Statistics and Computing **13**: 343–354, 2003

Disclosure risk assessment in statistical microdata protection via advanced record linkage

JOSEP DOMINGO-FERRER* and VICENÇ TORRA†

**Dept. Comput. Eng. and Maths—ETSE, Universitat Rovira i Virgili, Av Països Catalans 26, E-43007 Tarragona, Catalonia*
jdomingo@etse.urv.es
†*Institut d'Investigació en Intel.ligència Artificial—CSIC, Campus UAB, E-08193 Bellaterra, Catalonia*
vtorra@iiia.csic.es

The performance of Statistical Disclosure Control (SDC) methods for microdata (also called masking methods) is measured in terms of the utility and the disclosure risk associated to the protected microdata set. Empirical disclosure risk assessment based on record linkage stands out as a realistic and practical disclosure risk assessment methodology which is applicable to every conceivable masking method. The intruder is assumed to know an external data set, whose records are to be linked to those in the protected data set; the percent of correctly linked record pairs is a measure of disclosure risk. This paper reviews conventional record linkage, which assumes shared variables between the external and the protected data sets, and then shows that record linkage—and thus disclosure—is still possible without shared variables.

Keywords: statistical disclosure control, record linkage, disclosure risk for microdata, re-identification

1. Introduction

The main task of National Statistical Offices (NSO) is to collect information from individuals and organizations and disseminate this information to researchers, media and general public. Preserving respondent privacy is a legal obligation of NSOs which should not be considered less important than dissemination. The thorny issue is that there is a tension between the goal of accurate dissemination and respondent privacy preservation. The higher the dissemination accuracy, the higher the risk of disclosing respondent information which should stay confidential. Disclosure risk is defined as the risk of re-identification of particular individuals. Re-identification happens when some sensitive and confidential data that have been released are subsequently linked to a particular individual, which results in a confidentiality breach. To avoid re-identification, data are masked (i.e. distorted) before release. In this way, disclosure risk decreases. However, masked data must maintain the so-called analytical validity (Winkler 1995a), that is, statistical analyses on original confidential data and on masked data should yield similar results.

Among the existing re-identification methods, those based on record linkage are probably the most general and realistic ones, as they can be used with any method and in any disclosure scenario. Record linkage (Robinson-Cox 1998) is a technology whose original goal was, rather than assessing disclosure risk, to improve the quality of data by linking records in separate files that relate to the same individual or household.

When used by intruders as a method for re-identification of individuals in data files, record linkage is a threat to statistical confidentiality. Sensitive information can be disclosed as a result of released records being linked to other publicly available information. When used by data protectors themselves, record linkage is an invaluable tool to assess disclosure risk and thus the level of protection associated to masked data being released.

Recent developments in record linkage are discussed in Winkler (1995a, b), Robinson-Cox (1998), Rosman (1995), Bacher, Brand and Bender (2002). The latter reference compares

some of the existing methods. Usually, this kind of systems (e.g. Integrity (http://www.integrity.com)) use statistical and artificial intelligence methods to determine the matching between records and extract a unique identifier (or a set of variables acting as an identifier). Methods in the literature (see Newcombe *et al*. 1959, Winkler 1995a, Robinson-Cox 1998, Gill 2001) assume the existence of a set of common variables in the files to be linked. The main problem with such an assumption is that a matching procedure among pairs of records is not always enough to establish the link between records. As pointed out in Winkler (1995a), "the normal situation in record linkage is that identifiers in pairs of records that are truly matches disagree by small or large amounts and that different combinations of the non-unique, error-filled identifiers need to be used in correctly matching different pairs of records".

Example 1. Assume an external file A with the following variables:

- *Name_A*, whose values consist of a first name followed by a surname.
- *Income_tax*, whose values are the income tax paid on a certain year.

Assume a protected file B with the following variables:

- *Name_B*, whose values consist of a surname followed by an initial.
- *Net_income*, whose values are the net income on a certain year.

Note that files A and B do not share any variable (even the format for names is different). However, it would seem natural to link a record in A with the highest *Income_tax* and *Name_A* = "Henry Plantagenet" to a record in file B with the highest *Net_income* and *Name_B* = "Plantagenet H". Conventional record linkage cannot provide such a link.

1.1. *Our contribution*

Four record linkage situations are possible depending on the coincidence or non-coincidence of variables and terminology (terminology is the domain of the variables, i.e. the terms used to evaluate the individuals):

(i) Same variables and same terminology.
(ii) Same variables but different terminology.
(iii) Different variables but same terminology.
(iv) Different variables and different terminology.

Classical record linkage falls into situations (i) or (ii), although in the latter case only small terminology differences are allowed (small inconsistencies among names, missing values and the like). However, based on the above classification, other types of record linkage are conceivable: situation (ii) when the degree of non-coincidence on the terminology is not limited to small variations of names (e.g. completely different terms, due, for example, to the use of different granularities) and, of course, situations (iii) and (iv).

We study in this paper situation (iv), that is, record linkage when neither variables nor terminology are the same across the files to be linked. What will be assumed is that there is a set of shared individuals or entities across the files—without such a set of shared individuals, record linkage does not make sense.

There is no evidence in the literature of previous work devoted to record linkage without shared variables. Yet, it is a subject of interest for both statistical disclosure control and data mining, because it highlights relationships between individuals that would otherwise remain implicit and undiscovered in the files to be linked.

Section 2 discusses two simple benchmarks to evaluate the effectiveness of re-identification methods; these are random re-identification and re-identification based on one-dimensional ranking. Section 3 reviews classical record linkage between data sets which have some variables in common. Section 4 presents our approach for record linkage between data sets which have no variables in common. Section 5 contains some conclusions and suggestions for future work.

2. Benchmarking re-identification

For the sake of concreteness, we will concentrate in what follows on record linkage between a released unidentified file A and the intruder's identified file B. To keep benchmarking simple, we will assume in this section that both files correspond to the same set of n individuals. Two benchmarks will be described which should be useful in measuring the effectiveness of a record linkage method.

2.1. *Probability of random re-identification*

The difference between the proportion of records correctly re-identified by a record linkage method and the expected proportion of records correctly re-identified by a random strategy is a measure of the effectiveness of the method.

Let p_0 be the permutation expressing the correct correspondence between records in both files, i.e. record i in A and record $p_0(i)$ in B correspond to the same individual, for all $i = 1, \ldots, n$. Random re-identification can be viewed as an intruder picking a random permutation p of the set of individuals, such that the i-th individual in file A is linked to the j-th individual in file B, where $j = p(i)$. The following proposition, proven in the Appendix, gives the probability of correct re-identification when this random strategy is followed.

Proposition 1. *If A and B both contain n records corresponding to the same set of n individuals, the probability of correctly re-identifying exactly r individuals by a random strategy is*

$$\frac{\sum_{v=0}^{n-r} \frac{(-1)^{(n-r)}}{v!}}{r!} \tag{1}$$

2.2. *Re-identification via one-dimensional ranking*

Another benchmark that can be used is the proportion of records re-identified by a simple strategy such as one-dimensional ranking. This strategy is as follows:

- Project records of files A and B onto a single dimension each. There are several techniques for doing so. One possibility is to take the first principal component of each file. Another possibility is to take the sum of z-scores of each file: all variables in the file are standardized (by subtracting their average and dividing by their standard deviation) and the sum of the standardized variables gives a one-dimensional value for each record.
- Rank records in files A and B according to their one-dimensional projection.
- Link the i-th record in file A with the i-th record in file B.

Again, the difference in the proportion of records correctly linked by a record linkage method under test and the proportion correctly linked by one-dimensional ranking can be used as a measure of effectiveness.

3. Record linkage with common variables

In this section, we review the two most widely used approaches for re-identification between files sharing a set of variables. Let A and B be two files sharing a set of common variables. Both files are defined over the same set of individuals. We will consider a general case where it cannot be assured that the values for the same variables in both files are the same for the same individuals. In other words, even though variables are the same, values for a particular individual may differ due to errors.

We first review probabilistic-based record linkage. Then, distance-based record linkage is considered. Both approaches are subsequently compared.

3.1. *Probabilistic record linkage*

Probabilistic record linkage is described in Fellegi and Sunter (1969), Jaro (1989) and Winkler (1995b). In this section, we outline only some of its elements. See the above mentioned references for details.

Let us consider two files A and B with a single variable V each. Let a and b be records belonging to files A and B, respectively. Probabilistic record linkage applied to files A and B is based on the computation of an index $R(a, b)$ for each pair (a, b). Some index thresholds are then used to label the pair as a linked pair, a clerical pair or a non-linked pair. Equivalently, when the index is larger than, say, *linkThreshold*, the pair is linked; when the index is lower than, say, *nonLinkThreshold*, the pair is non-linked; when the index is between both thresholds the pair is classified as a clerical pair. A clerical pair is one that cannot be automatically classified as linked or non-linked; human inspection is needed to classify it.

The index $R(a, b)$ is computed as a log-likelihood ratio

$$R(a, b) = \log\left(\frac{P(a = b \mid (a, b) \in \mathbf{M})}{P(a = b \mid (a, b) \in \mathbf{U})}\right) \tag{2}$$

where **M** corresponds to the set of *matched pairs* and **U** corresponds to the set of *unmatched pairs*. Pairs in **M** are those that can be proven to be true matches (the ones that a perfect re-identification method would detect as corresponding to the same individual) and pairs in **U** are those that can be proven to be non-related (the ones that a perfect re-identification procedure would not relate).

When a set of variables are considered in both files rather than a single variable, an expression equivalent to Expression (2) is used. In this case, a and b correspond to vectors of values rather than values for a single variable V. It is usually assumed for computing $R(a, b)$ that different variables are statistically independent and thus products of conditional probabilities can be used. Alternative approaches not assuming statistical independence have also been considered in the literature (see Winkler 1995a).

To use probabilistic record linkage in an effective way, we need to set the thresholds (e.g. the values *linkThreshold* and *nonLinkThreshold*) and the conditional probabilities in Expression (2).

The thresholds are usually determined from the probabilities:

$$P(\text{linked pair} \mid \mathbf{U})$$
$$P(\text{non-linked pair} \mid \mathbf{M})$$

In plain words, thresholds are computed from:

1. The probability of linking a pair that is an unmatched pair (a *false positive* or *false linkage*).
2. The probability of not linking a pair that is a match pair (a *false negative* or *false unlinkage*).

Conditional probabilities in Expression (2) are usually estimated using the EM algorithm (Dempster, Laird and Rubin 1977). See the survey paper (Torra and Domingo-Ferrer 2003) for a description of the EM algorithm and its use in probabilistic record linkage.

3.2. *Distance-based record linkage*

This approach, described in Pagliuca and Seri (1999) in a very specific formulation, consists of computing distances between records in the two data files being considered. The method was applied in Pagliuca and Seri (1999) to disclosure risk assessment. An original data file A was considered together with a distorted version B of the same file. Record linkage was used to find out to what extent distorted records could be re-identified.

In general, for each record in file A, the distance to every record in file B is computed. Then the *nearest* and *second nearest* records in file B are considered. A record in file B is labeled as *linked* when the nearest record in file A turns out to be its corresponding original record (the one that generated the distorted record). A record in file B is labeled as *linked to 2nd nearest* when the second nearest record in file A turns out to be

the corresponding original record. In all other cases, records are not linked.

The distance-based approach requires that distances be standardized to avoid scaling problems. Also, an assumption on the weights of variables for computing the distance between a pair of records (equal weight for all variables according to Pagliuca and Seri (1999)) is required.

3.3. *Discussion*

Both record linkage methods aim at linking records in files A and B that correspond to the same individuals. As shown above, both approaches are radically different. The following aspects can be underlined:

- Distance-based record linkage methods are simple to implement and operate. The main difficulty with them consists of establishing appropriate distances for the variables under consideration. In Domingo-Ferrer and Torra (2002), distances were defined for categorical variables, for both ordinal and nominal scales. The advantage of distance-based record linkage is that it allows inclusion of subjective information (individuals or variables can be weighted) in the re-identification process.
- Probabilistic record linkage methods are less simple. However, they do not assume rescaling or weighting of variables and require the user to provide only two probabilities as input: the values $P(\text{linked pair} \mid \mathbf{U})$ and $P(\text{non-linked pair} \mid \mathbf{M})$.

It has been illustrated that both approaches lead to similar re-identification results for numerical data (Domingo-Ferrer and Torra 2001) as well as for categorical data (Domingo-Ferrer and Torra 2002). For a more detailed comparison of both methods, see the survey paper (Torra and Domingo-Ferrer 2003).

4. Record linkage without common variables

As explained in Section 1.1, re-identification without common variables requires some assumptions, which are next summarized:

Assumption 1. A set of common individuals is shared by both files.

Assumption 2. Data in both files contain, implicitly, similar structural information. In other words, even though there are no common variables, there is substantial correlation between some variables in both files.

Structural information of data files stands in our case for any organization of the data that allows explicit representation of the relationship between individuals. This structural information is obtained from the data files through manipulation of the data (e.g. using clustering techniques or any other data analysis or data mining technique). Comparison of the structural information implicit in both files is what allows two records that correspond to the same individual to be linked by the system.

Assumption 3. Structural information can be expressed by means of partitions.

In our approach, structural information is represented by means of partitions. Partitions obtained from data through clustering techniques make explicit the relation between individuals according to the variables that describe them. Common partitions in both files reflect the common structural information. We prefer partitions rather than other (more sophisticated) structures also obtainable with clustering methods, like dendrograms, because the former are more robust to changes in the data, as shown in Neumann and Norton (1986).

Although the main interest of our research is re-identification of individuals, the approach described below is not directly targeted to the re-identification of particular individuals. Instead, we try to re-identify groups of them. Due to this, we use the term of group-level re-identification; record-level re-identification is a particular case of group-level re-identification where one or more groups contain a single record (see Note 2 below).

4.1. *Theoretical background*

In this section, we review some of the theoretical results that belong to the area of aggregation of equivalence relations. Equivalence relations are relevant here as a way to express relationships between records (they are equivalent to partitioning records).

Definition 1. A binary relation R on a set A is an equivalence relation if and only if, for all a, b, c in A, the following conditions hold:

(a) *Reflexivity*: $R(a, a)$
(b) *Symmetry*: $R(a, b)$ if and only if $R(b, a)$
(c) *Transitivity*: $R(a, b)$ and $R(b, c)$ imply $R(a, c)$.

Definition 2. An aggregation function $\mathbb{C}$ over equivalence relations on a set A is a function that, given n equivalence relations $R_1, \ldots, R_n$, defines a new equivalence relation on A. Denote the aggregated relation by $\mathbb{C}(R_1, \ldots, R_n)$.

It is usual to add some constraints to define what an aggregation function is (see Godo and Torra (2000) for examples in the ordinal case). The additional constraints we impose are specified next:

Definition 3 (Consistent aggregation). An aggregation function $\mathbb{C}$ is said to be consistent if it satisfies the following two conditions:

1. For all $a, b \in A$ and all pairs of n-tuples $(R_1, \ldots, R_n)$ and $(R'_1, \ldots, R'_n)$ of equivalence relations on A,

$$\text{if } R_i(a, b) \Leftrightarrow R'_i(a, b) \text{ for } i = 1, \ldots, n, \quad \text{then}$$
$$R(a, b) \Leftrightarrow R'(a, b)$$

where $R = \mathbb{C}(R_1, \ldots, R_n)$ and $R' = \mathbb{C}(R'_1, \ldots, R'_n)$

2. For all $a, b \in A$ and all $(R_1, \ldots, R_n)$,
   (a) If $R_i(a, b)$ for $i = 1, \ldots, n$ then $R(a, b)$.
   (b) If $\neg R_i(a, b)$ for $i = 1, \ldots, n$ then $\neg R(a, b)$, where $\neg$ is the logical negation operator.

   where $R = \mathbb{C}(R_1, \ldots, R_n)$.

The first condition in the above definition is the so-called independence condition. It means that the value of the aggregated relation between a and b does only depend on the individual relations between these two elements. This is equivalent to saying that the aggregation can be computed over pairs. In other words, there exists a function F such that:

$$R(a, b) = F(R_1(a, b), \ldots, R_n(a, b))$$

The second condition is unanimity over pairs, which means that: (i) when all equivalence relations relate a and b, the aggregated relation also relates a and b; (ii) when all equivalence relations say that a and b are not related, the aggregated relation does not relate them.

Theorem 1 below states that consistent aggregation functions characterized in Definition 3 can be equivalently characterized as follows:

Definition 4 (Conjunctive aggregation). Let $R_1, \ldots, R_n$ be relations on a set A. Then an aggregation function $\mathbb{C}$ is a conjunctive aggregation function if there exists a nonempty subset N of $\{1, \ldots, n\}$ such that $R(a, b) \Leftrightarrow (R_i(a, b), \forall\, i \in N)$, where $R = \mathbb{C}(R_1, \ldots, R_n)$.

The following theorem states that conjunctive combination of partitions (i.e. intersecting the groups in the partitions) is the only consistent way to aggregate partitions. This is the basis of our method described in Section 4.2, which relies on conjunctive combination of partitions obtained using different clustering techniques.

Theorem 1 (Fishburn-Rubinstein 1986)**.** *If the set A has at least three elements*, *the set of consistent aggregation functions* (*Definition* 3) *equals the set of conjunctive aggregation functions* (*Definition* 4)*.*

4.2. *Group-level re-identification*

General structural information is identified in both files by means of clustering techniques. Moreover, as different clustering techniques identify different relationships between the individuals, several techniques with different parameterizations are applied in turn to both files. In this way, for each file and each technique, a partition of the individuals is obtained. This initial process is formalized below considering that data files are named A and B and, as usual, files are defined by a set of records that assign values to variables. We assume that the file for which individuals are known is file B.

Let us express files A and B as sets of records, i.e. $A = \{a_1, \ldots, a_{n(A)}\}$ and $B = \{b_1, \ldots, b_{n(B)}\}$, where the number of records ($n(A)$ and $n(B)$) is not necessarily the same in both files. Assumption 1 requires that a substantial number of individuals exist which are common to both files. However, no knowledge on which are those common individuals is assumed; therefore, at this stage, different names in each file will be used for denoting common individuals.

A clustering process is applied to each file. A set of t different clustering methods is considered (or the same clustering method with different parameters) and each method in the set is applied to each file. Each method induces a partition of the set of records in a file.

Let $\mathbf{CP} = \{CP_1, \ldots, CP_t\}$ be the set of clustering techniques considered. Let $\mathbf{C}_{i,A} = \{C_{i,A,1}, \ldots, C_{i,A,nc(i)}\}$ and $\mathbf{C}_{i,B} = \{C_{i,B,1}, \ldots, C_{i,B,nc(i)}\}$ be the groups obtained when the clustering technique CP_i is applied to files A and B, respectively. Here, $C_{i,A,j}$ and $C_{i,B,j}$ correspond to the j-th group obtained by the clustering technique CP_i when applied to data files A and B. For all i, j, groups $C_{i,A,j}$ and $C_{i,B,j}$ are subsets of the records in files A and B, respectively, i.e. $C_{i,A,j} \subseteq A$ and $C_{i,B,j} \subseteq B$. Note that we impose that the number of groups induced by a particular clustering technique CP_i be the same $nc(i)$ when applied to both data files. This constraint is necessary in order for groups obtained for file A to be identifiable with those obtained for file B.

Note 1 (On the number of groups). A substantial practical problem in performing a cluster analysis is deciding on the number of groups in the data. Hierarchical clustering methods give configurations for numbers of groups from one (the entire data set) up to the number of records (each group has only one record). Other algorithms find a best fitting structure for a given number of groups. The latter algorithms can be repeatedly used for different choices of the number of groups to give a variety of alternatives. See Section 1.7 of Hoppner *et al.* (1999), Chapter 10 of Duda, Hart and Stork (2001) or Anderberg (1973) and Hastie, Tibshirani and Friedman (2001) for further discussion. In this work, we use hierarchical clustering methods which give dendrograms for each file. For each hierarchical clustering technique CP_i used, both dendrograms are cut at the highest point (closest to their root) that gives partitions with the same number of groups $nc(i)$ for file A and file B. Cutting high in the dendrogram results in partitions with less groups, which seem more robust (less dependent on small changes in the data). In particular, this allows the method discussed in this paper to be used with files A and B with different number of records.

Once all partitions have been obtained for each CP_i, the structural information is built for each file. This information is a partition (according to Assumption 3) that should synthesize the common information extracted by all CP_i. To achieve this, we combine all partitions $\mathbf{C}_{i,A}$ and $\mathbf{C}_{i,B}$ by means of an aggregation function. Since Theorem 1 states that the only consistent aggregation functions are the conjunctive ones, we use the conjunction of all partitions. Thus, the structural information of a

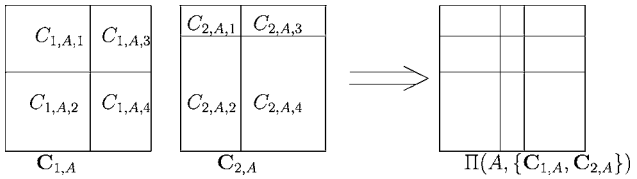

Fig. 1. *Conjunction of partitions:* $\Pi(A, \{\mathbf{C}_{1,A}, \mathbf{C}_{2,A}\}) = \{C_{1,A,j_1} \cap C_{2,A,j_2} \mid \text{for } 1 \leq j_i \leq nc(i)\}$

file is defined as the intersection of all partitions obtained by the clustering methods in **CP**.

Definition 5 (Structural information). Given a set of clustering methods **CP** and two files A and B, the structural information extracted by techniques in **CP** from files A and B is defined as $\Pi(A, \mathbf{CP})$ and $\Pi(B, \mathbf{CP})$, respectively, where

$$\begin{aligned}\Pi(A, \mathbf{CP}) &= \{\cap_{1 \leq i \leq t} C_{i,A,j} \mid \text{for } 1 \leq j \leq nc(i)\} \\ &= \left\{\pi_1^A, \ldots, \pi_{p(A)}^A\right\} \\ \Pi(B, \mathbf{CP}) &= \{\cap_{1 \leq i \leq t} C_{i,B,j} \mid \text{for } 1 \leq j \leq nc(i)\} \\ &= \left\{\pi_1^B, \ldots, \pi_{p(B)}^B\right\}\end{aligned}$$

Figure 1 illustrates the conjunction of partitions for file A in the particular case $t = 2$ (only two clustering techniques used). Tables 1 and 2 depict the structural information extracted from files A and B by a set of clustering methods. Remark that some of the $C_{i,A,j}$ in Table 1 may be empty, because the number of groups obtained with the i-th method CP_i is $nc(i) \leq p(A)$; a similar remark applies to the $C_{i,B,j}$ in Table 2.

From Definition 5, the proposition below follows:

Proposition 2. $\Pi(A, \mathbf{CP})$ *and* $\Pi(B, \mathbf{CP})$ *are partitions of the sets of records of files A and B, respectively.*

Using the equivalence of consistent and conjunctive aggregation (see Definitions 3 and 4 and Theorem 1), we can state the next proposition.

Table 1. *Partitions obtained from file A with clustering techniques* $\mathbf{CP} = \{CP_1, \ldots, CP_t\}$

FileA	CP_1	CP_2		CP_t
π_1^A	$C_{1,A,1}$	$C_{2,A,1}$	$\cdots$	$C_{t,A,1}$
$\cdots$	$\cdots$	$\cdots$		$\cdots$
$\pi_{p(A)}^A$	$C_{1,A,p(A)}$	$C_{2,A,p(A)}$	$\cdots$	$C_{t,A,p(A)}$

Table 2. *Partitions obtained from file B with clustering techniques* $\mathbf{CP} = \{CP_1, \ldots, CP_t\}$

FileB	CP_1	CP_2		CP_t
π_1^B	$C_{1,B,1}$	$C_{2,B,1}$	$\cdots$	$C_{t,B,1}$
$\cdots$	$\cdots$	$\cdots$		$\cdots$
$\pi_{p(B)}^B$	$C_{1,B,p(B)}$	$C_{2,B,p(B)}$	$\cdots$	$C_{t,B,p(B)}$

Proposition 3.

1. *All records in the same partition element* $\pi \in \Pi(A, \mathbf{CP})$ *are clustered together in all partitions* $\mathbf{C}_{i,A}$. *The same holds true for* $\Pi(B, \mathbf{CP})$.
2. *Two records in two different groups* π_i *and* π_j *in* $\Pi(A, \mathbf{CP})$ *are clustered in different partitions at least for one clustering method. The same holds true for* $\Pi(B, \mathbf{CP})$.

To perform record linkage between files A and B, we need to associate each group in a data file with a group in the other one. Therefore, we want a mapping $f_i : \mathbf{C}_{i,A} \rightarrow \mathbf{C}_{i,B}$ for all $i \in \{1, \ldots, t\}$ that assigns to each group in $\mathbf{C}_{i,A}$ a group in $\mathbf{C}_{i,B}$. Mappings f_i should be such that, when applied to Table 1, a table as similar as possible to Table 2 is obtained (note that $\mathbf{C}_{i,A}$ is the i-th column of Table 1 and $\mathbf{C}_{i,B}$ is the i-th column of Table 2). We define similarity between tables on a row basis: the j-th row in Table 1 (e.g. the one corresponding to π_j^A) should be similar to one of the rows in Table 2. Otherwise put, a π_k^B in $\Pi(B, \mathbf{CP})$ should exist which is similar to $f(\pi_i^A) = (f_1(c_{i,1}^A), \ldots, f_t(c_{i,t}^A))$ for all i.

Consistently with the above approach, we use a similarity function $S : [\mathbf{C}_{1,B} \times \cdots \times \mathbf{C}_{t,B}]^2 \rightarrow \mathbb{R}$ specified by:

$$S(X = (x_1, x_2, \ldots, x_t), Y = (y_1, y_2, \ldots, y_t)) = \sum_{i=1}^{t} s_i(x_i, y_i)$$

where $s_i(x, y) = 1$ if $x = y$, and 0 otherwise.

To complete the formalization of the re-identification process, we need a group-level re-identification function (i.e. one that relates the i-th partition of $\Pi(A, \mathbf{CP})$ with the k-th partition of $\Pi(B, \mathbf{CP})$). Let this function be

$$m : \{1, \ldots, p(A)\} \rightarrow \{1, \ldots, p(B)\}$$

Now, the group-level re-identification problem can be stated in the following way:

Problem 1 (Group-level re-identification). Find functions $f = (f_1, \ldots, f_t)$ and m such that $\sum_{i=1}^{t} S(f(\pi_i^A), \pi_{m(i)}^B)$ is maximized.

The following remarks related to Problem 1 are of interest:

1. The definition of the problem assumes that the file for which the identity of individuals is known is file B. Therefore, the re-identification function m maps groups in A into groups in B and the similarity function is computed by comparing partitions in B.
2. We place a restriction on f_i by requiring that $(f_i(C_{i,A,1}), \ldots, f_i(C_{i,A,nc(i)}))$ be a permutation of the vector $(C_{i,B,1}, \ldots, C_{i,B,nc(i)})$.
3. $p(A)$ can be different from $p(B)$. Therefore, m is not always a one-to-one function.
4. When the number of clustering methods increases (larger **CP**), the number of elements in $\Pi(A, \mathbf{CP})$ and $\Pi(B, \mathbf{CP})$

increases toward the number of records in the files (i.e. groups become smaller and smaller and tend to include a single record). Therefore, if a large number of clustering methods is used, group-level re-identification tends to become record-level re-identification.

5. The above statement of group-level re-identification is targeted to the case where no common variables nor terminologies exist between files A and B, but it also includes the case of common variables but different terminologies between A and B.

Note 2 (Group-level vs. record-level re-identification). The approach described above is meant as a group-level re-identification. That is, its outcome is a mapping between groups of records in file A and groups of records in file B. As a particular case, if groups in file B contain a single record, our approach leads to record-level re-identification (even if it may be a many-to-one or few-to-one re-identification, instead of one-to-one). Note that outliers in the released file A (usually the most interesting individuals for intruders) are likely to belong to single-record groups; thus, for outliers, record-level is likely to mean one-to-one re-identification. The way to reduce the size of groups and tend to record-level re-identification is to increase the number of clustering methods used. The explanation is that partitions resulting from the various clustering methods are conjunctively aggregated by intersecting groups, which increases the number of groups and decreases their size.

4.3. *Experimental results*

We demonstrate in this section the feasibility of our approach by reporting a practical application. To test the methodology, we have used part of a data set we used in our paper (Domingo-Ferrer and Torra 2001). The data set consists of 13 variables and 90 records that have been obtained using the Data Extraction System (DES) of the U.S. Census Bureau (2003). From the available data sources we chose the Current Population Survey corresponding to 1995—specifically, the file group "March Questionnaire Supplement—Person Data Files".

Variables in the data set are as follows: AFNLWGT (Final weight), AGI (Adjusted gross income), EMCONTRB (Employer contribution for health insurance), ERNVAL (Business or farm net earnings), FEDTAX (Federal income tax liability), FICA (Social Security retirement payroll reduction), INTVAL (Amount of interest income), PEARNVAL (Total personal earnings), POTHVAL (Total other persons income), PTOTVAL (Total personal income), STATETAX (State income tax liability), TAXINC (Taxable income amount), WSALVAL (Amount: Total wage and salary).

File A consists of variables AFNLWGT, EMCONTRB, PTOTVAL, TAXINC, POTHVAL, PEARNVAL and WSALVAL. File B consists of variables AGI, FEDTAX, STATETAX, INTVAL, FICA and ERNVAL. Table 3 shows the correlations among variables across files, to illustrate the level of similarity in the structural information in both files. It is enough that there be some variable pairs with high correlation, even if other variable pairs are uncorrelated.

We next illustrate the operation of our group-level re-identification method on files A and B. The following steps were performed:

1. Before splitting the original data set, a standardization of the domains of variables was performed. We standardized values x of variable V in the $[0, 1]$ interval as $x' = (x - \min(V))/(\max(V) - \min(V))$.
2. The partitions specified by Tables 1 and 2 were computed. To do so, $t = 6$ classification techniques were applied to both files. Each technique led to a dendrogram and for each dendrogram a partition was obtained. Dendrograms were constructed using SAHN (Everitt 1977) methods (i.e. sequential, agglomerative, hierarchical, non-overlapping methods). Several similarity functions (functions to compute similarities between variables/groups) and classification criteria (how to compute, from already known similarities, the similarity between a new group and the previous existing ones) were applied. Similarity functions used were based on Manhattan distance, on differences and on taxonomic distance; classification criteria used were the arithmetic average and centroid clustering. Detailed descriptions of these functions and criteria and their properties can be found in Everitt (1977). Once the partitions were obtained for the six methods, the intersection partitions in $\Pi(A, \mathbf{CP})$ and $\Pi(B, \mathbf{CP})$ were computed. The latter partitions together with the partitions obtained using the six classification methods are given in Tables 4 and 5.
3. The maximization problem (Problem 1) was heuristically solved, with the constraint that f_i's be one-to-one and group-level re-identification induced by m be such that a

Table 3. *Correlations between variables in files A and B*

	AFNLWGT	EMCONTRB	PTOTVAL	TAXINC	POTHVAL	PEARNVAL	WSALVAL
AGI	−0.111	0.469	0.782	0.972	0.027	0.774	0.773
FEDTAX	−0.100	0.304	0.741	0.977	0.114	0.703	0.703
STATETAX	−0.093	0.378	0.784	0.957	0.114	0.746	0.745
INTVAL	0.112	−0.130	0.255	0.155	0.648	0.035	0.035
FICA	−0.135	0.612	0.892	0.697	−0.185	0.956	0.956
ERNVAL	−0.095	0.629	0.922	0.707	−0.150	0.975	0.975

Table 4. *Intersection partitions $\pi_j^A \in \Pi(A, \mathbf{CP})$ and the corresponding groups $C_{i,A,j} \in \mathbf{C}_{i,A}$. The last six columns correspond to the six clustering techniques CP_i used: aa and cc are the classification criteria and stand for arithmetic average and centroid clustering, respectively; μ, δ and τ refer to similarity functions based, respectively, on Manhattan distance, differences and taxonomic distance. The column whose header is the i-th clustering technique CP_i contains the groups $C_{i,A,j} \in \mathbf{C}_{i,A}$. The re-identification function m maps partitions π_j^A to $\pi_{m(j)}^B$ (see Table 5). The 24 correctly re-identified records in the first column are shown within a box; the 5 correct group-level re-identifications which are not record-level are shown between parentheses*

π_j^A	$\pi_{m(j)}^B$	CP_1 aa,μ	CP_2 aa,δ	CP_3 aa,τ	CP_4 cc,μ	CP_5 cc,δ	CP_6 cc,τ
$\boxed{a_2}$	b_2	$C_{1,A,4}$	$C_{2,A,1}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,36}$	$C_{6,A,1}$
$\boxed{a_3}$	b_3	$C_{1,A,4}$	$C_{2,A,1}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,9}$	$C_{6,A,4}$
$\boxed{a_1}$	b_1	$C_{1,A,4}$	$C_{2,A,1}$	$C_{3,A,6}$	$C_{4,A,1}$	$C_{5,A,10}$	$C_{6,A,1}$
a_{82}	b_8	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,2}$	$C_{6,A,3}$
a_{83}	b_{47}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,12}$	$C_{6,A,3}$
(a_{81})	b_{78}, b_{81}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,11}$	$C_{6,A,1}$
$\boxed{a_{84}}$	b_{84}	$C_{1,A,3}$	$C_{2,A,2}$	$C_{3,A,3}$	$C_{4,A,2}$	$C_{5,A,19}$	$C_{6,A,2}$
a_{80}	b_{87}	$C_{1,A,3}$	$C_{2,A,2}$	$C_{3,A,3}$	$C_{4,A,2}$	$C_{5,A,10}$	$C_{6,A,2}$
a_{87}	b_{30}, b_{33}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,11}$	$C_{6,A,4}$
(a_{63})	$b_5, b_{20}, b_{49}, b_{63}$	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,39}$	$C_{6,A,3}$
a_{89}	b_{45}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,39}$	$C_{6,A,4}$
(a_{78})	b_{78}, b_{81}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,35}$	$C_{6,A,1}$
a_{54}	b_{60}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,6}$	$C_{6,A,3}$
$\boxed{a_{90}}$	b_{90}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,6}$	$C_{6,A,4}$
$\boxed{a_{51}}$	b_{51}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,38}$	$C_{6,A,3}$
a_{86}	b_{77}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,38}$	$C_{6,A,4}$
$\boxed{a_4}$	b_4	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,32}$	$C_{6,A,3}$
$\boxed{a_{88}}$	b_{88}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,31}$	$C_{6,A,4}$
$\boxed{a_6}$	b_6	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,15}$	$C_{6,A,3}$
a_{38}	b_{24}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,27}$	$C_{6,A,1}$
$\boxed{a_{39}}$	b_{39}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,14}$	$C_{6,A,1}$
$\boxed{a_{36}}$	b_{36}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,1}$	$C_{5,A,13}$	$C_{6,A,1}$
$\boxed{a_{40}}$	b_{40}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,21}$	$C_{6,A,1}$
$\boxed{a_{35}}$	b_{35}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,13}$	$C_{6,A,1}$
a_{41}	b_{72}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,14}$	$C_{6,A,1}$
a_{34}	b_{37}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,32}$	$C_{6,A,4}$
a_{42}	b_9, b_{74}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,22}$	$C_{6,A,4}$
a_{33}	b_{57}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,26}$	$C_{6,A,1}$
a_{44}	b_{21}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,29}$	$C_{6,A,4}$
a_{32}	b_{46}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,22}$	$C_{6,A,3}$
a_{46}, a_{62}	b_{67}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,17}$	$C_{6,A,3}$
a_{31}	b_{43}, b_{89}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,23}$	$C_{6,A,4}$
a_{47}	b_7, b_{13}, b_{66}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,7}$	$C_{6,A,3}$
a_{30}	b_{28}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,1}$	$C_{5,A,16}$	$C_{6,A,1}$
$\boxed{a_{50}}$	b_{50}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,11}$	$C_{6,A,3}$
a_{29}	b_{76}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,15}$	$C_{6,A,4}$
a_{52}	b_{22}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,30}$	$C_{6,A,1}$
a_{28}	b_9, b_{74}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,9}$	$C_{6,A,4}$
$a_{16}, \boxed{a_{53}}$	b_{53}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,1}$	$C_{5,A,30}$	$C_{6,A,1}$
$\boxed{a_{26}}$	b_{26}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,15}$	$C_{6,A,1}$
$\boxed{a_{55}}$	b_{55}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,28}$	$C_{6,A,1}$
$\boxed{a_{25}}$	b_{25}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,24}$	$C_{6,A,1}$
$\boxed{a_{56}}$	b_{56}	$C_{1,A,3}$	$C_{2,A,2}$	$C_{3,A,3}$	$C_{4,A,2}$	$C_{5,A,11}$	$C_{6,A,2}$
a_{24}	b_{54}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,25}$	$C_{6,A,3}$

(*Continued on next page.*)

Table 4. *(Continued)*

π_j^A	$\pi_{m(j)}^B$	CP_1 aa,μ	CP_2 aa,δ	CP_3 aa,τ	CP_4 cc,μ	CP_5 cc,δ	CP_6 cc,τ
a_{23}	b_{64}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,1}$	$C_{5,A,18}$	$C_{6,A,1}$
$\boxed{a_{58}}$	b_{58}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,1}$	$C_{6,A,1}$
a_{22}	b_{52}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,16}$	$C_{6,A,1}$
a_{21}	b_{80}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,2}$	$C_{4,A,1}$	$C_{5,A,17}$	$C_{6,A,1}$
a_{60}	b_{61}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,31}$	$C_{6,A,3}$
a_{19}	b_{79}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,17}$	$C_{6,A,1}$
$\boxed{a_{17}}$	b_{17}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,29}$	$C_{6,A,1}$
a_{66}, a_{73}	b_{82}, b_{83}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,9}$	$C_{6,A,3}$
a_{67}	b_{18}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,34}$	$C_{6,A,3}$
a_{14}	b_{70}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,1}$	$C_{5,A,19}$	$C_{6,A,1}$
a_{59}, a_{68}	b_{23}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,1}$	$C_{5,A,17}$	$C_{6,A,1}$
a_{18}	b_{69}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,11}$	$C_{6,A,3}$
a_{76}	b_{86}	$C_{1,A,6}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,11}$	$C_{6,A,4}$
a_{69}	b_7, b_{13}, b_{66}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,19}$	$C_{6,A,3}$
(a_{49})	$b_5, b_{20}, b_{49}, b_{63}$	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,4}$	$C_{6,A,3}$
a_{74}	b_{31}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,4}$	$C_{6,A,4}$
a_{70}	b_{68}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,7}$	$C_{6,A,4}$
a_{15}	b_{11}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,20}$	$C_{6,A,3}$
a_{45}	b_{29}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,20}$	$C_{6,A,4}$
$\boxed{a_{71}}$	b_{71}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,10}$	$C_{6,A,1}$
$\boxed{a_{27}}$	b_{27}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,8}$	$C_{6,A,3}$
a_{37}	b_{42}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,8}$	$C_{6,A,4}$
a_{72}, a_{79}	b_{16}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,19}$	$C_{6,A,1}$
(a_{20})	$b_5, b_{20}, b_{49}, b_{63}$	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,33}$	$C_{6,A,3}$
a_{43}	b_{65}	$C_{1,A,5}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,33}$	$C_{6,A,4}$
a_{75}	b_{14}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,9}$	$C_{6,A,1}$
a_{61}	b_{73}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,5}$	$C_{6,A,3}$
a_{65}	b_{10}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,5}$	$C_{6,A,4}$
$\boxed{a_{48}}$	b_{48}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,3}$	$C_{6,A,3}$
a_{64}	b_{43}, b_{89}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,3}$	$C_{6,A,4}$
a_{13}	b_{32}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,37}$	$C_{6,A,3}$
a_{77}	b_{44}	$C_{1,A,6}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,37}$	$C_{6,A,4}$
a_{11}	$b_5, b_{20}, b_{49}, b_{63}$	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,23}$	$C_{6,A,3}$
a_{12}, a_{57}	b_{19}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,1}$	$C_{4,A,1}$	$C_{5,A,19}$	$C_{6,A,1}$
a_5	b_{82}, b_{83}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,9}$	$C_{6,A,3}$
a_{10}, a_{85}	b_{75}	$C_{1,A,4}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,1}$	$C_{5,A,12}$	$C_{6,A,1}$
a_7	b_{62}	$C_{1,A,2}$	$C_{2,A,2}$	$C_{3,A,5}$	$C_{4,A,3}$	$C_{5,A,19}$	$C_{6,A,3}$
a_9	b_{41}	$C_{1,A,6}$	$C_{2,A,2}$	$C_{3,A,6}$	$C_{4,A,4}$	$C_{5,A,19}$	$C_{6,A,4}$
a_8	b_{15}	$C_{1,A,1}$	$C_{2,A,2}$	$C_{3,A,4}$	$C_{4,A,1}$	$C_{5,A,21}$	$C_{6,A,3}$

group $\pi_j^B \in \Pi(B, \mathbf{CP})$ receives at most $|\pi_j^B|$ mappings from $\Pi(A, \mathbf{CP})$.

The functions f_i in the solution are given in Table 6. The second column of Table 4 gives the group-level re-identification induced by the solution. From the first two columns of Table 4, it can be seen that, with just $t = 6$ clustering techniques, a group-level re-identification is obtained which in most cases is a record-level re-identification (groups with a single record are obtained in the second column).

After applying the method, we measured its effectiveness in terms of correct re-identifications. These can be counted from Table 4 as follows: a record a_i in the first column is correctly re-identified if b_i appears in the second column and same row. *This gives 29 correct group-level re-identifications for 90 records, 24 of which are record-level re-identifications (such that the group in the second column contains a single b_i). Thus, the percent of correct record-level re-identifications is 26.6%.* The 24 correctly re-identified records a_i are shown in Table 4 within a box; the 5 correct group-level re-identifications which are not record-level are shown between parentheses in that table.

Table 5. *Intersection partitions $\pi_j^B \in \Pi(B, \mathbf{CP})$ and the corresponding groups $C_{i,B,j} \in \mathbf{C}_{i,B}$. Meanings of aa, cc, μ, δ and τ as in Table 4*

π_j^B	CP_1 aa,μ	CP_2 aa,δ	CP_3 aa,τ	CP_4 cc,μ	CP_5 cc,δ	CP_6 cc,τ
b_2	$C_{1,B,1}$	$C_{2,B,2}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,26}$	$C_{6,B,2}$
b_3	$C_{1,B,2}$	$C_{2,B,2}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,22}$	$C_{6,B,2}$
b_1	$C_{1,B,1}$	$C_{2,B,2}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,8}$	$C_{6,B,2}$
b_{82}, b_{83}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,22}$	$C_{6,B,3}$
b_{78}, b_{81}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,14}$	$C_{6,B,2}$
b_{84}	$C_{1,B,4}$	$C_{2,B,1}$	$C_{3,B,3}$	$C_{4,B,3}$	$C_{5,B,22}$	$C_{6,B,4}$
b_{80}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,5}$	$C_{6,B,2}$
b_{85}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,28}$	$C_{6,B,2}$
b_{87}	$C_{1,B,2}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,22}$	$C_{6,B,2}$
b_{79}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,17}$	$C_{6,B,2}$
$b_5, b_{20}, b_{49}, b_{63}$	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,16}$	$C_{6,B,3}$
b_{43}, b_{89}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,4}$	$C_{4,B,1}$	$C_{5,B,16}$	$C_{6,B,1}$
b_{54}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,13}$	$C_{6,B,3}$
b_{90}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,6}$	$C_{6,B,2}$
b_{51}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,21}$	$C_{6,B,3}$
b_{86}	$C_{1,B,2}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,11}$	$C_{6,B,2}$
b_4	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,18}$	$C_{6,B,3}$
b_{88}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,35}$	$C_{6,B,2}$
b_6	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,15}$	$C_{6,B,3}$
b_{38}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,37}$	$C_{6,B,2}$
b_{39}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,24}$	$C_{6,B,2}$
b_{36}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,25}$	$C_{6,B,2}$
b_{40}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,15}$	$C_{6,B,2}$
b_{35}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,25}$	$C_{6,B,2}$
b_{41}	$C_{1,B,2}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,23}$	$C_{6,B,2}$
b_{34}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,16}$	$C_{6,B,2}$
b_{42}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,4}$	$C_{4,B,1}$	$C_{5,B,7}$	$C_{6,B,1}$
b_{30}, b_{33}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,27}$	$C_{6,B,2}$
b_{44}	$C_{1,B,2}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,30}$	$C_{6,B,2}$
b_{32}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,30}$	$C_{6,B,3}$
b_{46}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,9}$	$C_{6,B,3}$
b_{31}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,16}$	$C_{6,B,2}$
b_{47}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,12}$	$C_{6,B,3}$
b_{50}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,4}$	$C_{6,B,3}$
b_{29}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,20}$	$C_{6,B,2}$
b_{52}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,28}$	$C_{6,B,2}$
b_{28}	$C_{1,B,2}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,18}$	$C_{6,B,2}$
b_{53}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,4}$	$C_{6,B,2}$
b_{26}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,23}$	$C_{6,B,2}$
b_{55}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,33}$	$C_{6,B,2}$
b_{25}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,31}$	$C_{6,B,2}$
b_{56}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,29}$	$C_{6,B,2}$
b_{24}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,27}$	$C_{6,B,2}$
b_{57}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,39}$	$C_{6,B,2}$
b_{23}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,17}$	$C_{6,B,2}$
b_{58}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,1}$	$C_{6,B,2}$
b_{22}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,32}$	$C_{6,B,2}$
b_{59}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,29}$	$C_{6,B,2}$
b_{21}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,5}$	$C_{4,B,1}$	$C_{5,B,10}$	$C_{6,B,1}$
b_{60}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,6}$	$C_{6,B,3}$
b_{19}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,37}$	$C_{6,B,2}$
b_{62}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,34}$	$C_{6,B,3}$
b_{17}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,3}$	$C_{6,B,2}$
b_7, b_{13}, b_{66}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,38}$	$C_{6,B,3}$
b_{16}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,16}$	$C_{6,B,2}$
b_{67}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,17}$	$C_{6,B,3}$
b_{14}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,28}$	$C_{6,B,2}$
b_{68}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,38}$	$C_{6,B,2}$
b_{18}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,19}$	$C_{6,B,3}$
b_{76}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,4}$	$C_{4,B,1}$	$C_{5,B,13}$	$C_{6,B,1}$
b_{69}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,11}$	$C_{6,B,3}$
b_9, b_{74}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,4}$	$C_{4,B,1}$	$C_{5,B,38}$	$C_{6,B,1}$
b_{70}	$C_{1,B,2}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,34}$	$C_{6,B,2}$
b_{15}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,36}$	$C_{6,B,3}$
b_{45}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,4}$	$C_{4,B,1}$	$C_{5,B,36}$	$C_{6,B,1}$
b_{71}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,2}$	$C_{6,B,2}$
b_{27}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,7}$	$C_{6,B,3}$
b_{37}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,19}$	$C_{6,B,2}$
b_{72}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,1}$	$C_{4,B,2}$	$C_{5,B,24}$	$C_{6,B,2}$
b_{73}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,5}$	$C_{6,B,3}$
b_{75}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,12}$	$C_{6,B,2}$
b_{61}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,35}$	$C_{6,B,3}$
b_{65}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,26}$	$C_{6,B,2}$
b_{48}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,26}$	$C_{6,B,3}$
b_{64}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,9}$	$C_{6,B,2}$
b_{77}	$C_{1,B,6}$	$C_{2,B,1}$	$C_{3,B,4}$	$C_{4,B,1}$	$C_{5,B,21}$	$C_{6,B,1}$
b_{11}	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,20}$	$C_{6,B,3}$
b_{12}	$C_{1,B,1}$	$C_{2,B,1}$	$C_{3,B,3}$	$C_{4,B,2}$	$C_{5,B,34}$	$C_{6,B,2}$
b_{10}	$C_{1,B,5}$	$C_{2,B,1}$	$C_{3,B,6}$	$C_{4,B,2}$	$C_{5,B,5}$	$C_{6,B,2}$
b_8	$C_{1,B,3}$	$C_{2,B,1}$	$C_{3,B,2}$	$C_{4,B,4}$	$C_{5,B,10}$	$C_{6,B,3}$

Benchmarking the results. Since both files have the same number of records, it is easy to benchmark the above result with the strategies discussed in Section 2.

The first benchmark was random linkage. The probability of randomly obtaining 24 or more correct links out of 90 links (see Proposition 1) is $0.6175115 \cdot 10^{-24}$. Thus, the expected number of correct re-identifications for 90 records is $O(10^{-22})$.

Note 3. In fact, using Expression (1), one can see that the probability of randomly obtaining more than 3 correct links out of 90 links is already negligible.

The second benchmark consisted on using one-dimensional ranking for record linkage:

- For one-dimensional ranking based on the first principal component, 5 out of 90 records were correctly re-identified (5.5%). The re-identified pairs were (a_7, b_7), (a_{13}, b_{13}), (a_{31}, b_{31}), (a_{40}, b_{40}) and (a_{61}, b_{61}).
- For one-dimensional ranking based on the sum of z-scores, also 5 out of 90 records were correctly re-identified (5.5%). The re-identified pairs were (a_5, b_5), (a_7, b_7), (a_{13}, b_{13}), (a_{35}, b_{35}) and (a_{89}, b_{89}).

Table 6. *Functions f_i that maximize the similarity between partitions in Tables 4 and 5. Notation: f_i maps $\mathbf{C}_{i,A}$ to $\mathbf{C}_{i,B}$, so the groups $C_{i,A,j}$ on the left of the arrows for f_i are the groups in partition $\mathbf{C}_{i,A}$ (see Table 4) and the groups $C_{i,B,j}$ on the right of the arrows for f_i are the groups in partition $\mathbf{C}_{i,B}$ (see Table 5)*

f_1	$C_{1,A,1} \to C_{1,B,1}$	$C_{1,A,2} \to C_{1,B,3}$	$C_{1,A,3} \to C_{1,B,4}$	$C_{1,A,4} \to C_{1,B,5}$
	$C_{1,A,5} \to C_{1,B,6}$	$C_{1,A,6} \to C_{1,B,2}$		
f_2	$C_{2,A,1} \to C_{2,B,2}$	$C_{2,A,2} \to C_{2,B,1}$		
f_3	$C_{3,A,1} \to C_{3,B,1}$	$C_{3,A,2} \to C_{3,B,5}$	$C_{3,A,3} \to C_{3,B,3}$	$C_{3,A,4} \to C_{3,B,2}$
	$C_{3,A,5} \to C_{3,B,4}$	$C_{3,A,6} \to C_{3,B,6}$		
f_4	$C_{4,A,1} \to C_{4,B,2}$	$C_{4,A,2} \to C_{4,B,3}$	$C_{4,A,3} \to C_{4,B,4}$	$C_{4,A,4} \to C_{4,B,1}$
f_5	$C_{5,A,1} \to C_{5,B,1}$	$C_{5,A,2} \to C_{5,B,10}$	$C_{5,A,3} \to C_{5,B,29}$	$C_{5,A,4} \to C_{5,B,23}$
	$C_{5,A,5} \to C_{5,B,5}$	$C_{5,A,6} \to C_{5,B,6}$	$C_{5,A,7} \to C_{5,B,38}$	$C_{5,A,8} \to C_{5,B,7}$
	$C_{5,A,9} \to C_{5,B,22}$	$C_{5,A,10} \to C_{5,B,8}$	$C_{5,A,11} \to C_{5,B,11}$	$C_{5,A,12} \to C_{5,B,12}$
	$C_{5,A,13} \to C_{5,B,25}$	$C_{5,A,14} \to C_{5,B,24}$	$C_{5,A,15} \to C_{5,B,15}$	$C_{5,A,16} \to C_{5,B,28}$
	$C_{5,A,17} \to C_{5,B,17}$	$C_{5,A,18} \to C_{5,B,32}$	$C_{5,A,19} \to C_{5,B,34}$	$C_{5,A,20} \to C_{5,B,20}$
	$C_{5,A,21} \to C_{5,B,37}$	$C_{5,A,22} \to C_{5,B,9}$	$C_{5,A,23} \to C_{5,B,16}$	$C_{5,A,24} \to C_{5,B,31}$
	$C_{5,A,25} \to C_{5,B,13}$	$C_{5,A,26} \to C_{5,B,39}$	$C_{5,A,27} \to C_{5,B,27}$	$C_{5,A,28} \to C_{5,B,33}$
	$C_{5,A,29} \to C_{5,B,3}$	$C_{5,A,30} \to C_{5,B,4}$	$C_{5,A,31} \to C_{5,B,35}$	$C_{5,A,32} \to C_{5,B,18}$
	$C_{5,A,33} \to C_{5,B,2}$	$C_{5,A,34} \to C_{5,B,19}$	$C_{5,A,35} \to C_{5,B,14}$	$C_{5,A,36} \to C_{5,B,26}$
	$C_{5,A,37} \to C_{5,B,30}$	$C_{5,A,38} \to C_{5,B,21}$	$C_{5,A,39} \to C_{5,B,36}$	
f_6	$C_{6,A,1} \to C_{6,B,2}$	$C_{6,A,2} \to C_{6,B,4}$	$C_{6,A,3} \to C_{6,B,3}$	$C_{6,A,4} \to C_{6,B,1}$

Thus, the results obtained with our method are significantly better than those obtained using the benchmark strategies.

5. Conclusions and future work

This paper tackles the problem of re-identification of individuals when non-common variables are shared by two information sources. The results obtained are promising and show that re-identification is indeed possible in such a scenario. This is good news for data mining and less so for statistical disclosure control; indeed, data protectors should be aware that powerful record linkage tools are conceivable which do not require variables to be shared between the intruder's side information and the files released by NSOs.

In spite of the encouraging results obtained so far, more research is needed to reach more definitive and general conclusions. Lines for future work include the following:

- Carry out controlled experiments in a systematic way in order to precisely characterize the conditions under which the proposed method is successful.
- Characterize the clustering techniques which are most suitable for re-identification. Suitable techniques are those such that combining a small number of them yields a large number of single-record groups and a good group-level re-identification rate.
- The methodology introduced here has been applied to quantitative data. Nothing prevents it from being applied to categorical data (ordinal or nominal), since clustering techniques exist that deal with that kind of data. Therefore, the methodology presented can be used to link files containing both quantitative and categorical variables.

Appendix

Proof (Proposition 1): The probability of correctly re-identifying exactly r records is the probability that exactly r records are equally mapped by the intruder's random permutation p and by the correct permutation p_0. This probability is the same as the probability that a random permutation p of the records in A maps r records to fixed positions (specified by p_0). Without loss of generality, we can assume that the fixed positions are the original positions of the r records (this is just taking a particular p_0). Thus the probability of correctly re-identifying r records out of n can be computed as the probability that a permutation of n records leaves exactly r records in their original position. This is what is computed in the rest of the proof.

The number of possible re-identifications is the number of possible permutations p, that is, $n!$. Now let us compute the number of permutations such that there are exactly r elements unmoved. These permutations can be built by taking $n-r$ elements from the correct permutation and permuting them in such a way that none keeps its original position. The number of possible subsets of $n-r$ elements is $n!/((n-r)!r!)$. Call *permutations of $n-r$ elements without a fixed point* those permutations of a set of $n-r$ elements in which no element keeps its original position. According to Reinhard and Soeder (1997), the number of permutations of $n-r$ elements without a fixed point is

$$(n-r)! \sum_{v=0}^{n-r} \frac{(-1)^{(n-r)}}{v!}$$

Thus, the number of permutations such that there are exactly r elements in the correct position is:

$$\frac{n! \sum_{v=0}^{n-r} \frac{(-1)^{(n-r)}}{v!}}{r!} \tag{3}$$

Now the probability of finding at random a permutation with exactly r elements in the correct position can be computed by dividing Expression (3) by the total number $n!$ of permutations. □

Acknowledgments

Thanks go to Josep M. Mateo-Sanz for helping with some computations. This work has been partially supported by the European Commission under project no. IST-2000-25069 "CASC". Assistance of guest editor Leon Willenborg is gratefully acknowledged.

References

Anderberg M.R. 1973. Cluster Analysis for Applications. Academic Press, New York.

Bacher J., Brand R., and Bender S. 2002. Re-identifying register data by survey data using cluster analysis: An empirical study. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5): 589–608.

Dempster A.P., Laird N.N., and Rubin D.B. 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39: 1–38.

Domingo-Ferrer J. and Torra V. 2002. Validating distance-based record linkage with probabilistic record linkage. Lecture Notes in Computer Science 2504: 207–215.

Domingo-Ferrer J. and Torra V. 2001. A quantitative comparison of disclosure control methods for microdata. In: Zayatz L., Doyle P., Theeuwes J., and Lane J. (Eds.), Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, North-Holland, Amsterdam, pp. 111–134.

Duda R.O., Hart P.E., and Stork D.G. 2001. Pattern Classification, 2nd edition, Wiley, New York.

Everitt B. 1977. Cluster Analysis. Heinemann Educational Books Ltd., London.

Fellegi I.P. and Sunter A.B. 1969. A theory of record linkage. Journal of the American Statistical Association 64: 1183–1210.

Fishburn P.C. and Rubinstein A. 1986. Aggregation of equivalence relations. Journal of Classification 3: 61–65.

Godo L. and Torra V. 2000. On aggregation operators for ordinal qualitative information. IEEE Transaction on Fuzzy Systems 8(2): 143–154.

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.

Hastie T., Tibshirani R., and Friedman J. 2001. The Elements of Statistical Learning. Springer, Berlin.

Hoppner F., Klawonn F., Kruse R., and Runkler T. 1999. Fuzzy Cluster Analysis. Wiley, New York.

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: 414–420.

Neumann D.A. and Norton V.T. (Jr). 1986. Clustering and isolation in the consensus problem for partitions. Journal of Classification 3: 281–297.

Newcombe H.B., Kennedy J.M., Axford S.J., and James A.P. 1959. Automatic linkage of vital records. Science 130: 954–959.

Pagliuca D. and Seri G. 1999. Some Results of Individual Ranking Method on the System of Enterprise Accounts Annual Survey, Esprit SDC Project, Deliverable MI-3/D2.

Reinhard F. and Soeder H. 1997. Atlas des mathématiques, Librairie Générale Française, Paris.

Robinson-Cox J.F. 1998. A record-linkage approach to imputation of missing data: Analyzing tag retention in a tag-recapture experiment. Journal of Agricultural, Biological, and Environmental Statistics 3: 48–61.

Rosman D.L. 1995. The Linkage of Hospital and Police Information on Road Crash Casualties: An Investigation of Alternative Methods, Report N. RIIP-7.

Torra V. and Domingo-Ferrer J. 2003. Record linkage methods for multidatabase data mining. In: Torra V. (Ed.), Information Fusion in Data Mining, Springer, Berlin, pp. 99–130.

Winkler W.E. 1995a. Matching and record linkage. In: Cox B.G. (Ed.), Business Survey Methods, Wiley, New York, pp. 355–384.

Winkler W.E. 1995b. Advanced methods for record linkage. In: Proceedings of the American Statistical Association Section on Survey Research Methods, pp. 467–472.

`http://www.integrity.com`

U.S. Bureau of the Census Data Extraction System, 2003. `http://www.census.gov/DES/www/welcome.html`.