Privacy Preserving Probabilistic Record Linkage

Duncan Smith (Duncan.G.Smith@manchester.ac.uk)¹, Natalie Shlomo (Natalie.Shlomo@manchester.ac.uk)²

Keywords: Anonymisation, String comparators, Similarity scores, EM algorithm

1. INTRODUCTION

Probabilistic record linkage as set out in the seminal paper [1] is an important area of research in official statistics. This is especially true as more and more administrative sources are being used to improve the quality of surveys or to replace traditional censuses. Traditionally, all datasets are held within one organization, e.g. the National Statistics Institute, and record linkage is carried out on original variables, such as first name, last name, ID number, etc. without the need to anonymise these strings. Probabilistic record linkage joins all possible pairs on a set of matching variables (typically within blocks) and the pairs are classified to matches, non-matches and clerical review according to a test statistic based on the likelihood ratio under a Bayesian framework. The matching parameters for the linkage are estimated using an EM algorithm under the Binomial distribution for an agree/disagree (0,1) indicator function. String comparators are used to scale the likelihood ratios to improve the classification.

Data on individuals however can be contained in distinct databases held by different organisations and there may be a variety of data sharing agreements. A common practice is to suppress information that might directly identify an individual which inhibits the possibility of record linkage. For this reason, techniques have been developed to anonymize data in such a way that they can still be used for linkage. Exact matching can be carried out on anonymised strings and methods of classification based on ‘fuzzy’ matching have been introduced in the privacy preserving record linkage literature. A similarity score is calculated to measure the distance between two anonymised strings. One drawback is that when strings are anonymised for record linkage, it is not possible to carry out a clerical review of potential matches for the ambiguous cases and the possible linkages must be dichotomised into true matches and non-matches.

In this paper, we present a new method for classifying pairs into matches and non-matches using an EM algorithm under the Multinomial distribution where string comparators in the non-private setting or similarity scores in the private setting can be used to define classes within which matching parameters can be estimated. In this way, the string comparator/similarity score is included directly into the models for estimating matching parameters instead of through the ad-hoc method of scaling the likelihood ratios and classification into matches/non-matches is much improved. In addition, we examine methods for string anonymisation and propose a new method which provides protection against ‘attack’ scenarios on anonymised strings.

¹ The Cathie Marsh Institute of Social Research, University of Manchester
² Social Statistics, School of Social Sciences, University of Manchester
2. METHODS

2.1. Probabilistic Record Linkage

From two datasets A and B we produce the set of all possible matches: \( A \times B = \{(a, b); a \in A, b \in B\} \) typically within a block defined by an exact match on an error free variable, such as geographical area, to reduce the number of pairs that need to be investigated. We aim to classify the pairs into sets: Matches \( M = \{(a,b)\mid a = b, a \in A, b \in B\} \) and non-matches \( U = \{(a, b)\mid a \neq b, a \in A, b \in B\}. \) [1] define a decision rule based on the likelihood ratio of agreement \( m(\gamma)/u(\gamma) \) or disagreement \( \frac{1 - m(\gamma)}{1-u(\gamma)} \) where \( m(\gamma) \) is the probability of agreement for the comparison vector \( \gamma \) given a match and \( u(\gamma) \) is the probability of agreement for the comparison vector \( \gamma \) given not a match. In the simplest form, \( \gamma \) will be a vector of 1’s or 0’s where 1 denotes an agreement in matching variable \( q \) \((q=1,...,Q)\) and 0 otherwise. Under conditional independence, we can treat each matching variable separately and define \( m_q = P(\gamma_q^1 = 1 \mid (a,b) \in M) \) and \( u_q = P(\gamma_q^1 = 1 \mid (a,b) \in U) \). The two probabilities for each matching variable and the overall number of correct matches \( P(M) = P((a,b) \in M) \) is estimated using the EM algorithm under the Binomial distribution. Further details will be provided in the paper.

String comparators are used to scale the likelihood ratios. The Jaro string comparator is commonly used for official statistics and defined as:

\[
\Phi_q(X_a, X_b) = 1/3 \times \left( \frac{\#\text{common}}{\text{str_len1}} + \frac{\#\text{common}}{\text{str_len2}} + \frac{1 - \frac{\#\text{transposition}}{\#\text{common}}}{2} \right)
\]

where \( \text{str_len1} \) and \( \text{str_len2} \) length of strings, \( \#\text{common} \) is number of common letters and \#transposition where a letter can move one position left or right. The Jaro-Winkler string comparator is based on the Jaro string comparator but provides different weights depending on the position in the string. The string comparator is used to down-weight the agreement likelihood ratio so that if there is little difference in the string, the pair will not be regarded as a disagreement rather will receive a value at a proportional distance from the agreement likelihood ratio. The new likelihood ratio is defined as:

\[
\Phi_q(X_a, X_b) = \frac{m_q}{u_q} + (1 - \Phi_q(X_a, X_b)) \frac{1 - m_q}{1 - u_q}.
\]

Alternately, the log-likelihood can be adjusted as follows:

\[
\left( \frac{m_q}{u_q} \right)^{\Phi_q(X_a, X_b)} \left( \frac{1 - m_q}{1 - u_q} \right)^{(1 - \Phi_q(X_a, X_b))}
\]

2.2. String Anonymisation

Hash functions are used to anonymise strings. These hash functions convert strings to integers where equal strings produce equal hash values so linkage can be carried out on the equality or inequality of hash values. Strings are converted to tokens, typically bigrams, and each bigram is hashed [2]. For example, for the names John and Jon we obtain:

- 'john' → {‘jo’, ‘oh’, ‘hn’} → \{21299418, 21496024, 20971735\}
- 'jon' → {‘jo’, ‘on’} → \{21299418, 21889246\}

A similarity score such as the Dice Coefficient can be used:

\[
D_{a,b} = \frac{2 \times \#\text{common}}{\text{Total hashes}} = \frac{2}{5}
\]
[3] propose the use of Bloom filters (an array of 0 and 1) which can be represented as an integer in its binary form. m bits (all initially set to 0) and k hash functions are used to map an element to one of the m array positions. To add an element, produce k hash functions to get k array positions and set these positions to 1. To query whether an element is in the set, feed it to each of the k hash functions to get k array positions and check whether there is a zero. If there's a zero, the element is not in the set, otherwise if all 1’s it might be in the set since there may be false positives. The Dice coefficient can be estimated from a pair of Bloom filters. These methods may be open to attacks where intruders can learn the length of strings or identify tokens, eg. bigrams.

[4] proposed a technique called minwise hashing where many hash values are calculated for a set of tokens and the minimum hash values returned. The probability of a hash collision on the minimum hash value is the Jaccard Similarity Score defined as

\[ J_{A,B} = \frac{\#common\;hashes}{total\;discrete\;hashes} \].

Using the minwise hashing, the estimate of the Jaccard Similarity Score is the number of collisions on the minimum hash value where m is the number of hash functions: \( n \sim Bin(m, J_{A,B}) \) and the estimate is \( \hat{J}_{A,B} = \frac{n}{m} \). The variance is:

\[ \text{Var}(\hat{J}_{A,B}) = \frac{J_{A,B}(1-J_{A,B})}{m} \].

[5] propose to return only b-bits of the minimum hash value. In [6], we propose to return only 1-bit of the minimum hash value to produce a concatenated 1-bit minwise hashing. In this case, the estimate of the Jaccard Similarity score is \( \hat{J}_{A,B} = 2\frac{n}{m} - 1 \) with a variance of:

\[ \text{Var}(\hat{J}_{A,B}) = \frac{1-J_{A,B}^2}{m} \].

In the example in Table 1, we show the first 5 of the minwise hash values and the 1-bit minwise hash values. From table 1, for minwise hashing with 5 hash functions, the estimate of the Jaccard similarity score is 2/5 and for the concatenated 1-bit hashing 3/5, the true value is 1/4. More on string anonymisation will be covered in the paper.

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>H4</th>
<th>H5</th>
<th>…</th>
<th>Hm</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>451153726</td>
<td>1123790273</td>
<td>2501120381</td>
<td>2030682762</td>
<td>965995804</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>797504823</td>
<td>1123790273</td>
<td>262296169</td>
<td>1744666338</td>
<td>965995804</td>
<td></td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>…</td>
<td>Hm</td>
</tr>
<tr>
<td>S2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>…</td>
<td></td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.** Minwise hashes and 1-bit minwise hashes under a binary representation for S1={‘jo’,’oh’,’hn’} and S2={‘jo’,’on’}

2.3. Multinomial EM Algorithm

We extend the EM algorithm where instead of two categories agree/disagree for each matching variable and then scaling the likelihood ratios with string comparators, we define \( k \) categories, \( k=1,\ldots,K \) where each category represents a class based on an interval of string comparators (or similarity scores in the private setting). For example, 8 classes with (inclusive) upper bounds based on the string comparator/similarity score: [0.2,0.4,0.6,0.8,0.9,0.95,0.999,1].
If there is an agreement for variable \( q \) in class \( k \) of the string comparator/similarity score for a pair \( j \), then: \( \gamma'_{i,j} = 1 \), otherwise it is zero. The EM algorithm under the multinomial distribution is now used to estimate the match parameters for each variable \( q \) in class \( k \) (not shown here). Further details about the method will be in the paper.

3. EXPERIMENT

From a 1995 Israel Census Sample file, we select 700 records as a test file. From this file, 400 records were selected and varying levels of perturbation applied. We show in Figure 1 the low and high levels of perturbation. No blocking was carried out. Since we know the match status, we can evaluate the linkage using precision/recall plots where:

\[
\text{Precision} = \frac{tp}{tp + fp}, \quad \text{Recall} = \frac{tp}{tp + fn},
\]

where \( tp \) are the true positive, \( fn \) the false negatives and \( fp \) the false positives under varying levels of thresholds. The best approaches produce curves in the upper right sector of the plot. The results of the experiment are shown in Figure 1.

![Low Perturbation](image1)

![High Perturbation](image2)

**Figure 1.** Precision/Recall plots for low and high level of perturbation for: Binary EM algorithm (no use of similarity score); down-weighted likelihood ratio (LR); down-weighted log-likelihood ratio (Log LR); 8-bin EM algorithm with estimated Jaccard Score; 15-bin EM Algorithm with estimated Jaccard Score; 8-bin EM algorithm with Jaro String comparator; 8-bin EM algorithm with Jaro-winkler String comparator

4. CONCLUSIONS

From Figure 1, all approaches perform better with low levels of perturbation. The Binary EM without similarity scores performs the worst. Down-weighting log likelihood ratios outperforms down-weighting of likelihood ratios. Multinomial EM outperforms Binary EM with no clear difference between 8 classes and 15 classes Jaccard score schemes. The non-private setting with Jaro and Jaro-Winkler string comparators provide the best performance, although these are not privacy preserving.

Given that in the private setting, there is no possibility to carry out clerical review it is important that we improve the classification of pairs into matches and non-matches. The multinomial EM Algorithm approach shows an improvement over the traditional method of the Binomial EM algorithm and down-weighting of likelihood ratios.
ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 262608 (DwB - Data without Boundaries).

REFERENCES