Australian National University

COMP3430 / COMP8430
Data wrangling

Lecture 22: Privacy aspects in data wrangling and privacy-preserving record linkage (Lecturer: Peter Christen)

Based on slides by Dinusha Vatsalan (Data61 / CSIRO)

Australian National University

Lecture outline

- Privacy, confidentiality and security
- Privacy in data wrangling
- Privacy-preserving record linkage (PPRL)
  - Taxonomy of PPRL
  - PPRL techniques
- Bloom filter-based techniques
- Attacks on PPRL

Privacy, confidentiality, and security

- Three important and related concepts of data protection
- Privacy
  - Right of individual entities (e.g. customers or patients) to make decisions about how their personal data are shared and used
- Confidentiality
  - Obligation or responsibility of professionals and organisations who have access to data to hold in confidence
- Security
  - Means or tools used to protect the privacy of entities' data and to support professionals / organisations in holding data in confidence

Privacy by design

- Personal data are valuable for various applications, but are at risk of being used, stored, shared, exchanged, or revealed due to growing privacy concerns
  - Important to have proper systems in place that provide data protection
  - But allow applications and research studies utilise available information in data
- Standards and regulations are required
  - Safe environments to handle them in
  - Proper handling procedures and output
  - Safe storage
  - Privacy laws, such as recent European Union GDPR (General Data Protection Regulation)

Privacy in data wrangling

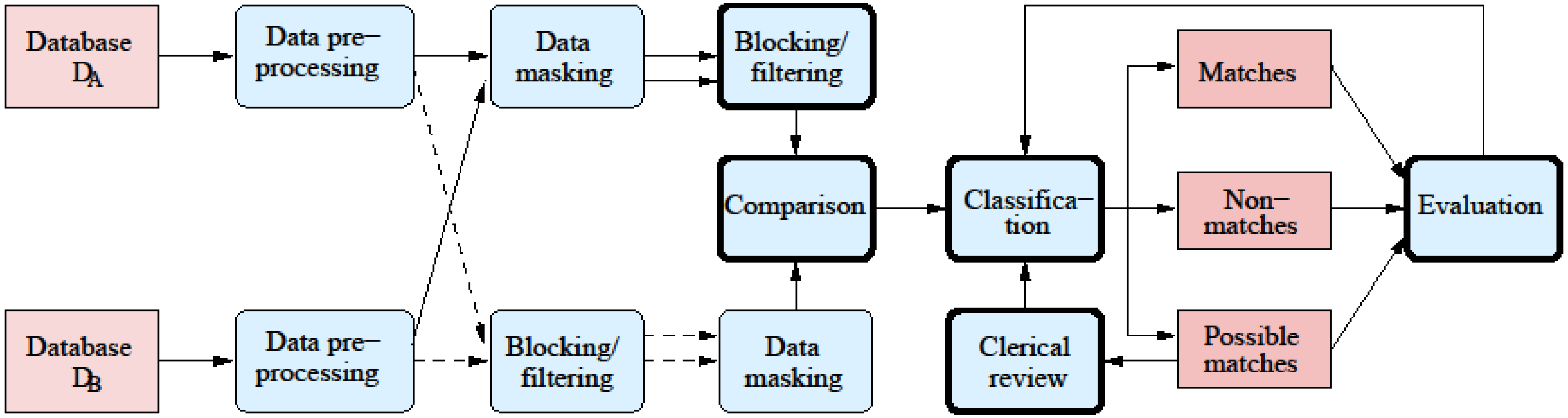
- Preserve privacy and confidentiality of entities represented by data during the data wrangling pipeline
- Privacy and confidentiality concerns arise when data are shared or exchanged between different organisations
  - Mainly the task of data integration / record linkage in the pipeline that requires data to be integrated from multiple sources held by different organisations
  - Require disclosure limitation to protect the privacy and confidentiality of sensitive data (such as personal names, addresses, etc.)

Disclosure limitations

- Filter or mask (encode or encrypt) raw data to block what is revealed or disclosed
- Disclosure-limited masking:
  - Using masking (encoding) functions to transform data such that there exists a specific functional relationship between the masked and the original data
  - Budget-constrained problem – the goal of masking functions is to achieve the maximum utility under a fixed privacy budget
  - Examples include noise addition, generalisation, and probabilistic filters

Privacy-preserving record linkage (PPRL)

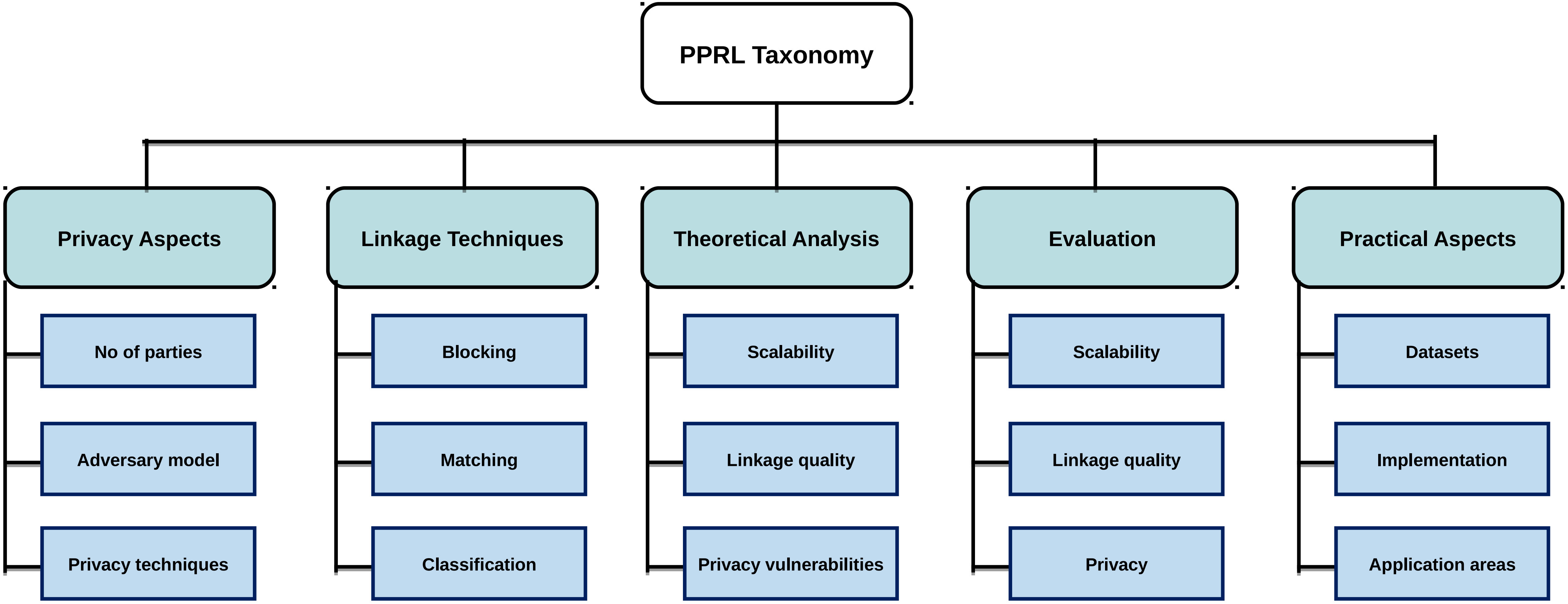
- Objective of PPRL is to perform linkage across organisations using masked (encoded) records
  - Besides certain attributes of the matched records no information about the sensitive original data can be learned by any party involved in the linkage, or any external party



PPRL: Example applications

- Health outbreak systems
  - Early detection of infectious diseases before they spread widely
  - Requires data to be integrated across human health data, travel data, consumed drug data, and even animal health data
- National security applications
  - Integrate data from law enforcement agencies, Internet service providers, and financial institutions to identify crime and fraud, or terrorism suspects
- Business applications
  - Compile mailing lists or integrate customer data from different sources for marketing activities and/or recommendation systems
- Neither of the parties is willing or allowed by law to exchange or provide their data between/to other parties

PPRL taxonomy



Vatsalan et al., A Taxonomy of PPRL techniques, JIS, 38(6),pp. 946-969, 2013

PPRL protocols

Three-party protocols **Two-party protocols**

- **Three-party protocols:** Use a linkage unit (LU) to conduct/facilitate linkage
- **Two-party protocols:** Only two database owners participate in the linkage
- **Multi-party protocols:** Linking records from multiple databases (with or without a LU) with the additional challenges of scalability and privacy risk (collusion between parties)

PPRL adversary models

- Honest-but-curious (HBC) or semi-honest model
  - Parties follow the protocol while being curious to learn about another party’s data
  - Most existing PPRL protocols assume HBC model
- Malicious model
  - Parties may behave arbitrarily, not following the protocol
  - Evaluating privacy under malicious model is difficult
- Advanced models
  - Accountable computing and covert model allow to identify if a party has not followed the protocol with a certain probability
  - Lower complexity than malicious and more secure than HBC

Attack models

- Dictionary attack
  - Mask a list of known values using existing masking functions until a matching masked value is identified (SHA or MD5)
  - Keyed masking approach, like HMAC, can overcome this attack
- Frequency attack
  - Frequency distribution of masked values is matched with the distribution of known values
- Cryptanalysis attack
  - A special type of frequency attack applicable to Bloom filters
- Collusion
  - A set of parties (in three-party and multi-party protocols) collude with the aim to learn about another party’s data

PPRL techniques

- Several techniques developed
  - Generalisation such as k-anonymity, noise addition and differential privacy; secure multi-party computation (SMC) such as homomorphic encryptions and secure summation; and probabilistic filters such as Bloom filters and variations
- First generation (mid 1990s): Exact matching only
- Second generation (early 2000s): Approximate matching but not scalable
- Third generation (mid 2000s): Take scalability into account

Australian National University

Secure hash encoding

- First generation PPRL techniques
- Use a one-way hash-encoding function (like SHA) to encode values and then compare the hash-encoded values to identify matching records
  - Only exact matching is possible
  - Single character difference in two values results in a pair of completely different hash-encoded values
    (for example, ‘peter → ‘10100…00101’, and ‘pete’ → ‘011101…11010’)
- Having only access to hash-codes will make it nearly impossible to learn the original values
  - Frequency attacks are still possible

Noise and differential privacy

- Add noise to overcome frequency attack at the cost of loss in linkage quality
- Differential privacy is an alternative to random noise addition
  - The probability of holding any property on the perturbed database is approximately the same whether or not an individual value is present in the database
  - Magnitude of noise depends on privacy budget and sensitivity of data

Generalisation techniques

- Generalises the records to overcome frequency attacks
- Example: k-anonymity
  - Ensure every combination of attribute values is shared by at least *k* records

Alice

Age	Postcode
27	2602
60	3042
50	3021
35	2616

Bob

Age	Postcode
[20,40]	26**
[46,80]	30**
[46,80]	30**
[20,40]	26**

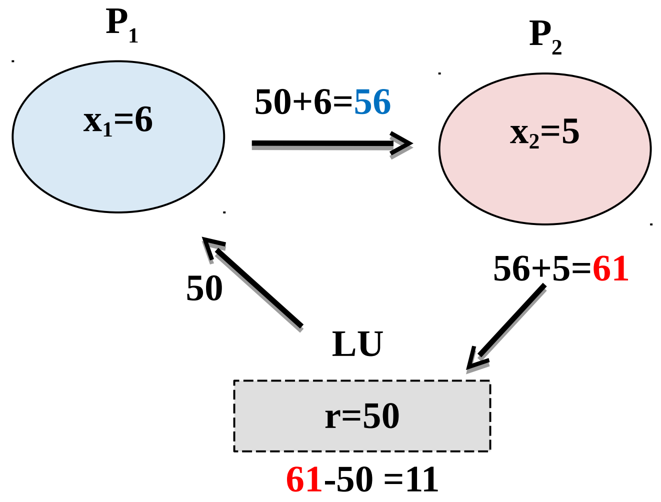
- Other techniques – value generalisation hierarchies and binning (as covered earlier in the course)

Encryption and SMC

- Commutative and homomorphic encryptions are used
- Computationally expensive
- Secure scalar product, secure set intersection, secure set union, and secure summation are the most commonly used SMC techniques
- Example:

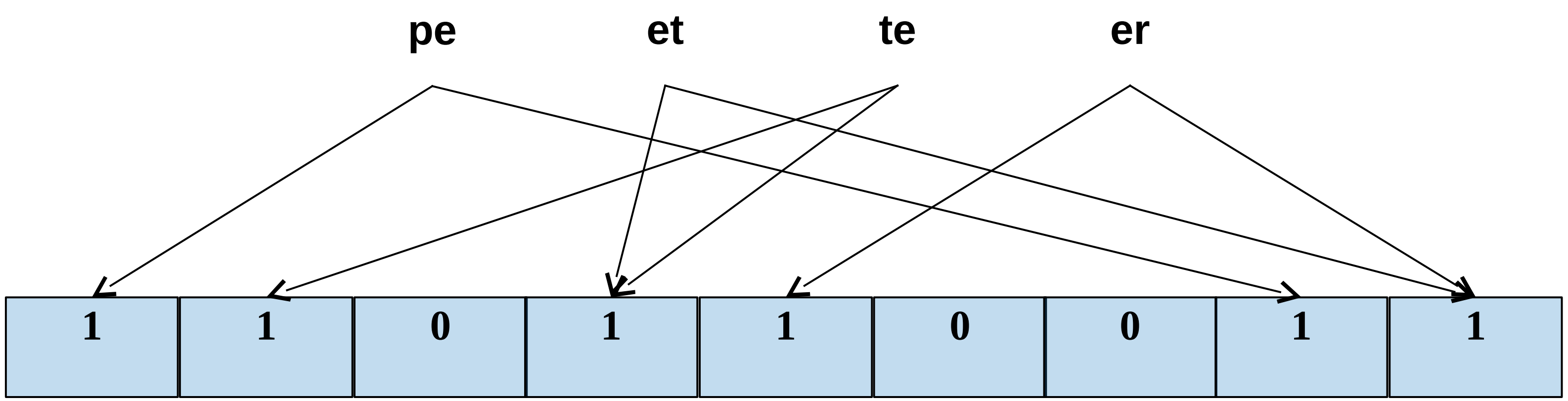
  Secure summation of values

  $x_1 = 6$ and $x_2 = 5$ using a LU



Bloom filters (1)

- Probabilistic data structure
  - Bit vector of *l* bits (initially all set to *0*)
  - *k* independent hash functions are used to hash-map each element in a set *S* into a Bloom filter by setting the corresponding bits to *1*



Encoding q-grams (with *q=2*) of string 'peter' into Bloom filters of length *l=9* bits using *k=2* hash functions

Bloom filters (2)

- Dice coefficient similarity of p BFs is calculated as:

$$Dice\_sim(b_1,\ldots,b_p) = \frac{p \times z}{\sum_i x_i},$$

where z is the number of common 1-bits in p BFs and x_i is the number of 1-bits in BF b_i

peter	→	b_1	1	1	0	1	1	0	0	1	1	$x_1 = 6$
pete	→	b_2	1	1	0	1	1	0	0	0	1	$x_2 = 5$
		b_1^b_2	1	1	0	1	1	0	0	0	1	$sim = 2 \times 5 / (6+5) = 0.9$

$z = 5$

Schnell et al., Privacy-Preserving Record Linkage using Bloom filters, BMC MIDM, 9(1), 2009

Bloom filters (3)

- Bloom filter-based matching
  - Similarity of Bloom filters can be calculated using a token-based similarity function, such as Jaccard, Dice, and Hamming
  - Dice is mostly used, as it is insensitive to many matching zeros
  - Similarity of Bloom filters ≥ similarity of input values (due to false positive rate of Bloom filters)
- False positive rate determines privacy and accuracy
  - The larger the false positive rate, the higher the privacy but lower the accuracy

Bloom filters (4)

- Attacks on Bloom filters
  - Susceptible to cryptanalysis attacks – mapping bit patterns to q-grams and values based on frequency and co-occurrence information
  - Several attack methods on basic Bloom filters have been developed
- We have recently developed a new efficient attack method that allows re-identification of frequent attribute values
  - Patterns in Bloom filter bit positions that have co-occurring patterns are identified using pattern mining
  - The most successful attack method so far
- Advanced Bloom filter hardening techniques are required

Christen et al., Pattern-Mining based Cryptanalysis of Bloom Filters for Privacy-Preserving Record Linkage, PAKDD, 2018

Bloom filters (5)

Plain–text database **V**

maude
mary
max
joan
john

Q–gram counts:

3: ma
2: jo
1: an, ar, au, ax, de, hn, oa, oh, ry, ud

(only shown for illustration, but not known to the attacker)

Encoded Bloom filter database **B**

1...																				
0	0	0	0	**1**	0	0	0	0	0	0	0	**1**	1	0	1	0	0	0	0	$\mathbf{b}_1$
1	0	1	0	0	0	0	1	0	**1**	0	0	0	0	1	0	0	0	0	1	$\mathbf{b}_2$
0	0	0	0	**1**	0	1	0	1	0	1	0	**1**	0	0	0	1	0	0	0	$\mathbf{b}_3$
0	0	0	0	**1**	1	1	0	0	0	0	1	**1**	0	0	1	0	1	1	0	$\mathbf{b}_4$
1	1	0	1	0	0	0	0	0	**1**	0	0	0	0	0	0	0	1	0	0	$\mathbf{b}_5$
jo	*oa*	*oh*	*oa*	*ma*	*au*	*ar*	*oh*	*ry*	*jo*	*ar*	*au*	*ma*	*ax*	*hn*	*ud*	*ry*	*ud*	*de*	*hn*	
	an					*de*									*ax*		*an*			
p_1				p_5					p_{10}				p_{13}							