B. König-Ries et al. (Hrsg.): Datenbanksysteme für Business, Technologie und Web (BTW 2023),
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2023 1

Value-specific Weighting for Record-level Encodings in Privacy-Preserving Record Linkage

Florens Rohde[1], Martin Franke[1], Victor Christen[1], Erhard Rahm[1]

Abstract: Privacy-preserving record linkage (PPRL) determines records representing the same entity while guaranteeing the privacy of individuals. A common approach is to encode plaintext data of records into Bloom filters that enable efficient calculation of similarities. A crucial step of PPRL is the classification of Bloom filter pairs as match or non-match based on computed similarities. In the context of record linkage, several weighting schemes and classification methods are available. The majority of weighting methods determine and adapt weights by applying the Fellegi&Sunter model for each attribute. In the PPRL domain, the attributes of a record are encoded in a joint record-level Bloom filter to impede cryptanalysis attacks so that the application of existing attribute-wise weighting approaches is not feasible. We study methods that use attribute-specific weights in record-level encodings and integrate weight adaptation approaches based on individual value frequencies. The experiments on real-world datasets show that frequency-dependent weighting schemes improve the linkage quality as well as the robustness with regard to threshold selection.

Keywords: Privacy-preserving record linkage; Bloom filter; Weighting; Value-specific

1 Introduction

Record linkage is an essential component in many data integration tasks with multiple data sources. It aims to detect records that belong to the same real-world entity such as a person. Typically, unique record identifiers are not available which would enable a join-like operation [Ch12]. Therefore, records are compared pairwise based on their attributes, such as first name, last name, date of birth and gender. The attribute similarities are used to classify pairs as match or non-match. Often weights are involved in this step to take the different discriminatory power and error rates of attributes into account [WT91]. For example, an equal date of birth is a stronger indicator for a match than an equal gender as there are much more values (and thus each value occurs less often) for date of birth than for gender.

Simple weight-based classification approaches only use attribute-specific weights that are equal for all values of a certain attribute. Thus, the very common last name *Smith* would result in the same weight as the rarer last name *Voigt*. Therefore, the use of value-specific weights based on the frequency of a specific attribute value can increase the linkage quality [WT91]. For uncertain duplicate candidates, e. g., due to a different address as in the following example (see Tab. 1 and 2), the likelihood of a match is higher if the agreeing

[1] University of Leipzig & ScaDS.AI Dresden/Leipzig {rohde,franke,christen,rahm}@informatik.uni-leipzig.de

attributes – here first and last name – are rare. This is reflected in a higher record similarity score (weighted average) due to increased weights of those attributes.

Tab. 1: Example of a similarity computation of two records with *common* first and last name.

	First name	**Last name**	**Date of birth**	**ZIP code**	**City**	**Total**
Record a	LISA	SMITH	23.09.1973	28451	LELAND	
Record b	LISA	SMITH	23.09.1973	28075	HARRISBURG	
Similarity	1.0	1.0	1.0	0.4	0.0	**0.79**
Weight	**12**	**13**	15	7	7	

Tab. 2: Example of a similarity computation of two records with *rare* first and last name.

	First name	**Last name**	**Date of birth**	**ZIP code**	**City**	**Total**
Record a	WYNONA	VOIGT	23.09.1973	28451	LELAND	
Record b	WYNONA	VOIGT	23.09.1973	28075	HARRISBURG	
Similarity	1.0	1.0	1.0	0.4	0.0	**0.85**
Weight	**20**	**25**	15	7	7	

To enable the assignment of globally unique record identifiers multiple data owners share their respective datasets with a trusted institution, called linkage unit, which is responsible for the actual linkage and determines pairs of records considered as a *match*. Using these identifiers the data owners can combine their respective data on matching entities. The exchange of sensitive data, such as identifying personal information, between the data owners or with the linkage unit is, however, restricted by law [CRS20]. Privacy-preserving record linkage (PPRL) addresses this challenge. It has been an active research subject for the last decades [VCV13]. To protect the sensitive data, it is encoded before being sent to the linkage unit which performs the linkage on the encoded data only. A variety of encoding techniques have been proposed, but the most popular and quasi-standard is based on Bloom filters [Gk21]. However, the initially proposed attribute-level encoding [SBR09], where each attribute is encoded in a separate Bloom filter, has been shown to be susceptible to frequency and pattern mining attacks [Vi22]. Therefore, state-of-the-art techniques combine multiple or all attributes into a joint record-level encoding to impede those attacks.

In general, Bloom filter based encodings (both attribute-level and record-level) allow for weighting attributes. Attribute-level encodings are very similar to traditional (plaintext) record linkage with regard to weighting. The attribute similarities can be aggregated to a record similarity, for example, by using a weighted average. The only difference effectively is the use of a similarity function that is suited for the encoded data structure. When using record-level encodings, the data owners can use different parameters per attribute to change the attributes' relative weight in the joint Bloom filter. However, weight adaptation and application in the PPRL context with record-level encodings differ from traditional record linkage as they must be applied by the data owners.

Specifically, we make the following contributions:

- We study the challenges that arise when applying value-specific weighting in the PPRL context to record-level encodings, e. g., the handling of name variations and missing values during the encoding phase.
- We modify record-level encoding techniques for PPRL to allow for frequency-dependent weight adaptation.
- We thoroughly evaluate these techniques and compare them to existing weighting approaches on attribute-level and record-level encodings. Moreover, we analyze the effects of using limited information on value frequencies as the complete information is considered sensitive in the PPRL context.

The paper is structured as follows. In the next section, we discuss Related Work. In Sect. 3 we describe the PPRL encoding and matching process. Then, we discuss weighting-based classification approaches in the PPRL context (Sect. 4) and present an extensive comparative evaluation of the different approaches using a real-world dataset (Sect. 5). Finally, we conclude our work in Sect. 6.

2 Related Work

The idea of assigning weights to different attributes when used for calculating similarities between records is part of the probabilistic record linkage approach proposed by Fellegi and Sunter in [FS69]. The weighting of attributes addresses the fact that each attribute has a different number of (possible) values and these values follow a certain distribution. Attributes can also be erroneous or out of date, with some attributes being affected more often than others. Consequently, for each attribute i two probabilities, namely the m- and u-probability, are determined as

$$m_i = P(a_i = b_i, a \in A, b \in B | a \equiv b)$$
$$u_i = P(a_i = b_i, a \in A, b \in B | a \not\equiv b)$$

where a is a record from database A, b is a record from database B and a_i and b_i are the values of attribute i of record a and b, respectively. With $\equiv$ we denote the equivalence relation, i. e., both records refer to the same entity. The m-probability specifies the probability that two records have the same value for attribute i, given the records refer to the same entity. Ideally, $m_i = 1$ if all true matches agree on attribute i. This is exactly the case if attribute i does not contain any errors. If, for example, 20 % of the duplicates have a non-equal value, for instance due to a typographical error, then $m = 0.8$. In contrast, the u-probability specifies the probability that two records have the same value for attribute i given the records refer to different entities. The u-probability is low if the attribute has a

wide range of possible values. In contrast, if, for example, an attribute has only two possible and equally likely values, then $u = 0.5$ as the chance that the attribute agrees for two random records is 50 %. u is typically frequency-dependent as a random agreement is more likely for common than for rare values.

Using the m- and u-probabilities the weight w_i for attribute i is calculated as

$$w_i = \begin{cases} w_m = \log_2\left(\frac{m_i}{u_i}\right) & \text{if } a_i = b_i \\ w_u = \log_2\left(\frac{1-m_i}{1-u_i}\right) & \text{if } a_i \neq b_i \end{cases} \quad (1)$$

The probabilistic record linkage approach by Fellegi and Sunter is the basis for many record linkage approaches and is still frequently used and adapted [Ch12; HSW07].

Herzog et al. [HSW07] propose a method to adjust match and non-match weights also based on the frequency of individual attribute values. Consequently, an attribute-specific and a value-specific weight is used. The authors provide a detailed discussion about the calculation of these weights. Similarly, Zhu et al. [Zh09] propose a scaling factor that is applied directly to the attribute weights of the Fellegi-Sunter approach. The scaling factor is calculated based on the present dataset without an external source of (name) frequencies.

Attribute weighting has been used in the PPRL domain as well. The record linkage and pseudonymization service Mainzelliste, which supports Bloom filter based matching, only uses the agreement weights to combine attribute similarity scores to a record similarity using the weighted average [Ro21]. In [Br17], weights are estimated based on partial agreement models for each individual attribute of a sensitive dataset. However, this approach can only be utilized for attribute-level encodings. Ranbaduge et al. proposed decay weights for record-level encodings based on time distances [RC18]. Value-specific weighting approaches, however, have received limited attention so far in PPRL. Giersiepen et al. apply the Fellegi-Sunter approach with frequency-dependent u-probabilities to encrypted attribute-level hashes [Gi10]. This approach is the standard procedure used by German cancer registries. To the best of our knowledge, no prior work has studied value-specific weight adaptation based on individual value frequencies for record-level encodings so far.

3 Background

The general privacy-preserving record linkage process is shown in Fig. 1. We follow a three-party protocol that uses a semi-trusted third party, called linkage unit (LU), to conduct the linkage [CRS20]. The protocol is based on an Honest-But-Curious adversary model which means that all parties follow the protocol but try to learn as much as possible about the sensitive data of others. To protect the privacy of individuals, the quasi-identifying attributes, such as names, dates of birth or addresses, are encoded by the data owners (DO). Often, a preprocessing step is performed before to reduce data quality problems and to

convert the data into a standardized format. Only the encoded quasi-identifiers are then shared with the LU. The LU compares records pairwise and classifies them as *Match* or *Non-Match*. The following subsections explain the matching and encoding phases in more detail.

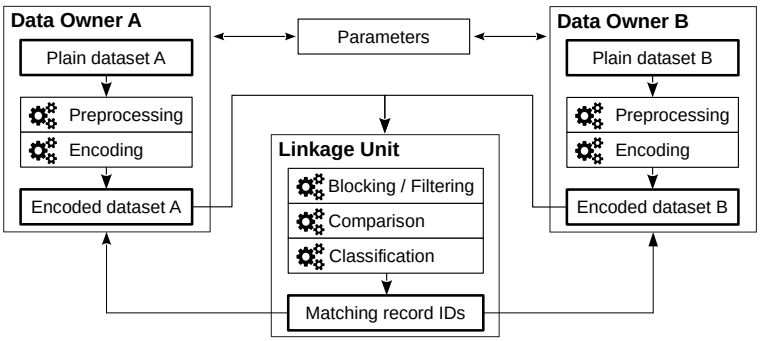

Fig. 1: Privacy-preserving record linkage protocol with two data owners and a semi-trusted third party as the linkage unit.

3.1 Encoding

In the encoding phase the plaintext is transformed into an encoded representation that cannot be reverted to its original form. An obvious solution is the use of cryptographic hash functions. However, simple hashes are only suitable for exact matching, as even small differences in the input result in very distinct hash values. Therefore, similarity-preserving encodings have been developed to enable approximate matching of records containing errors or inconsistencies, such as typos or outdated values.

The use of Bloom filters for PPRL has been proposed by Schnell and colleagues [SBR09]. It became the most popular encoding scheme for PPRL in research as well as in real applications [CRS20]. In general, quasi-identifying attributes are split into n substrings of length q (q-grams) to build a set of record features $F = \{e_1, \ldots, e_n\}$ being represented in a Bloom filter. The original strings can be surrounded by leading and trailing padding characters to ensure that all characters are included in the same number of q-grams, which has been shown to lead to a higher linkage quality [Fr21]. At first, a bit vector of size l is initialized with each bit set to zero. Moreover, k hash functions $h_1, \ldots, h_k$ are defined and used to hash (map) the elements of F into the bit vector. Therefore, *each* hash function is applied on *each* element of F and produces a position in the range $[0, l-1]$ as output. Finally, the bits at the resulting positions are set to one. Given that identical q-grams are mapped to the same bit positions, a high overlap of q-grams leads to similar Bloom filters making them suitable for determining the record similarity.

However, due to the deterministic encoding, frequent patterns in the plaintext values will lead to frequently set bit positions in the encoded data and thus enabling frequency attacks. This is true in particular for *attribute-level Bloom filter (ABF)* where a separate Bloom filter is used for each attribute. Consequently, frequently occurring plaintext attribute values can be aligned with frequently occurring Bloom filters. To hamper such attacks, state-of-the-art encodings combine multiple or all attributes into a joint Record-level encoding [Vi22]. The encoding procedure must not be known to the Linkage Unit because otherwise it could conduct dictionary attacks by encoding possible records, e.g., from a similar public dataset, in the same way and infer the membership of a possible record in the dataset. Therefore the encoding output must depend on secrets that are private to the data owners, e.g., by using keyed hash functions.

3.2 Matching

In the matching phase records are compared pairwise and classified as match or non-match. To reduce the quadratic complexity of comparing each record of one source with each record of the other source, blocking or filtering techniques can be used [Ch12]. Records that do not meet specific pre-defined blocking or filtering criteria are considered a non-match and thus, are not further compared. Possible blocking keys on plaintext are, for example, year of birth, geographical data items or phonetic encodings of the name. Blocking techniques for Bloom filter based PPRL using Locality-sensitive hashing have been proposed and evaluated [FSR18].

Similarities of Bloom filter encodings can be computed with set similarity measures. In this work we use the Dice coefficient [Di45] which is defined as $D(a,b) = (2 \cdot |a \cap b|)/(|a|+|b|)$ for Bloom filters a and b where $\cap$ denotes the intersection (logical AND) operation and $|\cdot|$ the hamming weight of a Bloom filter (number of 1-bits). The resulting similarity score is normalized in the range $[0, 1]$. When using ABF encodings, the attribute similarity scores have to be aggregated to a record similarity score, for instance, by computing a weighted average (see Sec. 4.1). If the record similarity score is above a predefined threshold t, the record pair is classified as a match, otherwise as a non-match.

4 Methods

In this section we describe how attribute weights can be applied in the PPRL context, followed by a discussion of methods to adapt the weights depending on the attribute values and their frequencies. Furthermore, we describe approaches to estimate weights and the limitations that arise when transferred to the PPRL domain using record-level encodings.

4.1 Weight application

Attribute weights can be applied in different ways in the PPRL process depending on the encoding strategy. If *attribute-level Bloom filter* (ABF) are used, the linkage unit can compare record pairs attribute-wise. In the probabilistic record linkage theory of Fellegi and Sunter [FS69] (positive) agreement and (negative) disagreement weights are assigned to each attribute depending on whether they are equal or not (see Equation (1)) The total weight is calculated by adding up the respective weights of all attribute pairs. However, this approach does not make use of approximate similarity functions.

Another approach is based on normalized attribute similarity scores in the range $[0, 1]$ [Ro21]. Those are aggregated into a single record similarity with a weighted average as follows

$$\text{sim}_{record} = \frac{\sum_{i=0}^{N-1} w_i \cdot sim_i}{\sum_{i=0}^{N-1} w_i} \tag{2}$$

where N is the number of attributes and the index i represents attribute i.

These techniques are equivalent to the application of weights on plaintext data in conventional record linkage as they can make use of attribute-level comparisons. Weights can be determined and applied at the linkage unit during the matching phase. When using record-level encodings, however, attribute weights must be incorporated in the encoding phase at the data owners.

Record-level Bloom filter (RBF) encodings, proposed by Durham et al. [Du14], use a sampling based approach. Initially, separate (attribute-level) Bloom filters are generated for each attribute. Based on the respective weights a proportional number of bits is sampled from each attribute-level Bloom filter to construct a record-level Bloom filter. Finally, the bits in the RBF are permuted to ensure that an attacker cannot easily reassign bits of the Bloom filters to specific attributes.

Following the *CLK-RBF* approach by Vatsalan et al. [Va14], weights can be reflected in the number of hash functions k_i that are used for each attribute i. The more hash functions are used for an attribute, the more bits in the final Bloom filter are set based on that attribute. Consequently, the influence of that attribute on the Bloom filter similarity is stronger. The number of set bit positions related to a certain attribute also depends on the (average) attribute length. Shorter values consist of fewer record features and thereby fewer bits are set. We compute k_i with the following equation to ensure that the average number of hash functions of each attribute with respect to the total number of hash functions is proportional to the relative weight of this attribute.

$$\frac{k_i \cdot n_i}{k \cdot \sum_{i=0}^{N-1} n_i} = \frac{w_i}{\sum_{i=0}^{N-1} w_i} \qquad \rightarrow \qquad k_i = \frac{w_i \cdot k \cdot \sum_{i=0}^{N-1} n_i}{n_i \cdot \sum_{i=0}^{N-1} w_i} \tag{3}$$

where w_i is the weight, n_i the average number of features, and k_i the number of hash functions for attribute i. k is the reference number of hash functions and determines the average fill rate (amount of 1-bits relative to the length l) of the Bloom filters.

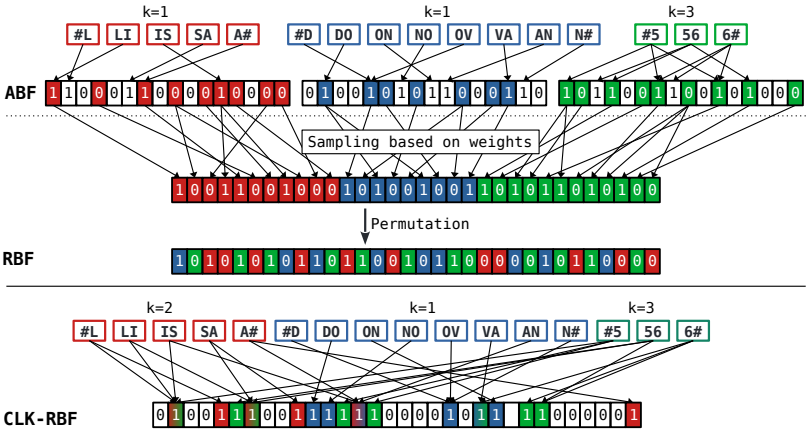

Fig. 2: Encoding of an example record using the weighted record-level techniques RBF and CLK-RBF.

4.2 Frequency-depending weight adaptation

In this section we describe methods to determine value-specific weights that reflect the relative significance of the respective attribute value based on its frequency.

Value frequencies can be incorporated in the Fellegi-Sunter approach by computing a value-dependent u-probability with $u_i = f_i/T$, where f_i is the absolute frequency of value i and T is the total number of values. Consider, for example, an attribute with three possible values 'A', 'B' and 'C' and their respective frequencies in the dataset are 100, 50 and 2 then $u_A = 100/152 \approx 0.66$ and $u_C = 2/152 \approx 0.01$. The likelihood that two random records agree on this attribute is much larger for the most frequent value 'A' than for the rarest value 'C'. Hence, the weights w_m and w_u are 0.45 and -1.77 for value 'A' and 6.10 and -3.30 for value 'C' (see Equation (1), assuming a constant $m = 0.9$).

Another approach to modify attribute-level weights is a value-dependent scaling factor S, so that $w' = S \cdot w$. This approach is independent of the method used to determine default weights. Furthermore, it is applicable to other parameters such as the number of hash functions k in CLK-RBF encodings. We therefore focus on this weight adaptation method.

Zhu et al. [Zh09] proposed a scaling factor defined as

$$S_{\text{Zhu},i} = \sqrt{\frac{T}{Q \cdot f_i}} \tag{4}$$

where T is the total number of values, Q is the number of unique values and f_i is the absolute frequency of the value i. For values that are more common than the average, S_{Zhu} is in $[0, 1)$ and for rare values the factor is larger than 1. In our previous example we would compute $S_A = \sqrt{152/(3 \cdot 100)} \approx 0.71$ and $S_C = \sqrt{152/(3 \cdot 2)} \approx 5.03$. Zhu's scaling factor has, however, two unfavorable properties: (1) The reference for $S = 1$ is fixed to the mean frequency. Since value frequency distributions, e. g., of names, typically have few very common values and many rare values, this reference can be quite low leading to $S < 1$ for many mid-common values. (2) The values of the scale factor are biased towards the lower bound. As a consequence, the scale factors are low for values that are not very common.

To address these issues, we propose an alternative scaling factor based on the inverse document frequency (idf) which is defined as $\text{idf} = \log_2 (T/f_i)$. To achieve $S = 1$ for a desired reference frequency f_{ref}, we define the scaling factor as

$$S_{\text{idf}} = 1 + \text{idf}(f_i) - \text{idf}(f_{\text{ref}}) \tag{5}$$

Moreover, we define f_{ref} as the frequency of the median attribute value, which is the value in the middle of the ordered list of values with repetition according to the respective frequency. For example, if we have a frequency distribution $[\{A, 4\}, \{B, 3\}, \{C, 1\}, \{D, 1\}, \{E, 1\}]$, then the (lower) middle of the list $[A, A, A, A, \mathbf{B}, B, B, C, D, E]$ is position 5 or value B. This results in $S_{\text{idf}} < 1$ for values that are more frequent than 3. For S_{Zhu} the reference (mean) frequency is $(4 + 3 + 1 + 1 + 1)/5 = 2$. The median-based approach results in half of the values having a scale factor of below 1 and half above 1.

In practice, the scaling factor S_{Zhu} can be unreasonably low or high. For instance, in one of the datasets used in our evaluation, we have $T = 200\,000$ records with $Q = 20\,060$ unique first names. For the most common name '*James*' ($f_i = 3401$ (1.7 ‰)) we get $S_{\text{Zhu}} = 0.05$ and for the rarest names with $f_i = 1$ we get $S_{\text{Zhu}} = 3.2$. This very large weight reduction for the name '*James*' would result in an almost complete disregard of this attribute in the classification which is not desirable. S_{idf} can even be negative for common values which makes normalization inevitable. Therefore, we normalize and restrict the scales as follows: (1) The scaling factor is normalized to the interval $[0, 2]$. We use a separate min-max normalization for scaling factors below and above 1 to ensure that this value is not modified. (2) S is restricted to a more narrow interval $[S_{\min}, S_{\max}]$, e. g., $[0.75, 1.5]$, to constrain the effect of the weight adaptation.

$$S_{\text{lowest}} = \min(S_i) \tag{6}$$

$$S_{\text{highest}} = \max(S_i) \tag{7}$$

$$S_{\text{norm},i} = \begin{cases} \frac{S_i - S_{\text{lowest}}}{1 - S_{\text{lowest}}} & \text{for } S_i < 1 \\ 1 + \frac{S_i - 1}{S_{\text{highest}} - 1} & \text{for } S_i \geq 1 \end{cases} \tag{8}$$

$$S_{\text{restricted},i} = \begin{cases} S_{\min} + S_{\text{norm},i} \cdot (1 - S_{\min}) & \text{for } S_{\text{norm},i} < 1 \\ 1 + (S_{\text{norm},i} - 1) \cdot (S_{\max} - 1) & \text{for } S_{\text{norm},i} \geq 1 \end{cases} \tag{9}$$

ALGORITHM 1: Computation of value-specific scale factors

Input: AF: Lookup table for attribute value frequencies
A_i: Attribute value i
Output: $scale_i$: Scale factor for value i

```
if A_i is in AF then
    freq_i ← GetFrequency(AF, A_i);
    scale_i ← ComputeScale(freq_i);          /* Eq. (4) (Zhu) or Eq. (5) (idf) */
    scale_i ← MinMaxNormalize(scale_i);                    /* Equation (8) */
    scale_i ← RescaleToBoundaries(scale_i);                /* Equation (9) */
else
    scale_i ← 1.0;
```

The scale factor is applied to weights at the linkage unit for ABF (see Algorithm 2) and at the data owner for RBF (see Algorithm 3) whereas in CLK-RBF the scale factor is applied to the number of hash functions k (see Algorithm 4). The weight adaptation technique based on a scaling factor requires only a few simple computations as can be seen in Algorithm 1. Therefore, it can be easily integrated into existing frameworks that already support (attribute-level) weighted Bloom filter encodings.

4.3 Weight estimation

In this section we describe how weights can be estimated and the issues that arise when applying these methods in the PPRL domain. As described in Sect. 2, a popular method to determine weights is based on the probabilistic approach of Fellegi and Sunter. The computation of the weights requires estimates of the m- and u-probabilities for the attributes. Given ground truth data, we can calculate $m = 1 - e$ where e is the error rate, i. e., the share of true duplicates with a different value for that attribute. In real-world use cases the error rate must be estimated based on expert and domain knowledge or be determined in a pre-study with a clerical review.

A naive approach to estimate attribute-specific u-probabilities is $u = 1/\#uniqueValues$, which means that the probability that two attribute values agree by chance is equal to the average relative frequency. We use a different approach which is sensitive to the frequency distribution by setting $u = \sum_{i=0}^{Q-1} p_i^2$, where Q is the number of distinct values and p_i the relative frequency of the i-th value. As an example, consider the attribute gender that can take the values '*female*', '*male*' and '*undesignated*'. The first two values are nearly equally frequent ($p = 0.48$), the last value, however, is far less common ($p = 0.04$). The estimated probability that the values of two random records agree is $1/3$ in the naive approach, but 46 % when considering the frequency distribution.

ALGORITHM 2: Linkage with attribute-level similarities and value-specific weighting

Input: **R**: Dataset with attribute-level encoded records
$\mathbf{w}_{default}$: Default attribute-specific weights
t: threshold
Output: **M**: Matching record pairs

Candidates ← *GenerateRecordPairsWithStandardBlocking*(**R**);
M ← [ ];
for *Candidate* ∈ **Candidates do**
  AP ← *GenerateAttributePairs*(*Candidate*);
  AS ← *ComputeAttributeSimilarities*(**AP**);
  w ← $\mathbf{w}_{default}$;
  for sim_i ∈ **AS do**
    if $sim_i = 1$ **then** /* Adapt only if attributes are equal */
      $scale \leftarrow GetScale(AP_i)$; /* Algorithm 1 */
      $w_i \leftarrow w_{default,i} \cdot scale$;
  $RS \leftarrow ComputeWeightedRecordSimilarity(\mathbf{w}, \mathbf{AS})$; /* Equation (2) */
  if $RS > t$ **then** **M**.*append*(*Candidate*) ;

ALGORITHM 3: Record-level Bloom filter (RBF) encoding with value-specific weighting

Input: R: Plaintext record
$\mathbf{w}_{default}$: Default attribute-specific weights
l_{RBF}, l_{ABF}: Length of the record-level / attribute-level Bloom filter
$\mathbf{k}_{ABF}$: Attribute-level number of hash functions
Output: $\mathbf{R_{BF}}$: Encoded Bloom filter record

B ← [ ]; **w** ← $\mathbf{w}_{default}$;
for $A_i \in R$ **do**
  $ABF_i \leftarrow GenerateBloomFilter(A_i, l_{ABF}, k_{ABF,i})$;
  $scale \leftarrow GetScale(A_i)$; /* Algorithm 1 */
  $w_i \leftarrow w_{default,i} \cdot scale$;
share ← *ComputeProportionalNumbersOfBits*(**w**, l_{RBF});
for ABF_i ∈ **ABF do** **B**.*append*(*SampleBits*(ABF_i, share$_i$) ;
$\mathbf{R_{BF}} \leftarrow Permute(\mathbf{B})$;

ALGORITHM 4: CLK-RBF encoding with value-specific weighting

Input: R: Plaintext record
$\mathbf{k_{default}}$: Default attribute-specific number of hash functions
l: Length of the record-level Bloom filter
Output: $\mathbf{R_{BF}}$: Encoded Bloom filter record

$\mathbf{R_{BF}}$ = *InitializeEmptyBloomFilter*(l);
for $A_i \in R$ **do**
  $scale \leftarrow GetScale(A_i)$; /* Algorithm 1 */
  $k_i \leftarrow k_{default,i} \cdot scale$;
  $\mathbf{BF_i} \leftarrow GenerateBloomFilter(A_i, l, k_i)$;
  $\mathbf{R_{BF}} \leftarrow \mathbf{R_{BF}} \cup \mathbf{BF_i}$;

The application of this approach in the PPRL domain comes with additional challenges, for attribute-specific weights as well as for frequency-dependent value-specific weights. The estimation of u-probabilities and the computation of weight scale factors are based on frequency distributions which are not readily available for the datasets to be linked as this information is considered sensitive. We discuss this restriction in Sect. 4.4.

Furthermore, in PPRL with record-level encodings, weights cannot be chosen depending on the agreement/disagreement of attributes. This is because weights must be applied at the data owners where the comparison result is not known yet. Therefore, the two weights of the Fellegi-Sunter model have to be combined into a single weight. Durham et al. [Du14] proposed the range $w = w_m - w_u$. The combined weight, however, can be dominated by w_m if the attributes have a large variety of values which is typically the case for names. As a consequence, we normalize w_m and w_u with respect to the maximum/minimum value across all attributes.

$$w_{mi,norm} = \frac{w_{mi}}{max(w_m)} \tag{10}$$

$$w_{ui,norm} = \frac{w_{ui}}{min(w_u)} \tag{11}$$

$$w_{i,norm} = max(w) \cdot (w_{mi,norm} - w_{ui,norm}) \tag{12}$$

4.4 Limited frequency information

Accurate global frequency distributions of attributes across all linked datasets are not readily available in the PPRL context. This information is considered sensitive as it could be used to perform frequency attacks on Bloom filter encodings. In the following, we discuss possible solutions to deal with this limitation.

Frequency distributions can be gathered from an external source on similar datasets, e. g., statistical data from a census of the same geographical region or be computed for the data to be linked. While the first approach is especially useful for smaller datasets where the calculated value counts may not represent the real-world frequency distribution well, the latter ensures that the used frequencies correspond to the actual properties of the dataset.

Each data owner could determine its own source-specific frequency distribution and compute weights based on it. This will result in different weights for identical values. We discuss these effects in Sect. 4.5.

Data owners cannot exchange the complete frequency information as this would leak information on rare values. However, the data owners might be willing and allowed to exchange and combine the relative frequencies of their most common values. While this can increase the linkage quality, it does not affect the privacy as the linkage unit does not learn this information. For S_{idf} a different reference frequency must be used as the median cannot be determined for an incomplete frequency distribution. We therefore propose using the

least frequent value in the list of most common attribute values as the reference. The scaling factor for this value as well as values not in the list is 1. For values on the list, it is below 1.

The limitation of frequency-dependent weight adjustments to common values will likely lead to smaller effects on the linkage result. Additionally, the weight application can be restricted to certain attributes, e. g. first and last name, because frequency information on other attributes is missing. However, it still could be beneficial with respect to the precision. We experimentally evaluate this effect in Sect. 5.

4.5 Effects of attribute differences in duplicates

Real-world data often contains typographical errors. Besides, names can have natural variations, for instance, the German last name 'Schmidt' with its variants '*Schmid*' or '*Schmitt*'. These varying values occur with different frequencies which leads to different frequency-dependent weights. The consequences of varying weights depend on the weight application technique that is used.

For the RBF approach a single different attribute weight changes the proportions of the weights and hence the sampling rates for all other attribute-level Bloom filters. If these Bloom filters have equal fill rates, the fill rate of the RBF does not change for different weights.

Using CLK-RBF with weight adaption, as described in Algorithm 4, the scaling factor of an attribute i is applied by changing the number of hash functions k_i for that attribute only. This does not affect the number of hash functions for the other attributes. Nevertheless, the fill rate of the final Bloom filter is changed as the total number of hash functions is modified.

In Tab. 3, we illustrate the effect of changing a single weight using the two encoding methods. Based on the default weights and the average number of features per attribute, we compute the sampling rates for RBF and the number of hash functions k for the CLK-RBF approach for an example record a, that has $S = 1$ for all attributes. Record b has the same last name and year of birth, but a different and very common first name. Therefore, we set the scaling factor $S_b(\text{FN}) = 0.5$. The sampling rates for all attributes of record b are changed due to the decreased sum of all weights. For CLK-RBF, however, only $k_b(\text{FN})$ is adapted. As a consequence, the generated Bloom filter encodings based on the RBF method are more affected by weight variations compared to the CLK-RBF approach.

The same applies if we compute a different scaling factor $S'_b(\text{FN}) = 0.67$ based on a different frequency information, e. g., when using source-specific frequency distributions (as described in the previous section) where in each distribution the name '*Lisa*' occurs often, but with different relative frequencies. If two sources encode the same record b based on their respective scaling factors S_b and S'_b, the resulting Bloom filters are different. In contrast, using the CLK-RBF approach, this affects only a few hash functions and thus the difference between the Bloom filters is lower.

Tab. 3: Example of a variation of a single weight on record-level encodings ($l = 1024$) based on RBF ($l_{ABF} = 256$) and CLK-RBF ($k = 20$).

	First name	**Last name**	**Year of birth**
Record *a*	LISE	DONOVAN	1956
Record *b*	LISA	DONOVAN	1956
n (avg. #features)	7	8	5
$\mathbf{w}_{\mathbf{default}}$	12	11	14
$\mathbf{S}_b$	0.5	1	1
$\mathbf{S}'_b$	0.67	1	1
RBF			
$\mathbf{w}_a (= \mathbf{w}_{\mathbf{default}})$	12	11	14
$\mathbf{w}_b$	6	11	14
$\mathbf{w}'_b$	8	11	14
% of sampling for *a* ($\mathbf{w}_a$)	$^{12}/_{37} = 32\,\%$	$^{11}/_{37} = 30\,\%$	$^{14}/_{37} = 38\,\%$
% of sampling for *b* ($\mathbf{w}_b$)	$^{6}/_{31} = 19\,\%$	$^{11}/_{31} = 36\,\%$	$^{14}/_{31} = 45\,\%$
% of sampling for *b* ($\mathbf{w}'_b$)	$^{8}/_{33} = 24\,\%$	$^{11}/_{33} = 33\,\%$	$^{14}/_{33} = 43\,\%$
CLK-RBF			
$\mathbf{k}_a$ **(for S = 1)**	18	15	30
$\mathbf{k}_b$ **(for $\mathbf{S}_b$)**	9	15	30
$\mathbf{k}'_b$ **(for $\mathbf{S}'_b$)**	12	15	30

To avoid different weights for attribute variations, such as '*Lisa*' and '*Lise*', we consider the use of frequency distributions based on generalizations of the plaintext values, e. g., using the Soundex phonetic encoding function [OR18]. Consequently, the weight for each value is computed based on the frequency of its generalized value (Soundex code). For example, the Soundex code for both '*Lisa*' and '*Lise*' is 'L200', and thus, the same weights are computed, although the values might have different frequencies.

4.6 Handling missing values

Apart from erroneous attributes, missing values often occur in real-world datasets and a strategy is needed to handle them. When working with plaintext or attribute-level encodings, the linkage unit can detect missing values. Multiple strategies can be used, e. g., ignoring the attribute in the similarity score aggregation or setting its similarity score to 0. When working with record-level encodings the linkage unit cannot detect missing values. If a data owner detects a missing value during the encoding phase the respective weight could be redistributed to the other attributes. However, as the missingness is source-specific, a true match with that value set would be encoded with different weights for all attributes. This would in turn lead to differences in the resulting Bloom filters and thus likely to misclassifications. We therefore do not adapt weights of missing values and simply treat them as empty attributes.

5 Evaluation

We evaluate the methods described in the previous section with respect to the linkage quality, while focusing on the following aspects: (1) Quantification of the effects of frequency-dependent weight adaptation. (2) Comparison of weight application approaches in PPRL. (3) Investigation of the effects of limited information on frequency distributions.

5.1 Datasets

To study the effects on real-world data, we use a dataset based on the North Carolina Voter Registration (NCVR) database (`https://www.ncsbe.gov/`) provided by Panse et al. [Pa21]. This dataset contains over 120 million historic voter records with person-related attributes such as first name (FN), middle name (MN), last name (LN), year of birth (YOB), place of birth (POB), city, ZIP code and sex. From that dataset we extracted a subset, tagged **F**, by

(1) Sampling 80 000 individual records (singletons) contained in the snapshot from '2021-01-01' into set A_S and B_S each, ensuring that $A_S \cap B_S = \emptyset$.

(2) Sampling 20 000 pairs of records a, b (duplicates) into sets A_D and B_D respectively, where a is from any snapshot between '2008-01-01' (inclusive) and '2021-01-01' (exclusive), and b is from snapshot '2021-01-01'. Moreover, $\forall a, b : (\text{YOB}(a) = \text{YOB}(b)) \wedge \exists \text{attr} \in \{\text{FN}, \text{MN}, \text{LN}, \text{POB}, \text{SEX}\} : \text{attr}(a) \neq \text{attr}(b)$.

(3) Constructing the final subsets as $\text{F}_A = A_S \cup A_D$ and $\text{F}_B = B_S \cup B_D$ respectively.

Based on **F** ('*Full*') we derive another dataset, tagged **R** ('*Reduced*'), where we removed the attributes middle name and place of birth, thus, making the dataset more challenging to match, see also Tab. 4.

Tab. 4: Description of used datasets.

Name	|A|	|B|	|A ∩ B|	Attributes
F	100k	100k	20k	FN, MN, LN, YOB, POB, CITY, ZIP
R	100k	100k	20k	FN, LN, YOB, CITY, ZIP

For our evaluations with external statistical information we use frequencies of first and last names from the 1990 US Census.[2]

5.2 Encoding

We set a fixed length of $l = 256$ for the attribute-level Bloom filter. The plaintext attributes are preprocessed by removing leading and trailing whitespace, conversion to lowercase

[2] `https://www.census.gov/topics/population/genealogy/data/1990_census.html`

and removal of diacritics before being split into overlapping bigrams using padding. The number of hash functions k_i is selected based on the average length of each attribute to achieve a unified average fill rate of the respective Bloom filter of approximately 40 %. RBF encodings are based on the same ABF parameters. We set the length of the record-level Bloom filter to $l = 1024$. For the computation of attribute-specific k_i in the CLK-RBF encoding we use the reference number of hash function $k = 12$ (**F**) and $k = 15$ (**R**) which results in an average Bloom filter fill rate of approximately 40 %.

Tab. 5: Attribute properties (availability, average length $\varnothing l$, m- and u-probability, normalized weight) and derived encoding parameters for Attribute-level Bloom filter (number of of hash functions k), Record-Level Bloom filter (share of each attribute) and CLK-RBF (number of hash functions k.)

Attr.	Properties					ABF	RBF		CLK-RBF	
	Avail.	∅l	m-prob	u-prob	w_{norm}	k	%(F)	%(R)	k(F)	k(R)
FN	99.99 %	6	0.9300	0.0027	12.83	18	19.95	24.07	15	18
MN	92.16 %	5.1	0.4380	0.0037	7.43	21	11.55	–	12	–
LN	100 %	6.4	0.7187	0.0010	11.14	17	17.32	20.90	11	15
YOB	100 %	4	0.9900	0.0135	14.44	26	22.45	27.09	24	29
CITY	99.97 %	8.9	0.6507	0.0193	6.63	13	10.31	12.44	6	7
ZIP	99.89 %	5	0.5318	0.0031	8.27	21	12.86	15.51	11	14
POB	79.13 %	2	0.6436	0.1513	3.57	43	5.55	–	10	–

5.3 Matching

In this work, we focus on the evaluation of comparison and classification rather than techniques to improve scalability. However, to run the experiments in a reasonable time, we use standard blocking to reduce the number of comparisons that need to be computed. To ensure the comparability of the results, we use the same blocking keys independent of the encoding method. For each record we generate blocking keys at the data owners based on the plaintext attribute combinations FN+YOB, LN+YOB and Soundex(FN)+Soundex(LN) and encode them using a cryptographic one-way hash function. These hashed blocking keys are transmitted together with the encoded records to enable blocking at the linkage unit. Additionally, we add a blocking key of the global record id that is unique for each duplicate pair based on the ground truth. This blocking key is used to ensure that no true duplicates are excluded from the comparison.

For attribute-level encodings, attribute pairs with one or both values missing in essential attributes (first name, last name and year of birth) are assigned a similarity score of 0. Other missing attributes are ignored in the weighted average aggregation.

We conduct additional experiments where a post-processing routine on the set of matches is applied, using a symmetric best match strategy (Max1-both) to restrict the result to 1:1 links. In many practical use cases this is reasonable when the sources can be considered duplicate-free and therefore each record of a database has at most one duplicate in the other database [Fr18].

5.4 Evaluation measures

We use the standard measures recall, precision and F1-score to evaluate linkage quality. Recall measures the proportion of found true matches from all true matches. Precision measures the proportion of found true matches from all found matches. The F1-score is the harmonic mean of these two measures.

$$\text{Rec.} = \frac{\#\text{TruePos.}}{\#\text{TruePos.} + \#\text{FalseNeg.}}, \qquad \text{Prec.} = \frac{\#\text{TruePos.}}{\#\text{TruePos.} + \#\text{FalsePos.}}, \qquad \text{F1} = \frac{2 \cdot \text{Rec.} \cdot \text{Prec.}}{\text{Rec.} + \text{Prec.}}$$

We evaluate these measures for similarity thresholds t in the range of $[0.7, 1.0]$ in steps of 0.01. In practical record linkage, however, ground truth data is not available and the used thresholds are rarely optimal. For a high linkage quality in real-world applications the results should be stable for a broader range of thresholds. We therefore introduce a loss measure L_M^d for the linkage quality that describes the maximal loss of measure M in the threshold range $[t_{\text{opt}} - d, t_{\text{opt}} + d]$. Furthermore, we report the area under the curve (AUC) of precision-over-recall as a threshold-independent measure.

5.5 Results

The threshold-dependent quality measures are reported for the classification thresholds t_{opt} that are optimal for this linkage configuration and used dataset with respect to the F1-score. First, we evaluate different weight adaptation methods based on the scaling factors S_{Zhu} and S_{idf} (see Tab. 6). We use ABF encodings as described in Algorithm 2 and the full frequency distribution computed for the respective datasets and test multiple scale factor intervals.

Tab. 6: Comparison of weight adaptation methods on Attribute-level Bloom filter.

DS	S	S_{min}	S_{max}	AUC	t_{opt}	Rec.	Pre.	F1	$L_{\text{F1}}^{0.01}$	$L_{\text{F1}}^{0.03}$	$L_{\text{F1}}^{0.05}$
R	–	–	–	0.841	0.85	0.786	0.787	0.787	0.029	0.089	0.180
	Zhu	0.5	2.0	0.869	0.82	0.813	0.813	0.813	0.032	0.094	0.189
		0.5	1.5	0.867	0.82	0.810	0.812	0.811	0.031	0.093	0.188
		0.75	1.5	0.856	0.84	0.786	0.817	0.801	0.022	0.080	0.160
	idf	0.5	2.0	**0.884**	0.84	0.830	0.839	0.835	0.030	0.091	0.184
		0.5	1.5	0.877	0.84	0.817	0.839	0.828	0.024	0.083	0.174
		0.75	1.5	0.866	0.85	0.795	0.837	0.815	0.019	0.087	0.156
F	–	–	–	0.918	0.83	0.809	0.914	0.859	0.008	0.050	0.151
	Zhu	0.5	2.0	0.933	0.79	0.853	0.904	0.878	0.019	0.080	0.210
		0.5	1.5	0.932	0.79	0.849	0.904	0.875	0.017	0.078	0.208
		0.75	1.5	0.927	0.81	0.834	0.906	0.868	0.014	0.066	0.185
	idf	0.5	2.0	**0.940**	0.81	0.863	0.913	0.887	0.017	0.072	0.191
		0.5	1.5	0.937	0.81	0.851	0.916	0.882	0.014	0.064	0.180
		0.75	1.5	0.931	0.82	0.838	0.914	0.875	0.012	0.058	0.167

All weight adaption configurations improve the linkage quality compared to static weights. However, the S_{idf}-based approaches generally show a higher rise of the AUC than the S_{Zhu}-based with a maximum improvement by 0.043 (**R**) and 0.022 (**F**) each with a constraint

Tab. 7: Comparison of averaging, attribute-specific and value-specific weighting (S_{idf} [0.5,2]) for different encoding methods.

DS	Enc.	Weighting	AUC	t_{opt}	Rec.	Pre.	F1	$L_{F1}^{0.01}$	$L_{F1}^{0.03}$	$L_{F1}^{0.05}$
R	ABF	–	0.777	0.86	0.609	0.803	0.693	0.015	0.045	0.052
		Attribute-specific	0.841	0.85	0.786	0.787	0.787	0.029	0.089	0.180
		Value-specific	0.884	0.84	0.830	0.839	0.835	0.030	0.091	0.184
	CLK-RBF	–	0.749	0.85	0.580	0.806	0.674	0.014	0.047	0.078
		Attribute-specific	0.837	0.83	0.782	0.767	0.775	0.028	0.093	0.221
		Value-specific	**0.875**	0.81	0.804	0.849	0.826	0.015	0.070	0.189
	RBF	–	0.787	0.86	0.605	0.828	0.699	0.007	0.024	0.043
		Attribute-specific	0.845	0.85	0.770	0.806	0.788	0.015	0.074	0.179
		Value-specific	0.675	0.76	0.373	0.973	0.539	0.001	0.001	0.001
F	ABF	–	0.900	0.80	0.808	0.857	0.832	0.009	0.046	0.134
		Attribute-specific	0.918	0.83	0.809	0.914	0.859	0.008	0.050	0.151
		Value-specific	0.940	0.81	0.863	0.913	0.887	0.017	0.072	0.191
	CLK-RBF	–	0.845	0.77	0.714	0.804	0.756	0.009	0.034	0.067
		Attribute-specific	0.917	0.80	0.824	0.903	0.861	0.019	0.079	0.192
		Value-specific	**0.938**	0.77	0.866	0.917	0.891	0.019	0.072	0.185
	RBF	–	0.874	0.77	0.793	0.815	0.804	0.020	0.120	0.291
		Attribute-specific	0.920	0.81	0.834	0.905	0.868	0.024	0.099	0.248
		Value-specific	0.801	0.77	0.677	0.958	0.793	0.005	0.023	0.054

interval of $[0.5, 2.0]$. The optimal F1-scores show similar increases by 0.048 (**R**) and 0.028 (**F**). Therefore, we use that weight adaption strategy for the following experiments.

We compare the results of the value-specific weight adaptation strategy for the encoding techniques ABF, CLK-RBF and RBF (see Tab. 7). We report two baselines, with and without attribute-specific weights. The latter is implemented by using the arithmetic mean of the attribute similarity scores (ABF), equal number of hash functions k for all attributes (CLK-RBF) and by sampling equal shares from each attribute (RBF). The value-specific weighting scheme achieves AUC improvements for CLK-RBF comparable to those of the attribute-level application despite the missing restriction of weight adjustments to equal attributes: +0.038 (**R**) and +0.021 (**F**) with respect to the attribute-specific weight baseline and +0.126 (**R**) and +0.093 (**F**) to the averaging baseline. However, with the sampling-based approach (RBF) AUC decreases for value-specific weighting. As we discussed in Sect. 4.5, this is because even a single different weight, e. g., due to a typo, leads to considerably dissimilar Bloom filters. Even with a low threshold of 0.76 the recall is as low as 0.373 (for **R**). Thus, we subsequently focus on the CLK-RBF encoding.

In general, we observe that weight adaptation methods lead to lowered optimal thresholds with increases in recall as well as in precision. While the first is expected when lowering the threshold, the rise of precision suggests that non-match candidates with comparatively high similarity due to common values are less often wrongly classified as matches as these attributes are weighted lower. Moreover, we note that the improved results are also equally or more stable regarding the threshold selection. For **R** with an increase of the F1-score by 0.051, L_{F1}^{d} is reduced from 0.028 to 0.015 in $d = 0.01$ and decreases by 0.023 in $d = 0.03$,

which indicates that a higher linkage quality can be achieved in real-world applications with non-optimal threshold selection (see also Fig. 3).

As explained above, the restriction of the linkage result to 1:1 links is reasonable in some applications and enhances the linkage quality. In order to study whether weight adaption further improves the results, we evaluate the weighting methods for CLK-RBF where the links have been post-processed before the linkage quality assessment (see Tab. 8). The results show increases of AUC, +0.026 (**R**) and +0.011 (**F**), indicating that the weight adjustment technique is beneficial under these conditions as well.

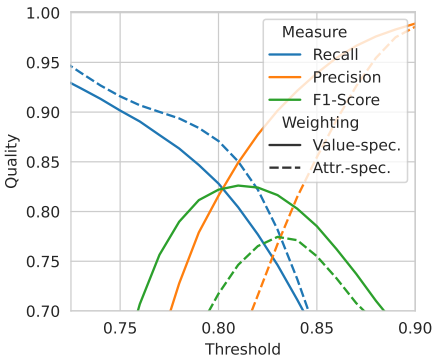

Fig. 3: Comparison of quality measures for attribute- and value-specific weighting on CLK-RBF for **R**.

Tab. 8: Comparison of attribute-specific and value-specific weighting (S_{idf} [0.5,2]) for CLK-RBF where the found matches have been restricted to 1:1 links in a postprocessing step (PP).

DS	PP	Weighting	AUC	t_{opt}	Rec.	Pre.	F1	$L_{F1}^{0.01}$	$L_{F1}^{0.03}$	$L_{F1}^{0.05}$
R	yes	Attribute-specific	0.867	0.81	0.833	0.799	0.816	0.016	0.058	0.117
		Value-specific	0.893	0.80	0.820	0.876	0.847	0.007	0.038	0.093
F		Attribute-specific	0.936	0.79	0.848	0.915	0.880	0.007	0.032	0.077
		Value-specific	0.947	0.76	0.881	0.927	0.904	0.012	0.045	0.101

Finally, we study how variations of available frequency information affect the results (see Tab. 9). Again, we report two baselines, with and without value-specific weight adaption, to allow for a comparison with the current state-of-the-art of attribute-specific weights and with value-specific weights under ideal conditions. The results with frequency distributions based on Soundex encodings instead of plaintext values show lower linkage quality, because weights of rare values can be decreased in this setting if these values share the encoding with a common value. Using source-specific frequency distributions the results are almost equal to those with access to the overall frequency information as the distributions are similar. When linking smaller datasets, the distributions will have larger differences, in particular for rare values. We therefore conduct additional experiments where we limit the available frequency information to the most frequent values, as described in Sect. 4.4. The results show that even with a limitation on the 20 most frequent values AUC increases by 0.027 (**R**) and 0.014 (**F**) compared to attribute-specific weights. However, when using external statistical data on the 100 most frequent first and last names only, the linkage quality improvements are comparatively low. The inclusion of information on the frequencies of additional (geographical) attributes could potentially improve the results.

Generally, we see that the quality improvements for **F** are lower than for **R** because the inclusion of information on value frequencies is more relevant in linkage scenarios where fewer attributes are available.

Tab. 9: Comparison of weighting methods with limited frequency information based on CLK-RBF.

DS	Weighting limitation	AUC	t_{opt}	Rec.	Pre.	F1	$L_{F1}^{0.01}$	$L_{F1}^{0.03}$	$L_{F1}^{0.05}$
R	Attribute-specific	0.837	0.83	0.782	0.767	0.775	0.028	0.093	0.221
	Value-specific	0.875	0.81	0.804	0.849	0.826	0.015	0.070	0.189
	⊢ Soundex-based	0.858	0.82	0.788	0.832	0.809	0.016	0.075	0.207
	⊢ Source-specific	0.875	0.81	0.791	0.864	0.826	0.009	0.058	0.171
	⊢ Top 10	0.858	0.83	0.789	0.836	0.811	0.014	0.077	0.218
	⊢ Top 20	0.864	0.82	0.825	0.813	0.819	0.032	0.117	0.286
	⌞ Top 100 Names (Census)	0.851	0.82	0.802	0.769	0.785	0.025	0.082	0.195
F	Attribute-specific	0.917	0.80	0.824	0.903	0.861	0.019	0.079	0.192
	Value-specific	0.938	0.77	0.866	0.917	0.891	0.019	0.072	0.185
	⊢ Soundex-based	0.930	0.78	0.852	0.915	0.882	0.016	0.073	0.197
	⊢ Source-specific	0.938	0.77	0.858	0.924	0.890	0.013	0.062	0.168
	⊢ Top 10	0.928	0.79	0.845	0.913	0.877	0.018	0.080	0.212
	⊢ Top 20	0.931	0.79	0.844	0.924	0.882	0.013	0.065	0.183
	⌞ Top 100 Names (Census)	0.929	0.79	0.841	0.906	0.872	0.016	0.067	0.168

6 Conclusion

Privacy-preserving record linkage enables the integration of sensitive data and thus, its comprehensive analysis. A main challenge is the classification of record pairs as match or non-match based on computed similarities between quasi-identifying attributes of these records. Several studies focus on attribute- and value-specific weighting methods for plaintext data. Nevertheless, only few works adapt these methods in the context of PPRL.

In this work, we apply existing record-level Bloom filter encodings and combine them with frequency-dependent weight adaptation approaches. We extensively evaluate our adapted encoding schemes and compare them with attribute- and record-level Bloom filter encodings. The results show that the modified CLK-RBF encoding outperforms the existing (record-level) methods and achieves comparable results to attribute-level weight application techniques regarding linkage quality and robustness. However, the latter require attribute-level encodings, which are susceptible to cryptanalysis and thus not secure in practical applications. While the weight adaptation disturbs certain frequent bit patterns in the record-level Bloom filters due to the reduced number of hash functions for frequent values, it introduces other frequent patterns in the encoded data as lower weights systematically result in lower fill rates.

In future work, we therefore plan to integrate Bloom filter hardening techniques in our approach to further improve the resistance against cryptanalysis. Furthermore, we will study approaches to estimate *attribute-specific* weights with limited information on frequency distributions and error rates.

Acknowledgements. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany (BMBF) and by the Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus for ScaDS.AI and by the BMBF for the SMITH consortium, grant number 01ZZ1803A.

References

[Br17] Brown, A. P.; Randall, S. M.; Ferrante, A. M.; Semmens, J. B.; Boyd, J. H.: Estimating parameters for probabilistic linkage of privacy-preserved datasets. BMC Medical Research Methodology 17/95, pp. 1–10, 2017, DOI: 10.1186/s12874-017-0370-0.

[Ch12] Christen, P.: Data Matching, Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer, 2012.

[CRS20] Christen, P.; Ranbaduge, T.; Schnell, R.: Linking Sensitive Data, Methods and Techniques for Practical Privacy-Preserving Information Sharing. Springer, 2020.

[Di45] Dice, L. R.: Measures of the Amount of Ecologic Association Between Species. Ecology 26/3, pp. 297–302, 1945, ISSN: 00129658, DOI: 10.2307/1932409.

[Du14] Durham, E. A.; Kantarcioglu, M.; Xue, Y.; Toth, C.; Kuzu, M.; Malin, B.: Composite Bloom Filters for Secure Record Linkage. IEEE Transactions on Knowledge and Data Engineering 26/12, pp. 2956–2968, Dec. 2014, DOI: 10.1109/TKDE.2013.91.

[Fr18] Franke, M.; Sehili, Z.; Gladbach, M.; Rahm, E.: Post-processing Methods for High Quality Privacy-Preserving Record Linkage. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology 2018. 2018, DOI: 10.1007/978-3-030-00305-0_19.

[Fr21] Franke, M.; Sehili, Z.; Rohde, F.; Rahm, E.: Evaluation of Hardening Techniques for Privacy-Preserving Record Linkage. In: 24th International Conference on Extending Database Technology (EDBT). Pp. 289–300, 2021, DOI: 10.5441/002/edbt.2021.26.

[FS69] Fellegi, I. P.; Sunter, A. B.: A Theory for Record Linkage. Journal of the American Statistical Association 64/328, pp. 1183–1210, 1969, DOI: 10.1080/01621459.1969.10501049.

[FSR18] Franke, M.; Sehili, Z.; Rahm, E.: Parallel Privacy-Preserving Record Linkage using LSH-based blocking. International Conference on Internet of Things, Big Data and Security (IoTBDS)/, 2018, DOI: 10.1007/978-3-030-00305-0_19.

[Gi10] Giersiepen, K.; Bachteler, T.; Gramlich, T.; Reiher, J.; Schubert, B.; Novopashenny, I.; Schnell, R.: Performance of record linkage for cancer registry data linked with mammography screening data. Bundesgesundheitsblatt, Gesundheitsforschung, Gesundheitsschutz 53/7, pp. 740–747, 2010, ISSN: 1436-9990, DOI: 10.1007/s00103-010-1084-1.

[Gk21] Gkoulalas-Divanis, A.; Vatsalan, D.; Karapiperis, D.; Kantarcioglu, M.: Modern Privacy-Preserving Record Linkage Techniques: An Overview. IEEE Transactions on Information Forensics and Security 16/, pp. 4966–4987, 2021, DOI: 10.1109/TIFS.2021.3114026.

[HSW07] Herzog, T. N.; Scheuren, F. J.; Winkler, W. E.: Data Quality and Record Linkage Techniques. Springer, 2007.

[OR18] Odell, M.; Russell, R.: The soundex coding system. US Patents 1261167/, 1918.

[Pa21] Panse, F.; Düjon, A.; Wingerath, W.; Wollmer, B.: Generating Realistic Test Datasets for Duplicate Detection at Scale Using Historical Voter Data. In: EDBT. 2021, DOI: 10.5441/002/edbt.2021.67.

[RC18] Ranbaduge, T.; Christen, P.: Privacy-Preserving Temporal Record Linkage. In: IEEE International Conference on Data Mining (ICDM). Pp. 377–386, 2018, DOI: 10.1109/ICDM.2018.00053.

[Ro21] Rohde, F.; Franke, M.; Sehili, Z.; Lablans, M.; Rahm, E.: Optimization of the Mainzelliste software for fast privacy-preserving record linkage. Journal of Translational Medicine 19/33, 2021, DOI: 10.1186/s12967-020-02678-1.

[SBR09] Schnell, R.; Bachteler, T.; Reiher, J.: Privacy-preserving record linkage using Bloom filters. BMC Med. Inf. & Decision Making 9/41, 2009, DOI: 10.1186/1472-6947-9-41.

[Va14] Vatsalan, D.; Christen, P.; O'Keefe, C. M.; Verykios, V. S.: An Evaluation Framework for Privacy-Preserving Record Linkage. Journal of Privacy and Confidentiality 6/1, pp. 35–75, 2014, DOI: 10.1016/j.chemosphere.2016.07.068.

[VCV13] Vatsalan, D.; Christen, P.; Verykios, V. S.: A Taxonomy of Privacy-Preserving Record Linkage Techniques. Information Systems 38/6, pp. 946–969, 2013, DOI: 10.1016/j.is.2012.11.005.

[Vi22] Vidanage, A.; Ranbaduge, T.; Christen, P.; Schnell, R.: A Taxonomy of Attacks on Privacy-Preserving Record Linkage. Journal of Privacy and Confidentiality 12/1, 2022, DOI: 10.29012/jpc.764.

[WT91] Winkler, W. E.; Thibaudeau, Y.: An application of the Fellegi-Sunter model of Record Linkage to the 1990 U.S. decennial census, Tech. Rep. RR1991/09, Washington, DC: US Bureau of the Census, 1991.

[Zh09] Zhu, V. J.; Overhage, M. J.; Egg, J.; Downs, S. M.; Grannis, S. J.: An Empiric Modification to the Probabilistic Record Linkage Algorithm Using Frequency-Based Weight Scaling. Journal of the American Medical Informatics Association 16/5, pp. 738–745, 2009, DOI: 10.1197/jamia.M3186.