Journal of Official Statistics, Vol. 18, No. 2, 2002, pp. 275–289

Local Recoding and Record Swapping by Maximum Weight Matching for Disclosure Control of Microdata Sets

Akimichi Takemura[1]

As a technique of disclosure control of microdata sets, we propose local recoding and record swapping based on the optimum matching of the records, where pairs of close records are formed and observed values are recoded or swapped within each pair. For optimally forming pairs we can employ Edmonds's algorithm (Edmonds 1965) of maximum weight matching. We illustrate the technique by applying it to the Japanese causes of death statistics data.[2]

Key words: Edmonds's algorithm; local suppression; nearest neighbor; NP-complete; perturbation.

1. Introduction

Global recoding is the obvious and the most important technique in disclosure control of microdata sets. In global recoding the observed values are grouped into broader intervals or categories. It is called global since the grouping is performed uniformly throughout the microdata set. In this article we consider local recoding (De Waal and Willenborg 1996; De Waal and Willenborg 1999), where each observed value is recoded into broader intervals or categories when necessary. We also consider record swapping (Schlörer 1981; Dalenius and Reiss 1982), because our technique can be equally applied to record swapping and local recoding.

As a means of performing local recoding and record swapping we propose matching or pairing of close individuals of a microdata set. When two individuals are grouped into a pair, we can locally recode or swap observations within the pair. The idea of local recoding is not necessarily tied to matching, and other techniques may be used to perform local recoding. One advantage of matching is that a well-known algorithm of optimum matching is available and local recoding and record swapping can be performed in a reasonable amount of computer time.

Many techniques have been proposed for disclosure control of microdata sets. The local suppression, where individual observations are marked as missing, is extensively discussed

[1] Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan. E-mail: takemura@stat.t.u-tokyo.ac.jp.
[2] This data set was used under permit No. 40, 1997, of the Management and Coordination Agency, Government of Japan, for the purpose of disclosure control experiments.
Acknowledgments: This article owes very much to contributions by Daishin Nakamura. First, he provided the author with a working program for full and approximate optimization based on Edmonds's algorithm for the two-sided matching. The material of Section 3 is largely due to him. He has also pointed out the NP-completeness of optimally forming triples. The presentation of the article has been greatly improved by detailed comments by three referees and an Associate Editor.

in Section 5.4 of Willenborg and de Waal (1996) and references therein. In Section 2.1 below we argue that local suppression is an extreme form of local recoding. In this sense local recoding is a more general technique of disclosure control than local suppression.

Addition of noise to original observations is dicussed by many authors including Fuller (1993) and Duncan and Pearson (1991). One conceptual difficulty of the addition of noise is that it is not clear how one can add noise to purely categorical variables. The post randomization method (PRAM; see Gouweleeuw et al. (1998)) is a probabilistic perturbation technique for categorical variables. Fienberg et al. (1998) give a survey of perturbation techniques for categorical data. One advantage of the present procedure is that local recoding and record swapping can be applied to a data set with both continuous and categorical variables.

The idea of pairing presented in this article is close to the idea of multivariate microaggregation in Mateo-Sanz and Domingo-Ferrer (1998). They use clustering algorithms whereas we use matching algorithms to form groups. One disadvantage of clustering might be the lack of a well-defined notion of optimality among various clustering algorithms. Various alternative techniques for implementing microaggregation are discussed in Defays and Anwar (1998) and references therein.

The organization of this article is as follows. In Section 2 we explain the idea of matching by a simple numerical example. In Section 3 we discuss full optimization and approximate optimization procedures based on Edmonds's algorithm. In Section 4 our procedures are applied to a real data set of considerable size. We shall show that computations can be done in a reasonable amount of time. Section 5 comprises discussion.

2. Simple Numerical Example

Here we discuss a simple numerical example at some length, because our idea and technique are best explained by an example.

2.1. Example data set

Consider a hypothetical population consisting of ten household records listed in Table 1. Table 1 presents the whole population and there is no complication associated with sampling, such as the distinction of the population unique and the sample unique. The variables observed are (1) Age of head of household, (2) Size of household, (3) Income,

Table 1. Hypothetical population of size ten

No.	Age	Size	Income	Occup.
1	47	4	490	A
2	52	3	720	B
3	38	4	480	A
4	43	5	610	C
5	46	3	870	B
6	35	3	540	A
7	43	4	640	C
8	51	2	560	A
9	44	6	580	A
10	33	3	380	A

Table 2. Result of obvious global recoding

No.	Age	Size	Income	Occup.
1	40	4	400	A
2	50	3	700	B
3	30	4	400	A
4	40	5	600	C
5	40	3	800	B
6	30	3	500	A
7	40	4	600	C
8	50	2	500	A
9	40	6	500	A
10	30	3	300	A

and (4) Occupation in three categories (A, B, or C). We consider these four variables as key variables which can be used to identify the individual household records.

From Table 1 we immediately see that all the households are population uniques. Therefore we need some disclosure control measures to avoid identification of households. It is reasonable to round the values of age and income. If we round the age down to 10's and the income to 100's, we obtain Table 2.

We see that even after this global recoding all the households remain population uniques. This can be understood by the following simple calculation. In Table 2 we count the number of categories present for each variable. The numbers are three for Age, five for Size, five for Income, and three for Occupation. Therefore the total number of possible combinations of the categories is

$$3 \times 5 \times 5 \times 3 = 225$$

We can think of ten households thrown into 225 boxes and it is likely that these households will fall into different boxes. The usual calculation of the ''birthday problem,'' i.e., the calculation of the probability of finding two people with the same birthday in a group of people, yields an approximate probability $e^{-(1+\cdots+9)/225} = e^{-0.2} = 0.82$ of ten households thrown into different boxes.

We proceed to more drastic global recoding: grouping the household sizes into two categories (≥ 4 or ≤ 3) and income into two categories (≥ 500 or < 500). The total number of combinations is reduced to $3 \times 2 \times 2 \times 3 = 36$ and the resulting table is Table 3.

We note that in Table 3 households No. 4 and No. 7 coincide and are no longer population uniques. However, the other eight households remain population uniques. At this point it seems to be very difficult to perform further global recoding without losing a substantial amount of information in the data set. This suggests that relying on global recoding alone may result in a microdata set which is too coarse.

Now let us consider the 2nd and the 5th household in Table 3. These two differ only in Age. Therefore we might *locally recode* Age for these two households and exhibit these households as follows:

No	Age	Size	Income	Occup.
2	40–50	3	500	B
5	40–50	3	500	B

Table 3. Result of further global recoding

No.	Age	Size	Income	Occup.
1	40	4	400	A
2	50	3	500	B
3	30	4	400	A
4	40	4	500	C
5	40	3	500	B
6	30	3	500	A
7	40	4	500	C
8	50	3	500	A
9	40	4	500	A
10	30	3	400	A

Then these two households are no longer population uniques. Alternatively we can *swap* the value of Age of these two households and exhibit these households as follows:

No	Age	Size	Income	Occup.
2	40	3	500	B
5	50	3	500	B

Then these two households are protected by perturbation of the observations.

Let us match the remaining six households into the following three pairs:

$$(1, 3), (6, 8), (9, 10)$$

and locally recode the observations into intervals or unions of categories. The result is shown in Table 4.

Note that Size and Income of the pair (9, 10) are denoted by ''*'' and locally suppressed. This is because merging two categories of a dichotomous variable (a variable which has been globally recoded into two categories) is equivalent to suppressing the observation. Therefore we can interpret local suppression as an extreme form of local recoding.

2.2. *Optimum matching based on distance function*

The matching of households in Table 4 was performed by inspection. Here we formulate the matching problem more precisely in order to perform the matching by a computer. The

Table 4. Local recoding by inspection from Table 3

No.	Age	Size	Income	Occup.
1	30–40	4	400	A
2	40–50	3	500	B
3	30–40	4	400	A
4	40	4	500	C
5	40–50	3	500	B
6	30–50	3	500	A
7	40	4	500	C
8	30–50	3	500	A
9	30–40	*	*	A
10	30–40	*	*	A

Table 5. Distances between ten households

	1	2	3	4	5	6	7	8	9	10
1	0	8	2	5	7	4	4	5	3	4
2	8	0	10	7	3	8	6	5	9	10
3	2	10	0	7	9	2	6	7	5	2
4	5	7	7	0	6	7	1	8	4	9
5	7	3	9	6	0	7	5	8	8	9
6	4	8	2	7	7	0	6	5	5	2
7	4	6	6	1	5	6	0	7	5	8
8	5	5	7	8	8	5	7	0	6	7
9	3	9	5	4	8	5	5	6	0	7
10	4	10	2	9	9	2	8	7	7	0

basic idea of the matching was to find close households. Therefore we introduce a distance function between the households. A distance function can be chosen in accordance with convenience. As a simple distance function we may use the Hamming distance, where we just count the number of variables with different values. It is probably better to consider relative importance of the variables and weight the variables accordingly.

Consider Table 2. Although the local recoding in Table 4 was obtained from Table 3, Table 3 is already too coarse and it seems to be better to perform local recoding to the values of Table 2. Concerning the distance function, we might argue as follows. Let us measure the difference of five years in the age as Distance ''1,'' since five years difference might be noticeable from the appearance. Then ten years difference in the age is measured as Distance 2. Concerning the size of the household, we measure the difference of 1 as just 1, since neighbours may know the exact household size. We measure the difference of 100 in the income as 1. Here we have in mind that in the case of Japanese households the difference of 100 in the annual income might be noticeable to neighbors. (The unit here is ten thousand yen.) Finally we measure the difference in the occupation as 2. As the total distance between two households, we add these individual distances for the four variables. Let $x = (x_1, x_2, x_3, x_4)$ and $y = (y_1, y_2, y_3, y_4)$ denote the values of Age, Size, Income, and Occupation of two households. Then the distance between x and y may be defined as

$$\text{dist}\,(x, y) = |x_1 - y_1|/5 + |x_2 - y_2| + |x_3 - y_3|/100 + 2I_{[x_4 \neq y_4]}$$

where $I_{[x_4 \neq y_4]}$ is the indicator function

$$I_{[x_4 \neq y_4]} = \begin{cases} 1, & \text{if } x_4 \neq y_4 \\ 0, & \text{if } x_4 = y_4 \end{cases}$$

Table 5 shows the distance matrix regarding the ten households. Using Table 5 we can list closest households (''nearest neighbors'') to each household, as shown in Table 6.

From Table 6 the average distance to the nearest neighbors is calculated as 2.4. It is noted that the relation to nearest neighbor may be ''one-sided.'' For example the nearest neighbor of household No. 9 is household No. 1, whereas the nearest neighbor of No. 1 is No. 3. If we allow this one-sidedness, we can match each household to its nearest neighbor and apply local recoding to each household. If there is more than one nearest household, we arbitrarily choose one of the nearest households. We call this type of local recoding

Table 6. Nearest neighbor and distance to nearest neighbor

	N.N.	Distance
1	3	2
2	5	3
3	1 or 6 or 10	2
4	7	1
5	2	3
6	3 or 10	2
7	4	1
8	1 or 2 or 6	5
9	1	3
10	3 or 6	2

''one-sided nearest neighbor local recoding'' or ''optimum one-sided matching.'' A resulting data set with local recoding is shown in Table 7.

In Table 7 each row corresponds to at least two households in the population. Therefore the one-sided nearest neighbor local recoding can withstand the ''fishing strategy attack'' (Müller et al. 1995), where an intruder chooses an arbitrary record of the microdata set and tries to identify this record in the population. On the other hand the one-sided nearest neighbor local recoding does not guarantee defense against the direct search attack (Müller et al. 1995), where an intruder possessing information on a household in the population tries to identify the household in the microdata set. For example household No. 9 of the hypothetical population corresponds only to the 9th row of Table 7 and in this sense the 9th row of Table 7 might be identified. This weakness clearly results from the one-sidedness of the matching.

We now allow only two-sided pairs and obtain optimum matching in the sense of minimizing the sum of the distances within the pairs. We call recoding by this type of two-sided matching ''two-sided nearest neighbor local recoding'' or ''optimum two-sided matching.'' This optimization problem is called ''maximum weight matching'' in the field of graph algorithms. In particular Edmonds's algorithm (Edmonds 1965) is a well-known algorithm for solving the maximum weight matching problem. In the next section we describe our maximization problem and Edmonds's algorithm more formally.

In our hypothetical example n is only ten and the total number in which five pairs out of

Table 7. One-sided matching to nearest neighbor

No.	Age	Size	Income	Occup.
1	30–40	4	400	A
2	40–50	3	700–800	B
3	30–40	4	400	A
4	40	4–5	600	C
5	40–50	3	700–800	B
6	30	3–4	400–500	A
7	40	4–5	600	C
8	50	2–3	500–700	A or B
9	40	4–6	400–500	A
10	30	3–4	300–400	A

Table 8. Two-sided matching to nearest neighbor

No.	Age	Size	Income	Occup.
1	30–40	4	400	A
2	50	2–3	700–800	B
3	30–40	4	400	A
4	40	4–5	600	C
5	50	2–3	700–800	B
6	30	3	300–500	A
7	40	4–5	600	C
8	40–50	2–6	500	A
9	40–50	2–6	500	A
10	30	3	300–500	A

ten households can be formed is

$$(n-1)\times(n-3)\times\cdots\times 3\times 1 = 9\times 7\times 5\times 3 = 945$$

Therefore we can check all 945 pairings and compute the sum of distances. Then the optimum matching is found to be

$$(1,3)\ (2,5)\ (4,7)\ (6,10)\ (8,9)$$

with the average distance of 2.8 within pairs. The resulting data set with local recoding is shown in Table 8.

From Table 7 and Table 8 we see that two-sided nearest neighbor local recoding leads to stronger protection accompanied by larger average distance within pairs. The advantage of the two-sided nearest neighbor local recoding is that it withstands both the fishing strategy attack and the direct search attack. From the computational viewpoint, the one-sided matching is very simple because we can treat each record separately, whereas the two-sided matching is more complicated and requires combinatorial optimization.

Once the two-sided optimum pairs have been obtained, the swapping can be done within these pairs. Since the pairs are formed optimally, the swapping is performed only between close records. We do not exhibit the result of swapping, since it can be immediately written down from Table 8. Note that if the matched records differ in more than one variable, the statistical agency can choose which observations to swap. Therefore the statistical agency can fine-tune the strength of protection by choosing the number of swapped observations.

The resulting optimum matching depends on the distance function. Further discussion on the choice of the distance function is given in Section 5.

3. Full Combinatorial Optimization by Edmonds's Algorithm and Its Approximation

In this section we explain our implementation of Edmonds's algorithm for our problem and an approximation to the full combinatorial optimization. The material in this section is largely due to Daishin Nakamura.

Let $G = (V, E)$ be a graph consisting of a set of vertices V and a set of edges E. A matching is a subset $\tilde{E} \subset E$ of the edges such that each vertex $v \in V$ is contained in at most one edge $e \subset \tilde{E}$. Suppose a weight w_e is associated with each edge $e \in E$. The problem of

maximum weight matching is to obtain a matching $\tilde{E}$ such that the sum of the weights of edges in $\tilde{E}$ is maximized:

$$\sum_{e \in \tilde{E}} w_e \rightarrow \max \tag{1}$$

Edmonds's algorithm (Edmonds 1965) is a remarkable algorithm for solving the maximum weight matching problem and is fully explained in a number of standard textbooks on graph theory (e.g., Gondran and Minoux 1984; Lawler 1976). The algorithm recursively forms vertices into smaller subgroups (called ''blossoms''), solves smaller problems, and attains the global optimum matching when the recursion terminates.

As in the example of the previous section, suppose that a data set X is given as an $n \times p$ matrix. For simplicity we assume that n is even. Choose some appropriate distance function $\mathrm{dist}(x_i, x_j)$ between two rows x_i, x_j of X. Then our goal is to form a complete matching of n rows such that the sum of distances within the pairs is minimized:

$$\sum \mathrm{dist}\,(x_i, x_j) \rightarrow \min \tag{2}$$

Here complete matching refers to the requirement that every row of X belong to some pair and hence $n/2$ pairs be formed. Choose M such that

$$M > \max_{1 \le i < j \le n} \mathrm{dist}\,(x_i, x_j) \tag{3}$$

and define the weight of the edge $e = (x_i, x_j)$ by

$$w_{ij} = M - \mathrm{dist}\,(x_i, x_j)$$

Then the minimization in (2) is reduced to the maximization in (1). Note that the optimum matching in (1) is automatically a complete matching if the weights w_e, $e \in E$, are all positive. Therefore by choosing M as in (3) we obtain a complete matching in (2) by solving (1).

Edmonds's algorithm requires an amount of time of the order of $O(n^4)$. It can be improved to $O(n^3)$ time using $O(n^2)$ amount of memory. In our application n is not small and the latter approach is not practical. As shown in Section 4 full optimization by Edmonds's algorithm is found to be too intensive for a data set of size $n > 10,000$. Hence there is a need for an approximate optimization.

Here we propose an approximation, which is found to work very well in our experiment in Section 4. Obviously more experimentation is needed to assess the overall quality of the following approximation. Let k be a small integer. We first construct a list of edges to the k nearest neighbors for each row of X. This requires $O(kn)$ amount of memory. Let G_k be a graph having the above kn edges. Note that the same edge can appear twice in the above list and therefore the number of the different edges of the graph G_k is at most kn. We apply Edmonds's algorithm to G_k and obtain an optimum matching for G_k. It may be the case that for small k, the resulting matching is not complete. In this case we increase k and perform the optimization again. Let k^* be the smallest k such that the resulting optimum matching is complete. We use this matching as an approximate solution to our problem. For finding k^* we could start with a fairly small value of k ($k = 5$ for example) and increase k if the resulting matching is not complete and decrease k if the resulting matching is complete. In practice it would be better to try some k much larger than k^*, possibly

Table 9. Ten records of the dataset

Sex	Age	Month	Major and subcause	Accident
1	56	3	E1	2
2	86	4	N4	2
2	68	2	L9	2
2	47	11	P0	2
2	80	9	B3	2
2	81	1	B9	2
1	78	3	C7	2
1	84	1	Q8	2
2	64	4	Q3	2
2	97	10	C7	2

with a randomly selected subset of rows of X, and see if the average distance of the optimum matching drastically decreases with larger k. If not, our approximate solution seems reasonable.

4. Experiment with Japanese Causes of Death Statistics

Here we apply the procedure of the previous section to a data set of considerable size and show that computations can be done in a reasonable amount of time. In the experiment it is found that the full combinatorial optimization using Edmonds's algorithm is computationally too intensive. We show that our approximation described in the previous section achieves almost the same optimization as the full optimization with a fraction of computation time. The source code of a working program by Daishin Nakamura for the computations of this section is available from the URL included in References.

4.1. The data set

The data set used is the death records for the year 1995 from the Ministry of Health and Welfare of Japan. This data is a ''census'' recording all deaths of Japanese nationals. Except for the classification of cause of death, which might be sensitive and requires a certain amount of medical knowledge, all variables are straightforward personal attributes. We prepared a file of 78,648 deaths in a certain prefecture during 1995. The variables we chose are the following: (1) sex (1 or 2), (2) age (in years), (3) month of death, (4) major cause of death, (5) subcause of death, (6) traffic accident or not (1 or 2). The major cause of death is coded by a single letter in the alphabet range A–Y and the subcause of death is coded by a single digit. Detailed description of the variables is not relevant for the present discussion. The first ten records of the data set are shown in Table 9.[3]

Among the 78,648 deaths, 17,090 deaths (21.72%) were unique with respect to these variables. In this article we only discuss the results of computations on this subset of 17,090 unique deaths. The distance function we chose is

$$\text{dist}\,(x, y) = 20I_{[x_1 \neq y_1]} + 2|x_2 - y_2| + |x_3 - y_3| + 3I_{[x_4 \neq y_4]} + I_{[x_4 = y_4]} \cdot I_{[x_5 \neq y_5]} + 10I_{[x_6 \neq y_6]} \tag{4}$$

[3] Actually the observations in Table 9 show simulated values different from the real values on the magnetic tape supplied by the Ministry of Health and Welfare. This is due to the condition of the special permit granted to us by the Management and Coordination Agency of the Japanese Government.

Table 10. Distribution of distances of optimally matched pairs

Distance	1	2	3	4	5	6	7	8	9	10
Number of pairs	4,896	1,878	1,531	142	60	15	9	4	1	3
Distance	11	12	13	14	15	16	17	18	19	20
Number of pairs	1	2	2	0	0	0	0	0	0	1

where

$$x_1 = \text{Sex},\ x_2 = \text{Age},\ x_3 = \text{Month},\ x_4 = \text{Major Cause},\ x_5 = \text{Subcause},\ x_6 = \text{Accident}$$

Here we measure a one month difference as 1. Then we count the difference in sex as 20, a one year difference in age as 2, and a difference of the major cause of death as 3. The difference of subcause is 1 provided that the major cause of death is the same, and traffic accident is ten. We choose these weights because they roughly reflect the relative importance or noticeability of these key variables.

The machine used to measure the processing time was equipped with an Intel Pentium Pro Processor with the clock speed of 200MHz and 64 MB of memory. We have first performed the one-sided matching among these 17,090 deaths. The CPU time needed was 224 seconds and the average distance within the pairs in the optimum one-sided matching was 1.49508.

The full optimization by Edmonds's algorithm took 328,163 CPU seconds (about four days) with the minimized sum of distances 14,418 or the average distance of $14,418/8,545 = 1.6873$. The distribution of the distances of the optimality matched pairs is tabulated in Table 10.

Although the exact optimization was possible, a processing time of four days is not practical. Therefore we applied the approximate optimization discussed in the previous section. Table 11 presents the results of the computation.

CPU seconds in Table 11 is the time for obtaining the maximum weight matching for kn edges. In addition it took about 340 CPU seconds to form the list of k neighbors for each of $n = 17,090$ rows of the data set.

For $k \leq 22$ there does not exist a complete matching. However, for $5 \leq k \leq 22$ all but two deaths are matched in pairs. $k^* = 23$ is the minimum k, for which the maximum weight matching becomes complete. In this matching the sum of distances is 14,423 with the average distance $14,423/8,545 = 1.6879$. This is almost the same as the fully optimized matching with the sum of distances 14,418. The distribution of distances of approximately optimized pairs with $k = k^* = 23$ is tabulated in Table 12, which is very close to Table 10.

The actual local recoding for the first 20 rows of the data set is shown in Table 13. The first set of columns, ''Original,'' shows the original rows and they are the same as in Table 9. The second set of columns, ''One-sided N.N,'' shows the result of the one-sided nearest neighbor local recoding. The third set of columns and the fourth set of columns show the results of fully optimized two-sided matching and approximately optimized two-sided matching, respectively. The last set of columns, ''Quadruples,'' show the results of tentative formation of quadruples by application of matching to matched pairs.

Table 11. Approximate optimization results for various k

k	1	2	3	4	5	6	7	8	9	10
Number of pairs	6,553	8,115	8,537	8,543	8,544	8,544	8,544	8,544	8,544	8,544
Sum of distances	9,273	13,803	15,297	14,872	14,692	14,578	14,519	14,467	14,446	14,427
CPU seconds	165	218	245	283	298	311	330	355	368	391
k	11	12	13	14	15	16	17	18	19	20
Number of pairs	8,544	8,544	8,544	8,544	8,544	8,544	8,544	8,544	8,544	8,544
Sum of distances	14,412	14,405	14,403	14,396	14,395	14,394	14,394	14,393	14,392	14,392
CPU seconds	420	438	456	469	482	498	544	539	556	576
k	21	22	23	24	25	26	27			
Number of pairs	8,544	8,544	8,545	8,545	8,545	8,545	8,545			
Sum of distances	14,392	14,392	14,423	14,423	14,423	14,423	14,423			
CPU seconds	595	614	631	650	669	688	707			

Table 12. Distribution of distances of pairs for $k = k^* = 23$

Distance	1	2	3	4	5	6	7	8	9	10
Number of pairs	4,899	1,869	1,534	148	55	13	13	2	4	3
Distance	11	12	13–20	21						
Number of pairs	3	1	0	1						

See Section 5 for discussion of forming the quadruples. In Table 13 the comma denotes ''or'' of the categories. If the main cause of death is locally recoded, then the subcause of the death becomes irrelevant and is denoted by an asterisk.

5. Discussion

In local recoding the observations are displayed as intervals or unions of categories when necessary. The presentation of this form might be unfamiliar to the users of the data set. Therefore statistical agencies may prefer swapping to local recoding for the convenience of users. On the other hand statistical agencies may want to avoid distorting the information in the data set. In the interval presentation of observations in local recoding, the observations are not distorted, whereas in swapping they *are*. In this sense local recoding is not a perturbation technique, whereas swapping is a perturbation technique.

From the viewpoint of usability, locally recoded data sets are not convenient for causal users, because the data sets cannot be directly analyzed by means of standard statistical packages. On the other hand, locally recoded data sets have the advantage of conveying the amount of information loss due to local recoding. The users can assess the influence of local recoding on their statistical analysis by doing their own simulations from the locally recoded data sets. They can randomly choose a possible value from each interval presentation of the observations and form a usual working data set for standard statistical packages. They can repeat this simulation run many times and observe the variability of the results of statistical analyses. For reporting purposes, the users can take the average of the simulation runs. We can interpret this simulation process by the users as follows: the users themselves form data sets with swapped observations without knowing the original data set. Therefore the process of swapping is shifted from the statistical agency to the users. If a single data set with swapped observations is published by the statistical agency, the users cannot numerically assess the information loss due to the swapping. In this sense, local recoding provides the users with more information than swapping. At the same time, from the viewpoint of protection, local recoding based on the two-sided matching is weaker than swapping, because it indicates which records are paired.

The choice of a distance function in our approach is based on convenience. The basic requirement for a distance function is that it should contain all relevant key variables and the weight of each key variable in the distance function should reflect the relative importance and noticeability of the variable. The distance function is just a tool for letting the computer do the matching. The resulting matching depends on the choice of the distance function in a rather unpredictable manner. From the viewpoint of protection, this dependence is an advantage to the statistical agencies in the case of swapping. By not

Table 13. Local recoding by one-sided and two-sided matchings

Original					One-sided N.N.			Two-sided Exact			Two-sided Approx.			Quadruples		
S	A	M	C	T												
1	56	3	E1	2	1	56	2-3 E 1 2	1	56	2-3 E 1 2	1	56	2-3 E 1 2	1	56	2-3 E 1, 3, 7 2
2	86	4	N4	2	2	86	4 N 3, 4 2	2	86	4 N 3, 4 2	2	86	2-4 N 4 2	2	86	2-5 N 3, 4 2
2	68	2	L9	2	2	68	2 L 1, 9 2	2	68	2-3 L 9 2	2	68	2-3 L 9 2	2	68	2-3 L 1, 4, 9 2
2	47	11	P0	2	2	47	11-12 P 0, 7 2	2	47	11-12 P 0, 7 2	2	47	11-12 P 0, 7 2	2	47	9-12 P 0, 7 2
2	80	9	B3	2	2	80	9 B 3, 4 2	2	80	9 B 3, 8 2	2	80	9 B 3, 4 2	2	80	9 B 3, 4, 7, 8 2
2	81	1	B9	2	2	81	1 B 7, 9 2	2	81	1 B 7, 9 2	2	81	1-2 B 0, 9 2	2	81	1-2 B 0, 2, 7, 9 2
1	78	3	C7	2	1	78	3 C 7, 8 2	1	78	3 C 7, 8 2	1	78	3 C 7, 8 2	1	78	2-4 C 1, 7, 8 2
1	84	1	Q8	2	1	84	1 Q 7, 8 2	1	84	1 Q 7, 8 2	1	84	1 Q 7, 8 2	1	84	1-2 Q 4, 7, 8, 9 2
2	64	4	Q3	2	2	64	3-4 Q 3, 7 2	2	64	4 Q, J * 2	2	64	4 Q, J * 2	2	64	3-4 Q, J * 2
2	97	10	C7	2	2	97	9-10 C 1, 7 2	2	97	9-10 C 1, 7 2	2	97	9-10 C 1, 7 2	2	97-98	6-10 C 1, 7 2
2	48	1	B3	2	2	48	1-2 B 3 2	2	48	1 B 2, 3 2	2	48	1 B 2, 3 2	2	48	1-2 B 1, 2, 3 2
2	11	10	P0	1	2	11	8-10 F, P * 1	2	11	8-10 F, P * 1	2	11	8-10 F, P * 1	2	11-12	8-10 D, F, P * *
2	80	8	E7	2	2	80	8 E 0, 7 2	2	80	8 E 0, 7 2	2	80	8 E 0, 7 2	2	80	8-9 E 0, 4, 7 2
1	77	10	K4	2	1	77	10-11 K 4 2	1	77	10 K 1, 4 2	1	77	10-11 K 4 2	1	76-77	10-12 K 4 2
2	47	7	D4	2	2	47	7-9 D 4 2	2	47	7-8 D 2, 4 2	2	47	7-8 D 2, 4 2	2	47	7-8 N, D, F * 2
2	80	11	K4	2	2	80	10-11 K 1, 4 2	2	80	11-12 K 1, 4 2	2	80-81	11 K 4 2	2	79-81	11-12 K 1, 4 2
1	80	1	K3	2	1	80-81	1 K 3 2	1	80-81	1 K 3 2	1	80-81	1 K 3 2	1	80-81	1-3 K 1, 3, 4 2
2	85	8	A8	2	2	85	8 K, A * 2	2	85	8 T, A * 2	2	85	8 T, A * 2	2	85	8 C, T, A, M * 2
1	55	10	C1	2	1	55	10-11 C 1 2	1	55	10-11 C 1 2	1	55	10-11 C 1 2	1	55	8-11 C 1 2
2	70	3	D4	2	2	70	3-4 D 4 2	2	70	3-4 D 4 2	2	70	3-4 D 4 2	2	70	2-4 D 4, 8, 9 2

disclosing the actual form of the distance function, the statistical agencies make it very difficult for intruders to undo the swapping.

In this article we have discussed forming pairs of individuals for disclosure control. For more security it might be more desirable to form groups of larger size. Unfortunately, it is generally known that the problem of optimally forming disjoint triples is an NP-complete problem and hence it is practically infeasible to obtain a fully optimized set of triples for large n. See the description of 3-dimensional matching problem and the exact cover by 3-sets problem on page 221 of Garey and Johnson (1979). This does not preclude the possibility that there might exist a satisfactory algorithm for approximate optimization. Even if this is the case, it may be hard to measure the performance of an approximate optimization algorithm in the absence of a full optimization algorithm.

For groups of size 2^h, $h = 2, 3, \ldots$, we might apply the optimum matching algorithm repeatedly. After forming matched pairs, we can introduce a distance measure between two pairs of rows of X and use the optimum matching algorithm again to form pairs of pairs or groups of size 4. If we repeat this process, we can form approximately optimized groups of size 2^h, $h \geq 2$. The ''quadruples'' of Table 13 show local recoding based on groups of size 4.

A more precise description of the procedure we used for forming quadruples of Table 13 is as follows. We started with the results of matching by approximate optimization with $k = k^* = 23$ as discussed in Section 4. Since there were an odd number (i.e., 8,545) of pairs, we took out one pair and worked with 8,544 pairs. We defined the distance between two pairs (x, x') and (y, y') as

$$\text{dist}_2((x, x'), (y, y')) = \text{dist}(x, y) + \text{dist}(x, y') + \text{dist}(x', y) + \text{dist}(x', y')$$

where dist( ) is given in (4). With this distance function $\text{dist}_2(\,)$ between two pairs we applied the approximate optimization. This time a complete matching of pairs was achieved with $k = k^* = 4$ neighbors. The quadruples of Table 13 show the result of local recoding based on this pairing of pairs.

6. References

Dalenius, T. and Reiss, S.P. (1982). Data-swapping: A Technique for Disclosure Control. Journal of Statistical Planning and Inference, 6, 73–85.

Defays, D. and Anwar, M.N. (1998). Masking Microdata Using Micro-aggregation. Journal of Official Statistics, 14, 449–461.

De Waal, A.G. and Willenborg, L.C.R.J. (1996). SDC Measures and Information Loss for Microdata Sets. Report. Dept. of Statistical Methods, Statistics Netherlands, Voorburg.

De Waal, A.G. and Willenborg, L. (1999). Information Loss Due to Global Recoding and Local Suppression. Netherlands Official Statistics, 14, 17–20.

Duncan, G. and Pearson, R.W. (1991). Enhancing Access to Microdata While Protecting Confidentiality: Prospects for the Future. Statistical Science, 6, 219–239.

Edmonds, J. (1965). Maximum Matching and a Polyhedron with 0–1 Vertices. Journal of Research of the National Bureau of Standards, Section B, Mathematical Sciences, 69, 125–130.

Fienberg, S.E., Makov, U.E., and Steele, R.J. (1998). Disclosure Limitation Using

Perturbation and Related Methods for Categorical Data. Journal of Official Statistics, 14, 485–502.

Fuller, W.A. (1993). Masking Procedures for Microdata Disclosure Limitation. Journal of Official Statistics, 9, 383–406.

Garey, M.R. and Johnson, D.S. (1979). Computers and Intractability. A Guide to the Theory of NP-Completeness. San Francisco: W.H. Freeman and Company.

Gondran, M. and Minoux, M. (1984). Graphs and Algorithms. (Translated by Steven Vajda). New York: Wiley.

Gouweleeuw, J.M., Kooiman, P., Willenborg, L.C.R.J., and de Wolf, P.P. (1998). Post Randomisation for Statistical Disclosure Control: Theory and Implementation. Journal of Official Statistics, 14, 463–478.

Lawler, E.L. (1976). Combinatorial Optimization: Networks and Matroids. New York: Halt, Rinehart and Winston.

Mateo-Sanz, J.M. and Domingo-Ferrer, J. (1998). A Method of Data-oriented Multivariate Microaggregation. In Conference Programme of Statistical Data Protection '98, Lisbon, March.

Müller, W., Blien, U., and Wirth, H. (1995). Identification Risks of Microdata. Evidence from Experimental Studies. Sociological Methods and Research, 24, 131–157.

Nakamura, D. (1998). A Program for Edmonds' Maximum Weight Matching Algorithm and Two-sided Nearest Neighbor Local Recoding. Available via Internet from http://www.e.u-tokyo.ac.jp/~takemura/localrec.html.

Schlörer, J. (1981). Security of Statistical Databases: Multidimensional Transformation. ACM Transactions on Database Systems, 6, 95–112.

Willenborg, L. and de Waal, T. (1996). Statistical Disclosure Control in Practice. Lecture Notes in Statistics 111. New York: Springer.

Received April 1999
Revised April 2001