P-Signature-Based Blocking to Improve the Scalability of Privacy-Preserving Record Linkage

Dinusha Vatsalan[1(✉)], Joyce Yu[1], Brian Thorne[2], and Wilko Henecka[1]

[1] CSIRO's DATA61, Eveleigh, NSW 2015, Australia
{dinusha.vatsalan,joyce.yu,wilko.henecka}@data61.csiro.au
[2] Hardbyte, Christchurch, New Zealand
brian@hardbyte.nz

Abstract. Integrating data from multiple sources with the aim to identify records that correspond to the same entity is required in many real-world applications including healthcare, national security, businesses, and government services. However, privacy and confidentiality concerns impede the sharing of personal identifying values to conduct linkage across different organizations. Privacy-preserving record linkage (PPRL) techniques have been developed to tackle this problem by performing clustering based on the similarity between encoded record values, such that each cluster contains (similar) records corresponding to one single entity. When employing PPRL on databases from multiple parties, one major challenge is the prohibitively large number of similarity comparisons required for clustering, especially when the number and size of databases are large. While there have been several private blocking methods proposed to reduce the number of comparisons, they fall short in providing an efficient and effective solution for linking multiple large databases. Further, all private blocking methods are largely dependent on data. In this paper, we propose a novel private blocking method addressing the shortcomings of existing methods for efficiently linking multiple databases by exploiting the data characteristics in the form of probabilistic signatures, and we introduce a local blocking evaluation framework for locally validating blocking methods without knowing the ground-truth data. Experimental results on large datasets show the efficacy of our method in comparison to several state-of-the-art methods.

Keywords: Entity resolution · Privacy · Scalability · Probabilistic signatures · Clustering

1 Introduction

Linking data from multiple sources with the aim to identify matching pairs (from two sources) or matching sets (from more than two sources) of records that correspond to the same real-world entity is a crucial data pre-processing

J. Garcia-Alfaro et al. (Eds.): DPM 2020/CBT 2020, LNCS 12484, pp. 35–51, 2020.
https://doi.org/10.1007/978-3-030-66172-4_3

task for quality data mining and analytics [3]. Various real-world applications require record linkage to improve data quality and enable accurate decision making. Example applications come from healthcare, businesses, the social sciences, government services, and national security.

Record linkage involves several challenges making the process not trivial. Due to the absence of unique entity identifiers across different databases, it is required to use the commonly available quasi-identifiers (QIDs), such as names and addresses, for linking records from those databases. QIDs generally contain personal and often sensitive information about the entities to be linked, which precludes the sharing of such values among different organizations for linkage due to privacy concerns. Known as privacy-preserving record linkage (PPRL) [16,19], this research has attracted increasing interest over the last two decades and has been employed in several real projects [2,6,13].

A prominent challenge of PPRL of multiple large databases is the quadratic complexity of similarity comparisons required between QIDs of records with the number of databases to be linked and their sizes. Blocking techniques are being used in the linkage to reduce the number of comparisons by grouping records according to a certain criteria and limiting the comparison only to the records in the same group [3]. However, existing private blocking methods do not perform well on low latency and high-scale data due to either (1) their dependency on data-sensitive parameters that need to be tuned for different datasets [7,8,10, 11,14,15,20,23], (2) they require external data of similar distribution [8,10,14, 20,23], (3) they require similarity computations for blocking itself which makes them not scalable to linking multiple large databases [1,8,10,14,15], (4) most of them are not developed for linking multiple databases (except [8,11,15]), or (5) they do not support efficient subset matching from any number of databases [8,11,15]. In this paper, we address these shortcomings by developing a novel private blocking method based on probabilistic signatures and proposing a local blocking evaluation framework for tuning data-dependent parameters.

The values in QIDs are often prone to data errors and variations, which impacts the quality of blocking as well as makes the linkage task challenging [3]. Probabilistic signatures (p-signatures) leverage the redundancy in data to reduce the impact of data quality issues on blocking. Subset of information contained in a record that can be used to identify the entity corresponding to that record is called as a signature. For example, 'John Smith' is a frequently occurred name, however, 'John Smith, Redfern' is more unique and more likely to correspond to the real-world entity as similar as 'John Smith, Redfern, NSW 2015'. Probabilistic identification of such signatures for linking records (in the non-PPRL context) has been studied in an existing work [24].

In this paper, we extend the idea of using p-signatures for efficient data-driven blocking for PPRL of multiple databases. Our approach does not depend on external data, and it does not require any similarity computations between records for blocking, as required by most of the existing methods [8,10,14,15]. In addition, our method enables subset matching for multi-party PPRL, which aims to identify matching records from any subset of multiple databases held

by different parties, for example, linking patients who have visited at least three out of ten hospitals. Existing blocking methods do not facilitate nor efficiently facilitate blocking for subset matching [8,11,15]. Moreover, existing methods are sensitive to errors and variations in the blocking keys. For example, if a record contains missing values in part of the blocking key, it will be misplaced in a wrong block, while with signatures the part with the missing value will not become a signature and thus will not be placed in a wrong bucket, improving the quality of blocking.

However, as with all existing methods, the blocking quality in terms of effectiveness of reducing the comparison space as well as coverage of true matches depends significantly on the signatures used. We use multi-signature strategy to improve blocking quality. Further, we propose a framework to locally evaluate the blocking quality guarantees individually by the database owners in order to choose an appropriate signature strategy (or parameter settings) depending on the datasets to be blocked. Our proposed local blocking evaluation metrics (which we refer to as **PQR**-guarantees metrics for **P**rivacy, **Q**uality, and **R**eduction guarantees of blocking) can be used to locally evaluate any blocking method for PPRL.

We provide a comparative evaluation of our proposed method with several state-of-the-art blocking methods for PPRL in terms of coverage of true matches, reduction in record comparisons, and privacy guarantees against frequency inference attacks [21] using large datasets. We also evaluate the effectiveness of our blocking method for multi-party PPRL using a black box clustering method [22] and compare the results with no blocking and using an existing multi-party blocking method [11]. Experimental results show that our method outperforms the state-of-the-art methods in terms of all three aspects.

Outline: We describe preliminaries in Sect. 2 and in Sect. 3 we present our protocol. In Sect. 3.1, we introduce a novel method to locally evaluate Privacy, Quality, and Reduction guarantees of any blocking methods. We analyze our protocol in terms of complexity, privacy in Sects. 3.2 and 3.3, respectively, and validate these analyses through an empirical evaluation in Sect. 4. Related work is reviewed in Sect. 5. Finally, we conclude the paper in Sect. 6.

2 Preliminaries

An outline of the general PPRL pipeline is shown in Fig. 1. Assume $P_1, \ldots, P_p$ are the p owners (parties) of the *deduplicated* databases $\mathbf{D}_1, \ldots, \mathbf{D}_p$, respectively. PPRL allows the party P_i to determine which of its records $r_{i,x} \in \mathbf{D}_i$ match with records in other database(s) $r_{j,y} \in \mathbf{D}_j$ with $1 \leq i, j \leq p$ and $j \neq i$ based on the similarity/distance between (masked or encoded) quasi-identifiers (QIDs) of these records. The output of this process is a set $\mathbf{M}$ of match clusters, where a match cluster $m \in \mathbf{M}$ contains a set of matching records of a maximum of one record from each database and $1 < |m| \leq p$. Each $m \in \mathbf{M}$ is identified as a set of

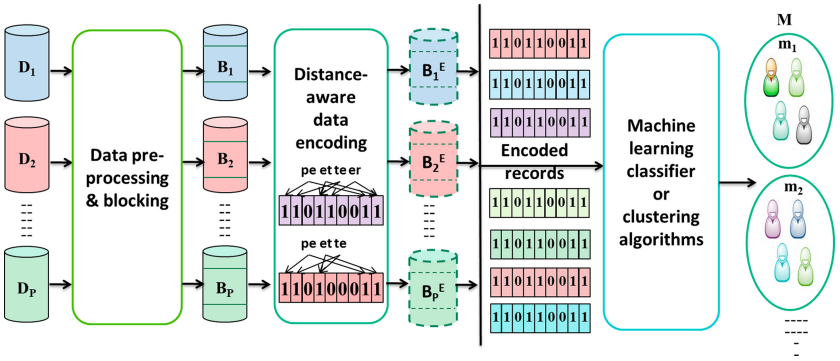

Fig. 1. General pipeline of the PPRL process

matching records representing the same entity. A linkage unit (LU) is generally employed to conduct PPRL using the encoded QID values of records sent by the database owners.

Assuming each of the p databases contains n records ($n \times p$ records in total), the number of similarity comparisons required is quadratic in both n and p (i.e. $n^2 \cdot p^2$). The quadratic comparison space is computationally expensive for clustering techniques with large-scale data. However, majority of the comparisons are between non-matches as record linkage is generally a class-imbalance problem [3]. Blocking aims at reducing the comparison space for linkage by eliminating the comparisons of record pairs that are highly unlikely to be matches. There are numerous blocking strategies [4] developed in the literature for record linkage and PPRL.

Generally, the records are grouped into blocks for each database D_i (denoted as $\mathbf{B}_i$) and the blocks of encoded records of each database (denoted as $\mathbf{B}_i^E$) are sent to a linkage unit (LU) to conduct the linkage of these encoded records from multiple databases using a clustering technique [22]. At the LU, the records are processed block by block (i.e. clustering is applied on each block $B \in \mathbf{B}$, where $\mathbf{B}$ contains the union of blocking keys in $\mathbf{B}_i^E$, with $1 \leq i \leq p$).

The existing blocking methods for PPRL require data dependent parameters to be tuned or external data of similar distribution for blocking. Exploiting the data characteristics, we propose a blocking method based on multiple signatures. Redundancy is one of the common data characteristics in real data as only some information in a record is sufficient to uniquely identify and link records. Such informative part in a record becomes a signature. Each unique signature becomes a blocking key in our blocking method.

Definition 1 (Signature). *Given records R with QIDs A, a signature s is a subset of information in a record $r \in R$, i.e. $s \subset \forall_{a \in A} r.a$, that can uniquely identify the corresponding entity with high probability.*

Example 1: A record r_1 with the values of QIDs $r_1.a_1$ = 'smith', $r_1.a_2$ = 'william', $r_1.a_3$ = 'redfern', $r_1.a_4$ = '2015', has the signature s_1 = 'smwr316',

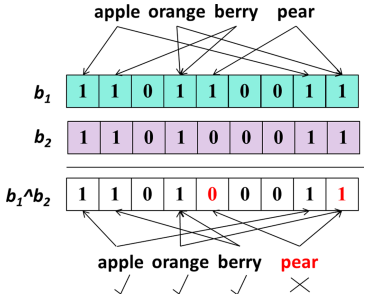

Fig. 2. An example encoding of two sets S_1 = {'apple', 'orange', 'berry', and 'pear'} and S_2 = {'apple', 'orange', 'berry'} into BFs b_1 and b_2, respectively, and membership test on the intersected BF $(b_1 \cap b_2)$. For example, 'pear' is not a member of $b_1 \cap b_2$.

where the signature is generated based on the concatenation of the first two characters of a_1, first character of a_2, none of the characters of a_4, and phonetic encoding of a_3.

Definition 2 (Signature strategy). *A signature strategy is a function $f(\cdot)$ of generating a signature for each record $r \in R$ from $\forall_{a \in A} r.a$.*

Example 2: $f(a_1[0:2] + a_2.phonetic())$ is a signature strategy, which returns the first two characters and phonetic code of QIDs a_1 and a_2, respectively.

We use multiple such signature strategies to improve the blocking quality (recall of true matches) at the cost of more record pair comparisons. For each of the signature strategies records containing the same signature are grouped into one block, and blocks of records of the same signature across multiple databases are compared and linked using clustering techniques [22].

In order to identify the common blocks (signatures) between two or multiple databases held by different parties without learning the non-common signatures of a party by other parties as well as any signatures by the LU, the parties encode their signatures into a Bloom filter (BF).

Definition 3 (BF encoding). *A BF b_i is a bit vector of length l bits where all bits are initially set to 0. k independent hash functions, $h_1, \ldots, h_k$, each with range $1, \ldots l$, are used to map each of the elements s in a set S into the BF by setting the bit positions $h_j(s)$ with $1 \leq j \leq k$ to 1.*

Figure 2 illustrates the encoding of two sets S_1 = {'apple', 'orange', 'berry', and 'pear'} and S_2 = { 'apple', 'orange', 'berry'} into two BFs b_1 and b_2, respectively, of $l = 9$ bits long using $k = 2$ hash functions. Collision of hash-mapping occurs where two different elements are mapped to the same bit position. Collision can result in false positives however providing privacy guarantees through the level of uncertainty about a true mapping at the cost of utility loss.

Definition 4 (Membership test). *Membership of an element s in a set that is encoded into a BF b can be tested by checking if $\forall_{i=1}^{k} b[h_i(s)] == 1$. If at least one of the hash functions returns 0, then the element could not have been a member of the set that is encoded into b.*

We use counting Bloom filter (CBF) [5] to count the number of parties/databases that have common signatures for multi-party PPRL.

Definition 5 (CBF encoding). *A counting Bloom filter (CBF) c is an integer vector of length l bits that contains the counts of values in each position. Multiple BFs can be summarized into a single CBF c, such that $c[\beta] = \sum_{i=1}^{p} b_i[\beta]$, where $\beta, 1 \leq \beta \leq l$. $c[\beta]$ is the count value in the β bit position of the CBF c. Given p BFs b_i with $1 \leq i \leq p$, the CBF c can be generated by applying a vector addition operation between the bit vectors such that $c = \sum_i b_i$.*

Secure summation protocols can be used to securely calculate the sum of p values $v_1, \cdots, v_p$ without learning the individual values v_i, but only the sum $\sum_{i=1}^{p} v_i$. v_i can either be a single numeric value or a vector of numeric values.

3 Methodology

In this section we describe the steps of our Privacy Preserving Probabilistic signature (**P3-SIG**) blocking method, which is outlined in Algorithm 1. It consists of three phases:

1. Signature generation: This phase involves identifying and agreeing on signature strategies and generating candidate signatures (lines 1–5 in Algorithm 1). The probability of a candidate signature to appear in records is bounded by the minimum and maximum size of resulting blocks (k_{min} and k_{max}, respectively) for privacy and comparison reduction guarantees, respectively. Signatures that appear in too many records are often redundant (non-informative) and signatures that appear in very few records can be uniquely re-identified against inference attacks.

The resulting candidate signatures are locally evaluated in order to select and agree on a set of good signature strategies to be used by all parties to generate signatures or blocking keys (lines 6–7 in Algorithm 1). We use multi-signature approach where multiple such good signature strategies are used to improve the coverage of true matches. Good signature strategies are determined considering three aspects: (1) comparison reduction, (2) coverage of true matches, and (3) privacy guarantees of the resulting blocks against frequency attack. We will describe the local blocking evaluation in terms of these three aspects in Sect. 3.1.

2. Common signatures identification: Once a set of good signature strategies are agreed upon by all parties, the parties individually generate the signatures for their records using the agreed signature strategies and hash-map the resulting signatures into a Bloom filter (BF) (lines 16–18). If the linkage task is to identify common blocks across all p parties, then the intersected BF of all parties' BFs is sufficient to calculate the common signatures/blocks. The

Algorithm 1. P3-SIG blocking (described in Sect. 3)

Input:
- $\mathbf{R}_i$: Party P_i's records, $1 \leq i \leq p$
- $\mathbf{S}'$: A set of signature strategies $f(\cdot)$
- $e(\cdot)$: A function to locally evaluate blocking
- s_m : Minimum subset size, with $2 \leq s_m \leq p$
- $h(\cdot)$: Hash functions for BF encoding
- l : Length of BF
- k : Number of hash functions

Output:
- $\mathbf{C}$: Blocks from all parties

Phase 1 (by each party P_i, with $1 \leq i \leq P$):

1: **for** $f \in \mathbf{S}'$ **do**: // Iterate strategies
2: $\mathbf{B}_f = \{\}$ // Initialize inverted index
3: **for** $r \in \mathbf{R_i}$ **do**: // Iterate records
4: $s = f(r)$ // Signature
5: $\mathbf{B}_f[s].add(r)$ // Store in inverted index
6: $e(\mathbf{B}_f)$ // Evaluate signature strategy
7: $f' = agree(\mathbf{S}', \forall_f e(\mathbf{B}_f))$ // Agree on a signature strategy

Phase 2 (by all parties P_i, with $1 \leq i \leq P$):

8: **for** $1 \leq i \leq P$ **do**: // Iterate P parties
9: $\mathbf{B}_i = \{\}; bf_i = []$ // Initialization
10: **for** $r \in \mathbf{R}_i$ **do**: // Iterate records
11: $s = f'(r)$ // Signature of r
12: $\mathbf{B}_i[s].add(r)$ // Store in inverted index
13: **for** $s \in \mathbf{B}_i$ **do** // Iterate signatures
14: **if not** $k_{min} \leq len(\mathbf{B}_i(s)) \geq k_{max}$ **do** // Larger and smaller blocks
15: $\mathbf{B}_i.remove(s)$ // Prune signatures
16: **for** $s \in \mathbf{B}_i$ **do**: // Iterate signatures
17: **for** $1 \leq j \leq k$ **do**: // Hash functions
18: $bf_i[h_j(s)] = 1$ // Set to 1 in BF
19: $cbf = sec\_sum(\forall_i bf_i)$ // Generate CBF

Phase 3 (by LU and by each party P_i, with $1 \leq i \leq P$):

20: $\mathbf{C} = \{\}$ // Initialization of $\mathbf{C}$
21: **for** $c \in cbf$ // LU iterates positions in CBF
22: **if** $c < s_m$ **then** // Count less than s_m
23: $c = 0$ // Set to 0
24: **else** // Count of at least s_m
25: $c = 1$ // Set to 1
26: $\forall_i \mathbf{C}.send\_to\_P_i()$ // LU sends Common BF to parties
27: **for** $1 \leq i \leq P$ **do** // All parties
28: **for** $s \in \mathbf{B}_i$ **do** // Iterate signatures
29: **if not** $\forall_{j=1}^{k} cbf[h_j(s)] == 1$ **then** // Membership test
30: $\mathbf{B}_i.remove(s)$ // Remove non-matching signatures
31: $\mathbf{B}_i.encode()$ // Encode records and BKVs
32: $\mathbf{B}_i.send\_to\_LU()$ // Send encoded blocks to LU
33: **for** $1 \leq i \leq P$ **do** // LU iterates parties
34: $C = \cup_i \mathbf{B}_i$ // Union of blocks from all parties
35: return $\mathbf{C}$ // Output $\mathbf{C}$

intersected BF contains 1 in positions that have 1 in all parties' BFs and 0 if at least one of the parties does not have 1 in those positions. An example is shown in Fig. 2 for two BFs.

However, for the linkage task of identifying all signatures/blocks that are common in at least s_m of p parties (for subset matching), we propose to use a CBF of p BFs which contains counts of 1-bits from all p BFs. A CBF is generated from all p BFs using a secure summation protocol (line 19). It contains the count values of common signatures (i.e. how many parties have those common signatures), which are in between 0 (if none of the p parties' BFs contain 1 in those bit positions) and p (if all p parties' BFs contain 1).

3. Blocks generation: The LU replaces all the count values in the generated CBF that are below the minimum subset size, s_m, to 0 as these are not common signatures across at least s_m parties, while count values above or equal to s_m are set to 1 (lines 21–25 in Algorithm 1). This implies that blocks need to be common across at least s_m parties for subset matching. The resulting CBF that contains 1s and 0s (which is essentially a BF) is sent to all the parties (line 26). The parties individually perform a membership test (as described in Sect. 2) on

the received CBF by checking all their signatures in order to determine if they are common or not (line 27–30 in Algorithm 1).

The encoded records belonging to each of the common signatures/blocks are sent to the LU to perform clustering on records belonging to the same blocks (lines 31–32). The union of blocks from all parties are stored in C and returned by the blocking method (lines 33–35), which will be used as an input to the clustering step.

3.1 Local Blocking Evaluation Framework

The performance of blocking (in terms of comparison space reduction, retaining true matches, and not being susceptible to frequency attacks) depends on the signature strategies (similar to most of the existing blocking methods). For such data-driven blocking techniques, we propose a framework to locally evaluate the blocking performance in order to choose and agree on a signature strategy that performs better in terms of all three aspects. This framework is applicable to any blocking method for local evaluation that provides minimum guarantees of the global blocking results.

Comparison Space Reduction: This refers to the global measure of *reduction ratio* of a blocking method [4]. The reduction ratio measures the percentage of record pair comparisons reduced after blocking from the total number of record pair comparisons. Different signature strategies generate different number and size of blocks and therefore vary by the reduction ratio. Performing blocking with many different strategies across parties and evaluating and comparing their reduction ratio to choose the best strategy is not trivial in a real application due to operational cost and privacy concerns. Therefore, we use a measure to locally evaluate and compare different signature strategies by each party individually on their records.

The statistics of the block sizes for each of the signature strategies can be compared to learn about their impact on the reduction ratio. We consider the average and maximum block sizes as local measures of reduction guarantees. We normalise these values in the range of $[0, 1]$ for comparative evaluation. The Reduction Guarantees metric RG is defined as $RG = 1 - m/n$, where m is the average or maximum block size and n is the total number of records in the dataset. For example, if a blocking strategy results in a maximum block size of $m = 1$ for a dataset of $n = 10000$ records, then $RG_{max} = 1 - 1/10000 = 0.9999$, while a maximum block size of $m = 10000$ results in $RG_{max} = 0.0$.

True Matches Preservation: This refers to the global measure of *pairs completeness* (or *recall*) of a blocking method [4]. Pairs completeness measures the percentage of true matches preserved in the candidate record pairs resulting from blocking in the total number of true matches. Smaller blocks favor the reduction ratio, however, they can have a negative impact on the pairs completeness as they have more likelihood of missing true matches (not grouped into the same block). We use Quality Guarantees (QG) metrics to locally evaluate the likelihood of not missing true matches in the candidate record pairs.

This likelihood is determined by the coverage of records in blocks. We measure the coverage by calculating the statistics of number of blocks per record (average and minimum). The larger the number of blocks where a record appears in, the more likelihood that it will be compared with a potential matching record in one of those blocks. Specifically, a signature strategy that leads records being appear in average m out of b total blocks and at least 1 block (minimum), then the QG metrics are calculated as $QG_{avg} = m/b$ and $QG_{min} = 1/b$.

Privacy Guarantees: While smaller blocks are preferred for reduction guarantees, and overlapping blocks are preferred for quality guarantees, these two have negative impact on the privacy guarantees. Based on the sizes of the blocks, the LU can perform a frequency inference attack by matching the frequency distribution of blocks to a known frequency distribution, as will be detailed in Sect. 3.3). A blocking method that generates blocks with low variance between their sizes is less susceptible to such frequency inference attacks. Moreover, too small blocks are highly vulnerable as they provide information about unique and rare values.

For Privacy Guarantees (PG) metrics, we calculate disclosure risk statistics [21] (average, maximum, and marketer risk) based on the probability of suspicion (P_s) for each record in blocks of a local database $\mathbf{D}$. P_s for a record r is calculated as $P_s(r) = 1/n_g$ where n_g is the number of possible matches in the global database $\mathbf{D}_G$ with r. We assume the worst case of $\mathbf{D}_G \equiv \mathbf{D}$, to calculate the minimum local privacy guarantees. Each of the records in a block of k records has the P_s of $1/k$ (i.e. each record matches with k records in the worst case). For example, if $k = 1$, then $P_s = 1.0$, whereas $k = 100$ gives $P_s = 0.01$ for all k records. Based on the P_s values, we calculate the maximum PG as $PG_{max} = max_{r_i \in \mathbf{D}}(P_s(r_i))$, average PG as $PG_{avg} = \sum_i^n P_s(r_i)/n$, and marketer PG as the proportion of records that can be exactly re-identified, i.e. $P_s = 1.0$, $PG_{mar} = |\{r_i \in \mathbf{D} : P_s(r_i) = 1.0\}|/n$ [21].

By locally evaluating and comparing the blocks generated by different blocking strategies using the privacy guarantees (PG), quality guarantees (QG), and reduction guarantees (RG) metrics, the parties can choose and agree on a strategy that can generate good blocking results in terms of the three aspects. We name the family of these metrics for local blocking evaluation as PQR-guarantees metrics, which refer to the Privacy, Quality, and Reduction guarantees of blocking methods.

3.2 Complexity Analysis

Assume p parties participate in the linkage of their respective databases, each containing n records, and b blocks are generated by the blocking function, with each block containing n/b records. Phase 1 has a linear computation complexity for each party as it requires a loop over all records in its database for multiple different signature strategies in a set of strategies, S', and calculating the Privacy, Quality, and Reduction Guarantees (PQR-guarantees) metrics as described in

Sect. 3.1 ($O(n \cdot |S'|)$). Agreeing on a signature strategy across multiple parties based on the PQR-guarantees metrics has a constant communication complexity.

In the second phase, encoding the candidate signatures into a BF of length l bits using k hash functions has a computation complexity of $O(b' \cdot k)$ (assuming b' candidate signatures) for each party, and generating a CBF using secure summation protocol is of $O(l)$ computation and communication complexity. In phase 3, the LU loops through the CBF to generate the intersected BF, which is $O(l)$, and sending to all parties is $O(l \cdot p)$ communication complexity. Each party individually performs membership test of their candidate signatures, which is of $O(b' \cdot k)$. Then the records containing any of the common signatures (assuming b common signatures/blocks) need to be retrieved and sent to the LU, which has a computation and communication complexity of $O(n \cdot b)$. At the LU, the number of candidate record pairs generated is $n^2/b \cdot p^2$. Similar to many existing methods, the reduction in the number of candidate record pairs depends on the number (b) and size of blocks (n/b on average) generated. Therefore, the proposed RG metric based on local block sizes can provide an estimate to locally evaluate the reduction in candidate record pairs.

3.3 Privacy Analysis

As with most existing PPRL methods, we assume that all parties follow the honest-but-curious adversary model [21], where the parties follow the protocol while being curious to find out as much as possible about other parties' data by means of inference attacks on (blocks of) encoded records or by colluding with other parties [21].

In Phase 2, the parties perform secure summation of their BFs, which does not leak any information about the individual BFs. However, secure summation protocols can be susceptible to collusion attacks where two or more parties collude to learn about another party's BF. There have been several extended secure summation protocols developed to reduce their vulnerability to collusion risk. For example, secret sharing-based protocol [17] generates p random shares r_i (one share per party) from the secret input value v_i, such that $\sum_i r_i = v_i$, and therefore even when some of the parties collude, without knowing the shares of other non-colluding parties the input value v_i of a party cannot be learned by the colluding parties.

In Phase 3, since the CBF contains only the summary information (count values), it provides more privacy guarantees than BFs against an inference attack by the LU.

Proposition 1. *The probability of inferring the values of individual signatures s_i of a party P_i (with $1 \leq i \leq p$) given a single CBF c is smaller than the probability of inferring the values of s_i given the corresponding party's BF b_i, $1 \leq i \leq p$.*

Proof. Assume the number of potential matching signatures from an external database that can be matched to a single signature $s \in s_i$ encoded into the BF

b_i through an inference attack is n_g. $n_g = 1$ in the worst case, where a one-to-one mapping exists between the encoded BF b_i and the candidate signatures (based on performing membership test). The probability of inferring the signature value s belonging to a party P_i given its BF b_i in the worst case scenario is therefore $Pr(s \in s_i|b_i) = 1/n_g = 1.0$. However, a CBF represents signatures from p parties and thus $Pr(s \in s_i|c) = 1/p$ in the worst case with $p > 1$. Hence, $\forall_{i=1}^{p} Pr(s \in s_i|c) < Pr(s \in s_i|b_i)$.

Finally, the parties send their blocks of encoded records to the LU. If one of the resulting blocks contains only one record, for example, then the likelihood of a successful inference of this record by the LU is higher than the inference of a record that belongs to a block of size 100. Similarly, a very large block can be uniquely identified by matching to a frequent value in the global database. Therefore, the variance between block sizes needs to be smaller to reduce the vulnerability of blocking methods to frequency inference attack. Our **P3-SIG** method prunes highly frequent ($> k_{max}$) and rare ($> k_{min}$) blocks to provide privacy guarantees, which can be locally evaluated as discussed in Sect. 3.1.

4 Experimental Evaluation

We conducted our experiments on three different datasets:

(1) **NCVR:** We extracted 4611, 46,116 and 461,116 records from the North Carolina Voter Registration (NCVR) database[1] for two parties with 50% of matching records between the two parties. Ground truth is available based on the voter registration identifiers. We generated another series of datasets where we synthetically corrupted/modified randomly chosen attribute value of records by means of character edit operations and phonetic modifications using the GeCo tool [18].

(2) **NCVR-Subset:** We sampled 10 datasets from the NCVR database each containing 10,000 records such that 50% of records are non-matches and 5% of records are true matches across each different subset size of 1 to 10 (1, 2, 3, $\cdots$, 9, 10), i.e. 45% of records are matching in any 2 datasets while only 5% of records are matching in any 9 out of all 10 datasets. This dataset is used to evaluate our method for multi-party PPRL with different subset sizes.

(3) **ABS Dataset:** This is a synthetic dataset used internally for linkage experiments at the Australian Bureau of Statistics (ABS). It simulates an employment census and two supplementary surveys. There are 120000, 180000 and 360000 records, respectively, with 50000 true matches.

We use six existing private blocking methods as the baseline approaches to compare our proposed approach (**P3-SIG**), which are three-party (two database owners and a LU) sorted neighbourhood clustering (SNC)-based blocking (**SNC-3P**) [20], two-party (without LU) SNC-based method (**SNC-2P**) [23], hierarchical clustering based approach (**HCLUST**) [14], k-nearest neighbourhood clustering-based method (**k-NN**) [10], Hamming LSH-based blocking

[1] Available from ftp://alt.ncsbe.gov/data/.

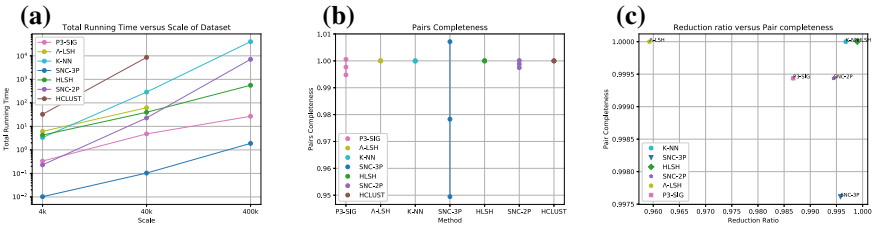

Fig. 3. Comparison of (a) Scalability, (b) Pairs Completeness, and (c) Reduction ratio vs. Pairs Completeness for two-database linking on **NCVR** dataset.

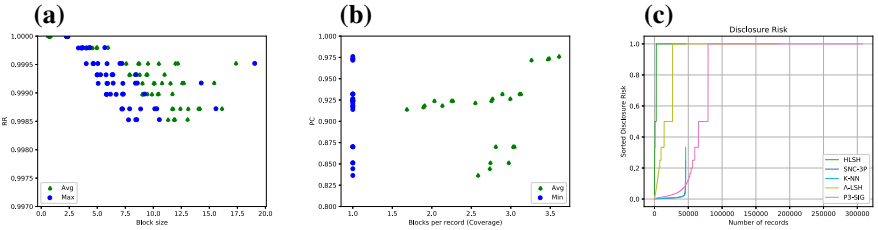

Fig. 4. Correlation between (a) local block sizes and reduction ratio (RR) metric and (b) local coverage and pairs completeness (PC) metric, and comparison of (c) disclosure risk of **P3-SIG** method with baseline methods on **NCVR** dataset.

method (**HLSH**) [7], and λ-fold LSH-based blocking method (λ**-LSH**) [11]. We choose methods for comparison that fall under different categories of shortcomings of existing methods as described in Sect. 1.

We evaluate the complexity (computational efficiency) using *runtime* required for the blocking and *reduction ratio* (RR) of record pair comparisons for the linkage (clustering). RR is calculated as $1.0 - \frac{number\ of comparisons\ after\ blocking}{total\ number\ of\ comparisons}$. The quality of the resulting candidate record pairs by a blocking method is measured using the *pairs completeness* (PC) for two-database linking and *set completeness* (SC) for multi-database linking [3,21]. They are calculated as the percentage of true matching pairs/sets that are found in the candidate record pairs/sets in the total number of true matching record pairs/sets, respectively. We evaluate privacy guarantees against frequency attack using block sizes and disclosure risk values [21], as described in Sect. 3.1.

We implemented our **P3-SIG** approach and the competing baseline approaches in Python 3.7.4[2], and ran all experiments on a server with 4-core 64-bit Intel 2.8 GHz CPU, 16 GBytes of memory and running OS X 10.15.1. For the baseline methods, we used the parameter settings as used by the authors in the corresponding methods. For **P3-SIG** method, the default parameter setting is length of BFs $l = 2048$, and number of hash functions $k = 4$. We evaluated multiple different strategies generated from the combinations of first character, first 2 characters, first 3 characters, all characters, phonetic encodings, q-grams,

[2] available in http://doi.org/10.5281/zenodo.3653169.

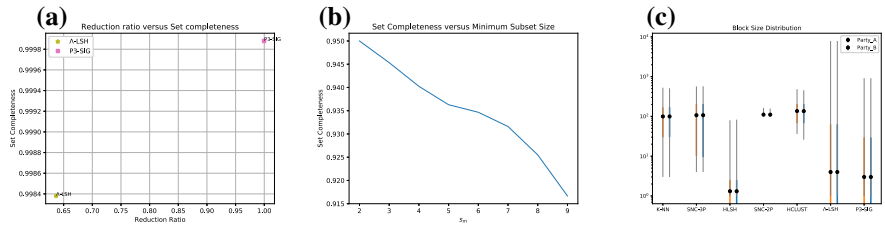

Fig. 5. (a) Reduction ratio vs. Set Completeness for multi-database linking on **ABS** dataset, (b) Set Completeness of **P3-SIG** blocking for subset matching of $p = 10$ databases against s_m on **NCVR**-subset dataset, and (c) comparison of block size distribution of **P3-SIG** method with baseline methods on **NCVR** dataset.

and 'None of the characters' in each of the QID (attribute) values of a record. Based on greedy search parameter tuning method, we used the numerical values in the first four attributes combined with the gender value as the default signature strategy for ABS dataset, and for the NCVR dataset we used the full first and last names, phonetic encoding of the first and last names along with the first one or two characters of first and last names and suburb values.

4.1 Discussion

We compare our method with the baseline methods in terms of runtime, pairs completeness (PC), and reduction ratio (RR) vs. pairs completeness (PC) in Fig. 3 for two-database linkage on the NCVR dataset. In terms of runtime, LSH-based methods and clustering based methods require more time followed by SNC-2P. Our method requires lower runtime than these methods, however SNC-3P is more efficient than our method. Our method however achieves higher RR and PC than the SNC-3P method. LSH-based blocking method generates higher quality blocking results, but they require higher computational cost for blocking. We were unable to conduct experiments for the λ-LSH and HCLUST methods on the largest dataset due to their memory and space requirements.

We next study the effectiveness of our local blocking evaluation framework. The correlation between the local block sizes and global reduction ratio (RR) metric as well the correlation between the local coverage values and global pairs completeness (PC) metric for a set of different signature strategies are shown in Figs. 4(a) and 4(b), respectively. As the results show, there exist a high correlation between them which reveals that local RG and QG metrics can be effectively used for blocking quality evaluation. PC and coverage values have a strong positive correlation, while RR and block sizes are negatively correlated. Figure 4(c) compares the maximum disclosure risk values calculated against a frequency attack in the worst case ($\mathbf{D} \equiv \mathbf{D}^G$) with baseline methods. The privacy guarantees (PG) results show that the disclosure risk values against a frequency inference attack are lower with our method.

We compare our method with the λ-fold LSH multi-party blocking method for multi-party linkage in Fig. 5(a). As can be seen, P3-SIG method outperforms

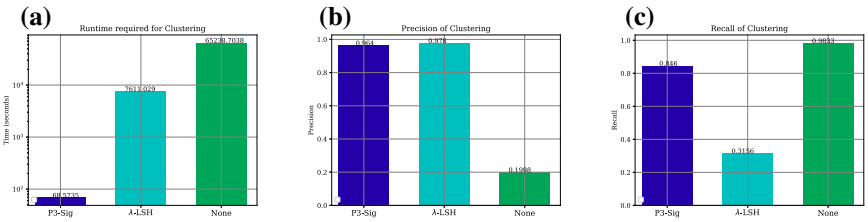

Fig. 6. Comparison of (a) runtime, (b) precision, and (c) recall of clustering for multi-party PPRL [22] with P3-SIG, λ-fold LSH [11], and no blocking on **NCVR** dataset.

λ-fold LSH for multi-party blocking in terms of higher blocking quality. Please note that λ-LSH method works efficiently on small datasets, however on large datasets it requires high runtime and memory space. Figure 5(b) shows the set completeness results for subset matching of $p = 10$ databases from NCVR-Subset dataset against different minimum subset sizes s_m. The larger the value for s_m is, the more difficult it is to find the set of records that match across at least s_m databases/parties. This reflects the challenge of subset matching in multi-party PPRL. These results show that P3-Sig can efficiently be used for multi-party linkage applications. Further, we compare the size of blocks generated by the different blocking methods in Fig. 5(c), which shows that the size of the blocks resulting from our method is similar to that of LSH-based methods, as they both generate overlapping blocks, however our method is more efficient and faster than these methods while achieving similar or superior blocking quality.

Finally, we evaluate our proposed P3-SIG method's performance on a recently developed incremental clustering method for multi-party PPRL [22] and compare with no blocking and λ-fold LSH multi-party blocking method in Fig. 6. The runtime of the PPRL reduces significantly using our method without impacting the linkage quality, which validates the efficacy of our blocking method for efficient clustering required by multi-party PPRL.

5 Related Work

Various blocking techniques have been proposed in the literature tackling the scalability problem of PPRL, as surveyed in [16,19,21]. Most of these methods require external data (reference values) of similar distribution as the original databases to be linked and employ a similarity comparison function to group similar records. For example, in [10] reference values are clustered using the k-nearest neighbor clustering algorithm and then the records are assigned to the nearest cluster. A token-based blocking method is proposed in [1], which requires calculating the TF-IDF distances of the hash signatures of blocking keys.

Similarly, sorted neighbourhood clustering is used in [20] and [23] to group similar reference and record values with and without a LU, respectively. Another method using hierarchical clustering to group similar reference values is proposed

in [14] where the records are then assigned to the nearest clusters and differential privacy noise is added to the blocks (clusters) to reduce the vulnerability to inference attacks.

Other set of methods rely on data-specific parameters that are highly sensitive to data. A private blocking method for PPRL of multiple database using Bloom filters and bit-trees is proposed [15]. This method is only applicable to Bloom filter encoded data. The method introduced in [7] uses a set of hash functions (Minhash for Jaccard or Hamming LSH for Hamming distances) to generate keys from records that are encoded into Bloom filters to partition the records, so that similar records are grouped into the same block [12]. [11] proposed a λ-fold LSH blocking approach for linking multiple databases. LSH provides guaranteed accuracy, however, this approach requires data dependent parameters to be tuned effectively and it can be applied only to specific encodings, such as Bloom filters or q-gram vectors.

6 Conclusion

We have presented a scalable private blocking protocol for PPRL that is highly efficient and improves blocking quality compared to existing private blocking approaches. In contrast to most of the existing methods that rely on a clustering technique for blocking records, our method uses signatures in the records to efficiently group records as well as to account for data errors and variations. Further, our blocking method is applicable to linking multiple databases as well as subset matching for multi-party PPRL. We also introduce a local blocking evaluation framework to choose good signature strategies/parameter settings in terms of privacy, blocking quality, and comparison reduction guarantees. Experiments conducted on datasets sampled from two real datasets show the efficacy of our proposed method compared to six state-of-the-art methods.

In future work, we aim to study optimisation techniques, such as Bayesian optimisation, to choose/tune signature strategies for optimal results. We also plan to study parallelisation to improve the scalability of blocking and linkage for multi-party PPRL. Finally, improving privacy guarantees for blocking methods needs to be explored in two different directions: (1) developing methods that provide formal privacy guarantees, such as output-constrained differential privacy [9], without significant utility loss, and (2) developing hybrid methods that combine cryptographic methods with probabilistic encoding methods (such as Bloom filter encoding) without excessive computational overhead.

Acknowledgements. This work was funded by the Australian Department of Social Sciences (DSS) as part of the Platforms for Open Data (PfOD) project. We would like to thank Waylon Nielsen and Alex Ware from DSS for their support on this work.

References

1. Al-Lawati, A., Lee, D., McDaniel, P.: Blocking-aware private record linkage. In: IQIS, pp. 59–68 (2005)

2. Baker, D., et al.: Privacy-preserving linkage of genomic and clinical data sets. Trans. Comput. Biol. Bioinform. **16**, 1342–1348 (2018)
3. Christen, P.: Data Matching. Data-Centric Systems and Applications. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31164-2
4. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. IEEE TKDE **24**(9), 1537–1555 (2012)
5. Cohen, W.W., Ravikumar, P., Fienberg, S.: A comparison of string distance metrics for name-matching tasks. In: IJCAI Workshop on Information Integration on the Web, pp. 73–78 (2003)
6. Condon, J.R., Barnes, T., Cunningham, J., Armstrong, B.K.: Long-term trends in cancer mortality for indigenous Australians in the northern territory. Med. J. Aust. **180**(10), 504 (2004)
7. Durham, E.A.: A framework for accurate, efficient private record linkage. Ph.D. thesis, Vanderbilt University, Nashville, TN (2012)
8. Han, S., Shen, D., Nie, T., Kou, Y., Yu, G.: Private blocking technique for multi-party privacy-preserving record linkage. Data Sci. Eng. **2**(2), 187–196 (2017). https://doi.org/10.1007/s41019-017-0041-5
9. He, X., Machanavajjhala, A., Flynn, C., Srivastava, D.: Composing differential privacy and secure computation: a case study on scaling private record linkage. In: ACM CCS, pp. 1389–1406 (2017)
10. Karakasidis, A., Verykios, V.S.: Reference table based k-anonymous private blocking. In: ACM SAC, Riva del Garda, pp. 859–864 (2012)
11. Karapiperis, D., Verykios, V.S.: A fast and efficient Hamming LSH-based scheme for accurate linkage. Knowl. Inf. Syst. **49**(3), 861–884 (2016). https://doi.org/10.1007/s10115-016-0919-y
12. Kim, H., Lee, D.: HARRA: fast iterative hashed record linkage for large-scale data collections. In: EDBT, Lausanne, Switzerland, pp. 525–536 (2010)
13. Kuehni, C.E., et al. and Swiss Paediatric Oncology Group (SPOG): Cohort profile: the Swiss childhood cancer survivor study. Int. J. Epidemiol. **41**(6), 1553–1564 (2011)
14. Kuzu, M., Kantarcioglu, M., Inan, A., Bertino, E., Durham, E., Malin, B.: Efficient privacy-aware record integration. In: ACM EDBT, Genoa, pp. 167–178 (2013)
15. Ranbaduge, T., Vatsalan, D., Christen, P., Verykios, V.: Hashing-based distributed multi-party blocking for privacy-preserving record linkage. In: Bailey, J., Khan, L., Washio, T., Dobbie, G., Huang, J.Z., Wang, R. (eds.) PAKDD 2016. LNCS (LNAI), vol. 9652, pp. 415–427. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31750-2_33
16. Schnell, R.: Privacy-preserving record linkage. In: Methodological Developments in Data Linkage, pp. 201–225 (2015)
17. Tassa, T., Cohen, D.J.: Anonymization of centralized and distributed social networks by sequential clustering. IEEE Trans. Knowl. Data Eng. **25**(2), 311–324 (2011)
18. Tran, K.N., Vatsalan, D., Christen, P.: GeCo: an online personal data generator and corruptor. In: ACM Conference in Knowledge Management (CIKM), San Francisco, pp. 2473–2476 (2013)
19. Trepetin, S.: Privacy-preserving string comparisons in record linkage systems: a review. Inf. Secur. J.: Global Perspect. **17**(5), 253–266 (2008)
20. Vatsalan, D., Christen, P.: Sorted nearest neighborhood clustering for efficient private blocking. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7819, pp. 341–352. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37456-2_29

21. Vatsalan, D., Christen, P., O'Keefe, C.M., Verykios, V.S.: An evaluation framework for privacy-preserving record linkage. JPC **6**(1), 35–75 (2014)
22. Vatsalan, D., Christen, P., Rahm, E.: Incremental clustering techniques for multi-party privacy-preserving record linkage. Data Knowl. Eng. **128**, 101809 (2020)
23. Vatsalan, D., Christen, P., Verykios, V.S.: Efficient two-party private blocking based on sorted nearest neighborhood clustering. In: ACM CIKM, San Francisco, pp. 1949–1958 (2013)
24. Zhang, Y., Ng, K.S., Churchill, T., Christen, P.: Scalable entity resolution using probabilistic signatures on parallel databases. In: ACM CIKM, Turin, Italy, pp. 2213–2221 (2018)