calculate similarities of attributes not used in sorting key

Experimental results







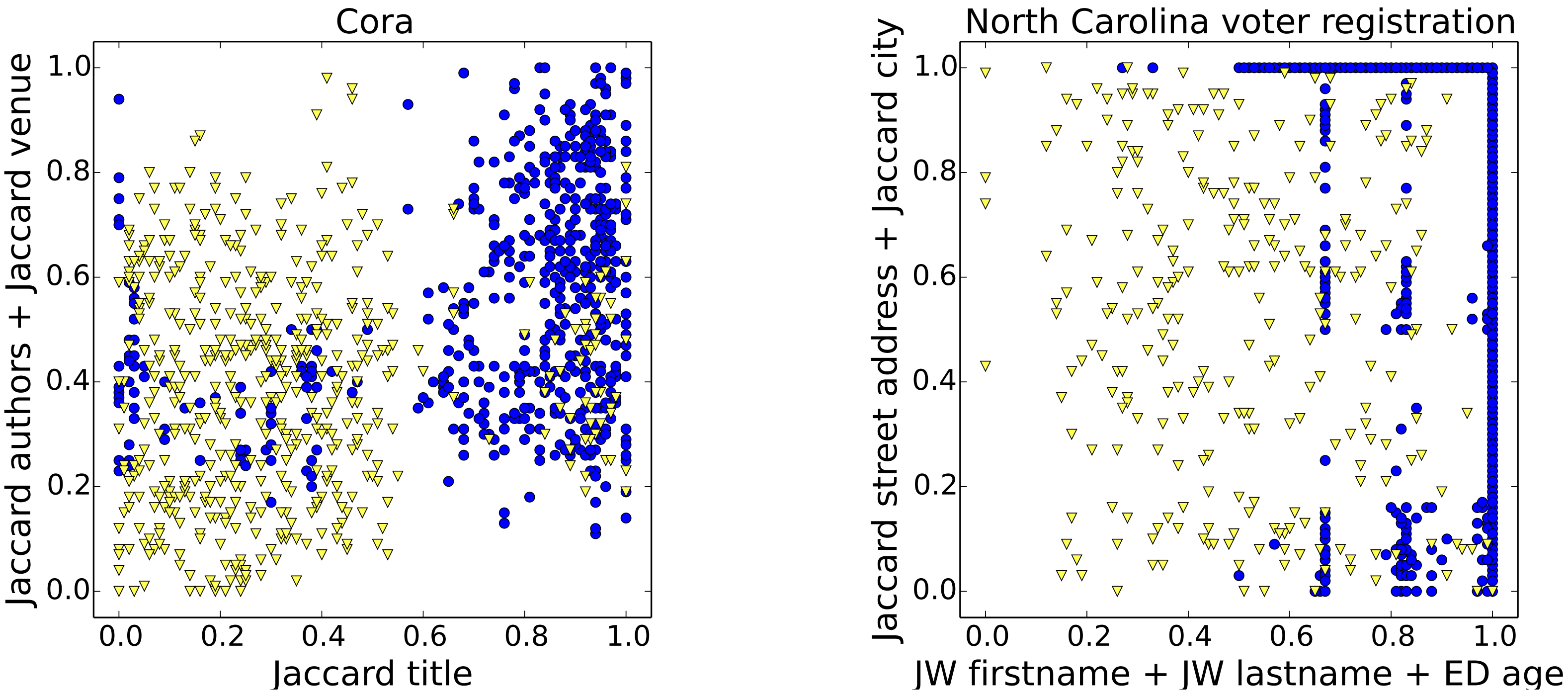
- Using North Carolina voter registration data (NC) and an Australian telephone directory (OZ)
- The average insertion time is almost constant for an increasing index size (both fixed and adaptive window)
- The average query time has a slight increase but flattens as the index size increases (fixed and adaptive)

Data linkage classification

- Traditional data linkage techniques classify each compared record pair individually using similarity thresholds (set manually or based on error estimates)
- Supervised techniques generally result in much better matching quality
- However, training data in the form of true matches and non-matches are rarely available in practice
  - They have to be manually generated, which is often difficult both in terms of cost and quality
- Two main challenges for generating training data
  1. How to ensure *good* examples are selected
  2. How to *minimise* the user's burden of labelling examples

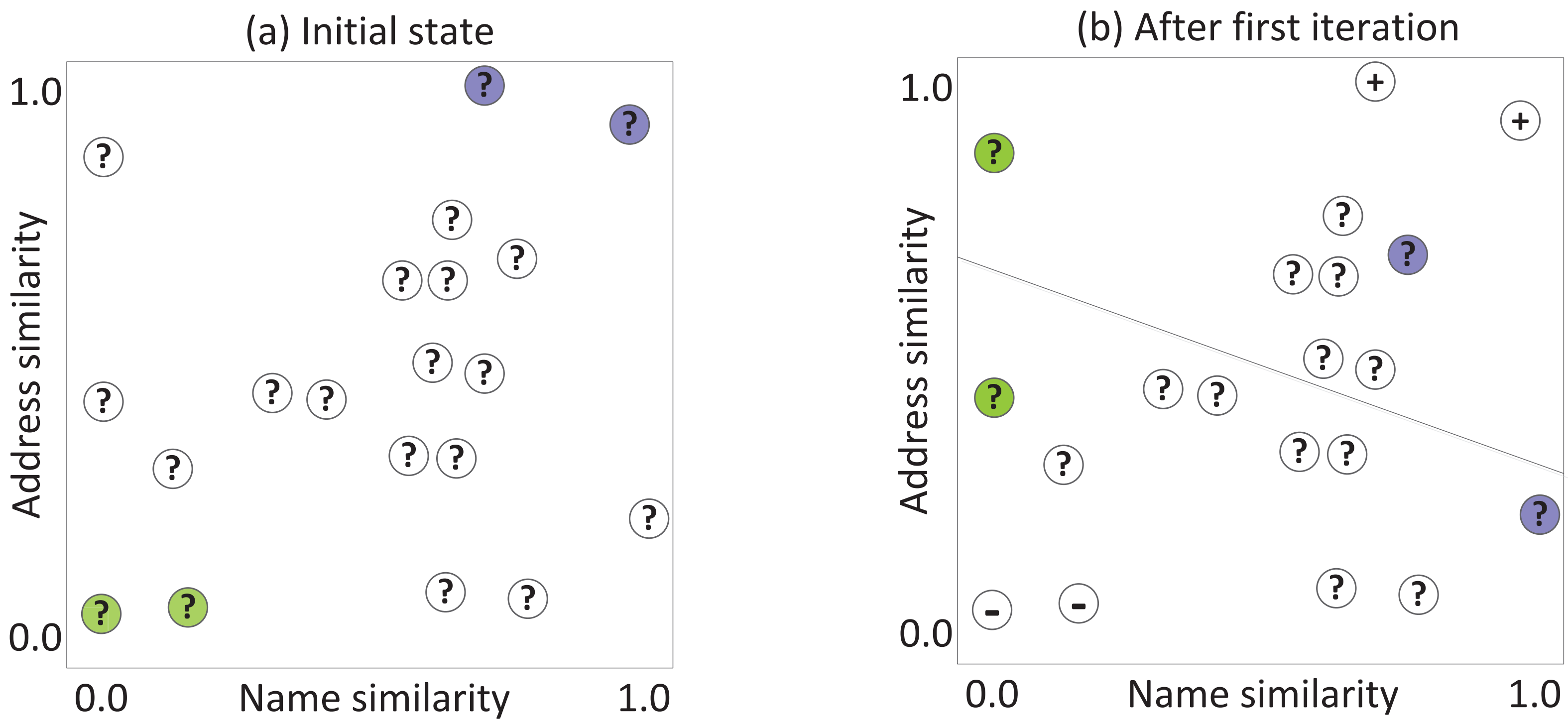
THE AUSTRALIAN NATIONAL UNIVERSITY

Active learning for data linkage: Monotonicity of similarities



- Assumption of existing approaches: the higher the overall similarity between two records is, the more likely they are a true match
- In practice, monotonicity does generally not hold!

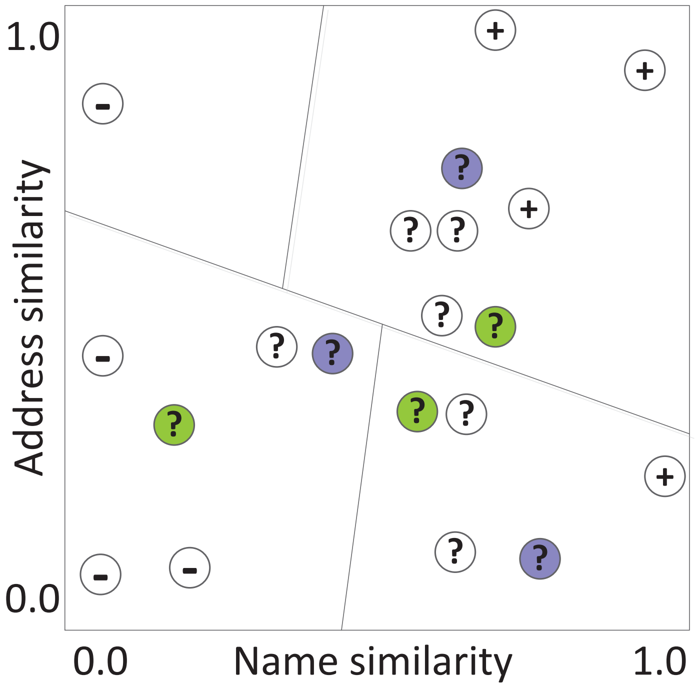
Adaptive and interactive training data selection (1) *(Christen et al., ICDM, 2015)*



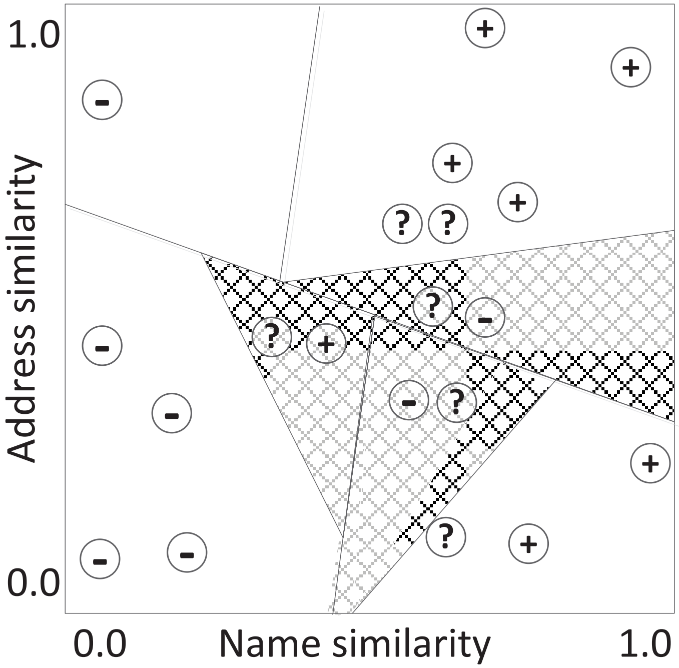
- We exploit the cluster structure of similarity vectors calculated from compared record pairs
- Number of examples selected for manual classification is calculated adaptively based on a sampling error margin

Adaptive and interactive training data selection (2)

(c) After second iteration

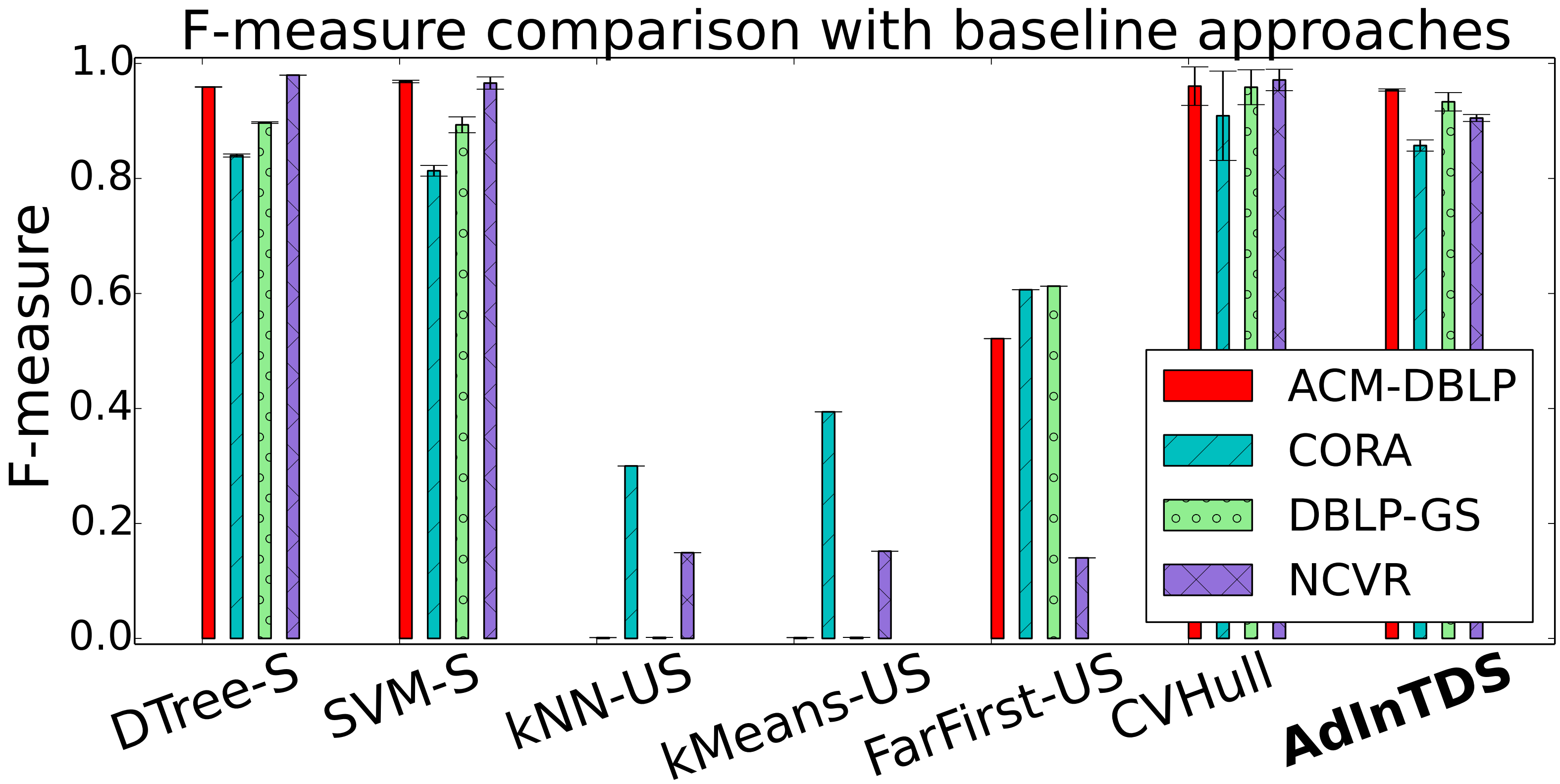


(d) After third iteration



- We recursively split the set of similarity vectors to find pure enough clusters for training
- We select clusters into the training set if they have a minimum purity, otherwise they are inserted into a queue for further recursive splitting

Experimental results



- Compared with fully supervised, unsupervised, and active learning (CVHull [Bellare et al., 2012]) techniques
- Our approach (AdInTDS) achieves a similar F-measure as CVHull, while requiring a much smaller budget

Privacy aspects in record linkage

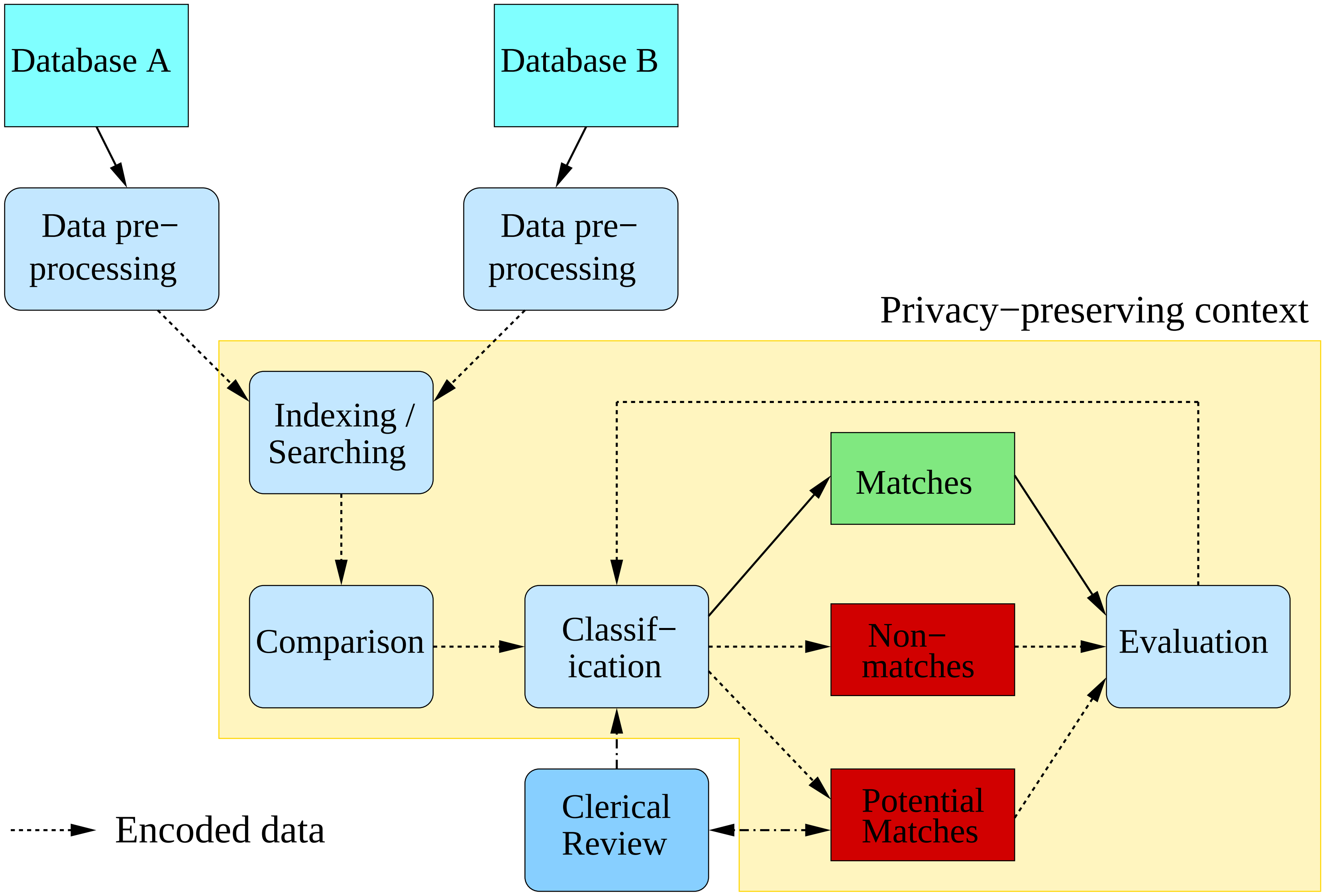
Example scenario: Crime investigation

- A national crime investigation unit is tasked with fighting against crimes that are of national significance (organised crime or money laundering)
- This unit will likely manage various national databases which draw from different sources (law enforcement and tax agencies, Internet service providers, and financial institutions)
- These data are highly sensitive; and storage, analysis and sharing must be tightly regulated (collecting such data in one place makes them vulnerable to outsider attacks and internal adversaries)
- Ideally, only linked records (such as those of suspicious individuals) are available to the unit

Privacy-preserving record linkage

- Objective: *To link data across organisations such that besides the linked records (the ones classified to refer to the same entities) no information about the sensitive source data can be learned by any party involved in the linking, or any external party.*
- Main challenges
  - Allow for approximate linking of values
  - Being able to asses linkage quality and completeness
  - Have techniques that are not vulnerable to any kind of attack (frequency, dictionary, crypt-analysis, etc.)
  - Have techniques that are scalable to linking large databases across multiple parties

The PPRL process



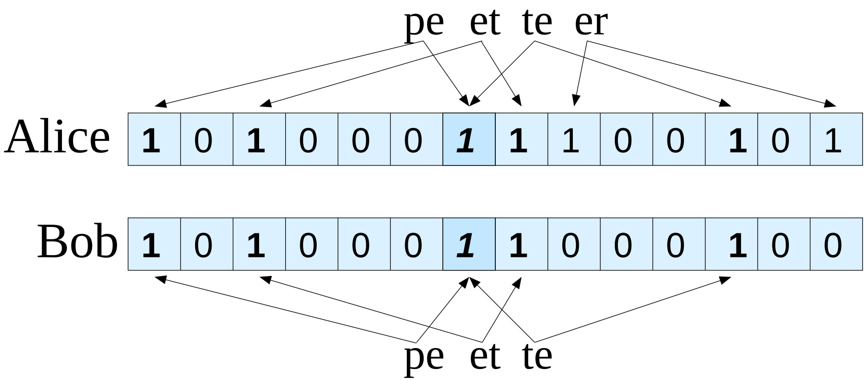
Hash-encoding for PPRL

- A basic building block of many PPRL protocols
- Idea: Use a one-way hash function (like SHA) to encode values, then compare hash-codes
  - Having only access to hash-codes will make it nearly impossible to learn their original input values
  - But dictionary and frequency attacks are possible
- Single character difference between two input values results in completely different hash codes
  - For example:

    ‘peter’ → ‘101010...100101’ or ‘4R#x+Y4i9!e@t4o]’
    ‘pete’ → ‘011101...011010’ or ‘Z5%o-(7Tq1@?7iE/’
  - Only exact matching is possible

Bloom filter encoding

(Schnell et al., BMC Med Inform Decis Mak, 2009)



pe et te er

Alice	**1**	0	**1**	0	0	0	***1***	**1**	1	0	0	**1**	0	1
Bob	**1**	0	**1**	0	0	0	***1***	**1**	0	0	0	**1**	0	0

pe et te

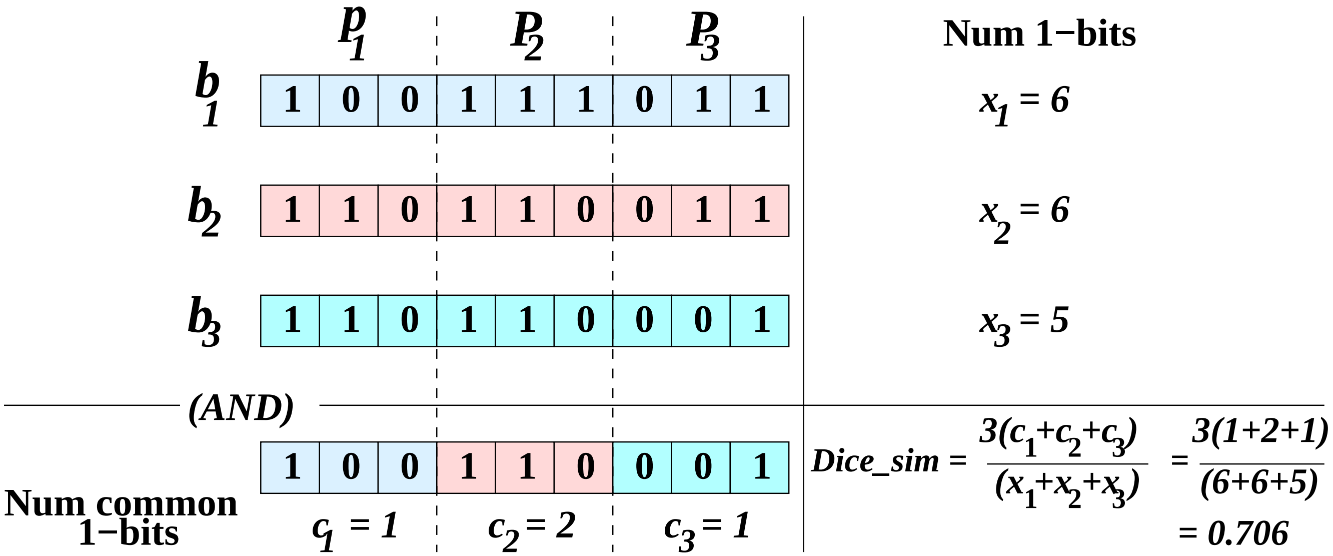
'peter': x_1=*7*, 'pete': x_1=*5*, *c*=*5*, therefore sim_{Dice} = 2×5/(7+5)= 10/12 = 0.83

- Bloom filters are bit vectors initially set to *0*-bits
- Use *k* hash functions to hash-map a set of elements by setting corresponding *k* bit positions to *1*
- A set of *q*-grams (from strings) are hash-mapped to allow approximate matching
- Dice similarity of two Bloom filters b_1 and b_2 is: $sim_{Dice}(b_1, b_2) = \frac{2 \times c}{(x_1 + x_2)}$, with: $c = |b_1 \cap b_2|$, $x_i = |b_i|$

Multi-party Bloom filter based PPRL

(Vatsalan and Christen, CIKM, 2014)

- Distribute similarity calculation across all parties:
  - Bloom filters are split into segments such that each party processes a segment to calculate the number of common *1*-bits in its segment
  - Secure summation is applied to sum the number of common *1*-bits (c_i) and total *1*-bits (x_i) in their Bloom filter to calculate the similarity



	P_1			P_2			P_3			Num 1−bits
b_1	1	0	0	1	1	1	0	1	1	$x_1 = 6$
b_2	1	1	0	1	1	0	0	1	1	$x_2 = 6$
b_3	1	1	0	1	1	0	0	0	1	$x_3 = 5$
(AND)	1	0	0	1	1	0	0	0	1	
Num common 1−bits	$c_1 = 1$			$c_2 = 2$			$c_3 = 1$			

$$Dice\_sim = \frac{3(c_1+c_2+c_3)}{(x_1+x_2+x_3)} = \frac{3(1+2+1)}{(6+6+5)} = 0.706$$

Research directions

To make sure everybody is awake.. :-)

Research directions (1)

- Linkage techniques for massive-scale Big data collections (parallel, distributed, cloud based)
- Linking data from many sources (computational challenges, as well as the issue of collusion between parties in PPRL)
- Linking dynamic data and linking data in real-time (dynamic indexing techniques and classification models)
- For historical data, the major challenge is data quality (automatic / semi-automatic data cleaning and standardisation techniques)
- How to employ collective / relational classification techniques for data with personal information?

Research directions (2)

- No training data in most applications
  - Active learning approaches
  - Visualisation for improved manual clerical review
- Frameworks for data linkage that allow comparative experimental studies
- Publicly available test data collections
  - Challenging (impossible?) to have true match status
  - Challenging as most data are either proprietary or sensitive
- Pragmatic challenge: Collaborations across multiple research disciplines

Research directions – PPRL (1)

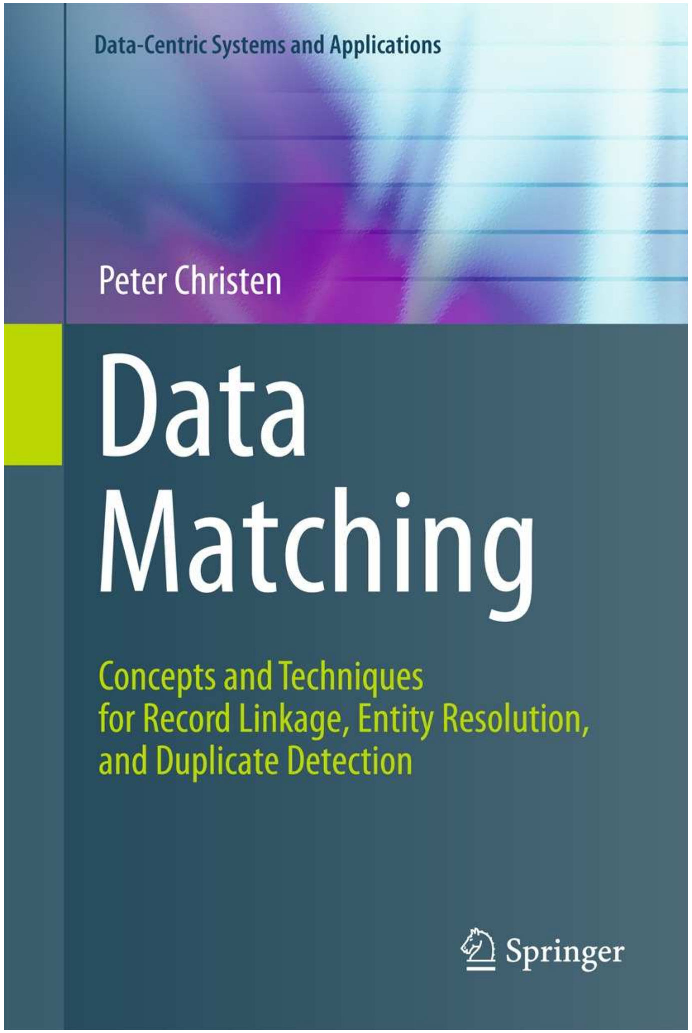
- Improved classification for PPRL
  - Mostly simple threshold-based classification is used
  - No investigation into advanced methods, such as collective / relational techniques
  - Supervised classification is difficult (no training data in many situations)
- Assessing linkage quality and completeness
  - How to assess linkage quality (precision and recall)?

    – How many classified matches are true matches?
    – How many true matches have we found?
  - Access to actual record values is not possible (as this would reveal sensitive information)

Research directions – PPRL (2)

- A framework for PPRL is needed
  - To facilitate comparative experimental evaluation of PPRL techniques
  - Needs to allow researchers to plug-in their techniques
  - Benchmark data sets are required (biggest challenge, as such data are sensitive!)
- PPRL on multiple databases
  - Most work so far is limited to linking two databases (often records from several organisations need to be linked)
  - Pair-wise linking does not scale up
  - Preventing collusion between (sub-groups of) parties becomes more difficult

Advertisement: Book 'Data Matching' (2012)


Data-Centric Systems and Applications
Peter Christen
Data Matching
Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection
Springer

The book is very well organized and exceptionally well written. Because of the depth, amount, and quality of the material that is covered, I would expect this book to be one of the standard references in future years.

William E. Winkler, U.S. Bureau of the Census.