Privacy-Preserving Record Linkage

Rob Hall and Stephen E. Fienberg

Department of Statistics, Machine Learning Department, and Cylab,
Carnegie Mellon University, Pittsburgh PA 15213, USA
rjhall@cs.cmu.edu, fienberg@stat.cmu.edu

Abstract. Record linkage has a long tradition in both the statistical and the computer science literature. We survey current approaches to the record linkage problem in a privacy-aware setting and contrast these with the more traditional literature. We also identify several important open questions that pertain to private record linkage from different perspectives.

1 Introduction

Record linkage is an historically important statistical problem arising when data about some population of individuals, is spread over several files. Most of the literature focuses on the two file setting. The record linkage goal is to determine whether a record from one file corresponds to a record of a second file, in the sense that the records describe the same individual. Winkler and others describe application areas, computational techniques and statistical underpinnings in detail in [19, 2, 38, 39]. The typical purposes of record linkage are:

- data integration, e.g, to create a public use file that will allow others to analyze the integrated data.
- as an intermediate step in performing a computation on the integrated data.

The overarching goal of privacy-preserving datamining (PPDM) [37] is to perform "data mining" computations on a set of data, in a manner that prevents both the computation, and the output of the computation from revealing "too much" sensitive information about the units represented in the data. Our goal in this paper is to detail recent advances at the intersection of record linkage and PPDM, largely as a followup to an earlier survey by Winkler [39, 40]. Whereas Winkler assumed that all the data files were accessible to the party running the computation, in our setting we remove this assumption. Instead, depending on the setting and the problem at hand, we are interested in access to files that may be somehow restricted, or not available at all.

Record linkage already plays a large role as a building block for privacy preserving statistical analysis. For example, numerous papers already tacitly assume that the files that are input to their procedures are a-priori matched, in the sense that the correspondence between the units is known [24, 23, 22, 12, 13]. We describe some key challenges at the interface of record linkage and PPDM

and show the steps various authors have taken to address them. We overview the basic record linkage approach and the secure multiparty computation literature with the intent of demonstrating some common failure modes of so called privacy preserving schemes. Then we survey the recent literature on privacy preserving schemes for performing record linkage, and conclude by outlining what we see are the key unsolved challenges in this area.

2 Record Linkage Overview

We begin by providing an overview of the record linkage problem in a non-private setting. We see the traditional approaches as being composed of a series of different steps, which we explain in turn. We then give these steps the privacy-preserving treatment in section 5. Currently, the literature on record linkage involving two files is fairly mature [39, 19], whereas the problem of linking many files has only begun to be studied (see for example the discussions in the context of merging files for the purposes of multiple capture-recapture [14, 19, 32]). Therefore in this paper we focus on the problem of record linkage between two files.

2.1 Problem Definition

Suppose there are two data files A and B, each of which contains possibly different numbers of records, say $a_i,\ (i = 1 \dots n)$ are the records belonging to file A and likewise $b_j,\ (j = 1 \dots m)$ are the records in B. The records are in essence vectors in which each component is a "field" or an attribute of the record, and we may regard the records as being the elements of the product space of the fields. For the purpose of this exposition we suppose that the fields in the two files are the same (or otherwise somehow the data has been cleaned ahead of time). When this is not the case then the problem is called "schema matching" —see [33] for a treatment of this topic. Suppose that there are p fields that are common to the files. We denote by a_i^k the k^{th} field of record a_i and likewise for B. In the database terminology, records correspond to rows of a file, whereas the fields correspond to columns. The goal of record linkage is to determine the pairs of records (a_i, b_j) corresponding to the same underlying individual.

Fellegi and Sunter formally studied this problem in their seminal paper [11]. They described an approach that partitioned the cartesian product of the files $A \times B$ into three disjoint sets: M the set of "matches", U the set of "non-matches", and C a set which requires human intervention in order to classify. The presence of this latter set is due to ambiguity in the data which is hard or impossible for an automated procedure to solve. For example, several people with a common first initial and last name may inhabit the same house, and so further data may be required to determine whether or not two records correspond to the same individual of such a household. The Felegi-Sunter approach [11] aims to minimize the cardinality of C, subject to a user-specified upper bound on the error rates in M and U. There are several modifications of this approach, a number of which are described in [19].

2.2 Computing Similarity of Record Pairs

In essence, most modern statistical record linkage techniques build on the Fellegi-Sunter idea and follow a common pattern. In a first stage, the cartesian product $A \times B$ is preprocessed and cleaned. Then some "similarity function" is applied to each element in the resulting file. Historically, the functions were indicators of whether corresponding fields of the records matched or not, i.e., whether their values for a particular fields were identical. These binary flags are referred to as the "match variables." Let $m_{i,j}$ be the vector of match variable corresponding to the pair (a_i, b_j). We may have:

$$m_{i,j} \in \{0,1\}^p,\ m_{i,j}^k = \mathbf{1}\{a_i^k = b_i^k\}$$

where we use $\mathbf{1}\{\cdot\}$ to mean the function that takes value 1 when the predicate in the braces is true, and 0 otherwise. In principle, there could be more match variables than fields, as multiple different similarity functions could be applied to different pairs of fields. For simplicity we omit a discussion of this variation here. The alternative to exact match indicators is to compute a distance function for the individual fields [8]. When fields are numeric then perhaps absolute or euclidian difference is appropriate. When fields are strings such as names and addresses, then string edit distances [21, 2] are useful. Such distance measures may be thresholded, i.e., reduced to binary match variables where the flag is "on" whenever the distance falls below some cutoff. In this case, we may have:

$$m_{i,j} \in \{0,1\}^p,\ m_{i,j}^k = \mathbf{1}\{d^k(a_i^k, b_i^k) < \tau^k\}$$

where $d^k(\cdot,\cdot)$ is the appropriate distance function for field k, and τ_k is some parameter that determines the thresholding. After this first step, there are $n \times m$ sets of match variables, corresponding to the pairs of elements in the product of the files. The match variables are either binary or real numbers depending on what kinds of similarity functions that were applied.

2.3 Parameter Estimation

In the second step, we estimate the parameters of two models, namely the conditional probabilities of the match variables, given that the records match: $p_\theta(m_{i,j}|(a_i, b_j) \in M)$ and the probability for the match variables given that the records don't match $p_\theta(m_{i,j}|(a_i, b_j) \in U)$. Here the notation $p_\theta(\cdot)$ is used to mean a probability density or mass function which is parameterized by some vector θ of parameters.

If there is plentiful labeled data (i.e., hand linked records of a similar nature) to use for estimation, then we may estimate these parameters analytically using a simple maximum likelihood approach [19]. In the absence of such data (the usual situation for PPDM) estimation is more problematic. Nevertheless, we can often use the EM algorithm [19]. Generally, there is not enough data to estimate a completely general model for the match variables, so instead some we impose additional structure [38]. Historically, a useful method was to restrict the models

to force conditional independence of the individual match variables. Winkler [40] provides a discussion of more structured approaches, and Ravikumar et al. [30] give a specific model with good performance.

2.4 Classification of Record Pairs

Since we are treating record linkage as a statistical problem, it is unlikely that every record pair will be labeled correctly as a link or a non-link. Nevertheless, we can tradeoff the amount of error in the final linkage against the amount of pairs sent for clerical review. As Felligi and sunter demonstrated, the classification of a particular pair (a_i, b_j) into M, U, C may be done by considering the likelihood ratio of $m_{i,j}$ under the two models:

$$r_{i,j} = \frac{p_\theta(m_{i,j}|(a_i, b_j) \in M)}{p_\theta(m_{i,j}|(a_i, b_j) \in U)}$$

As Fellegi and Sunter [11] show, the optimal decision rule is given by:

$$\psi(a_i, b_j) = \begin{cases} M & r_{i,j} > C_1 \\ C & C_0 \leq r_{i,j} \leq C_1 \\ U & r_{i,j} < C_0 \end{cases}$$

This rule is essentially a simple test of hypothesis. One chooses constants C_0, C_1 for user-specified error levels for false-links and false non-links [36]. The rule is optimal in the sense that among the classification rules with that achieve these error rates, this rule assigns the fewest records for clerical review.

2.5 Blocking

When the sizes of the data files to be linked are moderate (e.g., tens of thousands of records or more) then applying the above theory may be too inefficient, since we would have to consider hundreds of millions of pairs. A common way to deal with this problem is to perform a "blocking" phase in which we remove clear non-links, leaving blocks of potential links. The terminology goes back in some sense to the census uses where the population is divided into physical blocks, but also reflects the experimental design notion of "blocking" to remove heterogeneity.

The idea is that a "reliable" field such as zip code or gender may be used to quickly label some of the non-links. See [19] for discussion. The result is a tradeoff of computational efficiency versus accuracy in the final linkage, however the impact on the accuracy is usually fairly mild.

3 Overview of Privacy Preserving Data Mining

The field of "privacy preserving data mining" (PPDM) primarily focuses on performing useful data analysis in such a way as to mitigate the risk of releasing some private or secret information. On the surface, there are two distinct sets of

problems in this field. The first set includes problems of how two or more separate parties each with private data, may compute some function of the union of their data without having to reveal it. The second set focuses on how to determine whether the result of a computation alone constitutes an invasion of privacy (a identifiable release), and if so how to mitigate the release. When two parties need to link their private data and then perform some computation on the resulting linked records, both facets of PPDM are important to respect. In this section, we give a brief overview of the salient features of the field, the goal being to build enough sophistication to understand the subtleties of record linkage in a private setting.

3.1 Secure Multiparty Computation

Suppose two parties each hold a separate piece of private data which they would benefit from jointly analyzing. For example, the parties may be administrators of hospitals or government agencies, who are bound by law to not disclose the information of individuals in their databases. Nevertheless they may wish to join their data to that of some medical research center or another agency in order to fit a statistical model to the union of their data. Performing such computations is the concern of a mature area in the PPDM literature called "Secure Multiparty Computation" (SMC) see e.g., [27, 26] for an overview. The goal is to develop protocols consisting of local computations by individual parties, and the transmitting of messages between the parties. Depending on the demands of the parties involved, one of several models of security may be appropriate.

Perhaps the most well studied and rigorous formulation of a secure computation comes from cryptography [17, 16]. The idea is that the protocol should reveal no more information than would a fanciful "idealized" method in which the private data are presented to a completely trusted third party, who performs the computation and returns the results to each of the original parties. That is, to any specific party, the computation itself should reveal no more than whatever may be revealed by examining his input and output. An example of a protocol that would fail to meet this criteria is if one party was sent all the private inputs, performed the computation locally and then broadcast the results to the other parties. The reason this fails is because, in general, the party who does the computation cannot infer the other's data just from looking at his data and the result, and so the messages passed in the proposed protocol has revealed too much to him.

If it is understood that the parties will follow the protocol, but will try to covertly infer whatever they may from the messages, then this is called the "semi-honest" or "honest but curious" model. Using techniques from cryptography it is theoretically possible [16] to take a protocol for the semi-honest model and make it work under a malicious model, in which one of the parties tries to deviate from the protocol in order to reveal information. Generally though, when the task is inference on joint data, it seems likely that both parties would benefit from the collaboration, and hence the semi-honest model may be a reasonable assumption.

In order to build a protocol for a particular computation, we first make an assumption about the computational power available to the parties. Then we choose a "security parameter" (similar in idea to a key length) so that for a particular party, to determine the others' private inputs becomes a computationally intractable problem (e.g., similar to breaking public key encryption) [16].

An important theoretical result in this area is given by Yao [42] and similarly [18], which show that any function of the parties private inputs may be computed in this setting. The idea is that the parties arrange their computation into a large circuit consisting of wires and gates, then apply a generic protocol to evaluate it on their inputs. Details are given in [16] although for the time being, such a generic protocol is primarily of theoretical interest, since it is prohibitively expensive for all but very small computations. Nevertheless see [28] for an implementation of the generic protocol. An area of study is the construction of protocols for specific problems, which often result in faster and more practically applicable methods. A cornerstone of such techniques is homomorphic encryption [29] which allows parties to perform mathematical operations on each others' encrypted values.

3.2 Alternative Security Models

An alternative which results in fast and often simpler protocols is the "weak" security model given in [9] and studied in [37] section 5.1.3. The idea of this model is that any protocol is fine, so long as the output doesn't reveal exactly what any parties particular input was. Specifically, if there exists an infinitely large set which could be substituted for a parties input, and result in the same output, then the protocol is secure in this weak model. The authors acknowledge that this definition is weak since this infinite set may be e.g., a small ball centered around some point in space, and so may still reveal a great deal of information [37]. Furthermore this definition has no mention of information leakage due to the protocol itself, however it could be amended so that the definition must hold for the intermediate messages as well as the final output. An analysis of some weakly secure inner product protocols is given by [15], who conclude that the weaker model presents a far greater prospect of information leakage than does the cryptographic model.

A second recent alternative is the so-called "differential privacy" approach due to Dwork and colleagues, e.g., see [10]. A randomized algorithm achieves differential privacy if its distribution of outputs doesn't change greatly when the input database is changed by one record. This technique was developed to prevent datamining schemes from releasing information which would identify individuals in the data. Nevertheless it may be brought to bear on multiparty computation. For example, for the problem of record linkage it is conceivable that each party could use a randomized sanitization scheme on their data in order to achieve differential privacy. Then, the data could be revealed to the other parties, and then each party having his own copy of the complete sanitized data could run whatever record linkage or datamining algorithm he wanted to. The question which remains is whether differential privacy is a sufficiently strong

guarantee compared to the cryptographic model, and whether this randomized sanitization scheme would corrupt the data so much that the results would be meaningless.

Finally in some settings the existence of a trusted third party may be realistic. Several protocols make explicit use of such a party [41, 33, 5, 6, 34], in a more limited way.

4 Privacy Preserving Record Linkage

When the files to be matched are held by two different parties and are deemed to be sensitive or private, then we may elicit the use of secure protocols in order to perform the record linkage and whatever may be the final statistical computation. This intersection of record linkage and PPDM has been of great interest in the last decade. The purpose of this section is to first highlight some of the unique challenges posed in this setting, and then to survey the results of research which has sought to solve them.

When the goal is for two parties to integrate their private data, typically they will only care about the set of matching records. If it was the case that they also wanted to share the non-links then there would be no need for secrecy since in the end all the data would become visible to both parties. Protocols which compute the set of linked records and then output them to both parties are perhaps the most well studied part of record linkage in the PPDM literature. In this case, the goal is to perform record linkage without revealing anything about the non-linked records (besides of course, whatever may be inferred of them by means of the linked records). In the cryptographic model this means e.g., that the values of the match variables as well as the parameters of p_θ should not become known explicitly to either party. Even if the computation of the match variables is done securely, for any party to know the values constitutes a failure of security since in general these values are not implied exactly by the linkage itself. For example, while it may be the case that linked records have high similarity, the exact values must remain unknown to either party.

It is important to pay attention to these details, consider a simple model where we allow both parties to learn the similarity measures. Say the data are real vectors and the computed similarity scores are the square or absolute errors between the components. In this case for example the party who holds A may consider two of his distinct values $a_h^k,\ a_i^k$ along with the computed similarities $m_{h,j}^k,\ m_{i,j}^k$. Now he has two distinct points on the real line as well as the distance of b_j^k to each point. Therefore he may solve to recover exactly the value of b_j^k, this way he may reveal he entirety of B, and likewise the owner of B may reveal A. This is a simple example but it serves to illustrate the problems that might arise from revealing intermediate values.

Another important distinction between the private and the usual non-private setting is that resorting to human clerical workers for disambiguation seems tantamount to an invasion of privacy. Although recent methods have focused on performing pure statistical linkage with no need for human intervention, there

is a price to pay in the form of increased error rates. When the overarching goal is to perform some statistical analysis on the linked data, then the error in the linkage must be accounted for in order to obtain a valid analysis. This is in contrast to the usual setting where in essence the human-curated data may be treated as completely correct. Maintaining uncertainty about the linkage is an area which has begun to draw attention in the statistical literature, see e.g., [25].

When the goal is to perform some datamining task on the integrated data (e.g., [24, 23, 22, 12, 13]) then the data themselves are not part of the output. Instead, the final output of the protocol is e.g., a set of estimated regression coefficients on the integrated data, or some other such set of quantities. In this case, we need to take care to protect not only the non-links but also the linked data themselves. For instance, running a secure record linkage algorithm that outputs the links, and then using these data to fit a regression model does not constitute a secure protocol in the cryptographic model. The reason for this is that in general the data themselves are not implied by the regression output.

We repeat that, while in principle all the problems of privacy preserving datamining are solved by the generic protocol of Yao [42], the computational and communication demands of this method are too great in practice [37]. For this reason it is necessary in to devise protocols for the specific problem of record linkage, a problem that we now examine.

5 Methods in Privacy Preserving Record Linkage

While many authors in the literature propose end-to-end secure protocols for record linkage, oftentimes the individual steps may be seen as sub-protocols that are strung together into a secure protocol. Here we describe proposed methods for the steps identified in section 2. We begin, however, with a discussion of private exact matching, which is of historical importance.

5.1 Database Joins and Set Intersection

One of the earliest mentions of record linkage in a private setting is given in [1]. Here the author considers various classical problems from databases, ported to the private setting. The most relevant problem is the computing of a so called "equijoin." This may be considered a variant of record linkage in which two records link whenever they agree exactly on some specific subset of their fields. This then obviates both the need for parameter estimation and statistical inference of the joins, since a deterministic decision is made based upon the single match variable for each pair of records. The goal is to output the entire set of linked records, therefore it is not a concern if the match variables are revealed, since they are implied by the output.

A potential way to compute such an equijoin might be for both parties to apply some one-way hash function [17] to the fields of their records, then share them with each other and see which hashed values match. One might think

that if the hash function is computationally hard to invert then this protocol would be safe. As shown in [1], this naive method fails since the hash function is deterministic. First it may be possible for either party to mount a dictionary attack in which they hash every possible value a field may take on and then see which ones match up to the other party's data. Secondly, when this attack is infeasible the parties may still consider the frequencies with which the hashed values appear. Using this along with knowledge of the distribution of field values (say, estimated empirically from their data), they may be able to reveal some values with high confidence. The way [1] resolve the issue is through the use of a semantically secure [17] encryption scheme. Using such a scheme guarantees that both of these proposed attacks will fails, since it implies that the encryptions are random, and the distributions of them do not differ significantly when the plaintext values are changed. The original protocol must then be modified to accommodate randomness in the hashing. Agrawal's idea paved the way for interest in private record linkage. From a theoretical perspective it is good starting point, however two questions remained. The first is whether the overhead of using this encryption scheme is too great. For example, in order for encryption to be sufficiently hard to break, usually the keys must be chosen to be thousands of bits long. This means that there is a great deal of communication cost, as well as computation since basic mathematical operations on such large numbers may be costly. The second question which remains is whether this approach may be extended to support non-exact matching such as is usual in record linkage.

5.2 Record Pair Similarity

The question of non-exact matching is partially addressed in [5, 6, 34]. These works in essence compute similarity scores for pairs of records via a reduction to a secure set intersection protocol. The idea applies mainly to text data such as names and addresses. First such fields are broken up into a set of "n-grams" which are the substrings of length n. Then since each field is now represented by a set, the size of the intersection of such sets may be compared with the size of the union, to get a measure of the degree of overlap between the two sets. If the intersection is large then the two strings have a large number of common substrings and so are regarded as close to each other and a potential candidate for matching. In principle, the secure protocol of [1] could be used for computing the intersections, however the authors are concerned about the computational overhead. Therefore they resort to a variant of the naive insecure approach mentioned in [1], in which a deterministic one way hashing function is used. To overcome the security issues the authors here instead suggest that a trusted third party may be employed to look at the hashed values and report the cardinality of the intersections. While in principle this approach would be very efficient, it is perhaps conceptually unappealing since the assumption of a trusted third party may be too restrictive in a wide variety of real problems.

An alternative method to compute string similarity is given by [31]. They present a secure two party protocol which computes approximate inner products between real vectors. Their idea is that strings which consist of multiple words

may be represented in a vector space model by the well known TF-IDF transformation which was shown to be useful in record linkage [7]. Their approximation scheme makes use of a cryptographic protocol for secure set intersection, and therefore may be computationally demanding. Whats more, the approach is approximate and to increase the accuracy of the approximation requires increasing the size of the sets which get passed to the sub-protocol.

Another secure vector space method to compute edit distances is described by [33]. Their idea involves a so called metric embedding approach (see e.g., [4]). First some random set of strings is agreed upon by the two parties. Then each party computes the edit distance [2] of his records to each random string. With this in hand, the records may be described by a vector of real numbers in which each component is a distance to a random string. Then it may be shown that the euclidian distance between these vectors corresponds approximately to the string edit distance between the records. In principle, distances between strings could now be approximated by means of a secure inner product protocol, since if we use $\phi(\cdot)$ to denote the embedding we have:

$$d(a_i^k, b_j^k)^2 \approx ||\phi(a_i^k) - \phi(b_j^k)||_2^2 = ||\phi(a_i^k)||_2^2 + ||\phi(b_j^k)||_2^2 - 2\phi(a_i^k)^T\phi(b_j^k)$$

The last term is the inner product, and the other two terms may be computed locally by either party. The authors instead propose to use a third party protocol in which the embedded strings are sent to the third party for computation of these distances. It appears that despite the elegance of this approach, the third party would still be able to mount a frequency based attack on these embeddings. Nevertheless the metric embedding idea is compelling since it results in low-dimensional vectors [33], and so in principle it allows reduction of string edit distance computation to secure inner products which are already well-studied in the literature (e.g., [15]).

We note that all of the string similarity protocols make use of either set-intersection or inner products as a subprotocol. In essence any such protocol could be supplanted in place of the authors' suggestions, and the privacy guarantees and complexity of the resulting protocol would depend on those same characteristics of the sub-protocol. Therefore developing fast protocols for these two problems is important for the future of private record linkage. Although current protocols are reasonable in principle, remember that they will be run on every element of the direct product of the files, which could easily be millions of pairs for even modest size data.

Because a third party may decide whether or not certain similarity scores constitute a link, those protocols which use such a party evidently may output the linkage decision rather than just the similarity. For two party protocols it is less trivial to get the linkage classifications without revealing the similarity. One way, if the similarity scores are computed using a cryptographic protocol, would be to threshold it before it is allowed to be decrypted. For example reducing similarity to the inner product and using [15] results in an encrypted value held by one party, where it may only be decrypted by the other. In this case the

holding party may apply a certain sequence of operations to the ciphertext in order to reduce it to a binary flag corresponding to thresholding against some constant value. One such approach is via a reduction to the so called "millionaires problem" proposed by Yao, which in essence is a protocol to compute an inequality. See [3] for a recent approach.

5.3 Blocking

In the non-private setting, blocking [38] greatly reduces the number of record pairs to be classified. Several authors have ported this idea to the private setting. The idea of blocking is to use simple heuristics based on the record similarities to quickly remove obvious non-links from consideration. In the private setting, however, evaluating such heuristics may itself be a costly process.

One approach is given in [20]. In order to make the blocking step efficient the proposal is to first k-anonymize [35] the database rows, then share them. While the authors choose k-anonymity for its conceptual simplicity there is the prospect that other sanitization schemes could be used such as permuting with noise to achieve differential privacy [10]. After obtaining the sanitized version of the other party's data, the hope is that each party may infer a great deal of non-matches. However they won't be able to infer matches perfectly due to the corruption of the private data due to the sanitization. Therefore a second phase begins in which cryptographic protocols are used to resolve ambiguous record pairs. This way, the proposed scheme achieves a three-way tradeoff of computational overhead vs possible leakage of values vs accuracy of the solution. For example if the sanitization scheme leaves many values unchanged, then privacy is certainly breached, however the resulting accuracy of the linkage will be high, and the cost due to cryptographic protocols will be small. We note that since publication, there have been several published vulnerabilities in the k-anonymization framework [10].

Another paper which employs a blocking approach is [41]. Here the idea is to first transform the records into numeric vectors as in [33], and then perform a secure record linkage technique on these vectors. The protocol is structured in two rounds, the first of which is a blocking phase. The values are permuted and then shared so that the parties may quickly reject obvious non-matches. After this initial step, the remaining candidate record pairs are evaluated through a reduction to a secure inner product protocol as described above. The particular protocol they use may be considered as weakly secure [37].

Note that no matter the settings of the sanitization scheme, these methods will fail to meet the criteria of security in the cryptographic model. To achieve that standard, the sanitization scheme would have to render the data indistinguishable from any arbitrary dataset, and hence would render the blocking phase impossible. Therefore these approaches to blocking may only be used in a weaker security model. In principle it may be possible to do blocking in the cryptographic model, by using a cryptographic protocol for the blocking heuristic; however, this may not be significantly faster than not performing blocking at all, e.g., if such a protocol is costly relative to the full matching protocol for

a record pair. Nevertheless it is possible in practice that the guarantee afforded through the use of differential privacy [10] may be sufficient, so that a blocking scheme based on sanitized data may be feasible.

6 Prominent Unsolved Challenges

The main component of record linkage currently missing from the privacy-aware treatment is that of parameter estimation. All the works above made use of a-priori agreed upon thresholds for the similarity scores, and classify a record as a match if some a-priori agreed upon subset of fields are similar. This technique may result in good linkage under some conditions, however by sidestepping the difficult parameter estimation step, the result is a record linkage with no guarantees regarding error rates.

Another challenge which deserves attention is the development of techniques for record linkage which may propagate uncertainty through to subsequent statistical analysis. One approach is mentioned by Lahiri and Larsen [25] where the goal is to identify additional bias introduced by record linkage and remove it in the final calculation. More general techniques are required, but they may end up being different depending on the type of statistical analysis which is required. Such techniques will be very important, especially when the end result involves confidence intervals or hypothesis testing. The reason is that these are meant to come with well understood statistical guarantees (e.g., the probability of incorrectly rejecting a hypothesis is below some level α). When there is uncertainty in the data itself, then this uncertainty must be modeled in order to have such guarantees in the end.

In order for record linkage to be successfully applied to large databases, it will be important to increase the speed of the cryptographic underpinnings. While using clever protocols may reduce the number of operations (e.g., inner products) performed, ultimately the speed of these operations determines the feasibility of the secure approach.

Privacy-aware record linkage is a crucial problem lying at the intersection of statistics, computer science, and cryptography. We have provided an overview of the recent literature on the topic which builds on earlier reviews and the fundamental approach of Fellegi and Sunter pairs of data files. Extensions of of all of the methods described here to the case of linkage across multiple files, in the presence of measurement error remains a major statistical challenge.

Acknowledgement

This research was partially supported by Army contract DAAD19-02-1-3-0389 to Cylab at Carnegie Mellon University.

References

1. R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 86–97, New York, NY, USA, 2003. ACM.
2. M. Bilenko, R. J. Mooney, W. W. Cohen, P. Ravikumar, and S. E. Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.
3. I. Blake and V. Kolesnikov. Strong conditional oblivious transfer and computing on intervals. In *In Advances in Cryptology - ASIACRYPT 2004*, pages 515–529, 2004.
4. J. Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52(1):46–52, March 1985.
5. T. Churches and P. Christen. Blind data linkage using n-gram similarity comparisons. In H. Dai, R. Srikant, and C. Zhang, editors, *PAKDD*, volume 3056 of *Lecture Notes in Computer Science*, pages 121–126. Springer, 2004.
6. T. Churches and P. Christen. Some methods for blindfolded record linkage. *BMC Medical Informatics and Decision Making*, 4(1):9, 2004.
7. L. W. Cohen and W. W. Cohen. Data integration using similarity joins and a word-based information representation. *ACM Transactions on Information Systems*, 18:2000, 1998.
8. J. Domingo-Ferrer and V. Torra. Validating distance-based record linkage with probabilistic record linkage. In *CCIA '02: Proceedings of the 5th Catalonian Conference on AI*, pages 207–215, London, UK, 2002. Springer-Verlag.
9. W. Du, S. Chen, and Y. S. Han. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *In Proceedings of the 4th SIAM International Conference on Data Mining*, pages 222–233, 2004.
10. C. Dwork. Differential privacy: A survey of results. In M. Agrawal, D.-Z. Du, Z. Duan, and A. Li, editors, *TAMC*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2008.
11. I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.
12. S. Fienberg, W. Fulp, A. Slavkovic, and T. Wrobel. "secure" log-linear and logistic regression analysis of distributed databases. *Privacy in Statistical Databases: CENEX-SDC Project International Conference, PSD 2006*, pages 277–290, 2006.
13. S. Fienberg, A. Slavkovic, and Y. Nardi. Valid statistical analysis for logistic regression with multiple sources. In P. Kantor and M. Lesk, editors, *Proc. Workshop on Interdisciplinary Studies in Information Privacy and Security—ISIPS 2008*. Springer-Verlag, New York, 2009.
14. S. E. Fienberg and D. Manrique-Vallier. Integrated methodology for multiple systems estimation and record linkage using a missing data formulation. *AStA Adv. Stat. Anal.*, 93:49–60, 2009.
15. B. Goethals, S. Laur, H. Lipmaa, and T. Mielikainen. On secure scalar product computation for privacy-preserving data mining. In *ISISC, 2004*, 2004.
16. O. Goldreich. *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness*. Springer-Verlag, New York, 1998.
17. O. Goldreich. *Foundations of Cryptography: Volume 2 Basic Applications*. Cambridge University Press, 2004.
18. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229. ACM, 1987.

19. T. N. Herzog, F. J. Scheuren, and W. E. Winkler. *Data Quality and Record Linkage Techniques*. Springer, 1 edition, May 2007.
20. A. Inan, M. Kantarcioglu, E. Bertino, and M. Scannapieco. A hybrid approach to private record linkage. In *ICDE*, pages 496–505. IEEE, 2008.
21. M. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.
22. A. Karr, X. Lin, J. Reiter, and A. Sanil. Secure regression on distributed databases. *Journal of Computational and Graphical Statistics*, 14(2):263–279, 2005.
23. A. Karr, X. Lin, J. Reiter, and A. Sanil. Secure analysis of distributed databases. In D. Olwell, A. G. Wilson, and G. Wilson, editors, *Statistical Methods in Counterterrorism: Game Theory, Modeling, Syndromic Surveillance, and Biometric Authentication*, pages 237–261. Springer-Verlag, New York, 2006.
24. A. Karr, X. Lin, A. Sanil, and J. Reiter. Privacy-preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics*, 25(1):125–138, 2009.
25. P. Lahiri and M. Larsen. Regression analysis with linked data. *Journal of the American Statistical Association*, 100(469):222–230, 2002.
26. Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3):177–206, 2002.
27. Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):59–98, 2009.
28. D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay—a secure two-party computation system. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, pages 20–20, Berkeley, CA, USA, 2004. USENIX Association.
29. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT 1999*, pages 223–238, 1999.
30. P. Ravikumar and W. W. Cohen. A hierarchical graphical model for record linkage. In D. M. Chickering and J. Y. Halpern, editors, *UAI*, pages 454–461. AUAI Press, 2004.
31. P. Ravikumar, W. W. Cohen, and S. E. Fienberg. A secure protocol for computing string distance metrics. In *In PSDM held at ICDM*, pages 40–46, 2004.
32. M. Sadinle. Multiple record linkage: Generalizing the fellegi–sunter theory. Conflict Analysis Resource Center (CERAC), Bogota, Columbia, January 22 2010.
33. M. Scannapieco, I. Figotin, E. Bertino, and A. K. Elmagarmid. Privacy preserving schema and data matching. In C. Y. Chan, B. C. Ooi, and A. Zhou, editors, *SIGMOD Conference*, pages 653–664. ACM, 2007.
34. R. Schnell, T. Bachteler, and J. Reiher. Privacy-preserving record linkage using bloom filters. *BMC Medical Informatics and Decision Making*, 9(1):41, 2009.
35. L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.
36. V. Torra and J. Domingo-Ferrer. Record linkage methods for multidatabase data mining. In V. Torra, editor, *Information Fusion in Data Mining*, pages 99–130. Springer-Verlag, 2003.
37. J. Vaidya, Y. Zhu, and C. Clifton. *Privacy Preserving Data Mining (Advances in Information Security)*. Springer-Verlag, New York, 2005.
38. W. E. Winkler. Matching and record linkage. In *Business Survey Methods*, pages 355–384. Wiley, 1995.
39. W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

40. W. E. Winkler. Methods for record linkage and bayesian networks. Technical report, Series RRS2002/05, U.S. Bureau of the Census, 2002.
41. M. Yakout, M. J. Atallah, and A. K. Elmagarmid. Efficient private record linkage. In *ICDE*, pages 1283–1286. IEEE, 2009.
42. A. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.