# Local recoding by maximum weight matching for disclosure control of microdata sets

Akimichi Takemura
Faculty of Economics, University of Tokyo

February, 1999

**Abstract**

We propose "local recoding" as a new technique for controlling disclosure risk of microdata sets. Compared to the technique of global recoding, where the observed values are grouped into broader intervals or categories throughout the data set, in local recoding different grouping is performed for each observation when necessary. As a means of performing local recoding we propose to form pairs of close individuals and recode observed values within each pair. For optimally forming pairs we can employ Edmonds' algorithm (Edmonds (1965)) of maximum weight matching. We illustrate the technique by applying it to the Japanese vital statistics data.[1]

*Key words*: Edmonds' algorithm, local suppression, nearest neighbor, NP-complete, perturbation, swapping.

# 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 paper we propose "local recoding", where each observed value is recoded into broader intervals or categories when necessary.

As a means of performing local recoding we propose matching or pairing of close individuals of a microdata set. When two individuals are grouped into a pair, we can locally recode observations of these two individuals 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 can be performed in reasonable amount of computer time.

Although many techniques are proposed for disclosure control of microdata sets, the idea of local recoding appears to be new. The local suppression, where individual observations are marked as missing, is extensively discussed in Section 5.4 of Willenborg and

[1] This data set was used under permit No. 40, 1997, of Management and Coordination Agency, Government of Japan, for the purpose of disclosure control experiments.

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

Addition of noise to original observations is discussed by many authors including Fuller (1993) and Duncan and Pearson (1991). One conceptual difficulty of addition of noise is that it is not clear how one can add noise to purely categorical variable. The Post RAndomization Method (PRAM) introduced by Kooiman et al. (1997) is a probabilistic perturbation technique for categorical variables. One advantage of the present procedure is that local recoding can be applied to a data set with both continuous and categorical variables.

Another important technique of disclosure control is swapping discussed in Schlörer (1981) and Dalenius and Reiss (1982). The pairing technique proposed in this paper can also be used for the purpose of swapping rather than local recoding. This point is discussed in Section 5.

The idea of pairing of this paper is close to the idea of microaggregation in Mateo-Sanz and Domingo-Ferrer (1998). They use clustering algorithm whereas we use matching algorithm to form groups. One disadvantage of clustering might be the lack of well defined notion of optimality among various clustering algorithms.

The organization of this paper is as follows. In Section 2 we explain the idea of local recoding and matching by a simple numerical example. In Section 3 we discuss full optimization and approximate optimization procedures based on Edmonds' 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. Some discussions are given in Section 5.

# 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 10 household records of 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, 4. Occupation in 3 categories (A,B or C). We consider these 4 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 the age and the 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

Table 1: Hypothetical population of size 10

| 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 |

the number of categories present for each variable. The numbers are 3 for Age, 5 for Size, 5 for Income, and 3 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 10 households thrown into 225 boxes and it is likely that these households fall into different boxes. The usual "birthday problem" calculation yields an approximate probability of $e^{-(1+\cdots+9)/225} = e^{-0.2} = 0.82$ .

We proceed to more drastic global recoding: grouping the household size into 2 categories ( $\geq 4$ or $\leq 3$ ) and income into 2 categories ( $\geq 500$ 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 household No.4 and No.7 coincide and they are no longer population uniques. However other 8 households remain to be population uniques. At this point it seems to be very difficult to perform further global recoding without losing substantial amount of information in the data set. This suggests that relying on global

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 |

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* the 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 |

Then these two households are no longer population uniques.

Let us match remaining 6 households into the following 3 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 (variable which have 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 computer. The basic idea of the matching was to find close households. Therefore we introduce some distance function between households. Distance function can be chosen based on 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.

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 |

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 5 years in age as distance "1", since 5 years difference might be noticeable from the appearance. Then 10 years difference in age is measured as distance 2. Concerning the size of the household, we measure the difference of 1 as just 1, since neighbors may know the exact household size. We measure the difference of 100 in income as 1. Finally we measure the difference in occupation as 2. As the total distance between two households, we add these individual distances for the 4 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, Occupation) of two households. Then the distance between $x$ and $y$ may be defined as

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

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 between the 10 households. Using Table 5 we can list closest households ("nearest neighbors") from each household as shown in Table 6.

From Table 6 the average distance to 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 are more than one nearest household, we arbitrarily choose one of the nearest households. We call this type of local recoding "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"

Table 5: Distances between 10 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 |

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 |

(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 sum of the distances within the pairs. We call local recoding by this type of two-sided matching as "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' algorithm (Edmonds (1965)) is a well known algorithm for solving the maximum weight matching problem.

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 |

In our hypothetical example $n$ is only 10 and the total number of forming 5 pairs out of 10 households is

$$9 \cdot 7 \cdot 5 \cdot 3 = 945.$$

Therefore we can check all 945 pairings and compute the sum of distances. Then the optimum matching is obtained as

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

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

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 | 50 | 2-6 | 500 | A |
| 9 | 50 | 2-6 | 500 | A |
| 10 | 30 | 3 | 300-500 | A |

From Table 7 and Table 8 we see that two-sided nearest neighbor local recoding leads to stronger protection accompanied with 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 requiring combinatorial optimization.

# 3 Full combinatorial optimization by Edmonds' algorithm and its approximation

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

Let $G = (V, E)$ be a graph, where $V$ is the set of vertices and $E$ is the set of edges. Matching is a subset $\tilde{E}$ of $E$ such that each vertex $v \in V$ is contained in at most one edge $e \in \tilde{E}$. 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' algorithm (Edmonds (1965)) is a remarkable algorithm for solving maximum weight matching problem and is fully explained in a number of standard textbooks on graph theory (e.g. Gondran and Minoux (1984) or Lawler (1976)).

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 an appropriate distance function $\mathrm{dist}(x, y)$ between two rows 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:

$$\sum_{x,y} \mathrm{dist}(x, y) \ \rightarrow \ \min. \tag{2}$$

Here complete matching refers to the requirement that every row of $X$ belong to some pair and hence $n/2$ pairs be formed. Apart from the trivial difference in maximization of (1) and minimization in (2), there is no requirement on the number of edges in (1) whereas in (2) we require complete matching.

However this difference is superficial and the minimization problem in (2) can be easily reduced to the maximum weight matching in (1). Consider a complete graph

$$G = (\text{rows of } X, \ \text{pairs of rows of } X). \tag{3}$$

Let $M$ be a sufficiently large positive number and to each pair of rows $e = (x_i, x_j)$ assign the weight

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

Then minimization in (2) is reduced to maximization in (1). In the case where the distance function is nonnegative integer valued, it is shown in Appendix A that $M$ can be taken as

$$M = \frac{n}{2}U + 1, \qquad U = \max_{1 \le i < j \le n} \mathrm{dist}(x_i, x_j). \tag{4}$$

Edmonds' algorithm requires amount of time of order $O(n^4)$ . 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' algorithm is found to be too intensive for a data set of size $n > 10000$ . Hence there is a need for approximate optimization.

Here we propose an approximation, which is found to work very well in our experiment in Section 4. 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 subgraph of $G$ of (3) where the edges are restricted to 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' 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 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 vital statistics data

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

## 4.1 The data set

The data set used is the death records data for the year 1995 from the Ministry of Health and Welfare of Japan. This data set is a "census" recording all deaths of Japanese nationals. Except for the classification of cause of death, which might be sensitive and requires certain amount of medical knowledge, all other variables are straightforward personal attributes. We prepared a file of 78648 deaths in a certain prefecture during 1995. The variables we chose are the following 6 variables: 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 alphabet in the range A–Y and the subcause of the death is a single digit. Detailed description of the variables is not relevant for the present discussion. The first 10 records of the data set are shown in

Table 9: 10 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 |

Table 9.[2]

Among the 78648 deaths, 17090 deaths (21.72%) were unique with respect to these variables. In this paper we only discuss results of computations on this subset of 17090 unique deaths. The distance function we chose is

$$\begin{aligned}\mathrm{dist}(x,y) = 20I_{[x_1\neq y_1]} + 2|x_2 - y_2| + |x_3 - y_3| + 3I_{[x_4\neq y_4]} \\ + I_{[x_4=y_4]} \cdot I_{[x_5\neq y_5]} + 10I_{[x_6\neq y_6]},\end{aligned} \tag{5}$$

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 1 month difference as 1. Then we count the difference in sex as 20, 1 year difference of age as 2, 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 10.

The machine used to measure the processing time was equipped with Intel Pentium Pro Processor and 64 MB of memory. We have first performed the one-sided matching among these 17090 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' algorithm took 328163 CPU seconds (about 4 days) with the minimized sum of distances 14418 or the average distance of $1.6873 = 14418/8545$ . The distribution of the distances of the optimally matched pairs is tabulated in Table 10.

Although the exact optimization was possible, processing time of 4 days is not practical. Therefore we applied the approximate optimization discussed in the previous section. Table 11 presents the results of the computation.

[2] 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 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 | 4896 | 1878 | 1531 | 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 |

Table 11: Approximate optimization results for various $k$

| $k$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| number of pairs | 6553 | 8115 | 8537 | 8543 | 8544 | 8544 | 8544 | 8544 | 8544 | 8544 |
| sum of distances | 9273 | 13803 | 15297 | 14872 | 14692 | 14578 | 14519 | 14467 | 14446 | 14427 |
| 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 | 8544 | 8544 | 8544 | 8544 | 8544 | 8544 | 8544 | 8544 | 8544 | 8544 |
| sum of distances | 14412 | 14405 | 14403 | 14396 | 14395 | 14394 | 14394 | 14393 | 14392 | 14392 |
| CPU seconds | 420 | 438 | 456 | 469 | 482 | 498 | 544 | 539 | 556 | 576 |
| $k$ | 21 | 22 | 23 | 24 | 25 | 26 | 27 | | | |
| number of pairs | 8544 | 8544 | 8545 | 8545 | 8545 | 8545 | 8545 | | | |
| sum of distances | 14392 | 14392 | 14423 | 14423 | 14423 | 14423 | 14423 | | | |
| CPU seconds | 595 | 614 | 631 | 650 | 669 | 688 | 707 | | | |

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 $n = 17090$ 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 2 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 14423 with the average distance $14423/8545 = 1.6879$. This is almost the same as the fully optimized matching with the sum of distances 14418. 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 the same as Table 9. The

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 | 4899 | 1869 | 1534 | 148 | 55 | 13 | 13 | 2 | 4 | 3 |
| distance | 11 | 12 | 13-20 | 21 | | | | | | |
| number of pairs | 3 | 1 | 0 | 1 | | | | | | |

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 |

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 results of fully optimized two-sided matching and approximately optimized two-sided matching, respectively. The last set of columns "Quadruples" show the result of tentative formation of quadruples by application of matching to matched pairs. See Section 5 for discussion of forming quadruples. In Table 13 the comma "," denotes "or" of the categories. If the main cause of death is locally recoded, then the subcause of death becomes irrelevant and denoted by the asterisk "*".

# 5 Some discussion

In local recoding the observations are displayed as intervals or union of categories when necessary. The presentation of this form might be unfamiliar for the users of the data set. Another possibility is to use matching for the purpose of swapping of observations. If we just present one endpoint of the interval in local recoding, possibly always differently from the real values, we obtain swapping of observations (Schlörer (1981), Dalenius and Reiss (1982)). Once we obtain the two-sided optimum pairs, the swapping can be done within these pairs. Since the pairs are formed optimally, the swapping is performed only between close records.

On the other hand statistical agencies might prefer interval presentation of observations by local recoding rather than swapping, because in interval presentation the information is not distorted as in swapping. In this sense local recoding is not a perturbation technique, whereas the swapping is certainly a perturbation technique.

In this paper 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 forming disjoint triples is an NP-complete problem and hence it is practically infeasible to obtain 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 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 some 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.

More precise description of the procedure we used for forming quadruples of Table 13 is as follows. We started with the result of matching by approximate optimization with $k = k^* = 23$ as discussed in Section 4. Since there were odd number (i.e. 8545) of pairs, we took out one pair and worked with 8544 pairs. We defined the distance between two pairs $(x, x')$ and $(y, y')$ as

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

where dist() is given in (5). With this distance function $\mathrm{dist}_2()$ 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.

# A Proof of (4)

Let $U = \max_{1 \leq i < j \leq n} \mathrm{dist}(x_i, x_j)$ and let

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

where $M \geq U$. Then

$$M - U \leq w_{ij} \leq M, \qquad 1 \leq i < j \leq n.$$

A lower bound of the sum of distances for all complete matching is given by $(n/2)(M-U)$ and an upper bound for all non-complete matching is given by $((n/2) - 1)M$. Hence if $((n/2) - 1)M < (n/2)(M - U)$ or equivalently if

$$M > \frac{n}{2}U$$

then maximum weight matching results in a complete matching. Hence we can take

$$M = \frac{n}{2}U + 1.$$

Acknowledgment

This paper 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' algorithm for the two-sided matching. The material of Section 3 is largely due to him. He has also pointed out NP-completeness of optimally forming triples.

# References

[1] Dalenius, T. and Reiss, S.P. (1982). Data-swapping: A technique for disclosure control. *Journal of Statistical Planning and Inference*, **6**, 73–85.

[2] Duncan,G. and Pearson, R.W. (1991). Enhancing access to microdata while protecting confidentiality: Prospects for the future. *Statistical Science*, **6**, 219–239.

[3] Edmonds, J. (1965). Maximum matching and a polyhedron with 0–1 vertices. *J. Res. Natl. Bur. Stand. B*, **69**, 125–130.

[4] Fuller, W.A. (1993). Masking procedures for microdata disclosure limitation. *Journal of Official Statistics*, **9**, 383–406.

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

[6] Gondran, M. and Minoux, M. (1984). *Graphs and Algorithms.* (Translated by Steven Vajda), Wiley, New York.

[7] Kooiman, P., Willenborg, L.C.R.J., and de Wolf, P.P. (1997). PRAM: a method for disclosure limitation of microdata. Research paper no. 9705, Statistics Netherlands.

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

[9] Mateo-Sanz, J.M. and Domingo-Ferer, J. (1998). A method of data-oriented multivariate microaggregation. in *Conference Programme of Statistical Data Protection '98*, Lisbon, March 1998.

[10] Müller, W., Blien, U., and Wirth. H. (1995). Identification risks of microdata. Evidence from experimental studies. *Sociological Methods & Research*, **24**, 131–157.

[11] 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` .

[12] Schlörer, J. (1981). Security of statistical databases: multidimensional transformation. *ACM Transactions on Database Systems*, **6**, 95–112.

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