Record linkage methods for multidatabase data mining

Vicenç Torra[1] and Josep Domingo-Ferrer[2]

[1] Institut d'Investigació en Intel·ligència Artificial - CSIC
Campus UAB s/n, 08193 Bellaterra (Catalonia, Spain)
e-mail: vtorra@iiia.csic.es, http://www.iiia.csic.es/~vtorra/
[2] Dept. of Computer Science and Mathematics, Universitat Rovira i Virgili,
Av. Països Catalans 26, 43007 Tarragona (Catalonia, Spain)
e-mail: jdomingo@etse.urv.es, http://www.etse.urv.es/recerca/crises/

Abstract. This chapter reviews record linkage techniques, useful to link records in two different data files corresponding to the same individual. Both probability-based and distance-based are presented and compared.

1 Introduction

Record linkage is one of the existing preprocessing techniques used for data cleaning for distributed and non-homogoneous databases. Such databases contain information about the same individuals described using the same variables that, frequently, do not match due to accidental distortion of the data[1]. Record linkage techniques are applied in such cases to find the records that correspond to the same individuals and to make databases consistent. Multidatabase mining, that typically combines databases from different sources and, therefore, non-homogeneous also benefits from these tools.

This chapter describes existing mechanisms for re-identifying those pairs of records in two different data files corresponding to the same individual. We review in this chapter record linkage techniques in the case where the files share a set of variables. Techniques for files not sharing any variable have recently been proposed in [10] and will not be discussed here.

The structure of the chapter is as follows. Section 2 describes the notation that will be used in the rest of the paper. Then, in Sections 3 and 4 we review probabilistic and distance-based record linkage. Section 5 includes some technical issues concerning both record linkage approaches (*e.g.* standardization of variables and string comparison methods). The chapter ends with some conclusions.

[1] Sometimes differences are due to intentional distortion provoked *e.g.* for data anonymization

2 Notation

In this section we present the notation used throughout the chapter. We start describing the files and their records and, later on, the comparisons between pairs of records.

Let $\mathbf{A}$ and $\mathbf{B}$ be two data files defined, as usual, as sets of records. Let $r^{A,i}$ and $r^{B,i}$ denote the i-th record in files $\mathbf{A}$ and $\mathbf{B}$, respectively. We will use r^i when there is no possibility of confusion.

Records are defined in terms of variables (*i.e.* fields or attributes) and the values they take for those variables. Since files $\mathbf{F}$ contain a value for each record-variable pair, they can be modeled as a function:

$$\mathbf{V} : \mathbf{F} \rightarrow D(V_1) \times D(V_2) \times \cdots \times D(V_n)$$

where $\mathbf{V} = \{V_1, V_2, \cdots V_n\}$ denote the variables and $D(V_i)$ refer to the range of variable V_i (some fields like artificial intelligence often use the term domain of V_i). Without loss of generality, the n-dimensional function V can be assumed to be of the form:

$$\mathbf{V}(r) = (V_1(r)V_2(r) \cdots V_n(r))$$

where $V_i(\cdot) : \mathbf{F} \rightarrow D(V_i)$ is a one-dimensional function assigning a value for attribute V_i to a given record.

We shall use $\mathbf{V}^F(O)$ and V_i^F (for $i \in \{1, \ldots, n\}$) to specify the file when required.

For re-identification, we need to consider all pairs of records where one record is from the file $\mathbf{A}$ and the other is from the file $\mathbf{B}$. This is equivalent to considering the product of both files:

$$\mathbf{A} \times \mathbf{B}$$

Then, $r^{\mathbf{A}\times\mathbf{B}} = (a, b) \in \mathbf{A} \times \mathbf{B}$ denotes an arbitrary element of this set. Naturally, $a \in \mathbf{A}$ and $b \in \mathbf{B}$. We also assume the cardinality $|\mathbf{A} \times \mathbf{B}|$ to be N. Similar to the case of files, we will use $r^{\mathbf{A}\times\mathbf{B},i}$ to denote the i-th pair in $\mathbf{A} \times \mathbf{B}$ and use r^i when no possibility of confusion arises.

Two sets $\mathbf{M}$ and $\mathbf{U}$ are considered over the above Cartesian product, such that $\mathbf{M} \cup \mathbf{U} = \mathbf{A} \times \mathbf{B}$ and $\mathbf{M} \cap \mathbf{U} = \emptyset$). The first set $\mathbf{M}$ corresponds to the pairs such that both records (the one in $\mathbf{A}$ and the one in $\mathbf{B}$) correspond to the same individual; these are called the *matched pairs*. The second set $\mathbf{U}$ corresponds to the pairs such that both records correspond to different individuals; these are called *unmatched pairs*.

The procedure to link records of both files (*record linkage*) can be viewed as a classification of each record pair $(a, b) \in \mathbf{A} \times \mathbf{B}$ as either belonging to $\mathbf{M}$ or $\mathbf{U}$.

When re-identification is based on the assumption that both files share a set of variables, it is relevant to consider the comparison between the two

records that define a pair. To do such comparisons, we assume that the number of variables in both files is n and that both files present variables in the same order (*i.e.* $V_i^A = V_i^B$ for $i \in \{1, \ldots, n\}$). In that case, the pair (a, b) can, alternatively, be expressed as:

$$((V_1^A(a), V_2^A(a), \ldots, V_n^A(a)), (V_1^A(b), V_2^A(b), \ldots, V_n^A(b)))$$

Let us define the following function over $\mathbf{A} \times \mathbf{B}$

$$\gamma(a, b) = (\gamma_1(a, b), \ldots, \gamma_n(a, b))$$

Given a pair (a, b), we define $\gamma_i(a, b)$ as 1 if $V_i(a) = V_i(b)$, and as 0 if $V_i(a) \neq V_i(b)$.

If we do not care about the specific pair of records but only about the values, we shall use the following notation:

$$\gamma = (\gamma_1, \ldots, \gamma_n)$$

Note that γ is a vector in $\{0, 1\}^n$ (actually it can be viewed as a coincidence vector).

The set of all γ coincidence vectors is denoted by Γ. The maximum cardinality of this latter set is 2^n. Thus $\Gamma = \{\gamma^1, \gamma^2, \ldots, \gamma^{2^n}\}$ where the coincidence vectors γ^j for $j = 1, \ldots, 2^n$ are n-dimensional vectors:

$$\gamma^j = (\gamma_1^j, \ldots, \gamma_n^j)$$

3 Probabilistic record linkage

The goal of record linkage is to establish whether pairs of records $(a, b) \in \mathbf{A} \times \mathbf{B}$ either belong to the set $\mathbf{M}$ or to the set $\mathbf{U}$. This is, whether both records a and b correspond to the same individual or to different individuals. Record linkage can be achieved by means of the so-called linkage or decision rules. These rules classify pairs as linked (placing them in $\mathbf{M}$) or non-linked (placing them in $\mathbf{U}$). Moreover, as the available information is sometimes not enough to discriminate between matched and unmatched pairs, some decision rules consider an additional classification alternative: clerical pairs. This is, a pair is classified as clerical when it cannot be automatically classified neither in $\mathbf{M}$ nor in $\mathbf{U}$; classification of clerical pairs must be manually done by a human operator. According to the above discussion, the following classes are considered by decision rules: $\mathbf{DR} = \{\mathbf{LP}, \mathbf{CP}, \mathbf{NP}\}$.

1. **LP**: Set of linked pairs
2. **CP**: Set of clerical pairs
3. **NP**: Set of non-linked pairs

$Name^{\mathbf{A}}$	$Surname^{\mathbf{A}}$	$Age^{\mathbf{A}}$	$Name^{\mathbf{B}}$	$Surname^{\mathbf{B}}$	$Age^{\mathbf{B}}$
Joan	Casanoves	19	Joan	Casanovas	19
Pere	Joan	17	Pere	Joan	17
J.M.	Casanovas	35	J.Manel	Casanovas	35
Juan	Garcia	53	Juan	Garcia	53
Ricardo	Garcia	14	Ricard	Garcia	14
Pere	Garcia	18	Pere	Garcia	82
Juan	Garcia	18	Juan	Garcia	18
Ricard	Tanaka	14	Ricard	Tanaka	18

Table 1. Files **A** and **B**

In probabilistic record linkage, a basic assumption is that files share a set of variables. Taking this into account, decision rules rl are defined as mappings from the comparison space (the space of all comparisons Γ) into probability distributions over **DR**. If $\gamma \in \Gamma$ then $rl(\gamma) = (\alpha_1, \alpha_2, \alpha_3)$ where $\alpha_1, \alpha_2, \alpha_3$ are, respectively, the membership probabilities for $\{\mathbf{LP}, \mathbf{CP}, \mathbf{NP}\}$. Naturally, $\alpha_1 + \alpha_2 + \alpha_3 = 1$ and $\alpha_i \geq 0$.

Example 1. Let us consider the files **A** and **B** in Table 1. Both files contain 8 records and 3 variables (Name, Surname and Age). For the sake of understandability, the files are defined so that records in the same row correspond to matched pairs and records in different rows correspond to unmatched pairs. The goal of record linkage in this example is to classify all possible pairs so that pairs with both records in the same row are classified as linked pairs and all the other pairs are classified as non-linked pairs.

To do so, we consider all pairs $(a, b) \in \mathbf{A} \times \mathbf{B}$. These pairs and the corresponding $\gamma(a, b)$ are displayed in Table 2. In this example, $\Gamma = \{\gamma^1 = 000, \gamma^2 = 001, \gamma^3 = 010, \gamma^4 = 011, \gamma^5 = 100, \gamma^6 = 101, \gamma^7 = 110, \gamma^8 = 111\}$. Note that the number of different coincidence vectors (8) is much less than the number of pairs in $\mathbf{A} \times \mathbf{B}$ (64). Yet, in record linkage, the classification of any pair (a, b) in Table 2 is solely based on its corresponding coincidence vector $\gamma(a, b)$.

Let us consider in the example below a rule that will be used later on. Rather than being expressed in terms of a probability distribution on **DR**, the rule directly assigns a class to each pair of records using the following expression:

$$P(\gamma = \gamma(a, b) | (a, b) \in \mathbf{M}) / P(\gamma = \gamma(a, b) | (a, b) \in \mathbf{U}) \tag{1}$$

Example 2. Let (a, b) be a pair of records in $\mathbf{A} \times \mathbf{B}$ and let (lt, ut) be two thresholds (lower and upper) in $\mathbb{R}$ such that $lt < ut$. Then a possible decision rule is:

1. If $R_p(a, b) \geq ut$ then (a, b) is a Linked Pair (**LP**)

2. If $R_p(a,b) \leq lt$ then (a,b) is a Non linked Pair (**NP**)
3. If $lt < R_p(a,b) < ut$ then (a,b) is a Clerical Pair (**CP**)

where the index $R_p(a,b)$ is defined in terms of the vector of coincidences $\gamma(a,b)$ as follows:

$$R_p(a,b) = R(\gamma(a,b)) = \log(\frac{P(\gamma(a,b) = \gamma(a',b')|(a',b') \in \mathbf{M})}{P(\gamma(a,b) = \gamma(a',b')|(a',b') \in \mathbf{U})}) \quad (2)$$

Remark that, in the above example, $R(\gamma)$ does not really use the values in records a and b but only their coincidences. Therefore, two pairs (a,b) and (c,d) such that $\gamma(a,b) = \gamma(c,d)$ are classified in the same way. The rationale of Expression 2 is made clear in the rest of this section and the use of log is explained in Section 3.2. Nevertheless, note that this rule associates large values of R to those pairs whose γ is such that $P(\gamma = \gamma(a',b')|(a',b') \in \mathbf{M})$ is large and $P(\gamma = \gamma(a',b')|(a',b') \in \mathbf{U})$ is small. Therefore, larger values are assigned to $R(\gamma)$ when the probability of finding the coincidence vector γ is larger in **M** than in **U**. Otherwise, small values of R are assigned to coincidence vectors with larger probabilities in **U** than in **M**.

In what follows, we will use m^i and u^i to denote the conditional probabilities of the coincidence vector γ^i:

$$m^i = P(\gamma^i = \gamma(a',b')|(a',b') \in \mathbf{M}) \quad (3)$$

$$u^i = P(\gamma^i = \gamma(a',b')|(a',b') \in \mathbf{U}) \quad (4)$$

Example 3. Table 3 gives the computation of R for all pairs of records in Table 2. Probabilities $m^i = P(\gamma^i = \gamma(a',b')|(a',b') \in \mathbf{M})$ and $u^i P(\gamma^i = \gamma(a',b')|(a',b') \in \mathbf{U})$ have been estimated by the proportion of elements in either **M** or **U** with such coincidence vector γ^i. The table gives coincidence vectors ordered (in decreasing order) according to their R values.

In general, for any decision rule *rl*, the following two probabilities are of interest:

$$P(\mathbf{LP}|\mathbf{U}) = \mu \quad (5)$$

$$P(\mathbf{NP}|\mathbf{M}) = \lambda \quad (6)$$

Note that the above are the probabilities that the rule causes an error. In particular, the first probability corresponds to the classification as a linked pair of a pair that is not a matched pair. This situation corresponds to the so-called *false linkage*. The second probability corresponds to the classification as a non-linked pair of a matched pair. This situation corresponds to the so-called *false unlinkage*.

Example 4. Let (lt, ut) be the lower and upper thresholds used in the decision rule of Example 2. Then, the probabilities $\mu = P(\mathbf{LP}|\mathbf{U})$ and $\lambda = P(\mathbf{NP}|\mathbf{M})$ for this decision rule are equal to:

$$\mu = \sum_{i:\log(m^i/u^i)>ut} u^i$$

$$\lambda = \sum_{i:\log(m^i/u^i)<lt} m^i$$

Assume $lt = 1.5$ and $ut = 2.5$. Using the data from Examples 1 and 3, gives the following values for μ and λ:

$$\mu = 0/56 + 1/56 = 1/56 = 0.0178$$

$$\lambda = 0 + 0 + 0 + 1/8 = 0.125$$

In addition to the two conditional probabilities above, another probability is also relevant in decision rules: the probability of classifying pairs of records into the set **CP**. As this latter set corresponds to pairs that should be further revised, the smaller the probability, the better. Therefore, it is clear that, given the set of all decision rules with the same probabilities $P(\mathbf{LP}|\mathbf{U})$ and $P(\mathbf{NP}|\mathbf{M})$, we are interested in finding the one (or ones) with the smallest probability of classifying a pair as **CP**.

To that end, Fellegi and Sunter [11] considered the following definitions.

Definition 1. Let rl be a decision rule in the space Γ and let μ and λ be the two values in the interval $(0, 1)$ for its conditional probabilities $P(\mathbf{LP}|\mathbf{U})$ and $P(\mathbf{NP}|\mathbf{M})$ (Expressions 5 and 6). Then rl is a *rule with levels μ and λ* and is expressed by $rl(\mu, \lambda, \Gamma)$

Definition 2. Let **rl** be the set of all decision rules over Γ with levels μ and λ. Then $rl(\mu, \lambda, \Gamma)$ is the *optimal decision rule* if it satisfies:

$$P(\mathbf{CP}|rl) \leq P(\mathbf{CP}|rl')$$

for all $rl'(\mu, \lambda, \Gamma)$ in **rl**.

In these definitions, it is assumed that μ and λ lead to a non-empty set of decision rules. It is said that μ and λ are admissible when they satisfy simultaneously Expressions 5 and 6 and when the set of decision rules is not empty. See [11] for details on the admissibility of μ and λ.

Fellegi and Sunter define an optimal decision rule based on Expression 1 and, as will be seen later, the rule is similar to the one we have given in Example 2. This optimal decision rule is defined below:

Definition 3. [11] Let μ and λ be an admissible pair of error levels and σ be a permutation of $\{1, \ldots, |\Gamma|\}$ such that $\sigma(j) < \sigma(k)$ if:

$$\frac{P(\gamma^{\sigma(j)} = \gamma(a', b') | (a', b') \in \mathbf{M})}{P(\gamma^{\sigma(j)} = \gamma(a', b') | (a', b') \in \mathbf{U})} > \frac{P(\gamma^{\sigma(k)} = \gamma(a', b') | (a', b') \in \mathbf{M})}{P(\gamma^{\sigma(k)} = \gamma(a', b') | (a', b') \in \mathbf{U})} \quad (7)$$

and let $limit$ and $limit'$ be the indexes such that:

$$\sum_{i=1, limit-1} u^{\sigma(i)} < \mu \leq \sum_{i=1, limit} u^{\sigma(i)} \quad (8)$$

$$\sum_{i=limit'+1, |\Gamma|} m^{\sigma(i)} < \lambda \leq \sum_{i=limit', |\Gamma|} m^{\sigma(i)} \quad (9)$$

where u^i and m^i correspond to the conditional probabilities in Expression 3 and 4.

Then, the optimal decision rule ODR_p for the pair (a, b) is a probability distribution $(\alpha_1, \alpha_2, \alpha_3)$ on $\{\mathbf{LP}, \mathbf{CP}, \mathbf{NP}\}$ defined by $ODR_p(a, b) = ODR(\gamma(a, b))$ with ODR defined as follows:

$$ODR(\gamma^{\sigma(i)}) = \begin{cases} (1, 0, 0) & \text{si } 1 \leq i \leq limit - 1 \\ (P_\mu, 1 - P_\mu, 0) & \text{si } i = limit \\ (0, 1, 0) & \text{si } limit < i < limit' \\ (0, 1 - P_\lambda, P_\lambda) & \text{si } i = limit' \\ (0, 0, 1) & \text{si } limit' + 1 \leq i \leq |\Gamma| \end{cases} \quad (10)$$

and where P_μ and P_λ are the solutions of the equations:

$$u^{\sigma(limit)} P_\mu = \mu - \sum_{i=1}^{limit-1} u^{\sigma(i)} \quad (11)$$

$$m^{\sigma(limit')} P_\lambda = \lambda - \sum_{i=limit'+1}^{|\Gamma|} m^{\sigma(i)} \quad (12)$$

This decision rule is optimal. This is established in the next theorem:

Theorem 1. *[11] The decision rule in Definition 3 is a best decision rule on Γ at the levels μ and λ.*

According to the procedure outlined above, the classification of a pair (a, b) requires: (i) computing the coincidence vector γ; (ii) determining the position of this γ vector in Γ once elements in Γ are ordered according to Expression 7 (otherwise put, finding i such that $\gamma^{\sigma(i)} = \gamma$) and (iii) computing the probability distribution over $\mathbf{DR}$ for this $\gamma^{\sigma(i)}$.

3.1 Alternative expressions for decision rules

In the particular case that μ and λ satisfy the following equations:

$$\mu = \sum_{i=1,limit} u^{\sigma(i)} \tag{13}$$

$$\lambda = \sum_{i=limit',N_\Gamma} m^{\sigma(i)} \tag{14}$$

the decision rule in Expression 10 can be simplifiedas:

$$SimpODR(\gamma^{\sigma(i)}) = \begin{cases} (1,0,0) \text{ if } 1 \leq i \leq limit \\ (0,1,0) \text{ if } limit < i < limit' \\ (0,0,1) \text{ if } limit' \leq i \leq |\Gamma| \end{cases} \tag{15}$$

This rule uses σ, $limit$ and $limit'$ as given in Definition 3, and is also optimal under the established conditions for μ and λ in Expressions 13 and 14.

Nevertheless, when Equations 13 and 14 do not hold, Rule 15 is not applicable and the previous definition with probability distributions is needed. To avoid the use of such probability distributions, that make practical applications more complex, we can classify as clerical pairs (assign them to class **CP**) those pairs that lead to a $\gamma^{\sigma(i)}$ with $i = limit$ or $i = limit'$. This is equivalent to using the following decision rule:

$$AltDR(\gamma^{\sigma(i)}) = \begin{cases} (1,0,0) \text{ if } 1 \leq i \leq limit - 1 \\ (0,1,0) \text{ if } limit \leq i \leq limit' \\ (0,0,1) \text{ if } limit' + 1 \leq i \leq |\Gamma| \end{cases} \tag{16}$$

Note that in this rule μ is used as an error bound, because the probability $P(\mathbf{LP}|\mathbf{U})$ of the new rule is smaller than the probability of the previous rule and, thus, smaller than μ. This is proven in the next proposition. The same applies for λ.

Proposition 1. *Let* $P_{\text{ODR}}(\mathbf{LP}|\mathbf{U})$ *and* $P_{\text{AltDR}}(\mathbf{LP}|\mathbf{U})$ *be the probabilities of the optimal decision rule* ODR *in Definition 3 and of the decision rule in Equation 16. Then,*

$$P_{\text{AltDR}}(\mathbf{LP}|\mathbf{U}) < P_{\text{ODR}}(\mathbf{LP}|\mathbf{U}) = \mu$$

Proof. Let us consider the probability $P_{ODR}(\mathbf{LP}|\mathbf{U})$ when the decision rule in Definition 3 is used for a given admissible pair of errors μ and λ. In this case, $P_{ODR}(\mathbf{LP}|\mathbf{U})$ equals to:

$$P_{ODR}(\mathbf{LP}|\mathbf{U}) = \sum_{i=1}^{limit-1} u^{\sigma(i)} + P_\mu \cdot u^{\sigma(limit)}$$

According to Equation 11 in Definition 3, $P_\mu \cdot u^{\sigma(limit)}$ equals to $\mu - \sum_{i=1}^{limit-1} u^{\sigma(i)}$. Therefore,

$$P_{ODR}(\mathbf{LP}|\mathbf{U}) = \sum_{i=1}^{limit-1} u^{\sigma(i)} + \mu - \sum_{i=1}^{limit-1} u^{\sigma(i)} = \mu$$

Alternatively, when the decision rule in Definition 16 is used, the following conditional probability is used:

$$P_{AltDR}(\mathbf{LP}|\mathbf{U}) = \sum_{i=1}^{limit-1} u^{\sigma(i)}$$

According to Equation 8, the above probability is less than μ. Therefore:

$$P_{AltDR}(\mathbf{LP}|\mathbf{U}) = \sum_{i=1}^{limit-1} u^{\sigma(i)} < \mu$$

Proposition 2. *Let* $P_{\mathrm{ODR}}(\mathbf{NP}|\mathbf{M})$ *and* $P_{\mathrm{AltDR}}(\mathbf{NP}|\mathbf{M})$ *be the probabilities of the optimal decision rule* ODR *in Definition 3 and of the decision rule in Equation 16. Then,*

$$P_{\mathrm{AltDR}}(\mathbf{NP}|\mathbf{M}) < P_{\mathrm{ODR}}(\mathbf{NP}|\mathbf{M}) = \lambda$$

Nevertheless, the rule in Expression 16 also classifies as clerical pairs those pairs (a, b) with $\gamma(a, b) = \gamma^{\sigma(limit)}$ or $\gamma(a, b) = \gamma^{\sigma(limit')}$ when Expressions 13 and 14 hold. Since these pairs can be classified as **LP** and **NP** without violating the bounds

$$P(\mathbf{LP}|\mathbf{U}) \leq \mu$$

and

$$P(\mathbf{NP}|\mathbf{M}) \leq \lambda$$

, the previous rule can be rewritten as follows:

$$rule(\gamma^{\sigma(i)}) = \begin{cases} (1,0,0) \text{ if } 1 \leq i \leq limit \\ (0,1,0) \text{ if } limit < i < limit' \\ (0,0,1) \text{ if } limit' \leq i \leq N_\Gamma \end{cases} \tag{17}$$

This requires that the indices $limit$ and $limit'$ be determined according to the following inequalities:

$$\sum_{i=1}^{limit} u^{\sigma(i)} \leq \mu < \sum_{i=1}^{limit+1} u^{\sigma(i)} \tag{18}$$

$$\sum_{i=limit'}^{|\Gamma|} m^{\sigma(i)} \leq \lambda < \sum_{i=limit'-1}^{|\Gamma|} m^{\sigma(i)} \tag{19}$$

It is important to underline that these latter rules are non-optimal rules because the probability of classifying a pair as a clerical pair is larger than the one in Definition 3. However, from a practical point of view, the last rule is convenient and easy to use; for example, it was the rule used in [14].

For the application of the decision rules defined so far, we need to know the position of the coincidence vector $\gamma^{\sigma(i)}$ in the ordering obtained from Γ and also the indexes $limit$ and $limit'$. We give below an alternative definition that does not require these elements. This definition is equivalent to the one given above when appropriate thresholds are selected. This rule corresponds to the one presented in Example 2.

Definition 4. Let (a, b) be a pair of records in $\mathbf{A} \times \mathbf{B}$, let (lt, ut) be two thresholds (lower and upper) in $\mathbb{R}$ such that $lt < ut$, then the *Decision Rule* is defined as follows:

1. If $R_p(a, b) \geq ut$ then (a, b) is a Linked Pair (**LP**)
2. If $R_p(a, b) \leq lt$ then (a, b) is a Non linked Pair (**NP**)
3. If $lt < R_p(a, b) < ut$ then (a, b) is a Clerical Pair (**CP**)

where the index $R_p(a, b)$ is defined in terms of the vector of coincidences $\gamma(a, b)$ using Equation 1:

$$R_p(a,b) = R(\gamma(a,b)) = \log\left(\frac{P(\gamma(a,b) = \gamma(a',b')|(a',b') \in \mathbf{M})}{P(\gamma(a,b) = \gamma(a',b')|(a',b') \in \mathbf{U})}\right) \tag{20}$$

Proposition 3. *The decision rule in Definition 4 is equivalent to the decision rule in Expression 17 when lt and ut are defined as follows:*

$$ut = \log(m^{\sigma(limit)}/u^{\sigma(limit)})$$

$$lt = \log(m^{\sigma(limit')}/u^{\sigma(limit')})$$

where, as usual, $m^i = P(\gamma^i = \gamma(a', b')|(a', b') \in \mathbf{M})$, $u^i = P(\gamma^i = \gamma(a', b')|(a', b') \in \mathbf{U})$.

Example 5. Let us consider the probabilistic record linkage of records defined in terms of the three variables (*Name, Surname, Age*) as in Example 1. Let us consider the conditional probabilities m^i and u^i inferred from files **A** and **B** in Example 1 and computed in Example 3 (displayed in Table 3).

Now, let us compute the decision rule for $\mu = 0.05$ and $\lambda = 0.2$ and show its application to classify the pair *((J. Gomez 19), (P. Gomez 19))*.

First, to define the rule, we need to determine $limit$ and $limit'$ to apply Proposition 3. These values are set by Expressions 18 and 19 and the conditional probabilities in Table 3. Taking all this into account, we get

$$\sum_{i=1}^{2} u^{\sigma(i)} = 0 + 0.017 \leq 0.05 < 0 + 0.017 + 0.035 = \sum_{i=1}^{2+1} u^{\sigma(i)}$$

$$\sum_{i=5}^{|\Gamma|} m^{\sigma(i)} = 0+0+0+0+0.125 \leq 0.2 < 0+0+0+0+0.125+0.375 = \sum_{i=5-1}^{|\Gamma|} m^{\sigma(i)}$$

Therefore, $limit = 2$ and $limit' = 5$. Thus, $ut = \log(m^{\sigma(limit)}/u^{\sigma(limit)}) = \log(14) = 2.63$ and $lt = \log(m^{\sigma(limit')}/u^{\sigma(limit')}) = \log(3.5) = 1.25$.

According to this, the rule becomes:

1. If $R_p(a,b) \geq 2.63$ then (a,b) is a Linked Pair
2. If $R_p(a,b) \leq 1.25$ then (a,b) is a Non linked Pair
3. If $lt < R_p(a,b) < ut$ then (a,b) is a Clerical Pair

Now, we can consider any pair of records and classify them using this rule. If we take the pair (*J.Gomez*19), (*P.Gomez*19) we first compute the coincidence vector γ. We get $\gamma = (011)$. Then, we need to compute for this vector $R(011)$. Using the values m^i and u^i in Table 3 we get $R(011) = \log(14) = 2.63$. Therefore, the rule classifies the pair as a linked pair.

In this section we have seen how to define the decision rule and how to apply to a pair of records. However, this process requires several conditional probabilities to be determined. One possibility is to start with a pair of records for which the matched pairs are known (as in the examples in this section) and then estimate the probabilities by proportions of records. In the next sections we consider in more detail the computation of $R(a,b)$ and the estimation of the probabilities involved in $R(a,b)$.

3.2 Computation of $R_p(a,b)$

Some general aspects about the computation of $R_p(a,b)$ for a given pair (a,b) in $\mathbf{A} \times \mathbf{B}$ are described in this section. Specifically, the estimation of the probabilities involved in this computation is detailed in Section 3.3. In fact, according to the rule, the computation of $R_p(a,b)$ is solely based on the computation of R for $\gamma(a,b)$.

Due to the fact that the cardinality of Γ is typically quite large (recall that $|\Gamma| = 2^n$ where n is the number of variables), it is not appropriate to directly estimate the probabilities of all γ. To avoid this computation, it is usual to assume that the components of the vector $\gamma = (\gamma_1, \ldots, \gamma_n)$ are statistically independent. Under this assumption, the probabilities $P(\gamma = \gamma(a,b)|(a,b) \in \mathbf{M})$ and $P(\gamma = \gamma(a,b)|(a,b) \in \mathbf{U})$ can be expressed in the following way:

$$P(\gamma = \gamma(a,b)|(a,b) \in \mathbf{M}) = \prod_{i=1,n} P(\gamma_i = \gamma_i(a,b)|(a,b) \in \mathbf{M}) \qquad (21)$$

$$P(\gamma = \gamma(a,b)|(a,b) \in \mathbf{U}) = \prod_{i=1,n} P(\gamma_i = \gamma_i(a,b)|(a,b) \in \mathbf{U}) \qquad (22)$$

To simplify the notation, we shall use the following equivalences:

- $m(\gamma) = P(\gamma = \gamma(a,b)|(a,b) \in \mathbf{M})$
- $m_i(\gamma_i) = P(\gamma_i = \gamma_i(a,b)|(a,b) \in \mathbf{M})$
- $u(\gamma) = P(\gamma = \gamma(a,b)|(a,b) \in \mathbf{U})$
- $u_i(\gamma_i) = P(\gamma_i = \gamma_i(a,b)|(a,b) \in \mathbf{U})$

Using these equivalences, Equations 21 and 22 are rewritten as:

$$m(\gamma) = \prod_{i=1,n} m_i(\gamma_i)$$

$$u(\gamma) = \prod_{i=1,n} u_i(\gamma_i)$$

Therefore, under the same conditions of independence $R_p(a,b)$ can be rewritten as:

$$R_p(a,b) = R(\gamma(a,b)) = \log(\frac{P(\gamma(a,b) = \gamma(a',b')|(a',b') \in \mathbf{M})}{P(\gamma(a,b) = \gamma(a',b')|(a',b') \in \mathbf{U})}) \quad (23)$$

$$= \log(\frac{m(\gamma(a,b))}{u(\gamma(a,b))}) \quad (24)$$

$$= \log(\frac{\prod_{i=1,n} m_i(\gamma_i(a,b))}{\prod_{i=1,n} u_i(\gamma_i(a,b))}) \quad (25)$$

$$= \sum_{i=1,n} \log(m_i(\gamma_i(a,b))/u_i(\gamma_i(a,b))) \quad (26)$$

Note that the use of logarithm in $R(\gamma)$ simplifies its expression. Now, using that the following expressions about conditional probabilities hold for all $i \in \{1,\ldots,n\}$

$$P(\gamma_i = 1|(a,b) \in \mathbf{M}) + P(\gamma_i = 0|(a,b) \in \mathbf{M}) = m_i(1) + m_i(0) = 1$$

$$P(\gamma_i = 1|(a,b) \in \mathbf{U}) + P(\gamma_i = 0|(a,b) \in \mathbf{U}) = u_i(1) + u_i(0) = 1$$

we define

- $m_i = m_i(1)$
- $u_i = u_i(1)$

and express $m_i(0)$ and $u_i(0)$ as:

$$m_i(0) = 1 - m_i$$

$$u_i(0) = 1 - u_i$$

These definitions permits to express the conditional probabilities in an alternative and more compact way (note that here γ_i and $1 - \gamma_i$ is either 1 or 0):

$$P(\gamma = \gamma(a',b')|(a',b') \in \mathbf{M}) = \prod m_i^{\gamma_i}(1 - m_i)^{1-\gamma_i}$$

$$P(\gamma = \gamma(a', b')|(a', b') \in \mathbf{U}) = \prod u_i^{\gamma_i}(1 - u_i)^{1-\gamma_i}$$

By further defining $w_i(\gamma_i)$ as $\log(m_i(\gamma_i)/u_i(\gamma_i))$, we have that $R(a, b)$ in Expression 26 can be rewritten as:

$$R_p(a, b) = R(\gamma(a, b)) = \sum_{i=1}^{n} w_i(\gamma_i(a, b)) \tag{27}$$

Thanks to the above definitions, we only need m^i and u^i to compute $w_i(\gamma_i(a, b))$. To do so, two cases are considered:

Case $\gamma_i(a, b) = 1$**:** define $w_i(1) = log(m_i/u_i)$.
Case $\gamma_i(a, b) = 0$**:** define $w_i(0) = log((1 - m_i)/(1 - u_i))$.

Note that these expressions are correct because $w_i(1)$ is equal to $\log(m_i(1)/u_i(1)$ and $m_i = m_i(1)$ and $u_i = u_i(0)$. Similary, $w_i(0)$ is equal to $\log(m_i(0)/u_i(0)$ and, thus, considering the equalities $m_i(0) = 1 - m_i$ and $u_i(0) = 1 - u_i$ we get the expression above.

The terms $w_i(\gamma_i(a, b))$ are known as the *weights* of $\gamma_i(a, b)$. As the usual case is to have $m_i > u_i$, then, the variables with coincident values (*i.e.* with $\gamma_i = 1$) contribute positively to the value $R(\gamma)$. Instead, variables with non-coincident values (i.e., with $\gamma_i = 0$) contribute negatively to the value $R(\gamma)$.

In fact, expressions for $w_i(1)$ and $w_i(0)$ given above clearly show that, when $m_i > u_i$, the weights for $\gamma_i(a, b) = 1$ are positive (and thus contribute positively to $R_p(a, b)$) and the weights for $\gamma_i(a, b) = 0$ are negative (and thus contribute negatively to $R_p(a, b)$). This is stated in the next proposition.

Proposition 4. *Let* $m_i > u_i$*, then* $w_i(1) > 0$ *and* $w_i(0) < 0$.

Proof. $m_i > u_i$ implies $m_i/u_i > 1$, therefore $w_i(1) = \log \frac{m_i}{u_i} > \log 1 = 0$. Also, it implies $1 - u_i > 1 - m_i > 0$, therefore, $1 > \frac{1-m_i}{1-u_i}$ and, thus, $0 = \log 1 > \log \frac{1-m_i}{1-u_i}$

Let us now turn into the estimation of probabilities m_i and u_i for all $i \in \{1, \ldots, n\}$.

3.3 Estimation of the probabilities

The estimation of the probabilities involved in the computation of R_p is usually based on the EM (Expectation-Maximization) algorithm [4]. In this section, we describe this method. We start reviewing the maximum likelihood model, then we describe the EM algorithm and we finish with its application to the record linkage process. The section finishes with the computation of the thresholds.

Likelihood function and maximum likelihood The maximum likelihood is a method for estimating the parameters of a given probability density. Let us consider a probability density $f(z|\theta)$. This is, f is a parametric model of the random variable z with parameter θ (or, parameters, because θ can be a vector). Let $\mathbf{z} = \{z_1, \ldots, z_e\}$ be a sample of the variable z. Then, the *likelihood* of z under a particular model $f(z|\theta)$ is expressed by:

$$f(\mathbf{z} = (z_1, \ldots, z_e)|\theta) = \prod_{i=1}^{e} f(z_i|\theta)$$

This is, $f(\mathbf{z}|\theta)$ is the probability of the sample $\mathbf{z}$ under the particular model $f(z_i|\theta)$ with a particular parameter θ. The likelihood function is the function above when the sample is taken as constant and θ is the variable. This is denoted by $L(\theta|\mathbf{z})$. Thus,

$$L(\theta|\mathbf{z}) = \prod_{i=1}^{e} f(z_i|\theta)$$

Often, the *log-likelihood* function is used instead of the likelihood function. The former is the logarithm of the latter and is denoted by $l(\theta|\mathbf{z})$ (or, sometimes, by $l(\theta)$). Therefore,

$$l(\theta|\mathbf{z}) = \log L(\theta|\mathbf{z}) = \log \prod_{i=1}^{e} f(z_i|\theta) = \sum_{i=1}^{e} \log f(z_i|\theta)$$

Given a sample $\mathbf{z}$ and a model $f(\mathbf{z}|\theta)$, the maximum likelihood estimate of the parameter θ is the $\hat{\theta}$ that maximizes $l(\theta|\mathbf{z})$. Equivalently, the estimate is $\hat{\theta}$ such that

$$l(\theta|\mathbf{z}) \leq l(\hat{\theta}|\mathbf{z})$$

EM algorithm The EM algorithm [4] (where EM stands for Expectation-Maximization) is an iterative process for the computation of maximum likelihood estimates. The method starts with an initial estimation of the parameters and then in a sequence of two step iterations builds more accurate estimations. The two steps considered are the so-called Expectation step and Maximization step.

The algorithm is based on the consideration of two sample spaces $\mathcal{Y}$ and $\mathcal{X}$ and a many-to-one mapping from $\mathcal{X}$ to $\mathcal{Y}$. We use y to denote this mapping, and $X(y)$ to denote the set $\{x|y = y(x)\}$. Only data y in $\mathcal{Y}$ are observed, and data x in $\mathcal{X}$ are only observed indirectly through y. Due to this, x are referred to as complete data and y as the observed data.

Let $f(x|\theta)$ be a family of sampling densities for x with parameter θ, it is clear that the corresponding family of sampling densities $g(y|\theta)$ can be

computed from $f(x|\theta)$ as follows:

$$g(y|\theta) = \int_{X(y)} f(x|\theta)dx$$

Now, roughly speaking, the expectation step consists on estimating the complete data x and the maximization step consists on finding a new estimation of the parameters θ by maximum likelihood. In this way, the EM algorithm tries to find the value θ that maximizes $g(y|\theta)$ given an observation y. However, the method also uses $f(x|\theta)$.

EM algorithm for record linkage The application of EM to record linkage relies on the consideration of pairs of vectors $< \gamma(r), c(r) >_{r\in\mathbf{A}\times\mathbf{B}}$ as the complete data. Here, as usual, γ is the coincidence vector for $r \in \mathbf{A} \times \mathbf{B}$ and c (c for class) is a two dimensional vector $c = (c_m c_u)$ in $\{(10), (01)\}$ to indicate whether r belongs to $\mathbf{M}$ or $\mathbf{U}$. Then, for all pairs of records r in $\mathbf{A} \times \mathbf{B}$ we consider $(\gamma(r), c(r))$ where $c(r) = (10)$ if and only if $r \in \mathbf{M}$ and $c(r) = (01)$ if and only if $r \in \mathbf{U}$.

Incomplete data correspond to the case that some vectors c are unknown for some records r. Then, the expectation step assigns to the missing indicators fractions that sum to unity that are expectations given the current estimate of the parameters.

Here, the parameters θ consist of probabilities $m = (m_1, \ldots, m_n)$ and $u = (u_1, \ldots, u_n)$ with $m_i = P(1 = \gamma_i(a,b)|(a,b) \in \mathbf{M})$ and $u_i = P(1 = \gamma_i(a,b)|(a,b) \in \mathbf{U})$ (as defined in Section 3.2). Additionally, the parameters θ also contains p (the proportion of matched pairs $p = |M|/|M \cup U|$). Therefore, $\theta = (m, u, p)$.

Then, the log-likelihood for the complete data corresponds to [4]:

$$\ln f(\mathbf{x}|\theta) = \sum_{j=1}^{N} c(r^j)(\ln P\{\gamma(r^j)|\mathbf{M}\}, \ln P\{\gamma(r^j)|\mathbf{U}\})^T + \sum_{j=1}^{N} g(r^j)(\ln p, \ln(1-p))^T$$

This expression allows us to estimate the probabilities of assigning records in $\mathbf{A} \times \mathbf{B}$ either to the class $\mathbf{M}$ or $\mathbf{U}$ once an estimation for the parameters $\theta = (m, u, p)$ is given. In fact, the assignment does only depend on the corresponding coincidence vector. Therefore, the estimation is computed for the coincidence vectors $\gamma^j \in \Gamma$. This is the expectation step (see [14]), which yields the following assignment probabilities:

$$\hat{c}_m(\gamma^j) = \frac{\hat{p} \prod_{i=1}^{n} \hat{m}_i^{\gamma_i^j} (1 - \hat{m}_i)^{1-\gamma_i^j}}{\hat{p} \prod_{i=1}^{n} \hat{m}_i^{\gamma_i^j} (1 - \hat{m}_i)^{1-\gamma_i^j} + (1 - \hat{p}) \prod_{i=1}^{n} \hat{u}_i^{\gamma_i^j} (1 - \hat{u}_i)^{(1-\gamma_i^j)}} \quad (28)$$

$$\hat{c}_u(\gamma^j) = \frac{(1-\hat{p})\prod_{i=1}^{n}\hat{u}_i^{\gamma_i^j}(1-\hat{u}_i)^{1-\gamma_i^j}}{\hat{p}\prod_{i=1}^{n}\hat{m}_i^{\gamma_i^j}(1-\hat{m}_i)^{1-\gamma_i^j} + (1-\hat{p})\prod_{i=1}^{n}\hat{u}_i^{\gamma_i^j}(1-\hat{u}_i)^{(1-\gamma_i^j)}} \quad (29)$$

Then, in the maximization step, we need to calculate a new estimation of the parameters in θ. Therefore, we compute $\hat{m}_i$, $\hat{u}_i$ for all variables $i \in \{1, ..., n\}$ and recompute $\hat{p}$. This is done using the following equations (see [14]):

$$\hat{m}_i = \frac{\sum_{j=1}^{N}[\hat{c}_m(\gamma(r^j))\gamma_i(r^j)]}{\sum_{j=1}^{N}[\hat{c}_m(\gamma(r^j))]} \quad (30)$$

$$\hat{u}_i = \frac{\sum_{j=1}^{N}[\hat{c}_u(\gamma(r^j))\gamma_i(r^j)]}{\sum_{j=1}^{N}[\hat{c}_u(\gamma(r^j))]} \quad (31)$$

$$\hat{p} = \frac{\sum_{j=1}^{N}[\hat{c}_m(\gamma(r^j))]}{N} \quad (32)$$

Although the latter equations are written to consider all pairs of records in $\mathbf{A} \times \mathbf{B}$, it is advisable to accumulate the frequencies of each coincidence vector γ and use alternative expressions. If $fq(\gamma^j)$ is the frequency of the γ^j coincidence vector, then equations above for $\hat{m}_i$, $\hat{u}_i$ and $\hat{p}$ can be rewritten as:

$$\hat{m}_i = \frac{\sum_{j=1}^{2^n}[\hat{c}_m(\gamma^j)\gamma_i^j fq(\gamma^j)]}{\sum_{j=1}^{2^n}[\hat{c}_m(\gamma^j) fq(\gamma^j)]} \quad (33)$$

$$\hat{u}_i = \frac{\sum_{j=1}^{2^n}[\hat{c}_u(\gamma^j)\gamma_i^j fq(\gamma^j)]}{\sum_{j=1}^{2^n}[\hat{c}_u(\gamma^j) fq(\gamma^j)]} \quad (34)$$

$$\hat{p} = \frac{\sum_{j=1}^{2^n}[\hat{c}_m(\gamma^j) fq(\gamma^j)]}{\sum_{i=1}^{2^n} fq(\gamma^j)} \quad (35)$$

Initialization step In [14], it is stated that the algorithm is not very sensitive to initial values for m and u, although values $m_i > u_i$ are advisable. $m_i = 0.9$ was reported in [14] and $m_i = 0.9$ and $u_i = 0.1$ were used in [7] and [9].

4 Distance-based Record Linkage

This approach, described in [22] in a very restricted formulation for disclosure risk assessment, consists of computing distances between records in the two data files being considered. Then, the pair of records at minimum distance are considered linked pairs. We give below its formulation.

Let $d(a, b)$ be a distance function between records in file **A** and file **B**. Then, the distance-based record linkage is defined in the following way:

for all $a \in \mathbf{A}$
begin
  b' = arg $\min_{b \in \mathbf{B}} d(a, b)$
  $\mathbf{LP} = \mathbf{LP} \cup (a, b')$
  for all $b \in \mathbf{B}$ **such that** $b \neq b'$
  begin
    $\mathbf{NP} = \mathbf{NP} \cup (a, b)$
  end
end

Naturally, application of the approach relies on the existence of the distance function. Thus, a distance is assumed in each variable V_i. We denote this distance by d_{V_i}. The following distance has been considered:

Definition 5. Let $\mathbf{V} = \{V_1, V_2, \cdots V_n\}$ be the set of variables and let d_{V_i} be a distance on the range of $D(V_i)$. Then, assuming equal weight for all variables, the distance between records a and b is defined by:

$$d(a, b) = \sum_{i=1}^{n} d_{V_i}(V_i^A(a), V_i^B(b))$$

Several alternatives can be considered as within-variable distances d_V. In particular, depending on the type of variable, the following distances have been used [7], [9]:

Definition 6. 1. For a numerical variable V, the Euclidean distance d_E is used. Thus, $d_V = d_E$. In order to avoid scaling problems, it is convenient to standardize the Euclidean distance.

2. For a nominal variable V, the only permitted operation is comparison for equality. This leads to the following distance definition:

$$d_V(c, c') = \begin{cases} 0 \text{ if } c = c' \\ 1 \text{ if } c \neq c' \end{cases}$$

where c and c' correspond to categories for variable V.

3. For an ordinal variable V, let $\leq_V$ be the total order operator over the range of V. Then, the distance between categories c and c' is defined as the number of categories between the minimum and the maximum of c and c' divided by the cardinality of the range:

$$d_V(c, c') = \frac{|c'' : min(c, c') \leq_V c'' \leq_V max(c, c')|}{|D(V)|}$$

4.1 Discussion

The performance of probabilistic and distance-based record linkage is compared using data from a public repository. The results of two experiments, one for numerical data and one for categorical data, are explained below. In short, a data file has been masked (*e.g.*, distorted) using some data protection mechanisms and then record linkage programs have been applied to the pair (original file, masked file). The number of correctly linked recods give a measure of record linkage performance.

Two data files were obtained from the U. S. Census Bureau Data Extraction System (DES [5]). One with numerical data and the other with categorical data. In the numerical case, we used records from the Current Population Survey (corresponding to 1995). In the categorical case, data from the American Housing Survey 1993 was used. Details on the variable and record selection are explained in [7].

Masking methods were applied to both files to obtain several masked files. Masking methods are applied by National Statistical Offices to perturb the original data so that data is protected and the anonymity of the original respondents is assured. Several masking methods with several parameterizations were applied to the original data and for each pair (method, parameterization) a perturbed data file was obtained. On the basis of the two original data files (*i.e.*, numerical and categorical) two groups of masked files can be distinguished.

The next step is to apply probabilistic record linkage and distance-based record linkage to each pair of masked file and original file. Tables 5 and 6 give the percentage of re-identified records using distance-based and probabilistic record linkage for the numerical data. Tables 8 and 9 give the percentage of re-identified records, again for distance-based and probabilistic record linkage, for categorical data. Tables 7 and 7 give the difference between distance-based and probabilistic record linkage.

The results obtained in these tables show that in the numerical case, distance-based record linkage outperforms probabilistic record linkage. In the categorical case, however, probabilistic and distance-based record linkage lead to quite similar results, being the former slightly better[2].

[2] These results complement the ones in [7] and [9] where high correlations between record linkage and several information loss measures were obtained. Therefore, both methods were considered similar for measuring disclosure risk

5 Technical issues

In this section we review several aspects that have to be taken in account when building record linkage implementations and when applying these methods. First, we consider the standardization process. This process is applied to names and addresses so that the probability of linking matched pairs increases. Then, we describe the use of blocking variables to reduce the number of pairs to be considered. The section continues with a review of some algorithms for string comparison and for partial agreement. The last part of the section is a discussion about the need of taking advantage of variables with non-uniform probability distributions.

5.1 Standardization of variables

To increase the performance of record linkage, standardization of some of the variables is recommended (in particular, variables corresponding to names, addresses and places). This process is required so that different forms of the same name (*e.g.*, Robert, Bob), company names (*e.g.*, Limited, Ltd., LTD) and addresses (*e.g.* street, st.) are transformed into a single form. If this process is not accomplished in an effective way, it is possible to classify as unlinked pairs some pairs that correspond to the same individual. Standardization consists of the three procedures below:

1. Parse variables to build a uniform structure
2. Detect relevant keywords to help in the process of recognizing the components that form the values of a variable
3. Replace all the (common) forms of a word by a single one (for example, an abreviation).

The goal of parsing is to ensure that, when the value of the variable consists of several elements, these always appear in the same order. For example "Robert Green, PhD", "Dr. Bob Green" and "Green, Robert" are translated into "PhD Robert Green", "Dr. Bob Green" and " Robert Green", respectively, following a *title + name + surname* structure.

The detection of special keywords can help in this process. For example, detection of "Ms", "PhD" or "Dr" is usually an indication of the presence of a personal name and "Ltd" indicates the presence of a company name. Detection would trigger specific parsing routines when appropriate.

The third standardization procedure replaces variants of values by a standard form. Depending on the meaning and the values of the variable, this procedure can either be applied to the whole variable value (*e.g.*, to the string used to represent the name) or to components of the variable value. This latter case occurs when the variable corresponds to personal names and they include for example title and middle letters, or when the variable is an address with street names, numbers or a P. O. box.

The substitutions required by standardization can be efficiently implemented by building a database with lists of words and their corresponding standard form so that the forms that appear in the files can be *replaced* by the standard ones. It is important to note that this *standard* form does not need to be a "dictionary" form (the root of a word or any not-shortened version of the name) but only an abstract identifier. This abstract identifier can be useful when a single spelling can have different origins (*e.g.* Bobbie might refer to Robert but also to Roberta).

For details on standardization see [28]. Examples of name and address parsing are provided there.

5.2 Blocking variables

When the files to be linked contain a huge number of records, consideration of all possible pairs is rather costly. This is so because $|\mathbf{A}| \cdot |\mathbf{B}|$ pairs have to be considered, where $\mathbf{A}$ and $\mathbf{B}$ are the files to be linked. Moreover, when each individual appears once in a file, only $\min(|\mathbf{A}|, |\mathbf{B})|$ pairs can be effectively linked. To avoid most of the unsuccessful comparisons, the so-called blocking variables (or blocking variables) are sometimes considered.

The set of blocking variables is selected by the user among the most error-resilient variables present in both files (those variables most likely to maintain their values across files). Given a set of blocking variables, comparison between pairs is restricted to those pairs with equal values for all blocking variables. In this way, the number of comparisons is largely reduced. To do such a comparison, files are usually ordered according to blocking variables. In this way, records to be compared are found in an easy way.

Naturally, when the blocking variables also contain errors, some of the linked pairs are not detected. These pairs are the so-called *missed matches*. Therefore, an unsuitable selection of the blocking variables results in a large number of missed matches.

A typical example of a blocking variable is the ZIP code. Nevertheless, it is also possible to use some string variables. In this case, a good alternative is to use the first letter or a particular coding so that all the symbols with a similar sound are mapped onto the same block (for example, the SOUNDEX codification – see Section 5.3).

To mitigate the negative effects of selecting a particular set of blocking variables (it is almost impossible to find error-free variables!), a good strategy is to apply several times the record linkage method using in each iteration a set of blocking variables independent from those used in previous iterations. Nevertheless, this process increases the complexity of the procedure. In any case, the use of blocking variables corresponds to a compromise between a high-cost detailed analysis of all possible pairs with few missed matches and a low-cost analysis of only a few pairs with more missed matches.

Blocking variables can also be used in combination with the EM algorithm. In fact, Jaro [14] used blocking variables when storing frequency counts

in the EM algorithm in order to reduce computation. In such an approach, each file is partitioned into several blocks such that all records in a block have the same value for all the blocking variables. Then, pairs of records (one from each file) are only considered within the blocks with the same values for the blocking variables. For these pairs, coincidences are examined and counts are updated (the vector γ is computed and 1 is added to the frequency count for that particular configuration). Note that counters are not reset after a block is processed. In this way, counts represent the number of observations of each configuration over all blocks. However, it is important to note that, in this approach, counts are not the same that would be obtained without blocks. In fact, blocking reduces the number of unmatched pairs and, therefore, the probabilities u_i will be underestimated. To avoid underestimation, the probability

$$u_i = P(1 = \gamma_i(a,b) | (a,b) \in U)$$

is estimated by:

$$\hat{u}_i = P(1 = \gamma_i(a,b))$$

The above probability is the probability of $V_i(a) = V_i(b)$ and can be computed directly from the files **A** and **B** counting, for example, the number of pairs with the same value for the variable V_i:

$$\hat{u}_i = \frac{|\{V_i(a) = V_i(b)\}|}{N}$$

5.3 Partial coincidence and string comparison

In Section 3, we have described record linkage. We have only considered there the case of total coincidence between records (coincidence or non-coincidence). In practical situations, dealing with partial coincidence is also required. This topic is discussed in this section.

We start reviewing some algorithms for string comparison. First, we define the SOUNDEX method that transforms a string into a code that tends to bring together all variants of the same name. Therefore, the application of this method for string comparison leads to a Boolean comparison (strings are either encoded in the same or in a different way). Later on we review other string comparison methods that lead to values in the unit interval. The section finishes explaining how to adapt weights in probabilistic record linkage to accomodate partial coincidence.

SOUNDEX method A description of this method, originally developed by M. K. Odell and R. C. Russell [20,21], can be found in [15]. The method consists of transforming strings into a four character sequence. For example,

both strings "Smith" and "Smythe" are encoded as "S530". Then, comparison between strings is achieved by means of comparison of sequences.

This coding has been used to deal with surnames. Jaro [14] recommends its use as a blocking variable:

> "To maximize the chance that similarly spelled surnames reside in the same block, the SOUNDEX system can be used to code the names, and the SOUNDEX code can be used as a blocking variable. There are better encoding schemes than SOUNDEX, but SOUNDEX with relatively few states and poor discrimination helps ensure that misspelled names receive the same code" (p. 418 in [14])

However, in the case of non-blocking variables, [14] does not recommend this coding because nonphonetic errors result in different codes, and, therefore, variants of the same name may receive different codes. The coding is said to be quite effective [18] except when the names are predominantly of Oriental origin.

Now we turn to the description of the method. As said above, the method transforms any string into a sequence of one character and three digits. The encoding rules are as follows:

1. The first letter of the string is selected and used as the first character of the codification.
2. Vowels A, E, I, O, U and letter Y are not encoded. Letters W and H are also ignored.
3. All the other letters are encoded as follows:

B, F, P, V	encoded as 1
C, G, J, K, Q, S, X, Z	encoded as 2
D, T	encoded as 3
L	encoded as 4
M, N	encoded as 5
R	encoded as 6

4. When the coding results into two or more adjacent codes with the same value only one code is kept. The others are removed. *E. g.,* "S22" is reduced to "S2" and "S221" to "S21".
5. All strings are encoded into a string with the following structure: `Letter, digit, digit, digit`. Additional elements are truncated and in case the string is too short, additional "0" are appended.

Table 11 displays some examples taken from Knuth [15]. Examples of pairs of surnames that do not lead to the samme codification include *(Rogers, Rodgers)* and *(Tchebysheff, Chebyshev)*.

There exist other methods that proceed in a way similar to SOUNDEX by transforming a large number of strings into a single codification. These

methods are classified in [24] as hashing techniques (a description of hash functions, very common in data structures, can be found *e.g.* in [1]). For example, Blair [3] builds the so-called r-letter abbreviations. This procedure transforms all strings s to r-letter strings removing $length(s) - r$ irrelevant characters. In this method, relevance of a character is computed in terms of relevance of letters (*e.g.*, "A" has relevance 5 and "B" relevance 1) and relevance of position (*e.g.* relevance of second position is larger than relevance of first position). Some example codings for 4-letter abbreviations are: *Euler* and *Ellery* are transformed to `ELER` and *Tchebysheff* and *Chebyshev* are transformed to `ESHE`. *Rogers* is translated to `OERS` and *Rodgers* can either be translated to `OERS` or `GERS` (letters "O" and "G" in *Rodgers* have the same importance but only one of them can be deleted).

Bigrams An alternative method for measuring string similarity is the one based on the comparison of the so-called bigrams. A bigram is defined as a pair of consecutive letters in a string. Therefore, the word `bigram` contains the following bigrams: `bi, ig, gr, ra, am`. The value of the function $simB$ applied to two strings *s1* and *s2* is a value in the $[0, 1]$ interval corresponding to the number of bigrams in common divided by the mean value of bigrams in both strings:

$$simB(s1, s2) = \frac{|bigrams(s1) \cap bigrams(s2)|}{(|bigrams(s1)| + |bigrams(s2)|)/2}$$

where $bigrams(s)$ corresponds to the bigrams in string s.

Naturally, this function defines a similarity function and is equal to 1 when both strings are equal.

As said above, bigrams correspond to two consecutive characters. In fact, the literature also considers the general structure of n-grams (n consecutive characters in a string – a substring of length n). Similarity measures have been considered for n-grams with $n > 2$. In particular, there exists one such measures for the so-called trigrams (n-grams for $n = 3$).

Jaro algorithm The algorithm, introduced in [13], consists of the following steps when applied to strings $s1$ and $s2$:

1. Compute the length of the strings $s1$ and $s2$ ($strLen1 = length(s1)$, $strLen2 = length(s2)$).
2. Find the number of common characters. These characters are the ones that appear in both strings at a distance that is at most $minLen/2$ where $minLen = \min(strLen1, strLen2)$:

   $$common = \{c | c \in chars(s1) \cap chars(s2) \text{ and } pos(s1) - pos(s2) \leq minLen/2\}$$

3. Find the number of transpositions among the common characters. A transposition happens whenever a common character from one string does

not appear in the same position as the corresponding character from the other string. Let *trans* be the number of transpositions.

Then, the Jaro similarity is defined as follows:

$$jaro(s1, s2) = \frac{1}{3}\left(\frac{common}{strLen1} + \frac{common}{strLen2} + \frac{1}{2}\frac{trans}{common}\right)$$

McLaughlin, Winkler and Lynch have studied this similarity and defined some enhancements which, when combined with record linkage, improve the performance of the latter. The enhancements and results are given in [23].

Dynamic programming methods Another approach for computing similarities between strings is based on dynamic programming. Below we describe the algorithm for computing the Levenshtein distance between two strings [16] (see also [2] or [24]). This distance is defined on any pair of strings (not necessarily of the same length) and gives the same weight (assumed to be 1 in the algorithm below) to insertions, deletions and substitutions:

for $i = 0$ **to** m **do**
begin
  d[i,0] = i;
end
for $i = 0$ **to** n **do**
begin
  d[0,i] = 0;
end
for $j = 1$ **to** n **do**
begin
  for $i = 0$ **to** m **do**
  begin
    if $s1[i] == s2[j]$ **then**
      begin
        $d[i, j] = d[i - 1, j - 1]$;
      end
    else
      begin
        $d[i, j] = 1 + \min(d[i - 1, j], d[i, j - 1], d[i - 1, j - 1])$;
      end
    end if
  end
end
return $d[m, n]$;

From the above distance, a similarity can be computed using a non-increasing function such that $f(0) = 1$. Improvements of this method exist so that the computation time and the working space requirement are reduced. See [24] for details.

Adapting weights w_i If partial coincidence is considered when comparing variable values, the weights attached to variables (w_i following the notation in Section 3.2) must be updated according to the coincidence. Usually, the update is proportional to the similarity (for example, multiplying the weight by the similarity – $w_i'(\gamma_i(a,b)) = w_i(\gamma_i(a,b)) \cdot similarity(V_i(a), V_i(b))$). Moreover, to improve the performance of the method, updating the similarity function by applying a particular transformation f is sometimes required. Therefore, an expression similar to the one below is used:

$$w_i'(\gamma_i(a,b)) = w_i(\gamma_i(a,b)) \cdot f(similarity(V_i(a), V_i(b)))$$

A particular example of the transformation function is the following one used in [23]:

$$f(x) = \begin{cases} x^{0.2435} & \text{if } x > 0.8 \\ 0.0 & \text{if } x \leq 0.8 \end{cases}$$

When a file with known coincidences is available, it is possible to learn these functions from the examples in that file.

Variables with values not following uniform probability distributions An important aspect to be considered when defining matching probabilities is that not all values in the range of a variable occur with the same probability. This is obvious in the case of names and surnames (*e.g.*, the probability of finding the Japanese surname Tanaka in Catalonia is very low). Newcombe [17] introduced a method to take into account the frequencies when computing the weights. The intuitive idea is that coincidence on surnames with a large frequency is less relevant than coincidence on less frequent surnames. Therefore, the probability of being a linked pair when a less frequent surname is detected is larger than when it is a high frequency surname.

6 Conclusions

In this chapter we have reviewed main record linkage techniques (probabilistic and distance-based ones) and we have compared their results for both numerical and categorical data. While distance-based record linkage seems to be more appropriate for numerical data, probabilistic based one seems more appropriate in the case of categorical data.

Acknowledgment

Partial support of the European Community under the contract "CASC" IST-2000-25069 and of the Spanish Ministry of Science and Technology under the project "STREAMOBILE" (TIC2001-0633-C03-01/02) is acknowledged.

References

1. Aho, A. V., Hopcroft, J. E., Ullman, J. D., (1988), Data Structures and Algorithms, Addison-Wesley, USA
2. Baeza-Yates, R., Ribeiro-Neto, B., (1999), *Modern Information Retrieval*, Addison-Wesley, England.
3. Blair, C. R., (1960), A Program for Correcting Spelling Errors, *Information and Control*, 3 60-67.
4. Dempster, A. P., Laird, N. M., Rubin, D. B., (1977), Maximum Likelihood From Incomplete Data Via the EM Algorithm, *Journal of the Royal Statistical Society*, 39, 1-38.
5. DES, (2002), Data Extraction System, U. S. Census Bureau, http://www.census.gov/DES/www/welcome.html
6. Domingo-Ferrer, J., Torra, V., (2001), Disclosure Control Methods and Information Loss for Microdata, 91-110, in *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. M. Zayatz (Eds.), Elsevier.
7. Domingo-Ferrer, J., Torra, V., (2001), A Quantitative Comparison of Disclosure Control Methods for Microdata, 111-133, in *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, P. Doyle, J. I. Lane, J. J. M. Theeuwes, L. M. Zayatz (Eds.), Elsevier.
8. Domingo, J., Torra, V., (2002), Aggregation techniques for statistical confidentiality, in *Aggregation operators: New trends and applications*, (R. Mesiar, T. Calvo, G. Mayor, eds.), Heidelberg: Physica-Verlag, pp. 261-271.
9. Domingo-Ferrer, J., Torra, V., (2002), Validating distance-based record linkage with probabilistic record linkage, *Lecture Notes in Computer Science*, vol. 2504, pp. 207-215, 2002.
10. Domingo-Ferrer, J., Torra, V., (2003), Disclosure risk assessment in statistical disclosure control via advanced record linkage, *Statistics and Computing* (to appear).
11. Fellegi, I. P., Sunter, A. B., (1969), A theory for record linkage, *Journal of the American Statistical Association*, 64:328, 1183-1210.
12. Gill, L., (2001), *Methods for Automatic Record Matching and Linking and Their Use in National Statistics*, National Statistics Methodology Series no. 25, London: Office for National Statistics.
13. Jaro, M. A., (1978), *UNIMATCH: A record linkage system: User's Manual*, U. S. Bureau of the Census, Washington DC.
14. Jaro, M. A., (1989), 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.
15. Knuth, D. E., (1973), *The Art of Computer Programming Vol. 3: Sorting and Searching*, Reading, MA: Addison-Wesley.
16. Levenshtein, V. I., (1965), Binary codes capable of correcting deletions, insertions, and reversals, *Doklady Academii nauk SSSR*, 163:4 845–8 (in Russian) (also in *Cybernetics and Control Theory*, 10:8 (1966) 707-10).
17. Newcombe, H. B., Kennedy, J. M., Axford, S. J., James, A. P., (1959), Automatic linkage of vital records, *Science*, 130, 954-959.
18. Newcombe, H. B. (1967), Record linking: the design of efficient systems for linking records into individuals and family histories, *American Journal of Human Genetics*, 19:3, part I.

19. Newcombe, H. B., (1988), *Handbook of Record Linkage*, Oxford University Press.
20. Odell, M. K., Russell, R. C., (1918), U. S. Patents 1261167
21. Odell, M. K., Russell, R. C., (1922), U. S. Patents 1435663
22. Pagliuca, D., Seri, G., (1999), *Some Results of Individual Ranking Method on the System of Enterprise Accounts Annual Survey*, Esprit SDC Project, Deliverable MI-3/D2.
23. Porter, E. H., Winkler, W. E., (1997), Approximate string comparison and its effect on an advanced record linkage system, Report RR97/02, Statistical Research Division, U. S. Bureau of the Census, USA.
24. Stephen, G. A., (1994), *String Searching Algorithms*, World Scientific Publishing Co, Singapore.
25. Torra, V., (2000), Towards the re-identification of individuals in data files with common variables, *Proc. of the 14th European Conference on Artificial Intelligence (ECAI2000)*, Berlin, Germany.
26. Torra, V., (2000a), Re-identifying individuals using OWA operators, *Proc. of the 6th Int. Conference on Soft Computing*, Iizuka, Fukuoka, Japan.
27. Torra, V., (2000b), On the use of aggregation operators in Data Mining, submitted.
28. Winkler, W. E., (1993), Matching and record linkage, Report RR93/08, Statistical Research Division, U. S. Bureau of the Census, USA.
29. Winkler, W. E., Thibaudeau, Y., (1991), An application of the Fellegi-Sunter model of record linkage to the 1990 U. S. Decennial Census, Report, Statistical Research Division, U. S. Bureau of the Census, USA.

$Name^{\mathbf{A}}$	$Surname^{\mathbf{A}}$	$Age^{\mathbf{A}}$	$Name^{\mathbf{B}}$	$Surname^{\mathbf{B}}$	$Age^{\mathbf{B}}$	$\gamma(a,b)$	$\gamma(a,b)$
Joan	Casanoves	19	Joan	Casanovas	19	101	γ^6
Joan	Casanoves	19	Pere	Joan	18	000	γ^1
Joan	Casanoves	19	J.Manel	Casanovas	35	010	γ^3
Joan	Casanoves	19	Juan	Garcia	53	000	γ^1
Joan	Casanoves	19	Ricard	Garcia	14	000	γ^1
Joan	Casanoves	19	Pere	Garcia	82	000	γ^1
Joan	Casanoves	19	Juan	Garcia	18	000	γ^1
Joan	Casanoves	19	Ricard	Tanaka	18	000	γ^1
Pere	Joan	17	Joan	Casanovas	19	000	γ^1
Pere	Joan	17	Pere	Joan	18	110	γ^7
Pere	Joan	17	J.Manel	Casanovas	35	000	γ^1
Pere	Joan	17	Juan	Garcia	53	000	γ^1
Pere	Joan	17	Ricard	Garcia	14	000	γ^1
Pere	Joan	17	Pere	Garcia	82	100	γ^5
Pere	Joan	17	Juan	Garcia	18	000	γ^1
Pere	Joan	17	Ricard	Tanaka	18	000	γ^1
J.M.	Casanovas	35	Joan	Casanovas	19	010	γ^3
J.M.	Casanovas	35	Pere	Joan	18	000	γ^1
J.M.	Casanovas	35	J.Manel	Casanovas	35	011	γ^4
J.M.	Casanovas	35	Juan	Garcia	53	000	γ^1
J.M.	Casanovas	35	Ricard	Garcia	14	000	γ^1
J.M.	Casanovas	35	Pere	Garcia	82	000	γ^1
J.M.	Casanovas	35	Juan	Garcia	18	000	γ^1
J.M.	Casanovas	35	Ricard	Tanaka	18	000	γ^1
Juan	Garcia	53	Joan	Casanovas	19	000	γ^1
Juan	Garcia	53	Pere	Joan	18	000	γ^1
Juan	Garcia	53	J.Manel	Casanovas	35	000	γ^1
Juan	Garcia	53	Juan	Garcia	53	111	γ^8
Juan	Garcia	53	Ricard	Garcia	14	010	γ^3
Juan	Garcia	53	Pere	Garcia	82	010	γ^3
Juan	Garcia	53	Juan	Garcia	18	110	γ^7
Juan	Garcia	53	Ricard	Tanaka	18	000	γ^1
Ricardo	Garcia	14	Joan	Casanovas	19	000	γ^1
Ricardo	Garcia	14	Pere	Joan	18	000	γ^1
Ricardo	Garcia	14	J.Manel	Casanovas	35	000	γ^1
Ricardo	Garcia	14	Juan	Garcia	53	010	γ^3
Ricardo	Garcia	14	Ricard	Garcia	14	011	γ^4
Ricardo	Garcia	14	Pere	Garcia	82	010	γ^3
Ricardo	Garcia	14	Juan	Garcia	18	010	γ^3
Ricardo	Garcia	14	Ricard	Tanaka	18	000	γ^1
Pere	Garcia	18	Joan	Casanovas	19	000	γ^1
Pere	Garcia	18	Pere	Joan	18	101	γ^6
Pere	Garcia	18	J.Manel	Casanovas	35	000	γ^1
Pere	Garcia	18	Juan	Garcia	53	010	γ^3
Pere	Garcia	18	Ricard	Garcia	14	010	γ^3
Pere	Garcia	18	Pere	Garcia	82	110	γ^7
Pere	Garcia	18	Juan	Garcia	18	011	γ^4
Pere	Garcia	18	Ricard	Tanaka	18	001	γ^2
Juan	Garcia	18	Joan	Casanovas	19	000	γ^1
Juan	Garcia	18	Pere	Joan	18	001	γ^2
Juan	Garcia	18	J.Manel	Casanovas	35	000	γ^1
Juan	Garcia	18	Juan	Garcia	53	110	γ^7
Juan	Garcia	18	Ricard	Garcia	14	010	γ^3
Juan	Garcia	18	Pere	Garcia	82	010	γ^3
Juan	Garcia	18	Juan	Garcia	18	111	γ^8
Juan	Garcia	18	Ricard	Tanaka	18	001	γ^2
Ricard	Tanaka	14	Joan	Casanovas	19	000	γ^1
Ricard	Tanaka	14	Pere	Joan	17	000	γ^1
Ricard	Tanaka	14	J.Manel	Casanovas	35	000	γ^1
Ricard	Tanaka	14	Juan	Garcia	53	000	γ^1
Ricard	Tanaka	14	Ricard	Garcia	14	101	γ^6
Ricard	Tanaka	14	Pere	Garcia	82	000	γ^1
Ricard	Tanaka	14	Juan	Garcia	18	000	γ^1
Ricard	Tanaka	14	Ricard	Tanaka	18	110	γ^7

Table 2. Product space $\mathbf{A} \times \mathbf{B}$ and corresponding Γ vectors

$Name^{\mathbf{A}}$	$Surname^{\mathbf{A}}$	$Age^{\mathbf{A}}$	$Name^{\mathbf{B}}$	$Surname^{\mathbf{B}}$	$Age^{\mathbf{B}}$	γ^i		$\mathbf{M/U}$	m^i	u^i	m^i/u^i	$\log(m^i/u^i)$
Juan	Garcia	53	Juan	Garcia	53	111	γ^8	**M**	2/8	0/56	∞	∞
Juan	Garcia	18	Juan	Garcia	18	111	γ^8	**M**				
Pere	Joan	17	Pere	Joan	18	110	γ^7	**M**	3/8	2/56	10.5	2.35
Juan	Garcia	53	Juan	Garcia	18	110	γ^7	**U**				
Pere	Garcia	18	Pere	Garcia	82	110	γ^7	**M**				
Juan	Garcia	18	Juan	Garcia	53	110	γ^7	**U**				
Ricard	Tanaka	14	Ricard	Tanaka	18	110	γ^7	**M**				
Joan	Casanoves	19	Joan	Casanovas	19	101	γ^6	**M**	1/8	2/56	3.5	1.25
Pere	Garcia	18	Pere	Joan	18	101	γ^6	**U**				
Ricard	Tanaka	14	Ricard	Garcia	14	101	γ^6	**U**				
Pere	Joan	17	Pere	Garcia	82	100	γ^5	**U**	0/8	1/56	0	$-\infty$
J.M.	Casanovas	35	J.Manel	Casanovas	35	011	γ^4	**M**	2/8	1/56	14	2.63
Ricardo	Garcia	14	Ricard	Garcia	14	011	γ^4	**M**				
Pere	Garcia	18	Juan	Garcia	18	011	γ^4	**U**				
Joan	Casanoves	19	J.Manel	Casanovas	35	010	γ^3	**U**	0/8	11/56	0	$-infty$
J.M.	Casanovas	35	Joan	Casanovas	19	010	γ^3	**U**				
Juan	Garcia	53	Ricard	Garcia	14	010	γ^3	**U**				
Juan	Garcia	53	Pere	Garcia	82	010	γ^3	**U**				
Ricardo	Garcia	14	Juan	Garcia	53	010	γ^3	**U**				
Ricardo	Garcia	14	Pere	Garcia	82	010	γ^3	**U**				
Ricardo	Garcia	14	Juan	Garcia	18	010	γ^3	**U**				
Pere	Garcia	18	Juan	Garcia	53	010	γ^3	**U**				
Pere	Garcia	18	Ricard	Garcia	14	010	γ^3	**U**				
Juan	Garcia	18	Ricard	Garcia	14	010	γ^3	**U**				
Juan	Garcia	18	Pere	Garcia	82	010	γ^3	**U**				
Pere	Garcia	18	Ricard	Tanaka	18	001	γ^2	**U**	0/8	3/56	0	$-\infty$
Juan	Garcia	18	Pere	Joan	18	001	γ^2	**U**				
Juan	Garcia	18	Ricard	Tanaka	18	001	γ^2	**U**				
Joan	Casanoves	19	Pere	Joan	18	000	γ^1	**U**	0/8	36/56	0	$-\infty$
Joan	Casanoves	19	Juan	Garcia	53	000	γ^1	**U**				
Joan	Casanoves	19	Ricard	Garcia	14	000	γ^1	**U**				
Joan	Casanoves	19	Pere	Garcia	82	000	γ^1	**U**				
Joan	Casanoves	19	Juan	Garcia	18	000	γ^1	**U**				
Joan	Casanoves	19	Ricard	Tanaka	18	000	γ^1	**U**				
Pere	Joan	17	Joan	Casanovas	19	000	γ^1	**U**				
Pere	Joan	17	J.Manel	Casanovas	35	000	γ^1	**U**				
Pere	Joan	17	Juan	Garcia	53	000	γ^1	**U**				
Pere	Joan	17	Ricard	Garcia	14	000	γ^1	**U**				
Pere	Joan	17	Juan	Garcia	18	000	γ^1	**U**				
Pere	Joan	17	Ricard	Tanaka	18	000	γ^1	**U**				
J.M.	Casanovas	35	Pere	Joan	18	000	γ^1	**U**				
J.M.	Casanovas	35	Juan	Garcia	53	000	γ^1	**U**				
J.M.	Casanovas	35	Ricard	Garcia	14	000	γ^1	**U**				
J.M.	Casanovas	35	Pere	Garcia	82	000	γ^1	**U**				
J.M.	Casanovas	35	Juan	Garcia	18	000	γ^1	**U**				
J.M.	Casanovas	35	Ricard	Tanaka	18	000	γ^1	**U**				
Juan	Garcia	53	Joan	Casanovas	19	000	γ^1	**U**				
Juan	Garcia	53	Pere	Joan	18	000	γ^1	**U**				
Juan	Garcia	53	J.Manel	Casanovas	35	000	γ^1	**U**				
Juan	Garcia	53	Ricard	Tanaka	18	000	γ^1	**U**				
Ricardo	Garcia	14	Joan	Casanovas	19	000	γ^1	**U**				
Ricardo	Garcia	14	Pere	Joan	18	000	γ^1	**U**				
Ricardo	Garcia	14	J.Manel	Casanovas	35	000	γ^1	**U**				
Ricardo	Garcia	14	Ricard	Tanaka	18	000	γ^1	**U**				
Pere	Garcia	18	Joan	Casanovas	19	000	γ^1	**U**				
Pere	Garcia	18	J.Manel	Casanovas	35	000	γ^1	**U**				
Juan	Garcia	18	Joan	Casanovas	19	000	γ^1	**U**				
Juan	Garcia	18	J.Manel	Casanovas	35	000	γ^1	**U**				
Ricard	Tanaka	14	Joan	Casanovas	19	000	γ^1	**U**				
Ricard	Tanaka	14	Pere	Joan	17	000	γ^1	**U**				
Ricard	Tanaka	14	J.Manel	Casanovas	35	000	γ^1	**U**				
Ricard	Tanaka	14	Juan	Garcia	53	000	γ^1	**U**				
Ricard	Tanaka	14	Pere	Garcia	82	000	γ^1	**U**				
Ricard	Tanaka	14	Juan	Garcia	18	000	γ^1	**U**				

Table 3. Product space $\mathbf{A} \times \mathbf{B}$ and corresponding Γ vectors

$Name^{\mathbf{A}}$	$Surname^{\mathbf{A}}$	$Age^{\mathbf{A}}$	$Name^{\mathbf{B}}$	$Surname^{\mathbf{B}}$	$Age^{\mathbf{B}}$	γ^i		$\mathbf{M/U}$	$m^{\sigma(i)}$	$u^{\sigma(i)}$	$m^{\sigma(i)}/u^{\sigma(i)}$
Juan	Garcia	53	Juan	Garcia	53	111	$\gamma^\sigma(1)$	**M**	2/8	0/56	∞
...											
J.M.	Casanovas	35	J.Manel	Casanovas	35	011	$\gamma^\sigma(2)$	**M**	2/8	1/56	14
...											
Pere	Joan	17	Pere	Joan	18	110	$\gamma^\sigma(3)$	**M**	3/8	2/56	10.5
...											
Joan	Casanoves	19	Joan	Casanovas	19	101	$\gamma^\sigma(4)$	**M**	1/8	2/56	3.5
...											
Pere	Joan	17	Pere	Garcia	82	100	$\gamma^\sigma(5)$	**U**	0/8	1/56	0
Joan	Casanoves	19	J.Manel	Casanovas	35	010	$\gamma^\sigma(6)$	**U**	0/8	11/56	0
...											
Pere	Garcia	18	Ricard	Tanaka	18	001	$\gamma^\sigma(7)$	**U**	0/8	3/56	0
...											
Joan	Casanoves	19	Pere	Joan	18	000	$\gamma^\sigma(8)$	**U**	0/8	36/56	0
...											

Table 4. Product space $\mathbf{A} \times \mathbf{B}$ and corresponding Γ vectors

1.19	0.93	1.39	2.17	1.92	2.43	2.50	0.69	0.95	3.90	1.52	5.01	6.07	7.51	9.02
19.34	16.80	19.22	17.99	19.76	17.43	20.81	17.78	20.41	17.10	17.82	15.93	16.85	23.78	23.49
22.88	22.77	14.26	13.72	29.70	11.73	13.20	13.00	13.12	31.73	49.38	14.66	15.65	12.82	51.03
54.31	54.72	22.21	27.70	56.38	19.13	19.21	11.96	36.06	47.26	9.66	10.01	58.97	23.85	61.53
35.37	45.54	66.97	58.51	12.67	60.11	67.90	69.19	7.02	77.34	75.42	3.16	5.57	85.19	87.14
97.37	97.84	97.96	97.66	97.58	97.39	97.63	97.79	3.43	4.26	7.66	3.94	4.02	4.54	3.37
3.16	3.65	4.15	3.70	3.97	4.13	3.72	4.27	3.88	4.03	4.55	4.83	4.35	43.05	3.02
2.13	1.36	1.10	0.93											

Table 5. Average percentage of linked records using distance-based record linkage: numerical variables

0.15	0.08	0.11	0.12	0.07	0.25	0.25	0.09	0.09	0.38	0.20	0.52	0.85	1.08	2.79
4.70	13.60	3.44	3.44	6.67	5.45	4.15	3.35	13.90	2.08	3.98	2.00	2.37	18.29	22.75
16.69	22.78	1.88	1.38	29.06	1.14	1.20	1.22	0.99	36.92	27.29	0.50	4.66	0.85	33.04
33.70	37.41	22.39	29.03	42.00	6.93	6.24	3.52	39.76	57.47	2.34	0.97	56.84	24.48	60.69
46.98	56.22	64.79	65.28	2.90	66.56	67.63	66.35	1.90	71.32	71.85	0.77	1.26	74.13	73.03
74.07	74.07	74.40	75.28	75.99	78.96	79.78	88.06	0.62	0.67	1.52	0.69	0.62	0.25	0.50
0.61	0.44	0.67	0.53	0.34	0.46	0.66	0.38	0.41	0.42	0.52	0.38	0.38	64.88	0.48
0.25	0.29	0.15	0.22											

Table 6. Average percentage of linked records using probabilistic record linkage: numerical variables

1.045	0.847	1.283	2.050	1.852	2.183	2.249	0.595	0.860	3.517	1.323	4.497	5.225	6.425	6.227
14.643	3.204	15.780	14.550	13.095	11.984	16.653	14.431	6.508	15.026	13.836	13.929	14.484	5.489	0.741
6.190	-0.003	12.381	12.341	0.635	10.595	11.997	11.786	12.130	-5.185	22.090	14.153	10.992	11.971	17.989
20.608	17.315	-0.185	-1.336	14.378	12.196	12.963	8.439	-3.704	-10.215	7.315	9.048	2.130	-0.635	0.847
-11.614	-10.675	2.183	-6.772	9.775	-6.455	0.265	2.842	5.119	6.019	3.571	2.394	4.312	11.058	14.114
23.294	23.770	23.558	22.381	21.587	18.439	17.857	9.735	2.804	3.585	6.138	3.254	3.399	4.286	2.870
2.553	3.214	3.479	3.175	3.624	3.664	3.056	3.889	3.466	3.611	4.034	4.444	3.968	-21.826	2.540
1.878	1.071	0.952	0.701											

Table 7. Difference between the average percentage of linked records using distance-based and probabilistic record linkage: numerical variables

57.2	35.2	22.5	17.1	12.6	9.2	6.8	2.2	1.1	72.2	37.2	34	33.2	32.3	31.3
30.2	27.2	15.6	99.4	89.9	76.7	55.6	35.5	22.6	6.8	2.3	0.9	99.5	98.4	96.7
91.7	80.2	49.3	24.3	20.2	16.7	98.6	95.6	94.3	93.9	91.4	89.6	88	87.7	84.9
100	99.7	99.7	99.4	99	98.5	98.1	98	98.1	90.6	74.6	58.9	35.7	25.9	17.9
10.5	5.8	5	96.6	87.8	83.6	80.5	79.7	75.8	50.3	27.5	25.5	50.2	26.3	17.6
10.1	6.6	4.3	3.6	0.3	0.3	87.6	74.6	57	44.2	30.9	26.5	21.3	18.6	16.9
95.2	93.7	92.1	90.7	74	72.3	68.6	60.8	56.8	98.6	86.1	65.1	32.6	10.3	8.3
0.9	0.3	0.3	96.5	91.2	81.5	71.1	56.3	42.3	24	21.5	19.7	99.6	99.2	96.9
94.6	91.2	86.3	82.4	75.3	68.7	97.8	93.8	91.6	90.5	88.2	82.8	79.2	78	75.3
98.7	97	92.5	91.7	87.7	86.1	86.2	84.8	83.3	99.2	98.5	97.3	97.3	96.4	97.5
95.8	96	95.3	85.3	61.7	44.7	20	9.5	7.8	4.2	4.2	4.2	92.5	79.3	65.2
47.3	33.8	26.1	24.4	20.7	17.7	97	90.9	87.4	82.5	77.2	71.6	66.1	55.4	47.6

Table 8. Average percentage of linked records using distance-based record linkage: categorical variables

53	79.9	82.8	82.4	69.1	69	66.6	57	59.8	19.5	44.8	42.8	41	38.6	37.2
36.7	36.7	59.3	41.5	40.2	38	37.9	38.5	46.4	88.2	85.6	78.2	11.8	10.4	9.3
8.3	9.3	14.8	44.8	57.6	55.9	41.4	41.1	43.8	42.1	41.3	40.2	37.8	39.1	38.7
11.8	11.7	11.4	11.5	10.9	10.1	9.6	10.2	9.8	38.9	31.8	28	24.6	23.3	19
26.6	50.9	42.5	11.4	8.8	7.4	5.4	5.2	3.8	7.7	35.6	32.5	100.1	69.9	57.7
44.7	27.9	16.1	7.9	5.1	5.1	100.1	93.2	74.9	55.3	42.8	35.7	31.8	32.9	30.5
100.1	99.3	99.2	98.3	95.4	93.6	88.9	82.6	78	100.1	100.1	91.7	73	83.5	69.5
35.5	5.1	5.1	99.9	99.3	96.2	83.3	68.5	51.8	29.5	30	33.8	99.9	99.7	99.3
98.6	97.9	93.4	89.4	84.4	78.5	96.6	92.1	89.7	88.1	84.3	80.3	78.9	75.9	73.4
98.8	95.2	93.6	90.9	90.6	86.5	85.3	82.9	80.1	99.7	99.3	98	97.5	97.6	96.7
95.9	96.1	94.8	100.1	89.1	74.9	49.3	42.9	45.8	68.8	4.5	18.8	100.1	91.4	78.3
57.4	40.4	35	38.6	44	39.8	99.7	96.8	95	91.7	86.2	79.8	73	67.6	57.1

Table 9. Average percentage of linked records using probabilistic record linkage: categorical variables

4.2	-44.7	-60.3	-65.3	-56.5	-59.8	-59.8	-54.8	-58.7	52.7	-7.6	-8.8	-7.8	-6.3	-5.9
-6.5	-9.5	-43.7	57.9	49.7	38.7	17.7	-3	-23.8	-81.4	-83.3	-77.3	87.7	88	87.4
83.4	70.9	34.5	-20.5	-37.4	-39.2	57.2	54.5	50.5	51.8	50.1	49.4	50.2	48.6	46.2
88.2	88	88.3	87.9	88.1	88.4	88.5	87.8	88.3	51.7	42.8	30.9	11.1	2.6	-1.1
-16.1	-45.1	-37.5	85.2	79	76.2	75.1	74.5	72	42.6	-8.1	-7	-49.9	-43.6	-40.1
-34.6	-21.3	-11.8	-4.3	-4.8	-4.8	-12.5	-18.6	-17.9	-11.1	-11.9	-9.2	-10.5	-14.3	-13.6
-4.9	-5.6	-7.1	-7.6	-21.4	-21.3	-20.3	-21.8	-21.2	-1.5	-14	-26.6	-40.4	-73.2	-61.2
-34.6	-4.8	-4.8	-3.4	-8.1	-14.7	-12.2	-12.2	-9.5	-5.5	-8.5	-14.1	-0.3	-0.5	-2.4
-4	-6.7	-7.1	-7	-9.1	-9.8	1.2	1.7	1.9	2.4	3.9	2.5	0.3	2.1	1.9
-0.1	1.8	-1.1	0.8	-2.9	-0.4	0.9	1.9	3.2	-0.5	-0.8	-0.7	-0.2	-1.2	0.8
-0.1	-0.1	0.5	-14.8	-27.4	-30.2	-29.3	-33.4	-38	-64.6	-0.3	-14.6	-7.6	-12.1	-13.1
-10.1	-6.6	-8.9	-14.2	-23.3	-22.1	-2.7	-5.9	-7.6	-9.2	-9	-8.2	-6.9	-12.2	-9.5

Table 10. Difference between the average percentage of linked records using distance-based and probabilistic record linkage: categorical variables

Surnames		Coding
Euler	Ellery	E460
Gauss	Ghosh	G200
Hilbert	Heilbronn	H416
Knuth	Kant	K530
Lloyd	Ladd	L300
Lukasiewicz	Lissajous	L222

Table 11. SOUNDEX Codification