Source Data Perturbation in Statistical Disclosure Control

Menno Cuppen

august 2000

dr. Leon Willenborg (CBS)

dr. Wim Pijls (EUR)

Abstract

When tables of quantitative data are generated from a datafile, the release of those tables should not reveal information concerning individual respondents. This disclosure of individual respondents in the microdata file can be prevented by applying disclosure control methods at the table level, but this may create inconsistencies across tables. Alternatively, disclosure control methods can be executed at the microdata level, but these methods change the data permanently and do not account for specific table properties. These problems can be circumvented by assigning a weight factor to each respondent in the microdata file. Upon tabulation, each contribution of a respondent is weighted multiplicatively by the respondent's weight factor. This approach is called Source Data Perturbation (SDP) because the data is perturbed at the microdata level, not at the table level. It should be noted, however, that the original microdata is not changed. Moreover, the weight factors can be chosen such that the tables generated from the microdata are safe, and the information loss is minimized.

Keywords: Confidentiality, Disclosure, Source Data Perturbation, Noise addition, Table protection

Acknowledgement: I would like to thank dr. Leon Willenborg of Statistics Netherlands and dr. Wim Pijls of the Erasmus University Rotterdam for their assistence. Furthermore, I would like to thank Statistics Netherlands for the opportunity to perform this research, and the researchers of the department of Statistical Methods for their support.

Contents

Chapter 1

Introduction

When data are released in the form of tables of magnitude data, each cell in the released table(s) represents the aggregate on some response variable (e.g. profits) over all respondents in that cell. Some cell values may be dominated by one or a few respondents. This may lead to disclosure of some respondent's contribution, as some of the dominating contributors can combine forces to disclose information about other respondents. To prevent any entity's value from being identified, disclosure limitation can be executed at the table level (e.g. by cell suppression), or at the level of the underlying microdata. A recently proposed method is to provide protection by adding multiplicative noise to the respondent microdata prior to tabulation (see Evans, Zayatz, Slanta [7]). To add noise, each contributing establishment in the microdata is assigned a multiplier, and this multiplicative weight factor should provide protection upon tabulation. Because the noise is added at the microdata level, and because several other perturbation methods exist, the expression *Source Data Perturbation* (SDP) is used when referring to the use of multiplicative noise at the microdata level.

The amount of noise added should depend on the desired safety level of the tables that are to be released. Therefore, a set of tables that are demanded to be safe is defined. This set of tables is used as a calibration set, i.e. these tables are used to find the amount of perturbation that is applied to the respondent microdata. After the perturbation process, the tables in this set are supposed to be safe. However, next to tables of the calibration set also other tables may be generated. Sensitive cells, which are cells that

contain contributors that are at risk to be disclosed, should receive a lot of perturbation, while nonsensitive cells should receive as little perturbation as possible. Definitions for "disclosure" as well as for "sensitive cells" will be given in Section 2.1. The method proposed in [7] achieves these desired properties, but in this method the exact amount of noise added is chosen arbitrary while in general, this amount should depend on the sensitivity of the cells in the tables to be released. This means that the amount of noise added should not be too small, but only just enough to assure protection of individual respondents in sensitive cells. Therefore the information loss suffered by the perturbation should be minimized. The information loss can be evaluated by measuring the variance added to the tables due to applying the noise.

Therefore, first the method of Zayatz, Evans, and Slanta will be implemented and evaluated. This method, that will be referred to as the ZES method, only looks at the individual respondents in the microdata, i.e., it does not look at the table level. Generally, SDP methods should first look at the table level to see how much perturbation is necessary. Then, given the results of this first step, the second step should be the translation of these results into multiplicative weight factors that are to be assigned to the individual respondents.

In the first step, the desired perturbation levels that are necessary to make the table safe can be found using several approaches. First, a measure for the safety of perturbed tables is developed. Measures for the safety of not-perturbed tables do not directly apply to perturbed tables, because they do not account for the protection offered by the perturbation. This measure can be used to deduce the amount of noise needed in the cells of a table. Also, an approach based on the method of Iterative Proportional Fitting is introduced. In this method, important cell values are fixed at their true values. Because this method controls the marginal cells, it will be referred to as the MARG method.

In the second step, the new sensitivity measure for perturbed tables can be used to find the amount of noise that the ZES method should add to the cells. The original ZES method skips the first stage, but the extension to the ZES method does not. Second, the desired perturbation in a cell could

be proportionally spread over all contributors to that cell. Third, we also consider rewriting the problem as an optimization problem, in which the multiplicative weight factors are the variables for which the problem is to be optimized. The optimization problem involves minimizing an objective function that measures the difference between the *desired* cell totals and the *weighted* cell totals. The desired cell totals were found in stage one. The weighted cell totals are dependent of the perturbation factors chosen and the optimal perturbation factors are found when the weighted cell totals match the desired cell totals. Also, the optimization problem can be formulated to minimize information loss under the constraint of safety.

The SDP problem can be summarized as follows: For generating safe and consistent tables from microdata, multiplicative weight factors can be assigned to the respondents in the microdata, such as in [7]. This method can be generalized by choosing these perturbation factors dependent of some safety measure. Also other methods can be used to find desired perturbation factors at the table level. These then have to be translated into perturbation factors for the multiplicative noise approach at the respondent level. These methods will be proposed and evaluated.

In this report, in Chapter 2 first some background on statistical agencies, Statistical Disclosure Control (SDC), terminology, and currently used disclosure control methods is given. Then some cell sensitivity measures are given, to provide some intuition on the disclosure prevention problem. After that, in Chapter 3 the basic principles of SDP methods will be considered. Also the ZES method will be discussed. New SDP methods to perturb tables are proposed in Chapter 4. Then, in Chapter 5, some methods to translate these perturbation levels into multiplicative weight factors at the respondent level are discussed. To know where protection is needed, some disclosure scenarios are discussed in Chapter 6. In Chapter 7, the SDP methods will be evaluated and compared using real-world data. The properties of the data are discussed and also some attention is paid to the implementation of the testing program SoDaP.

Chapter 2

Statistical Disclosure Control

In this chapter some background on Statistical Disclosure Control (SDC) and the role of statistical offices is given. Some conventional and some newly developed disclosure control methods are discussed. Also attention is paid to some safety measures, especially to the (n,k)-dominance rule. First however, some terminology is introduced to familiarize the reader with the vocabulary used.

2.1 Terminology

Respondents are the entities under analysis that have submitted their information to the statistical office. These include companies, institutions, individual people, etc. For each respondent, scores on a number of *attributes* or *variables* are given. In this report, the term *attribute* and the term *variable* are used as synonyms. Attributes that can be measured on a metrical scale, such as income or profits, are referred to as *quantitative* variables. Attributes that are categorical, such as sex, are referred to as *qualitative* attributes. The datafile which contains the records describing respondents by their scores on the attributes is referred to as the *microdata* file. From the microdata file, *tables* are generated. Tables are spanned by several attributes, such as industrial classification, measure of size, geographical location, etc. Tables consist of *cells*, which contain the value on some response variable for some specific combination of values of the spanning variables. For instance, a cell may give the profits of chicken farms in Belgium, in a table spanned by attributes representing industrial classifica-

tion and geographical location. *Marginal cells* are cells that represent row totals and column totals in the table. In the previous example, a marginal cell could represent the aggregate profits of all chicken farms in Holland, Belgium, Germany, etc. A table is *additive* if the interior cells sum up to the marginal cell values. Some disclosure control methods may cause tables to be no longer additive, as can be seen in Section 2.4.1.

To define statistical disclosure, Eurostat [9] uses the following definition:

Definition 1 *Statistical disclosure occurs, if the dissemination of a statistic enables the external user of the data to obtain a better estimate for a confidential piece of information than would be possible without it.*

Accordingly, the publication of data by the statistical agency should not result in giving more information about individual respondents than was already common knowledge.

A *disclosure scenario* describes the strategy that possible intruders may follow. Intruders are malignant data users that try to disclose information. To know how disclosure control methods should be applied, it is necessary to be aware of the possibilities of the intruders. Only then effective countermeasures can be taken.

A very important definition is that of a *sensitive* cell:

Definition 2 *In a sensitive cell, the contribution of an individual respondent contributing to that cell can be disclosed, i.e. it can be approximated to within unacceptable narrow ranges.*

So in a sensitive cell, too much information concerning an individual respondent can be deduced, in any case more than was already common knowledge. Sensitive cells are identified by sensitivity measures, and how these measures work is explained in Section 2.6.

Data can be released in three different forms: as public use microdata, tables of frequency count data, and tables of magnitude data. Microdata contains records at the respondent level (naturally names and other direct identifiers are not included). Microdata safety requirements depend on the

recipients of the data. In tables of frequency count data, each cell contains the number of units of interest over some qualitative variables. In this case, a cell total may represent the *number* of chicken farms in Belgium, in the previous example. These counts of qualitative attributes are considered to be quantitative data. As mentioned before, in tables showing magnitude data, the various cells contain aggregate quantities concerning respondents such as business establishments, farms or institutions. In this report, mainly tables of magnitude data are discussed. However, the effect of the perturbation can be extended to tables of frequency count data, see Section 3.1. Multiplicative noise is not applicable to qualitative variables.

2.2 Statistics Netherlands - Centraal Bureau voor de Statistiek

The task of a national statistics office is to produce and publish statistical information concerning national society. Statistical information published by Statistics Netherlands (or, for that matter, by any other statistical office) has to meet certain requirements to prevent individual information of the respondents from being disclosed. These requirements are imposed by law, and by public opinion. When a statistical agency looses its 'safe' reputation due to providing insufficient protection to individual respondents, those respondents may no longer be willing to provide their exact figures on some key variable, but rather would give some figure roughly equal to the original figure to protect their valuable information, if they would reply at all. For instance, most companies are reluctant to give their true R&D budget, as it may influence their competitive advantage. However, companies may be legally obliged to respond.

The law defines which information is allowed to be released, and also to whom that information is allowed to be released. In general, the receiving party has to meet some statistical confidentiality requirements. For instance, microdata is only permitted to be released to institutions that perform statistical or scientific research. Whether applying institutions satisfy these criteria is determined in the Netherlands by the Central Committee for Statistics. When releasing any information, the statistical office has to

ensure that measures have been taken to prevent any individual respondent from being recognized. Because the law cannot outline all cases that may be encountered, the Director-General of Statistics Netherlands, advised by the department of Statistical Methods at Statistics Netherlands, also has the authority to judge which data is allowed to be released.

The release of business microdata is subjected to very strict requirements, not only because of the legal response requirement, but also because of the fact that companies are often easily identified. This is a consequence of the properties of the population: the population is relatively small, the population is likely to be skewed by one or a few entities, the composition of the population is common knowledge, identifying attributes can take a relatively large range of possible values, and many companies are highly visible in the public eye.

The department of Statistical Methods at Statistics Netherlands performs research into Statistical Disclosure Control (among other fields of research) and has developed software to evaluate disclosure risks and to implement data protection measures. This software, named ARGUS (see [10] and [11]) serves as a tool to support the various rules (see Section 2.6 for these rules) provided by the Statistical Disclosure Control practices. For more details, see [12].

2.3 Notation

To avoid ambiguities about the mathematical notation used in this report, some definitions concerning these matters are given in this section.

The notation used throughout this report is as follows:

X_i = contribution of respondent i

X_{ij} = contribution of respondent j to cell i

Y_i = estimate of the contribution of respondent i

$|C|$ = number of cells in the set of tables

$|R|$ = number of respondents in the microdata

$T = T_C = \sum_{i=1}^{\infty} X_i$ = the cell total of cell C (the subscript C is

omitted when possible). The ∞-sign is used for notational convenience. For $|C| \leq i < \infty$, $X_i = 0$.

T_N = the perturbed cell total, which can either be perturbed up or down. The former is denoted by T_N^+, the latter by T_N^-

$e = T_N - T$ = added noise to a cell

$T_{a:b} = \sum_{i=a}^{b} X_i$ = the sum of contributions a to b (inclusive)

$S(C)$ = the sensitivity of cell C. If $S(C) > 0$, the cell is considered to be sensitive

$m_i = 1 \pm r_i$ = multiplier (perturbation factor) of respondent i

μ is the mean of the (to be) assigned perturbation factors

p measures the accuracy with which a contribution can be estimated. The measurement is in terms of distance in percentages to the real value, *i.e.* $p = \frac{T-D}{X_1} - 1$ if coalition D tries to estimate the contribution 1. Small p implies a good approximation.

q = the prior knowledge of the coalition. q is defined in the same terms as p

n = the maximum size of the coalition of intruders *plus one*, *i.e.* the coalition is formed by $n-1$ respondents (parameter of the (n,k)-dominance rule)

k = a restriction on the relative mass of the n largest contributions to a cell (parameter of the (n,k)-dominance rule)

w_i = the sample weight of respondent i. In sample surveys, each respondent's data is weighted inversely proportional to the respondent's probability of being included in the sample.

$D = \sum_{i=2}^{n} X_i$ = joint contribution of the coalition of intruders

$R = \sum_{i=n+1}^{\infty} X_i$ = joint contribution of the remaining contributors

2.4 Currently used disclosure control methods

Because the publication of sensitive cells is prohibited by law, and because respondents themselves do not appreciate disclosure, several solutions are

available to protect sensitive cells. These include at the table level cell suppression, table redesign and rounding. At the microdata level these include global recoding, local suppression, top and bottom coding and microaggregation.

2.4.1 Table level

- Cell suppression: sensitive, unsafe cells are replaced by missing values ('**x**' for instance). These are primary suppressions. However, when marginals (which are row and column totals) are also published, the suppressed values can be approximated using these marginals, so extra cells have to be suppressed to prevent the primary suppressed cells from being approximated too close. These are called secondary suppressions, and they prevent safe cells from being published. The secondary suppressions are chosen in such a way that the information loss is minimized, and also such that the interval of possible values for each sensitive cell value is sufficiently large. To this end a safety interval is defined. This is necessary, because else sensitive cell totals may be estimated very accurately, in spite of secondary suppressions. Consider the left-hand side of Table 2.1. Suppose the upper right and the lower left (boldfaced) cells are sensitive. These are primary suppressed. Suppose a suppression pattern such as in the right-hand side of Table 2.1 is applied, where x_p stands for a primary suppression and x_s stands for a secondary suppression.

0	1	**1**	2
1000	0	1	1001
1000	1000	0	2000
2000	1001	2	

0	x_s	$\mathbf{x}_p$	2
x_s	0	x_s	1001
$\mathbf{x}_p$	x_s	0	2000
2000	1001	2	

Table 2.1: Primary and secondary suppressions

Now, by using the row totals and column totals, ranges for the sensitive cells can be deduced. For all cells, the deducible ranges are given in Table 2.2

The sensitive upper right cell can be deduced to within $\frac{2}{1} - 1 = 100\%$

0	[0,2]	**[0,2]**	2
[999,1001]	0	[0,2]	1001
[999,1001]	[999,1001]	0	2000
2000	1001	2	

Table 2.2: Ranges for all cells

of its value, which is not a very good approximation. However, the sensitive lower left cell can be deduced to within $\frac{1001}{1000} - 1 = 0.1\%$ of its real value, which is very close to the real value. This example illustrates that safety ranges need to be used when using cell suppression to prevent this kind of situations. Besides the suppression of nonsensitive cells, another disadvantage is that finding the optimal set of secondary suppressions is not trivial, and keeping track of suppressed cells across various tables generated from the same microdata to keep those tables consistent which each other may be very difficult.

- Table redesign: if there are too many sensitive cells, categories of the spanning variables can be combined to reduce the level of detail in the table. For example, two distinct rows "chicken farms" and "turkey farms" could be combined into one row. This results in higher levels of aggregation in the cells, thus protecting individual contributors better. Unfortunately, the resulting table may be no longer of any value to their users, as a result of the information loss incurred.

- Rounding: cell values are rounded to a given base value (e.g. multiples of 10). This is a form of adding noise that may cause the table to be no longer additive, since the rounded values no longer add up to the rounded marginals. This may induce disclosure.

2.4.2 Microdata level

Some disclosure control methods employed at the microdata level are:

- Global recoding: several categories of a variable are combined into one single category. This is done for the entire dataset, not only the unsafe parts. This also reduces the amount of detail in the data. For

instance, the categories 'cats' and 'dogs' could be combined into the category 'pets'.

- Local suppression: one or more values in an unsafe combination of values are suppressed (replaced by a missing value). E.g. in the combination "occupation" = "mayor" and "residence"= "Amsterdam", "occupation" = "mayor" is replaced by "occupation" = "missing", because since there is only one mayor living in Amsterdam, this combination uniquely identifies him (Prior knowledge is that mayors are forced to live in the city they run, and that each city only has one mayor. Also instead of "occupation", "residence" could be suppressed). This can be done for any record in the data, thus making it possible to minimize the number of local suppressions.

- Top and bottom recoding: in stead of global recoding, the top (or bottom) categories of a variable form a new category. For instance, a record showing an income of 1.000.000 is replaced by the mean for the upper tail of the income distribution, thus aggregating over the top income categories.

- Microaggregation: values for a variable are sorted and grouped, and each score in a group is replaced by the group average. Therefore totals and averages are preserved by this kind of noise addition.

- Data swapping: for some groups of records that match on certain demographic characteristics quantitative data is exchanged. The demographic match of characteristics is essential because it doesn't make sense to replace rural population data with urban population data and vice versa.

- PRAM: The Post Randomization Method changes for each record the score on a number of variables. This is done according to some probability mechanism. Because the transition probabilities are known, the original data can unbiasedly be estimated from the observed data moments in the perturbed data. For instance, "gender" = "male" is with some probability a changed to "gender" = "female" for some record in the microdata. See [13] for details.

Naturally combining several methods is also possible.

In general, there are two categories of approaches: data limitation (by limiting the amount of data that is released, or by restricting the users given access to the data), and data masking (by changing the data). The former limits the amount of delivered information by giving less detailed data, or by limiting the access to the data, while the latter limits the amount of released information by giving less accurate data. Combining both approaches offers the opportunity to make a trade off between accuracy and detail.

For more details concerning these and other methods, see for instance Willenborg and De Waal [18], the "Manual on Disclosure Control Methods" by Eurostat [9], or for an early overview "Report on Statistical Disclosure and Disclosure-Avoidance Techniques" [16].

2.5 Research into disclosure control

In the last decade, very powerful computers became widely available, and statistical agencies have been shifting from releasing data printed in tables towards releasing data electronically in tables or public use microdata files. As the demand for more (quantitatively) and more detailed (qualitatively) data increases and disclosure risks increase as technology advances, also the effectiveness of Statistical Disclosure Control should increase. One way to maintain control over which information is released, is to let users do their research on-site at the premises of statistical agencies, at stand-alone computers or isolated networks (with no outside network connections, no disk drives etc.). Also users can send their research requests or prespecified programs to the statistical agency, which looks into it and, if approved, returns the desired data. Another development is that users can ask for tabulations via some interface tool (e.g., Statline developed by Statistics Netherlands), which then composes the table directly from the microdata. As all returned tables have to satisfy safety measures, while it is not exactly clear which tables are going to be asked for in the future, it is important to find some disclosure control measure that provides means to generate safe and consistent tables, without changing the underlying microdata permanently.

To protect data before release, current research is shifting from data-hiding to data-masking. The objective is to prevent disclosure, while not

destroying the means and covariance structures of arbitrary subdomains of the data. The general idea is that by preserving the statistical properties of the data, data users can draw reliable conclusions from the released (but edited) data. Naturally, they are not supposed to be able to draw reliable conclusions concerning individual respondents. For some of the new data-masking methods, see [19] and [13].

2.6 Sensitivity measures

Sensitive cells are cells that are at risk of disclosing some information about respondents. To measure the degree of risk, to identify sensitive cells, and to support decisions about which disclosure control measures are to be used, sensitivity criteria were developed. Upper sensitivity measures focus on the degree to which a narrow upper estimate of the contribution of an individual respondent can be obtained by using the published cell total. In general, when an upper sensitivity measure $S(C)$ is used, the cell C is considered sensitive if $S(C) > 0$. Lower sensitivity measures, which focus on lower estimates of individual contributors, can also be used, and are deduced analogously to upper sensitivity measures.

The most frequently used upper sensitivity rule is the 'n-respondent, k-percent'-dominance rule, which demands that for a cell to be nonsensitive, the n largest contributors do not contribute more than k percent of the cell total. This can be stated as follows:

$$S_d(C) = \sum_{i=1}^{n} X_i - k \sum_{i=1}^{\infty} X_i = T_{1:n} - kT \tag{2.1}$$

where X_i is the contribution of the i^{th} largest contributor. The contributors are ordered according to size, implying $X_i \geq X_j$ if $i \leq j$. Also $T_{1:n}$ is the joint contribution of the largest n contributors, and T is the cell total or the sum of all contributions to that cell. Equation (2.1) measures the concentration within cell C, i.e., the n largest contributors should not be responsible for more than k percent of the cell total. This implies that $T_{1:n} \leq kT$. Therefore, if $S_d(C) < 0$, then cell C is considered safe.

In some cells, one respondent may contribute a relatively large value as compared to the other contributors, and in these cells the cell total approx-

imates the value of the dominating contributor. These cells are considered sensitive by the dominance rule. As most cell values are usually determined by variables as industrial classification and regional information, the set of contributing establishments to a certain cell is considered common knowledge. Hence, in the case of one dominating contributor, disclosure is very well possible. In the case of n dominating contributors, the disclosure scenario is that $n-1$ dominating contributors may pool their knowledge to identify the n-th contributor, but as n increases the probability of cooperation decreases, also lowering the probability of disclosure. Usually, cooperation between three or four and more entities is considered unlikely.

For the (n,k)-dominance rule, frequently used values are $n = 3$ and $k = 70\%$, implying that cooperation between 2 entities is considered possible, but cooperation between 3 entities is considered unlikely to happen.

In Table 2.3, a sensitive cell, according to a (3,70%)-rule is shown. For $k = 70\%$, this cell fails the dominance rule, since entities A, B and C together contribute 80% of the cell total.

Entity	Value
A	4000
B	2500
C	1500
D	850
E	600
F	550
total	10000

Table 2.3: The structure of a sensitive cell

As the (n,k)-dominance rule is the sensitivity measure used at Statistics Netherlands, it will be investigated in more detail. Afterwards, the pq-prior/posterior rule will be described in less detail. First, however, some results on the safety of individual respondents are deduced.

2.6.1 The safety of individual respondents

The contribution of a respondent to a cell can be estimated by other contributors to that cell. These respondents form a coalition and this coalition

can subtract its joint contribution D from the published cell total T. This results in an *upper* estimate of the largest contribution. To measure the effectiveness of this estimation, the following definition is used:

Definition 3 *A respondent is (p,n)-safe from a coalition of $n-1$ respondents if any coalition of $n-1$ respondents can not estimate that respondent's contribution with an accuracy of up to p percent.*

The p-percent safety requirement states that the estimation Y_a of the contribution X_a of respondent a should be at a distance of p percent of X_a. Thus the accuracy p is defined by

Definition 4 $Y_a = T - D = (1+p)X_a \Leftrightarrow p = \frac{T-D}{X_a} - 1.$

if Y_a is an upper estimate of X_a. This definition implies that small p gives a good approximation. For analyzing the safety of individual respondents, the following theorem is very useful:

Theorem 1 *If the largest respondent in a cell is (p,n)-safe from a coalition of the $2^{nd}, \ldots, n^{th}$ respondents, then every respondent in that cell is (p,n)-safe from every coalition of size $n-1$.*

To see this, some lemmas are deduced and finally combined to prove this theorem.

First, consider the case where the estimated respondents are contributors 1 resp. a. The size of the coalition is $n-1$.

Lemma 2 *If the largest contribution in a cell is (p,n)-safe from a given coalition of size $n-1$, then any respondent in that cell is (p,n)-safe from that coalition.*

Proof: The estimation Y_1 of X_1 by a coalition D_{n-1} of size $n-1$, where D_{n-1} denotes the sum of the contributions of the $n-1$ intruders, is

$$Y_1 = T - D_{n-1}$$

so respondent 1 is not (p,n)-safe if

$$Y_1 \leq X_1 + pX_1 \Leftrightarrow T - X_1 - D_{n-1} \leq pX_1 \tag{2.2}$$

In the general case, where the coalition estimates X_a, the estimate is

$$Y_a = T - D_{n-1}$$

This means that respondent 1 is not (p, n)-safe if

$$Y_a \leq X_a + pX_a \Leftrightarrow T - X_a - D_{n-1} \leq pX_a \tag{2.3}$$

As the contributions X_i are ordered according to size,

$$X_a \leq X_1 \Rightarrow T - X_a - D_{n-1} \geq T - X_1 - D_{n-1} \tag{2.4}$$

Suppose equation (2.2) fails, i.e. the largest possible coalition cannot estimate the largest contribution to within p percent of its real value. Then, using the ordering on the X_i's, and using equations. (2.2), (2.3) and (2.4), we find

$$T - X_a - D_{n-1} \geq T - X_1 - D_{n-1} \geq pX_1 \geq pX_a \Rightarrow$$

$$T - X_a - D_{n-1} \geq pX_a$$

This implies that whenever a coalition of $n-1$ contributors cannot approximate the largest contributor, then such a coalition of $n-1$ respondents cannot approximate any contributor. □

Second, consider the case where the size of the coalition is $n-1$, and the coalition is formed by the $2^{nd}, \ldots, n^{th}$ contributors respectively any $n-1$ contributors, trying to estimate X_1.

Lemma 3 *If the largest contribution in a cell is (p, n)-safe from a coalition of size $n-1$ of contributors $2, \ldots, n$ to that cell, then the largest respondent is (p,n)-safe from any coalition of size $n-1$.*

Proof:

The estimation Y_1 of X_1 by a coalition of respondents $2, \ldots, n$ is

$$Y_1 = T - T_{2:n}$$

so respondent 1 is not (p, n)-safe if

$$Y_1 \leq X_1 + pX_1 \Leftrightarrow T - T_{2:n} \leq X_1 + pX_1 \Leftrightarrow \sum_{i=n+1}^{\infty} X_i \leq pX_1 \tag{2.5}$$

In the general case, where any coalition estimates X_1, the estimate is

$$Y_1 = T - D_{n-1}$$

where D_{n-1} denotes the sum of the contributions of the $n-1$ intruders. This means that respondent 1 is not (p,n)-safe if

$$Y_1 \leq X_1 + pX_1 \Leftrightarrow T - X_1 - D_{n-1} \leq pX_1 \tag{2.6}$$

As the contributions X_i are ordered according to size,

$$T_{2:n} \geq D_{n-1} \Leftrightarrow T - X_1 - D_{n-1} \geq T - X_1 - T_{2:n} \tag{2.7}$$

because the left-hand side of (2.7) always contains the largest $n-1$ available respondents. Suppose equation (2.5) fails, i.e. the largest possible coalition cannot estimate the largest contribution to within p percent of its real value. Then, using the ordering on the X_i's, and using equations. (2.5), (2.6) and (2.7), we find

$$T - X_1 - D_{n-1} \geq T - X_1 - T_{2:n} \geq pX_1 \Rightarrow T - X_1 - D_{n-1} \geq pX_1$$

This implies that whenever a coalition of the $2^{nd}, \ldots, n^{th}$ contributors cannot approximate the largest contributor, then no coalition of $n-1$ respondents can approximate the largest contributor. □

Finally, combining Lemmas 3 and 2 gives that if the largest contribution is (p,n)-safe from a coalition of the $2^{nd}, \ldots, n^{th}$ respondents then the largest contribution is (p,n)-safe from any coalition, and this implies that any respondent is safe from any coalition. This yields Theorem 1.

2.6.2 The protection offered by the (n,k)-dominance rule to individual respondents

Although the (n,k)-dominance is quite standard, the protection level it offers at the respondent level is not directly clear. This is because the (n,k)-dominance rule does not entirely account for the internal structure of the cell. It compares the relative size of the sum of the n largest contributors to the size of the cell total, but it does not account for the relative size of largest contributor versus the other $n-1$ largest contributors or versus the remaining contributors.

contributor	cell 1	cell 2
1	70%	40%
2	5%	35%
remainder	25%	25%

Table 2.4: The compositions of two cells with equal sensitivity but different internal structures

In Table 2.4, the compositions of two cells with equal sensitivity are shown, but intuitively, the left-hand cell is more sensitive than the right-hand cell. The reason for this is as follows. Suppose that the largest 'coalition' which is likely to be formed has size one for both cells, i.e., we are interested in the estimate contributor 2 can make for contributor 1. In the left-hand cell, the subtraction of contribution 2 gives an estimate of contribution 1 with an accuracy of at most $p_{left} = \frac{100-5}{70} - 1 = 36\%$. In the right-hand cell, this estimate is at best $p_{right} = \frac{100-35}{40} - 1 = 63\%$. Intuitively, the conclusion is that the left-hand cell is more sensitive than the right-hand cell, as $p_{left} < p_{right}$ and a low p implies a good approximation. This is in contradiction with the conclusion we draw from the dominance rule, since according to this rule both cells are equally sensitive.

The contradiction is caused by the fact that the (n,k)-dominance rule only looks at the sum of the dominant contributions and the cell total, while protection against a coalition should be offered by the remaining, non-dominating respondents to the cell. Therefore, the protection level offered also depends on the internal cell structure. Hence, the dominance rule does not specify the accuracy with which individual contributors to an unsafe cell can be deduced. We can however deduce a minimum protection level guaranteed by using an (n, k)-dominance rule.

Theorem 4 *If an (n, k)-dominance rule is satisfied, this implies that all individual respondents are $(\frac{1}{k} - 1, n)$-safe*

This means that if an (n, k)-dominance rule states that a cell is safe, then each individual respondent in that cell is $(\frac{1}{k} - 1, n)$-safe. Hereby the dominance rule is translated into a safety requirement on individual respondents.

Proof: A cell is safe according to the dominance rule if

$$T_{1:n} < kT \Leftrightarrow X_1 + T_{2:n} < kT \tag{2.8}$$

The (p, n)-safety requirement states that every respondent is safe if

$$T - T_{2:n} > X_1(1 + p)$$

If $p = \frac{1}{k} - 1$, this results in

$$T - T_{2:n} > X_1(\frac{1}{k}) \Leftrightarrow$$

$$X_1 + kT_{2:n} < kT \tag{2.9}$$

Combining this with equation (2.8) gives

$$X_1 + kT_{2:n} < X_1 + T_{2:n} < kT$$

because $k < 1$. This means that if the (n, k)-dominance rule is satified, every respondent is $(\frac{1}{k} - 1, n)$-safe.

As was shown in Table 2.4, the worst case scenario is that the largest contribution makes the cell sensitive on its own, i.e. the cell is not only sensitive according to an $(2, k)$-rule, but is also sensitive for an $(1, k)$ rule. So, in case $n = 1$, equation (2.8) reduces to

$$X_1 + 0 < kT \Leftrightarrow X_1 < kT$$

and equation (2.9) also reduces to

$$X_1 + 0 < kT \Leftrightarrow X_1 < kT$$

This means that in the worst case, the (n, k)-dominance rule exactly guarantees $(\frac{1}{k} - 1, n)$ safety to individual respondents. The quality of the estimation that is made by the coalition is

$$p_{min} = \frac{T - D}{X_1} - 1 = \frac{T - 0}{kT} - 1 = \frac{1}{k} - 1 \qquad \square \tag{2.10}$$

The conclusion is that using a (n,k)-dominance rule does not uniquely determine a protection level provided to individual respondents. The protection depends on the internal structure of the cell. However, using a

(n,k)-dominance rule implies a minimal protection level, being the protection level offered to a respondent in the intuitively most unsafe cell at a dominance rule safety level of k. Therefore, p is fixed at this minimum protection level p_{min}, and used for every cell. In this way, the (n, k)-dominance rule indirectly imposes a restriction on the safety of individual respondents, independently of the internal cell structures.

2.6.3 The (p,q)-prior/posterior rule

When using the (p,q)-prior/posterior rule, it is assumed that each respondent has some prior knowledge about the contributions of the other respondents. More precisely, it is assumed that each respondent can estimate all the other contributions with an accuracy of up to q percent. This expresses the idea that the intruder is not totally uninformed about the order of magnitude of the contributions of other respondents. The intruder can combine its prior knowledge with the published cell totals (the posterior knowledge) to make an estimate of the contributions of other respondents. The (p,q)-prior/posterior rule identifies a cell being sensitive if a intruder, using his prior and posterior knowledge, is able to estimate another respondent's contribution with an accuracy of up to p percent, where $p < q$. (remember that small p, q imply a good estimate. When $q < p$, the intruder already knows more than allowed). In this disclosure scenario, the intruder can subtract his own contribution, and his estimates of the other contributions from the published cell total to estimate the contribution of the largest contributor. The estimate of the contribution of respondent i is at least equal to $X_i - qX_i$, $i = 3, \ldots, \infty$. Therefore,

$$E[X_1] = \sum_{i=1}^{\infty} X_i - X_2 - \sum_{i=3}^{\infty} X_i(1-q) = X_1 + q\sum_{i=3}^{\infty} X_i$$

is an upperbound for the estimate of the largest contribution. Again, the cell is considered sensitive if $E[X_1] \leq X_1 + pX_1$, and again it suffices to only consider the case for which the second-largest contributor tries to disclose the largest contributor, by an argument similar to that for the (n,k)-dominance rule. Note that if $q = 1$, i.e. the intruders do not have any prior knowledge, this degenerates to (2.5). The sensitivity measure corresponding to the

(p,q)-prior/posterior rule is

$$S_{p;q}(X) = pX_1 - \sum_{i=3}^{\infty} qX_i \tag{2.11}$$

For a detailed analysis of the mathematical properties of upper sensitivity measures see Cox [6].

Chapter 3

Source Data Perturbation

3.1 Basic principles

Often, many tables are generated from one microdata file. Safety regulations demand that these tables are safe, and this can be accomplished by applying disclosure control methods at the table level or at the microdata level (see Section 2.4 for an overview of these methods). These methods have some disadvantages. Table-level methods do not recognize the interrelationship between tables imposed by the underlying microdata. This means that, for example, when two (almost) equal cells are found in two tables that are generated from the same microdata, one may be altered in some way by a disclosure control measure, and the other may be altered in yet another way by another disclosure control measure. Hence, applying disclosure control methods on a table-by-table basis is likely to create inconsistencies. This creates very complex problems when suppressing cells: once a cell is suppressed, it also has to be suppressed in every other table it appears. The data disseminator has to keep track of which cells were suppressed, and the problem of minimizing the number of secondary suppressions can also get very complicated.

On the other hand, disclosure control methods that operate at the microdata level guarantee consistency across tables. However, now it is not clear when generated tables are sufficiently protected by the microdata level disclosure control methods. It is desired that each table is safe according to some safety measure, while the data still must contain some information

value for their users. This means that the data must not be changed too much as a result of the disclosure control methods used. Microdata disclosure control methods do not account for the specific properties of the tables involved. Another disadvantage of using disclosure control measures at the microdata level is that the data is changed permanently.

Therefore, the conclusion is that table level-methods do not account for their underlying microdata (and hence for other tables), and that microdata level-methods do not account for the tables generated from that microdata. Therefore some link has to be set up between the tables and the microdata, and disclosure control measures have to be developed using this link. To this end, the data disseminator defines a calibration set of tables that are to be published, and that therefore have to be safe. Then each table is investigated on how much it has to be altered to be safe, and this implies that the respondents contributing to that table have to be changed also. This is done by multiplying their contribution by some weight factor, and this weight factor is chosen according to some safety measure that evaluates the safety of the now perturbed tables. This way, each respondent is assigned its own weight factor. In any case, *the original microdata is not changed.* The perturbation of the data is applied upon tabulation, and the only alteration of the microdata is the addition of the perturbation factors, but these could also have been saved to a separate file.

The multiplicative weight factors can only be applied to quantitative variables. The multipliers can also be applied to frequency counts, but not to categorical variables. In this case, multiplicative noise factors can best be seen as compared to sample weights. In a frequency count table, the number of observations in a cell is multiplied by the sample weight to obtain an estimation of the population total for that cell. It makes sense to perturb this estimation by multiplicative noise. However, a perturbed frequency count table may be subject to table rounding, because the counts are no longer integers (see Section 2.4.1). Therefore, perturbing frequency count tables may have consequences.

If the attributes can only be represented by a discrete set of values, e.g. male or female, it makes no sense to perturb these scores because a person is either male or either female, not something in between. Therefore

categorical variables cannot be perturbed. For SDP however, this is not a problem when generating tables for publication, because the cells in those tables are always filled with scores on quantitative attributes. The scores on the qualitative attributes are used to span the tables. Therefore, these scores are not supposed to be perturbed.

In general, SDP methods have the following properties:

- a multiplicative weight factor is added to the microdata
- when generating tables from the microdata, contributions are weighted by this factor
- weighted tables generated from the microdata are consistent and safe
- sensitive cells are perturbed by the weight factors; nonsensitive cells are left relatively unaffected
- no bias is introduced into the table data
- the original microdata is not changed
- the weight factors can only be applied to quantitative variables

In short, the microdata is perturbed by assigning a weight factor to each respondent in the microdata file. Because the perturbation factors are applied at the microdata level, the tables generated from the microdata are mutually consistent. Also, because the perturbation is implied by some safety measure that evaluated the disclosure risks of the tables in the predefined set of tables, these tables are safe and can be published. Naturally, the safety of the entire set of tables is checked prior to publication, because a *combination* of tables may still induce disclosure. See Section 6.2 for more details.

The safety measure used has to account for the fact that the contribution of individual respondents is protected by the use of protective noise, as intruders can deduce less information from perturbed figures than from the real figures. The (n,k)-dominance rule and the (p,q)-prior/posterior rule introduced in Section 2.6 do not account for the use of protective noise. However, it is desirable to base the safety level of a table on the (n,k)-dominance

rule, as it is currently the sensitivity measure at Statistics Netherlands. This issue will be elaborated in Section 4.1.

Generally, SDP methods consist of two stages:

1. Find the amount of perturbation needed for the tables in the predefined set of tables

2. Find the amount of perturbation needed for all individual respondents in the microdata, given stage 1.

Several methods can be used to determine the how much the tables in the set of tables should be defined. These methods will be described in Chapter 4. Also, several methods can be used to translate this table level perturbation to the microdata level. These will be described in Chapter 5. First however, the standard ZES method will be described. This method can be seen as a member of the family of SDP methods, although it skips the first stage.

3.2 The ZES method

In this section, the method of Zayatz, Evans, Slanta [7] is evaluated. This method assigns respondents weight factors that alternately are equal to (1 - μ) and (1+μ), where μ is the mean percentage of noise added. However, not exactly μ percent of noise is added, but rather a percentage that is drawn from some distribution centered around μ. Because of the random way of assigning perturbation factors, noise in nonsensitive cells is expected to cancel out, while this is not the case for sensitive cells, as will be shown. First the percentage of noise added is considered to be given (e.g. 10 percent), but later we will try to find ways to let μ depend on the predefined set of tables that are to be generated from the microdata. However, if no set of tables is predefined, the ZES method is very appropriate.

To perturb an entity's data by an amount of noise of 10%, a multiplier of about 0.9 or about 1.1 could be assigned to each entity. Then, upon tabulation each entity's contribution is multiplied with this multiplier. To hide the exact amount of perturbation for possible intruders, multipliers are

with equal probability (i.e. $\frac{1}{2}$) drawn from some distribution centered around $(1-\mu)$ or from some distribution centered around $(1+\mu)$. To ensure that the expected value of the applied multipliers equals 1, it is very important that the distribution at $(1-\mu)$ and the distribution at $(1+\mu)$ together form a symmetrical distribution around 1. Hence, the perturbed value is computed by

$$\mathsf{perturbed\ value} = \mathsf{true\ value} * \mathsf{multiplier} \tag{3.1}$$

where multiplier is drawn from a bimodal distribution, for instance

$$\mathsf{multiplier} \sim \frac{1}{2}N(1-\mu,\sigma) + \frac{1}{2}N(1+\mu,\sigma)$$

where $N(\mu,\sigma)$ is the Normal distribution. The multiplier of respondent i is denoted by $m_i = 1 \pm r_i$.
This is the case for census data; for sample survey data see Section 3.2.2. An example of a bimodal distribution is sketched in Figure 3.1.

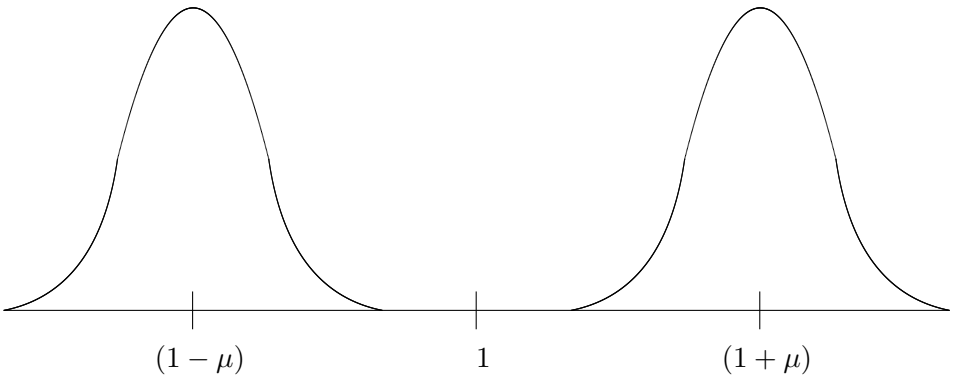


Figure 3.1: bimodal distribution centered around $(1-\mu)$ and $(1+\mu)$

Any distribution can be used to compose such a bimodal distribution, such as the normal distribution, or the beta distribution. For details on generating random variates, see Law and Kelton [14]. Zayatz, Evans, and Slanta [7] try several distributions and find no consistent differences.

3.2.1 The influence of perturbation on sensitive cells versus nonsensitive cells

By definition, a nonsensitive cell contains a relatively large number of contributors (at least 4, when n = 3 in the dominance rule), and because the probability of receiving positive directed noise equals the probability of negative directed noise, it is to be expected that about half of the contributions is inflated and that the other half is deflated. In short, the alternately addition of positive and negative directed noise cancels itself out when aggregating perturbed company values, and consequently the expected value of the noise added is 0 (i.e., the expected value of the used multipliers equals 1). However, the addition of noise induces an increase of variance, see Section 3.2.3.

For sensitive cells, where one company is dominating the other contributors, the relative large size of the dominating entity assures that the overall amount of noise added to the cell is also relatively large. For example, in Table 3.1, company A dominates companies B, C, and D, by contributing 80% of the cell total. Naturally, the more dominant the dominator is, the more the total change in cell value resembles the amount of noise added to the dominating company.

Sensitive cell				Nonsensitive cell			
resp.	value	weight	perturbed value	resp.	value	weight	perturbed value
A	8000	0.89	7120	H	2700	0.89	2403
B	850	1.11	943.5	I	2400	1.11	2664
C	600	1.12	672	J	2600	1.12	2912
D	550	0.91	500.5	K	2300	0.91	2093
total	10000	(-7.6%)	9236	total	10000	(+0.7%)	10072

Table 3.1: influence of perturbation on sensitive and nonsensitive cell

It is not guaranteed that sensitive cells *always* receive a high amount of noise. Suppose for example a cell contains two dominant, equally sized contributors. If one of these two is perturbed upward, and the other is perturbed downward, then the cell is not likely to receive much perturbation. However, as we will see in Section 7.2, in general sensitive cells are much more likely to receive noise than nonsensitive are.

3.2.2 Census data versus sample survey data

With census data, the sample consists of all companies present in the population. Because it is quite an effort to include all companies, often only part of the population is included in the sample. In a sample survey, the contribution of each company is weighted inversely proportional to the probability of inclusion in the sample. The sample weights are deduced from the inclusion probabilities, but they are also corrected for nonrespons. Large companies have a larger probability of being included in the sample, so they end up having smaller sample weights. The motivation for this is that large companies are rather unique and ignoring them would prohibit the sample from accurately representing the real population. On the other hand, there are a lot of smaller companies, and one such small company could represent a number of other small companies with similar properties. Suppose for example that in a population there are 5 companies of size (approximately) 8000 employees. Then if one of those companies is chosen to enter the sample, its contribution is multiplied by 5 as to represent the total value of companies that are of that size. Thus companies with large weights are already somewhat protected because their large weight implies that there are a lot of other companies of similar size in the population, thus lowering the probability of identification. Large companies with small weights need some extra protection.

When adding noise to sample survey data, these weights are accounted for by adding noise as follows:

$$\mathsf{perturbed\ value} = \mathsf{true\ value} * [\mathsf{multiplier} + (\mathsf{weight}\ -\ 1)] \tag{3.2}$$

The sample weight of respondent i will be denoted by w_i.
Intuitively, noise is only added to the company in the sample, and not to the other (weight - 1) multiples that are not included in the sample. For companies that are unique in the population (when using the dominance rule the risky ones), the weight factor will be close to 1, degenerating to the census case. Contributions by companies with small weights in this fashion receive a lot of noise, while companies with large weights are not severely perturbed, see example 4.

As is clear from example 4, dominant company A receives a lot of noise, while companies B, C and D who have large weights receive a very small

	true contributions				perturbed contributions				
resp.	value	x	weight		value	x	multiplier + weight-1		change
A	8000	x	1	= 8000	8000	x	0.89 + 0	= 7120	11%
B	850	x	6	= 5100	850	x	1.11 + 5	= 5194	1.8%
C	600	x	8	= 4800	600	x	1.12 + 7	= 4872	1.5%
D	550	x	10	= 5500	550	x	0.91 + 9	= 5451	0.9%
cell total	10000			23400				22636	3.3%

Table 3.2: including sample weights

amount of noise. As company A does not get much protection from its weight, this is a favorable effect.

Notice however that this cell is not sensitive: for the weighted contributions, company A contributes 34% of the cell total, hence explaining the low cell value change of 3.3%.

In the next section the statistical properties of this method are mathematically elaborated.

3.2.3 The effects of applying multiplicative noise

Now the method of perturbation is intuitively clear, it is time to mathematically prove the demonstrated properties of the ZES method. Also, because the perturbation of the data induces a loss of information, the increase of the variance due to the addition of the noise is evaluated as a measure of this information loss.

Since the distribution of the multipliers is symmetric around 1 and the multipliers are in expectation equal to one, the expected value of the amount of noise added to a respondent is zero. Now we will show that the expected value of the amount of noise in any cell is also zero.

Theorem 5 *The ZES perturbation procedure does not introduce any bias into the cell values.*

Proof: Suppose we are dealing with census data (i.e. all companies are included). Because $E[m_k] = 1$ for all respondents k,

$$E[T_N] = E[\textstyle\sum_k m_k X_k] = \sum_k X_k E[m_k] = \sum_k X_k = T$$

for each k in each cell in each table. This is the case for census data. For the more general case of sample survey data, let $\widehat{T} = \sum_k X_k w_k$ be the unperturbed cell estimate. Let $\widehat{T}_N = \sum_k (m_k + w_k - 1)X_k$ be the noise-added estimate. Now $E[\widehat{T}_N] = E[\sum_k (m_k + w_k - 1)X_k t_k]$ where $t_k = 1$ if the k^{th} sampling unit is chosen, 0 otherwise. Assume $P[t_i = 1] = \pi_i$ and $P[t_i = 0] = 1 - \pi_i$. Remember that the sample weights w_k are the reciprocals of π_k, i.e. $w_k = \frac{1}{\pi_k}$. So $E[\widehat{T}_N] =$

$$= \sum_{i=0}^{1} E\left[\sum_k (m_k + w_k - 1)X_k t_k \mid t_k = i\right] P\left[t_k = i\right]$$

$$= E\left[\sum_k (m_k + w_k - 1)X_k t_k \mid t_k = 1\right] P\left[t_k = 1\right]$$

$$+ E\left[\sum_k (m_k + w_k - 1)X_k t_k \mid t_k = 0\right] P\left[t_k = 0\right]$$

$$= E\left[\sum_k (m_k + w_k - 1)X_k \pi_k + 0 * (1 - \pi_k)\right]$$

$$= \sum_k X_k \pi_k (w_k - 1) + X_k \pi_k E\left[m_k\right]$$

$$= \sum_k X_k \pi_k (w_k - 1) + X_k \pi_k = \sum_k X_k w_k \pi_k = T \qquad \square$$

Now it is proved that $E[T_N] = E[\widehat{T}_N] = T$ for census data and for sample survey data. For theory on computing expectations by conditioning, see e.g. Ross [17].

To compute the increase of variance induced by the perturbation, let $e = \widehat{T}_N - \widehat{T}$. We have already proved that $E[e] = 0$. Also, since $\mathrm{COV}\left[\widehat{T}, e\right] = 0$,

$$\sigma^2\left[\widehat{T}_N\right] = \sigma^2\left[\widehat{T} + e\right] = \sigma^2\left[\widehat{T}\right] + \sigma^2\left[e\right] + 2\mathrm{COV}\left[\widehat{T}, e\right]$$

$$= \sigma^2\left[\widehat{T}\right] + \sigma^2\left[e\right] + 0 = \sigma^2\left[\widehat{T}\right] + E\left[e^2\right] - (E[e])^2 = \sigma^2\left[\widehat{T}\right] + E\left[e^2\right]$$

An unbiased estimator of the variance is

$$\widehat{\sigma}^2\left[\widehat{T}_N\right] = \widehat{\sigma}^2\left[\widehat{T}\right] + e^2 = \widehat{\sigma}^2\left[\widehat{T}\right] + \left(\widehat{T}_N - \widehat{T}\right)^2$$

Notice that for census data, the first part of the right-hand term disappears, as the census value is not an estimate like the sample survey value is. Hence, for census data the unbiased estimator equals e^2. For sample survey data, $\widehat{\sigma}^2\left[\widehat{T}\right]$ can be found using the Horvitz-Thompson estimator $\widehat{T}_{HT}$ for T:

$$\widehat{T}_{HT} = \sum_{i=1}^{n} \frac{X_i}{\pi_i}$$

given that $\pi_i > 0, (i = 1, 2, \ldots, N)$, where n is the sample size and N is the population size. Formulas for the variance of $\widehat{T}_{HT}$ are given in Cochran [5].

Applying the multiplicative noise to the data results in altered tables. How much the tables are altered, can be computed by calculating the loss of information.

The increase of variance can be used to measure the information loss. As the added noise increases the variance of the cell totals by e^2, the information loss of cell i amounts to $e_i^2 = (\widehat{T}_i^N - \widehat{T}_i)^2$. This was shown in the previous section. The total loss of information can be defined as the sum of the information losses over all cells.

Chapter 4

Perturbing tables

The first step of SDP methods is to look at the tables in the predefined set of tables to determine the amount of perturbation needed for those tables to be safe. The desired perturbed cell totals in those tables can be found using several methods. First, a sensitivity measure for perturbed cells is deduced. Using this sensitivity measure, the amount of perturbation needed for each cell can be determined. Also, an approach based on the method of Iterative Proportional Fitting can be used to determine the desired cell totals. Because this method controls the marginal cells, it will be referred to as the MARG method.

4.1 Measuring the safety of perturbed cells

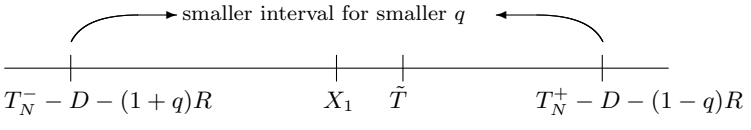
When noise is added to a cell, the dominance rule is not applicable anymore, since it presumes unperturbed data. The dominance rule does not account for the extra protection offered by the added noise. In fact, the sensitivity of a perturbed cell, measured by the dominance rule, differs only slightly from the sensitivity of the original cell, and the direction of the difference depends on the randomness of the noise assignment process. So according to the dominance rule, noise addition could in fact increase the sensitivity of a cell. Therefore a suitable safety measure for perturbed cells has to be found. This safety measure should be deduced from the dominance rule, as the dominance rule is the sensitivity measure used at Statistics Netherlands. A solution to this problem is to use the (p, n)-safety requirement, implied by using a (n,k)-dominance rule (see Theorem 1). This requirement states that

the largest contributor of a cell is not allowed to be approximated to within p percent of its value, and the value of p is deduced from the (n, k)-dominance rule used. The link between k and p was shown in Section 2.6.2. To find a sensitivity measure for perturbed cells, a reasoning is followed similar to that of the (p,q)-prior/posterior rule. That is, the (n,k)-dominance rule is used to define how close the largest contribution is allowed to be approximated, and then a reasoning similar to that of the (p,q)-prior/posterior rule is used to get to that approximation. First, it is shown how to find a lower and an upper estimate for the largest contribution, such that the lower estimate is as far from the largest contribution as the upper estimate is, i.e. X_1 lies in the middle of the interval $(Y_1^{lower}, Y_1^{upper})$. If this is the case, inflation has the same effect as deflation. Then a sensitivity measure for perturbed cells can be deduced.

Suppose a cell has a cell total T. Suppose a coalition of the 2^{nd} to n^{th} largest contributors wishes to disclose the the contribution of the largest contributor. Also suppose the perturbed cell total equals either T_N^+ or T_N^-, for inflated or deflated cell totals respectively. The coalition of intruders can subtract their own contributions from the perturbed cell total, and they can also subtract their estimates of the remaining contributors from the perturbed cell total. The estimation of contribution i is at least equal to $X_i(1-q)$, and at most equal to $X_i(1+q)$. Thus, the coalition can construct an interval around X_1:

$$X_1 \in \left[\left(T_N^- - D - (1+q)R\right), \left(T_N^+ - D - (1-q)R\right)\right]$$

By this definition, the interval around X_1 is smaller for smaller q, which means that if the coalition can make a relatively good approximation of the contributions, they can, evidently, also make a relatively good approximation of X_1. This can be visualized as follows:



where

$D = \sum_{i=2}^{n} X_i$ = joint contributions of the coalition of intruders

$R = \sum_{j=n+1}^{\infty} X_j$ = joint contributions of remaining contributors

This interval can be shown (in expectation) to be symmetrical around X_1, that is, the distance of X_1 to the upperbound equals the distance of X_1 to the lowerbound of the interval:

$$
\begin{aligned}
& X_1 - lowerbound = \\
& = X_1 - T_N^- + D + (1+q)R \\
& = T - T_N^- + qR
\end{aligned} \tag{4.1}
$$

For the upperbound,

$$
\begin{aligned}
& upperbound - X_1 = \\
& = T_N^+ - D - (1-q)R - X_1 \\
& = T_N^+ - T + qR
\end{aligned} \tag{4.2}
$$

The expressions in equations (4.1) and (4.2) are equal, if the multipliers are chosen from a symmetrical distribution with mean 1. In that case, the probability that a cell total is perturbed upwards equals the probability that a cell is perturbed downwards. In both cases the amount of noise is equal, that is, $T_N^+ - T = T - T_N^-$. So, the safety interval is in expectation symmetrical around X_1. This is very useful, since now inflation and deflation of the cell total have the same effect on the safety of the largest respondent. This is why the reasoning similar to that of the (p,q)-rule is used, since else the interval around X_1 would be shifted to the right. This is caused by the fact that the intruders subtract their contributions from the cell total, but do not take the remaining contributors into account. Therefore, their estimate lies above the true value X_1, and the interval is not centered around X_1 (it lies upward, towards T). If that were the case, the amount of deflation would have to be larger than the amount of inflation to provide sufficient distance to X_1. This would destroy the unbiasedness of the noise.

To translate the safety interval found into a sensitivity measure, we can use the fact that the estimate of X_1, deduced by the intruders should not approximate X_1 up to p percent. On the upperside of the interval, the cell is safe if

$$T_N^+ - D - (1-q)R > X_1 + pX_1 \tag{4.3}$$

The information available to the coalition of intruders in the left-hand side of equation (4.3) is compared to the safety requirement in the right-hand side of equation (4.3). In the case of sample survey data, determining the information that is available to the intruders is somewhat more intricate. The coalition knows the weighted and perturbed cell total $\widehat{T}_N$, its own (*unweighted*) contributions D, and its estimation of the remaining contributors $(1-q)\widehat{R}$. Note that the respondents represented by the sample weights of the largest n contributors are included in the set of remaining contributors $\widehat{R}$ for as far as the intruders know. Therefore,

$$\widehat{R} = \sum_{i=n+1}^{\infty} w_i X_i + \sum_{i=1}^{n} (w_i - 1) X_i$$

Now (4.3) evaluates to

$$\widehat{T}_N^+ - D - (1-q)\widehat{R} > X_1 + pX_1 \Leftrightarrow$$

$$\sum_{i=1}^{\infty} (w_i + r_i) X_i - \sum_{i=2}^{n} X_i - (1-q)\widehat{R} - X_1 - pX_1 > 0 \Leftrightarrow$$

$$-\sum_{i=1}^{\infty} r_i X_i + pX_1 + q\sum_{i=1}^{n} X_i - q\sum_{i=1}^{\infty} w_i X_i < 0 \Leftrightarrow$$

$$-\sum_{i=1}^{\infty} r_i X_i + pX_1 - q(\widehat{T} - D - X1) < 0$$

The cell total was perturbed upwards to begin with, so $\sum_{i=1}^{\infty} r_i X_i > 0$. Therefore, $-\sum_{i=1}^{\infty} r_i X_i < 0$.

This expression could be used as a sensitivity measure $S_N(C)$ for perturbed cells. Note that for $w = 1$ (the census case) and if $E[r_i] = 0$, this degenerates into the sensitivity measure $S_{p;q}(C)$ for the (p,q)-prior/posterior rule of Section 2.6.3. The deduced sensitivity measure $S_N(C)$ accounts for the protection offered by the noise added to the data:

$$S_N^+(C) = -\sum_{i=1}^{\infty} r_i X_i + pX_1 - q(\widehat{T} - D - X_1)$$

In fact this measures the distance between the upperbound of the desired safety interval and the actually offered upperbound of the safety interval. Note that $S_N^+(C)$ is decreasing in r_i, the added noise. This means that noise

addition induces a decrease of the sensitivity of the cell. Also note that $S_N^+(C)$ is increasing in p, which implies that a more strict safety demand results in a higher sensitivity. Moreover, $S_N^+(C)$ is decreasing in q and in w, as $(\widehat{T} - D - X_1)$ is always greater than or equal to 0. This implies that these parameters hamper disclosure. This reflects that the sample weights also provide some protection to individual respondents. Similar reasoning can be applied to the lowerside of the interval, and this results in S_N^-.

$$S_N^-(C) = \sum_{i=1}^{\infty} r_i X_i + pX_1 - q(\widehat{T} - D - X_1)$$

The cell total was perturbed downwards to begin with, so $\sum_{i=1}^{\infty} r_i X_i < 0$. This means that $S_N^-(C)$ is decreasing in r. Therefore, $S_N^-(C)$ is equal to $S_N^+(C)$. As the probability of a cell total being inflated equals the probability of a cell total being deflated, and as the amount of inflation in expectation equals the amount of deflation, it suffices to use

$$S_N(C) = -\sum_{i=1}^{\infty} r_i X_i + pX_1 - q(\widehat{T} - D - X_1)$$

Again, for $S_N(C) > 0$, cell C is considered to be sensitive.

If the reasoning of this section is used to extend the reasoning of section 2.6.2, the worst case accuracy of equation (2.10) is

$$\begin{aligned} p_{min} &= \tfrac{T-D-(1-q)R}{kT} - 1 = \tfrac{T-0-(1-q)(1-k)T}{kT} - 1 \\ &= \tfrac{1-(1-q)(1-k)}{k} - 1 \end{aligned} \tag{4.4}$$

This reflects that if the intruder can make a good approximation before any publication of data (i.e., q is small), the quality of the estimation is higher. The parameter q is not given from the (n,k)-dominance rule. However, since p can be deduced from k (see Section 2.6.2), and since it is required that $p < q \leq 1$, we can choose q dependent of p, for instance $q = p + \frac{1-p}{2}$.

The desired cell totals can be deduced from this sensitivity measure. For each sensitive cell, this sensitivity measure gives a safety interval. The desired cell total is one of the two endpoints of the interval. By choosing either one with equal probability, roughly half of the weighted cell totals in the row/column will be higher than its real value, while roughly the other half will be smaller than its original value. Therefore, the row/columntotals

are not expected to deviate too much from their real values, which is a desirable effect. Suppose all weighted cell totals would be chosen to be equal to their accompanying upperbound, then all the protective noise would accumulate in the marginal cells. This is not very desirable. Naturally, relative differences in cell total sizes also have to be accounted for. The decision whether a cell total is chosen higher or lower than its original cell total, can be based on some kind of pattern. For instance, some of the patterns possible are:

+	-	+	-
-	+	-	+
+	-	+	-
-	+	-	+

or

+	+	-	-
-	-	+	+
-	-	+	+
+	+	-	-

On the upperside of the interval, the cell is safe if

$$S_N^+(C) < 0 \Leftrightarrow$$

$$\sum_{i=1}^{\infty} r_i X_i > pX_1 - q(\widehat{T} - D - X1) \tag{4.5}$$

The right-hand side of this inequality is the desired amount of perturbation. The desired cell total b then is equal to $b = \widehat{T} + pX_1 - q(\widehat{T} - D - X1)$. On the lowerside, the cell is safe if

$$S_N^-(C) < 0 \Leftrightarrow$$

$$\sum_{i=1}^{\infty} r_i X_i < q(\widehat{T} - D - X1) - pX_1 \tag{4.6}$$

In this case, the desired cell total b is $\widehat{T} - pX_1 + q(\widehat{T} - D - X1)$. Of course, if the cell isn't sensitive, the length of the safety interval is zero. The desired amount of perturbation then is zero and the desired cell total is identical to the original cell total.

4.2 Using IPF: the MARG method

The perturbed cell totals can also be found using a method that controls the perturbation of the marginal cells. These cells are considered more important than interior cells, because they represent the total of some category.

Therefore, a small amount of noise is added to the marginal cells. It is important that the sum of the row totals still equals the sum of the column totals, since else the table is inconsistent. The perturbation put into the marginal cells then is spread over the corresponding rows and columns, making sure that more sensitive cells receive more of the perturbation. This can be done by Iterative Proportional Fitting (IPF). IPF, also known as raking, is often used in the field of input-output analysis to estimate the input coefficients matrix A given some total interindustry sales U_i by sector i and total interindustry input purchases V_j by sector j. Entry a_{ij} of A then represents the sales from sector i to sector j. For the exact economical background of this procedure, see (for instance) [3] or [15]. In the statistical context, IPF is used to make tables additive, i.e. given row totals U_i and column totals V_j, all the entries in row i should add up to U_i and all entries in column j should add up to V_j. For mathematical details on IPF and other methods for making tables additive, see Fagan and Greenberg [8]. The standard IPF method is described in appendix A.1. In this report, the two-dimensional IPF method is described. However, the method can be applied to tables of all dimensions.

In our context, IPF can be of use in the following way: the vectors U_i and V_j form the *desired* marginal row totals and column totals. Hence, by this approach it is possible to add noise to interior cells, while controlling the amount of perturbation imposed on the original values of the marginal cells, which is very desirable, because we wouldn't like the marginal cells to receive too much noise. Often, marginal cells are considered to be of more importance than the interior cells. The desired marginal cell totals can be specified to values very close to their real values. Then the noise imposed by the deviation made is spread along the corresponding row/column, and this can be done in such a way that somewhat more sensitive cells receive more of the noise than less sensitive cells. Also, cells that are desired to receive no noise at all can be forced to equal their true values. These cells can for instance be identified by a certain minimum level of aggregation. This approach is discussed in Section 4.2.1, and will be referred to as the MARG method.

Another approach is to apply IPF *after* the ZES method is applied.

Since marginal and interior nonsensitive cells at a high level of aggregation are considered important, these are fixed at their real values. Then IPF is used to make the tables additive. This approach is proposed in Zayatz, Evans, and Slanta [7], and will be discussed in Section 4.2.2. This approach will be referred to as the MARG after ZES or ZES/MARG method.

4.2.1 The MARG method

To find the target rowtotals $\overline{\mathbf{u}}$, each i^{th} row total could be multiplied by a perturbation factor p_i. To ensure that the table converges for these new marginals, the new target column totals have to be deduced from these chosen row totals. Therefore, suppose each row i, $i = 1, \ldots, g$ of the original matrix $\mathbf{A}$ is multiplied by a perturbation factor p_i. By this multiplication, each row is perturbed by a perturbation factor, hence each row total is perturbed by that factor. This means u_i becomes $p_i u_i$ for $i = 1 \ldots g$. The column totals are perturbed by a factor dependent of $p_1, \ldots, p_g$, i.e. $v_j = \sum_{k=1}^{g} p_k a_{kj}$, $j = 1, \ldots, h$. Note that it's not possible to just 'pick ' perturbation factors for the column totals as we did for the row totals; these two depend on each other, so if this dependency is neglected, no solution can be reached and the IPF method does not converge. The exact choice for the values of the p_i's should depend on the sensitivity of the marginal of the corresponding row. Generally, marginal cells are not expected to be very sensitive, so they should only be perturbed by a relatively small amount. However, occasionally marginal cells may be sensitive, implying that the corresponding row is entirely dominated by a few respondents, and sensitive cells in this row should receive more noise, as they are expected to be very sensitive. Naturally it is also possible to pick column total perturbation factors and let the row totals depend on that choice. This depends on the dimensions of the matrix involved. If the matrix is very rectangular (that is, g $\gg$ h), the smallest side of the matrix may accumulate a lot of noise.

The desired marginal cell totals could also be found by adding some noise to the marginal cell of the marginal cells, i.e. the total of all contributions to the table. This is the sum of the row totals and also the sum of all column totals. The perturbation added to this 'super' marginal cell then is spread over all marginals. Then the perturbation in all row totals and column totals

is defined, and can be spread over the table. This is a top-down approach, that works very natural in multi-dimensional tables.

The IPF method tries to estimate the target matrix $\overline{\mathbf{A}}$ given $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$, the (perturbed) totals (see Appendix A.1). However, we are not interested in $\overline{\mathbf{A}}$, that is, we do not want the perturbation to be equally spread over all cells of $\mathbf{A}$. We would rather have the sensitive cells receiving a relatively larger share of the perturbation. To this end, each cell is weighted by a factor z_{ij} that is dependent on its sensitivity. In fact the r's of stage 1 become

$$r_{ij} = \frac{u_i}{\sum_{j=1}^{h} z_{ij} a_{ij}}$$

while in stage 2 the s's become

$$s_{ij} = \frac{v_j}{\sum_{i=1}^{g} z_{ij} a_{ij}}$$

The size of the scalar assigned depends on the sensitivity of the cell. Therefore, the weight factor z_{ij} is chosen as

$$z_{ij} = 1 + S_{ij} \tag{4.7}$$

where S_{ij} denotes the safety level of cell a_{ij}.

$$S_{ij} = \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{\infty} x_i}$$

following an (n, k)-dominance rule. So, because s_{ij} and t_{ij} are the scalars assigned to $v_{ij}a_{ij}$, the a_{ij} are multiplied by $s_{ij}v_{ij}$ and $r_{ij}v_{ij}$. However, any safety measure can be used.

Also, we can demand that some cell totals receive no perturbation at all, by replacing those cell totals by zeros in the target matrix and subtracting them from their target marginals, and then, after the procedure, the original cell totals are put back into the then perturbed matrix. These cells can be chosen on the basis of their sensitivity or their level of aggregation. So, the level of perturbation depends on the sensitivity, and the decision regarding whether to perturb a cell or not to perturb it, could be chosen to depend on the level of aggregation.

4.2.2 MARG after adding multiplicative noise

An objection to the ZES method is that nonsensitive cells are also perturbed by the multiplicative noise factors. By fixing nonsensitive cells at their true cell totals, and by raking to the true marginals, this problem can be solved. For example, nonsensitive cells that contain more than 1000 contributors could be fixed at their true values. Such cells are considered more important because they represent a lot of respondents, and for the same reason marginal cells are also more important to table users than interior cells are. Therefore, such nonsensitive cells are left totally unperturbed. Their noise is spread over the other cells. Since nonsensitive cells at a high level of aggregation are not supposed to receive much noise, this additional noise to the other cells is not expected to be large.

4.2.3 MARG on several tables simultaneously

All tables in the predefined set could be raked individually, but this could create inconsistencies across tables. Therefore, all cells could be raked simultaneously, to prevent the creation of inconsistencies (see [7]). This is done by constructing a n-dimensional supertable, where n is the total amount of distinct categorical variables that appear in any of the tables in the predefined set. Each (perturbed) cell value then is inserted into the interior of the supermatrix, dependent of its values on the n categorical values. This implies that each cell value exactly fits into one interior cell of the supermatrix. This supermatrix is raked to the fixed marginal values, which gives an adjustment factor for each cell. This adjustment factor then should be applied to all respondents contributing to that cell. A practical drawback of this approach is that the supertable may be very large, and that it contains a lot of empty cells.

Chapter 5

Perturbing microdata

When, using one of the methods of the previous chapter, perturbed tables are found, these have to be translated into perturbation factors for the individual respondents in the microdata. This can be done by using the sensitivity measure for perturbed cells which was developed in Section 4.1. Also, the perturbation factors could be found by solving an optimization problem. This approach will be described in Section 5.2. Finally, given a perturbed cell total, the respondent's perturbation factors can be found by proportionally spreading the cell perturbation factor over all respondents that contribute to the cell. This will be described in Section 5.3.

5.1 Using the S_N sensitivity measure

Now that a sensitivity measure for perturbed cells has been developed (see Section 4.1), this measure can be used to find the noise needed for each sensitive cell. Because of the random assignment of the noise, the safety of a noise added cell cannot be guaranteed. However, it can be demanded that the probability that it turns out to be safe is larger than some threshold probability d:

$$P\left[S_N(C) \leq 0\right] > d$$

This implies

$$P\left[\sum_{i=1}^{\infty} r_i X_i > pX_1 - q(\widehat{T} - D - X1)\right] > d$$

So, the parameters needed are the μ and the σ of the Normal distribution, and the threshold factor d. The standard deviation σ can be chosen small, for instance equal to 0.02. If σ is too small, the perturbation factors become too predictable. On the other hand, if σ is chosen too large, the control over the randomly chosen perturbation factors is lost. The threshold factor d can be set arbitrarily at 95 or 90%. The choice of the mean μ is derived below.

If the cell is sensitive, then the first contributor must be a dominant contributor, which implies that its noise also dominates the noise factors of the other contributors. Because we cannot make any assumptions on the direction of perturbation of the other contributors to the cell, we assume the other contributions are not perturbed at all. On average this is true, because $E[r_i] = 0$. Therefore we can use

$$P\left[r_1 X_1 > pX_1 - q(\widehat{T} - D - X1)\right] > d \Leftrightarrow$$

$$P\left[r_1 > \frac{pX_1 - q(\widehat{T} - D - X1)}{X_1}\right] > d \tag{5.1}$$

This can be rewritten, using the fact that the perturbation parameter r is $\mathrm{N}(\mu, \sigma)$ distributed: Suppose the standard deviation used in the perturbation process is σ. Let $X \sim N(0, \sigma)$. Let X_d be the value for which $P[X > X_d]$ equals d percent (this X_d can be found by using the inverse Normal distribution). Also, $r \sim N(\mu, \sigma)$, so $r - \mu \sim N(0, \sigma)$. This implies that

$$P[r - \mu > X_d] = d$$

so now (5.1) can be used to find μ:

$$\left.\begin{array}{r} P[r > \frac{pX_1 - q(\widehat{T} - D - X1)}{X_1}] = d \\ P[r - \mu > X_d] = d \end{array}\right\} \Leftrightarrow$$

$$\mu = \frac{pX_1 - q(\widehat{T} - D - X1)}{X_1} - X_d \tag{5.2}$$

Substituting (2.10) into this expression gives

$$\mu = \frac{\left(\frac{1}{k} - 1\right) X_1 - q(\widehat{T} - D - X1)}{X_1} - X_d \tag{5.3}$$

So, the mean of the noise required for this cell is decreasing in k (high k implies that safety demands are not very strict) and in q (high q implies that the intruders are rather clueless about the real value of the other contributions).

Since the distribution of the r_i's is given, it may be useful to derive the distribution of $\sum_{i=1}^{\infty} r_i X_i$. The sum of independently Normal distributed random variates is also Normal distributed, its mean being the weighted sum of all individual means, and its variance being the quadratically weighted sum of individual variances (see Section 8.3).

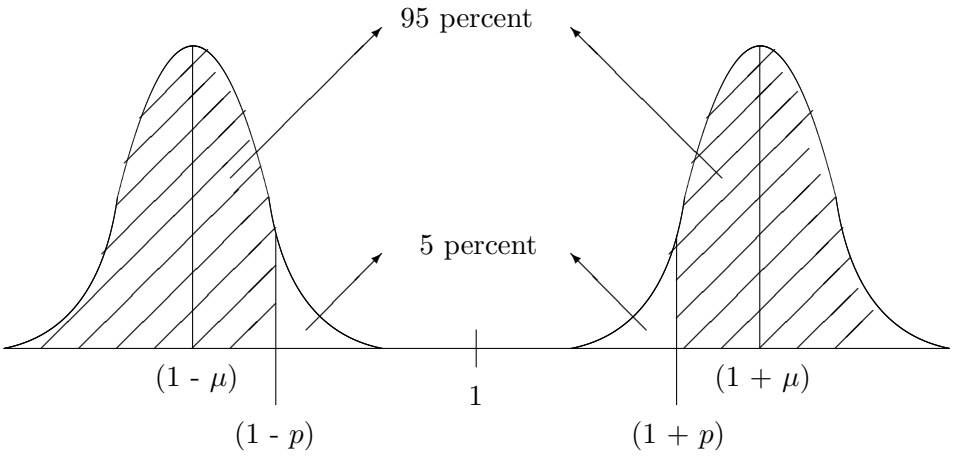


Figure 5.1: Choosing μ when $d = 95$ percent

If μ is chosen according to equation (5.3), d percent of the randomly generated multipliers will deviate sufficiently from 1, i.e., d percent of the multipliers will have a deviation larger than p percent from 1. For $d = 95$ percent, this is visualized in Figure 5.1.

By this procedure, each cell in each table is assigned some μ that reflects the noise level that is appropriate for that particular cell. Ideally, we would like to have one μ for which all cells are safe to be released. Therefore it seems logical to take the largest μ found, and to use this $\mu_{\max}$ as a basis for the ZES method. This approach will be referred to as the ZES(μ) method. Alternative approaches may assign different μ's to different classes of respondents. For instance, respondents that contribute to sensitive cells may

be assigned a larger μ than respondents that never contribute to sensitive cells. These and other possibilities are sketched in Section 5.3. For these alternative possibilities, this approach increasingly deviates from the original ZES method.

5.2 Using optimization methods

Given the desired cell totals over all tables in the predefined set of tables, the perturbation factors for the respondents can be chosen such that the perturbed contributions sum up to the desired cell totals. The weighted cell totals, i.e., the sum of the weighted contributions, have to resemble the desired cell totals as closely as possible, hence this problem can be rewritten as a minimization problem. This approach will be referred to as the optimization approach, or for short the OPT approach.

5.2.1 Problem formulation

The OPT approach can be formulated as a minimization problem. The weighted cell totals have to fit to the desired cell totals as good as possible. Therefore, we would like to solve the system of linear equations

$$\sum_{j=1}^{|R|} m_j X_{ij} = b_i \qquad (1 \leq i \leq |C|)$$

This means that for each cell i, the sum of the weighted contributions $\sum_j m_j X_{ij}$ has to equal the desired cell total b_i. Because this is very likely to be an inconsistent system of equations, i.e., there is no weight vector m_j that can simultaneously satisfy all $|C|$ equations, we have to minimize the residuals $e_i = \sum_j m_j X_{ij} - b_i$. One way to do this is to minimize $\sum_{i=1}^{|C|} |e_i|$. This is a so called ℓ_1-problem. The problem can also be solved using other norms, which lead to minimizing $\sum_{i=1}^{|C|} (e_i)^2$ (the ℓ_2-problem) or minimizing $\max_{1 \leq i \leq |C|} |e_i|$ (the ℓ_∞-problem). The ℓ_2-problem is often solved using least squares. Now, first the ℓ_1-problem formulation is investigated, and after that a ℓ_2-problem formulation is discussed.

The ℓ_1-problem formulation

The ℓ_1-optimization problem can be written as

$$\min \sum_{i=1}^{|C|} \left| b_i - \sum_{j=1}^{|R|} m_j X_{ij} \right|$$

where

$|C|$ = number of cells in optimization

$|R|$ = number of weightfactors/respondents in optimization

b_i = desired cell total of cell i, $1 \leq i \leq |C|$

m_j = weight factor of respondent j, $1 \leq j \leq |R|$

$$X_{ij} = \begin{cases} X_j & \text{if respondent j contributes to cell } i \\ 0 & \text{otherwise} \end{cases}$$

X_j = contribution of respondent j

$0 \leq m_j^{\min} \leq m_j \leq m_j^{\max}$, $1 \leq i \leq |C|$, $1 \leq j \leq |R|$

This problem can be solved using linear programming (LP), but since e_i can be negative as well as positive, the problem has to be reformulated. To this end, the following formulation is used. To this end, introduce a variable $y_{|R|+1}$ and write $x_j = y_j - y_{|R|+1}$. Then define $X_{i,|R|+1} = -\sum_{j=1}^{|R|} X_{ij}$, to create an additional column in the matrix X. Each entry in this column is the negated cell total for the cell corresponding to the row it is in. Also write $|e_i| = u_i + v_i$, where $u_i = e_i$ and $v_i = 0$ if $e_i \geq 0$, but $u_i = 0$ and $v_i = -e_i$ if $e_i \leq 0$. Hence, the reformulation is

$$\min \sum_{i=1}^{|C|} u_i + \sum_{i=1}^{|C|} v_i$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^{|R|+1} y_j X_{ij} - u_i + v_i = b_i \quad (1 \leq i \leq |C|) \\ u \geq 0, v \geq 0, y \geq 0 \end{cases}$$

This reformulation is elaborated in Cheney & Kincaid [4].

The ℓ_2-problem formulation

The ℓ_2-problem formulation is

$$\min \sum_{i=1}^{|C|} \left(b_i - \sum_{j=1}^{|R|} m_j X_{ij} \right)^2$$

An extra constraint can be added to this model:

$$\sum_{i=1}^{|R|} m_i = |R|$$

ensuring that the mean of the multipliers equals 1.
This problem can be solved using least squares or (convex) nonlinear programming (NLP).

Also, another problem formulation can be applied. This problem formulation focuses on the dilemma of losing information versus the amount of disclosure protection offered. It minimizes information loss, under the restriction of safety:

$$\min \sum_{i=1}^{|C|} (e_i)^2 = \sum_{i=1}^{|C|} \left(\sum_{j=1}^{|R|} m_j X_{ij} - T_i \right)^2$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{|R|} m_i = |R| \\ \text{cell } i \text{ is safe} \qquad (1 \leq i \leq |C|) \end{cases}$$

Cell i is safe if either (4.5) or (4.6) is satisfied. Information loss e_i was defined in Section 3.2.3.

5.3 Assigning perturbation factors proportionally

After the first stage of a SDP method, for each cell a perturbed cell total is given. The noise in this perturbed cell total can be spread proportionally over all respondents that contribute to that particular cell. This does not necessarily means that all contributors to the cell are perturbed by the perturbation factor assigned to the cell total. The contributors could also be assigned a perturbation factor proportional to their size or sensitivity. Also some respondents could be kept unaffected by the multiplicative noise (i.e. their perturbation factor is set equal to 1). Then the remaining respondents receive all the noise imposed on the cells. Also, strata (i.e., classes) of respondents could be defined. All respondents in a stratum then receive equivalent multipliers. For short, the translation of the perturbation imposed on a cell can be done in may different ways.

Chapter 6

Disclosure scenarios for SDP

6.1 Disclosure scenarios

Using SDP can also have its drawbacks, and it is important to be aware of them when using SDP methods. One problem arises when there are several tables that are spanned by the same variables. This may be the case when in one table spanned by region and industrial classification, profits are given, and in another table, spanned by the same variables, some other variable like returns are given. As in the perturbation process each company is assigned its own multiplier that is used when generating all tables, both tables may be combined to find information about individual data and multipliers. Once a multiplier is disclosed, that information can be used to disclosure more information. Another example is formed by trend statistics, where for each year some variable of interest is given in the cells. As, for each company, the same multiplier is used every year, the ratio of, for instance, profits in 1996 and profits in 1997 is not protected. For short, always using the same multiplier when perturbing a company's contributions also has its drawbacks. It can be shown that trends and ratios are not protected if each respondent is assigned a unique multiplier.

Another issue is that before a set of perturbed tables is released, the disclosure risks have to be evaluated. Although sufficient perturbation is applied to the individual cells, some cell totals may be combined to approximate individual contributions. A checking mechanism to evaluate these disclosure risks is sketched. It is investigated under what circumstances a

respondent's multiplier can be disclosed, and how this can be prevented.

To know to what extent Statistical Disclosure Control methods should be applied, it is necessary to know what the possibilities of the intruders are. If such disclosure scenarios are known, appropriate measures can be taken to prevent those disclosure scenarios from being successful.

6.2 Checking the safety of a set of perturbed tables

In this subsection, a checking mechanism to evaluate disclosure risks is sketched. It has to prevent that cell totals can be combined to approximate individual contributions.

Suppose respondents A, B, and C all contribute to cells 1, 2, and 3. Suppose these cells are composed in the following way:

$$\begin{aligned} m_A X_A + m_B X_B &= T_1 \\ m_A X_A + m_C X_C &= T_2 \\ m_B X_B + m_C X_C &= T_3 \end{aligned} \tag{6.1}$$

This is a "dangerous" combination of cells. For instance, a respondent can find it's multiplicative weight factor using the information of the published cell totals T_1, T_2, and T_3 (6.1) and its knowledge of its own contribution to the cells and of who are the other contributors to the cells. Suppose $X_C = 120$, $T_2 = T_3 = 180$, and $T_1 = 240$. Then, from cells 2 and 3,

$$\begin{aligned} & m_B X_B + m_C * 120 = 180 \\ & m_A X_A + m_C * 120 = 180 \\ & \text{———————————————————} + \\ & m_A X_A + m_B X_B + m_C * 240 = 360 \end{aligned}$$

Using cell 1, contributor C can deduce its weight:

$$\begin{aligned} & m_A X_A + m_B X_B + m_C * 240 = 360 \\ & m_A X_A + m_B X_B = 240 \\ & \text{———————————————————} - \\ & m_C * 240 = 120 \end{aligned}$$

$$\Leftrightarrow m_C = \frac{1}{2}$$

Using this information, contributor C knows $m_A X_A$ and $m_B X_B$ and it can approximate X_A with an accuracy of $(m_A - 1)$ percent and X_B with an accuracy of $(m_B - 1)$ percent. As the weights are usually close to 1, this would imply a disclosure of information. The protective powers of the multiplicative weights lie in aggregation, and are not intended to work on an individual basis. The situation would be worse if all the noise factors have the same deviation from 1, i.e. if they are all equal to $(1 - \mu)$ or $(1 + \mu)$. If this were the case, contributor C could deduce μ and that would be very dangerous. For instance, in the example used above, C knows that m_A is either $\frac{1}{2}$ or $1\frac{1}{2}$. Then C can use T_2 and deduce that m_A is either 80 or 240. Using possible prior knowledge, C might guess which value is the real one. This is why the multipliers are drawn from some distribution centered around $(1 - \mu)$ or $(1 + \mu)$ respectively in the ZES method.

To prevent weights from being disclosed, the system of equations (6.1) has to be solved. Generally speaking, (6.1) can be written as

$$\mathbf{Xm} = \mathbf{T} \Leftrightarrow \begin{bmatrix} X_A & X_B & 0 \\ X_A & 0 & X_C \\ 0 & X_B & X_C \end{bmatrix} \begin{bmatrix} m_A \\ m_B \\ m_C \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

This system can only be solved if the determinant of $\mathbf{X} \neq 0$. So,

$$\begin{vmatrix} X_A & X_B & 0 \\ X_A & 0 & X_C \\ 0 & X_B & X_C \end{vmatrix} \neq 0$$

and this evaluates into

$$-2X_A X_B X_C \neq 0$$

So, if any of the contributions X_A, X_B, or X_C is zero, insufficient information is available to disclose the multipliers. This is logical, since a contribution can only be disclosed via a cell to which it contributes. Three equations are needed to solve for three variables, so $\mathbf{X}$ has to be square.

By this exercise, it becomes clear that cells with few respondents contributing to them should not be published, especially if these respondents are found together in the same cells. For these "sparsely" populated cells,

too much information can be disclosed by individual respondents or coalitions. For cells with a higher number of contributors, it is more difficult to know which respondents are contributing to those cells, and moreover $\mathbf{X}$ is larger and harder to solve. Also, larger-sized coalitions are needed, while the probability of cooperation between a number of respondents decreases as the number of respondents increases. It is not easy to predict the prior knowledge a intruder might have about which respondents contribute to a certain cell, but it is to be expected that the quality of this knowledge is decreasing in the number of contributors to the cell. Therefore, it might only be necessary to check combinations of cells that have a small number of contributors. For instance, cells with more than 100 contributors could be ignored. This number is dependent of the number of tables that are to be published and is also dependent of their properties.

6.3 Trends and ratios

Inherently to using a unique multiplicative perturbation factor for each respondent, trends and ratios are not always protected, as will be shown. Suppose two cells are composed as follows:

$$m_A X_A + m_B X_B = T_1$$
$$m_A Z_A + m_B Z_B = T_2$$

Suppose $T_1 = 40$ and $T_2 = 42$. Assume contributor B is the intruder. Contributor B knows its own contributions, which are, say, $X_B = 20$ and $Z_B = 10$. Therefore B knows

$$m_A X_A + m_B * 10 = 40$$
$$m_A Z_A + m_B * 20 = 42$$

If contributor B has discovered its own weight, then it can deduce the ratio of X_1 and Z_1. Suppose $m_B = 0.9$. Then

$$\frac{m_A X_A}{m_A Z_A} = \frac{X_A}{Z_A} = \frac{22}{33} = \frac{2}{3}$$

The ratio of X_A and Z_A is not protected. The intruder now knows that the profits of year Z are $\frac{3}{2}$ times larger than those of year Z, i.e. an increase of 50%. Therefore, if it is desirable to protect ratios and trends, then the multiplier m_i should not be applied to all attributes of respondent

i. The multiplier m_i should be considered as a base multiplier, and each attribute of respondent i should be multiplied with a weight factor that is slightly different than m_i. This is in fact a form of *additive* noise on top of multiplicative noise. For instance, suppose the base multiplier m_A of respondent C is 1.1. Suppose X_A is multiplied by 1.105 and that Z_A is multiplied by 1.098. Then the ratio is perturbed by a factor $\frac{1.105}{1.098} = 1.006$. This is half a percent, which not very much on an increase of 50%. However suppose that the ratio is very small, for example 1.05, i.e. an increase of 5%. An adjustment of 0.6% on an increase of 5% is a perturbation of 10%. This means that smaller ratios can be better protected than larger ratios.

An alternative approach is to use two distinct multiplicative weight factors for each contribution. The first one is the old base multiplier, that is applied to all attributes of the respondent. The second one is a multiplier that belongs to a certain attribute, i.e., each attribute has it's own multiplier. Now contribution X_i of respondent i becomes $m_x m_i X_i$ in stead of $m_i X_i$, Y_i becomes $m_y m_i Y_i$ etc.

The disadvantage of this additive noise approach is that the weighted attributes may no longer add up to their totals. Attention should be paid to this issue. Suppose respondent i has three attributes X_i, Y_i, and Z_i. Suppose Z_i is defined as $Z_i = X_i + Y_i$. If X_i and Y_i are both multiplied by m_i, then the perturbed values sum up as they are supposed to:

$$m_i X_i + m_i Y_i = m_i(X_i + Y_i) = m_i Z_i$$

If they are not perturbed by the same factor, this is not the case. The solution is to the m_z implied by m_x and m_y:

$$m_x X + m_y Y = m_z Z \Leftrightarrow m_z = \frac{m_x X + m_y Y}{Z}$$

Evidently, m_x and m_y should be chosen such that m_z is somewhere near m_x and m_y.

Chapter 7

Results from real data

In this chapter, the methods described in the previous chapter are evaluated by applying them to real data. First the properties of the data will be described. Then some results of the various SDP methods are presented and evaluated. The implementation of the testing program SoDaP is discussed in the appendix.

7.1 The data

Two datafiles were available for testing purposes. Datafile A contains about 65,000 respondents and four variables, being measure of size, geographic location, industrial classification, and returns. Datafile B is the result of a sample survey which contains 10,664 respondents and a wide variety of variables. For small companies (10 to 50 employees), not all companies of the population were included in the sample survey for the sake of efficiency, i.e. a sample was taken. Larger companies (more than 50 employees) were all included in the sample. To compensate for not included companies (10 to 50 employees), and for nonresponding companies (in the entire sample), each respondent included in the sample was assigned a sample weight. This implies that for the smaller companies, the sample weights are generally higher than those assigned to the larger companies, because for the larger companies the entire population, excluding nonrespondents, is included. The entire population size is 46,932, which is the sum of the sample weights. Datafile A is assumed to contain census data, as no weights were given. Datafile B is very appropriate for the generation of a set of tables, since it contains a

large amount of variables that may be used to span and fill those tables. From datafile A, only a few tables can be generated, which are used for small-scale testing purposes.

Both datafiles are in ASCII-format and are provided with a metadata file, specifying which variables are in the accompanying microdata file. Also, to maintain compatibility with the ARGUS-software, the metadata files contain information concerning field starting positions and widths, and missing value indicators. Therefore, all meta- and microdata files can also be read by the ARGUS-software, providing the possibility to obtain feedback on the correctness of the testing program. The functional design and the data structures of the testing program are described in the appendix.

7.2 The results

In this section, the effects of applying SDP methods on tables generated from real microdata are evaluated. First, the effects of SDP are investigated for a single table, and then for a set of tables.

7.2.1 The effects on a single table

To see what happens when a SDP method is applied, a simulation is used to evaluate the behaviour of cells in a table generated from the microdata when SDP was applied. The table used is generated from the data of microdata file A, and shows returns of companies, grouped by SBI (standard industrial classification) and region. This table is replicated 1000 times, while applying the ZES method. In each replication new perturbation multipliers are drawn independently from a bimodal Normal distribution with a mean of 10 percent of perturbation and a small standard deviation of 0.02. After the simulation, for each cell the average of the 1000 noise-added cell values in the 1000 replications is computed by adding up over all replications and dividing by 1000. This figure then is divided by the real, unperturbed cell value and the result of this operation is shown in Table 7.1.

As can be seen in Table 7.1, the ratios are all very close to 1, indicating that the cell totals are equally probable to be perturbed upward as to be perturbed downward. Also, 46 of the 92 interior cells are perturbed down-

	region	1	2	3	4
SBI2	0.9998	1.0002	0.9998	0.9993	1.0002
15	1.0001	1.0005	0.9998	1.0001	1.0000
16	**0.9992**	**0.9997**	**0.9953**	**1.0017**	**0.9997**
17	1.0001	1.0006	1.0003	1.0005	0.9996
18	0.9995	1.0012	0.9995	0.9992	0.9997
19	0.9997	**1.0018**	0.9994	0.9996	0.9997
20	0.9998	1.0008	0.9994	0.9993	1.0002
21	0.9999	1.0012	0.9997	1.0001	0.9992
22	1.0002	0.9993	1.0000	1.0001	1.0009
23	0.9984	**1.0006**	**0.9964**	**0.9974**	**1.0061**
24	0.9997	1.0006	1.0000	0.9992	1.0001
25	1.0002	1.0013	1.0005	0.9992	1.0002
26	1.0003	1.0006	1.0002	0.9999	1.0005
27	0.9982	**1.0038**	0.9997	**0.9960**	1.0005
28	0.9997	0.9994	0.9997	0.9999	0.9997
29	1.0006	0.9999	1.0001	0.9999	1.0017
30	**0.9982**	**0.9973**	**0.9990**	1.0011	**0.9970**
31	0.9994	**0.9993**	0.9982	1.0003	0.9996
32	0.9987	**0.9976**	**1.0045**	0.9999	**0.9974**
33	0.9992	0.9961	1.0007	0.9999	0.9987
34	1.0012	1.0002	1.0000	0.9994	**1.0016**
35	0.9999	1.0000	1.0000	0.9998	1.0001
36	1.0002	1.0008	1.0001	1.0000	1.0002
37	0.9999	0.9995	0.9997	1.0014	0.9991

Table 7.1: Average perturbed value over 1000 replications, divided by original value. Sensitive cells are set in boldface.

ward while the other 46 interior cells were perturbed upward, which is to be expected. The largest value observed is 1.0061, the smallest value equals 0.9960. A (3,70%)-dominance rule was used to evaluate the safety of the cells. The ratios of sensitive cells do not differ significantly from those of the nonsensitive cells.

We can also look at the standard deviations for the 1000 perturbed observations. These standard deviations give an impression of how much noise would typically be present in a cell after a single application of the perturbation factors. The standard deviation of the perturbed cell value is equal to the standard deviation σ of the added noise, e.

$$\sigma[T^N] = \sigma[e]$$

These are standardized by the true cell value T. Table 7.2 shows the values

of $\frac{\sigma[T^N]}{T}$, which can be seen as the coefficients of variation (CV) of the perturbed value given the true value. This can be denoted by $CV[T^N|T]$.

	region	**1**	**2**	**3**	**4**
SBI2	0.0000	0.0076	0.0058	0.0097	0.0098
15	0.0057	0.0158	0.0097	0.0105	0.0112
16	**0.0602**	**0.0615**	**0.0956**	**0.0706**	**0.0932**
17	0.0115	0.0491	0.0169	0.0164	0.0201
18	0.0107	0.0269	0.0187	0.0176	0.0186
19	0.0160	**0.0638**	0.0382	0.0273	0.0189
20	0.0070	0.0195	0.0136	0.0127	0.0125
21	0.0119	0.0268	0.0235	0.0200	0.0228
22	0.0070	0.0160	0.0128	0.0103	0.0142
23	0.0443	**0.0726**	**0.0798**	**0.0490**	**0.0990**
24	0.0171	0.0317	0.0182	0.0265	0.0275
25	0.0077	0.0236	0.0143	0.0138	0.0135
26	0.0088	0.0237	0.0119	0.0141	0.0168
27	0.0391	**0.0572**	0.0337	**0.0705**	0.0263
28	0.0047	0.0131	0.0111	0.0086	0.0072
29	0.0086	0.0175	0.0141	0.0075	0.0211
30	**0.0565**	**0.0957**	**0.1000**	0.0331	**0.0621**
31	0.0131	**0.0479**	0.0251	0.0208	0.0247
32	0.0418	**0.0511**	**0.0662**	0.0444	**0.0631**
33	0.0151	0.0479	0.0384	0.0139	0.0318
34	0.0450	0.0248	0.0232	0.0210	**0.0587**
35	0.0135	0.0244	0.0255	0.0211	0.0257
36	0.0077	0.0176	0.0114	0.0162	0.0146
37	0.0143	0.0295	0.0299	0.0289	0.0239

Table 7.2: CV's after ZES

Again, sensitive cells are set in boldface. Clearly, for sensitive cells higher CV's are observed. This means that in sensitive cells, the variability of the amount of perturbation is higher than in nonsensitive cells. This implies that sensitive cells are more likely to receive a lot of noise after a single application of the ZES method than nonsensitive cells are. This is a desirable effect.

We can also look at the distribution of the amount of perturbation added to the various types of cells. In every replication, the absolute percentage of noise added to each cell value is computed. After the simulation, for each cell the average percentage of noise added over all 1000 replications is computed. Then, for each type of cell, the average amount of perturbation (in percentages) is computed, and also the maximum and the minimum levels of perturbation found per type of cell are given. The results are shown in

Table 7.3. Evidently, sensitive cells receive more perturbation than non-sensitive cells. Also, interior cells receive more noise than marginal cells. This is logical, because marginal cells are at a higher level of aggregation and are therefore likely to be less sensitive than the interior cells they are marginal for. Nevertheless, marginal cells can also be sensitive, so on average they will receive more perturbation than non-sensitive cells. Because the mean percentage of noise added to individual respondents is 10 percent, cells are not expected to be changed by more than 10 percent. Only if the standard deviation used is large, cell values can be changed more vigorously. The information loss in a table is measured by

$$\sum_{i=1}^{|C|} |e_i| = \sum_{i=1}^{|C|} |T_N - T| \tag{7.1}$$

This figure then is divided by the number of cells in the table. To make comparison between the various methods possible, this figure can be perturbed by the average amount of perturbation in all nonnegative cells. This figure is given between brackets.

percentage noise in:	average	max	min
marginal cells (28)	1.55	5.56	0.41
interior cells (92)	2.74	9.83	0.58
sensitive cells (21)	6.43	9.83	4.18
nonsensitive cells (99)	1.62	4.60	0.41
all nonzero cells (120)	2.46	9.83	0.41
Information Loss: $1.21 \cdot 10^5$ $(4.92 \cdot 10^4)$			

Table 7.3: Amount of perturbation in cells, over 1000 replications of ZES. Information loss is measured by (7.1)

The ZES method does not guarantee that the perturbed table is safe. In fact, for this table only one cell is provided with enough perturbation to be evaluated 'safe' by the S_N sensitivity measure.

If the sensitivity of cells is determined by the S_N safety measure of Section 4.1, rather than by the (n,k)-dominance rule, the 1000 simulations result in Table 7.4. Note that according to Table 7.4 there are only 18 sensitive cells found by the S_N sensitivity measure. The distinction between sensitive cells

percentage noise in:	average	max	min
marginal cells (28)	1.55	5.56	0.41
interior cells (92)	2.74	9.83	0.58
sensitive cells (18)	6.79	9.83	4.83
nonsensitive cells (102)	1.70	4.60	0.41
all nonzero cells (120)	2.46	9.83	0.41
Information Loss: $1.21 \cdot 10^5$ $(4.92 \cdot 10^4)$			

Table 7.4: Amount of perturbation in cells, over 1000 replications of ZES using the S_N sensitivity measure

and nonsensitive cells is more clear: note that the most severely perturbed nonsensitive cell receives less noise than any of the sensitive cells. Because the average perturbation of sensitive cells increases while three sensitive cells moved to the nonsensitive cells, these moving cells are less sensitive than the average sensitive cell. However, they clearly are more sensitive than the average nonsensitive cell is, because the average perturbation of nonsensitive cells increased. Inspection learns that all three cells have a sensitivity close to 70 percent (by the dominance rule), which is just above the threshold of 70 percent. Because the S_N safety measure accounts better for the internal cell structure than the dominance rule does, and because in Table 7.4 the distinction between nonsensitive cells and sensitive cells is more clear than in Table 7.3, we can conclude that the ZES method accounts for the internal cell structure.

percentage noise in:	average	max	min
marginal cells (28)	1.41	4.88	0.37
interior cells (92)	2.47	8.21	0.55
sensitive cells (18)	5.79	8.21	4.40
nonsensitive cells (102)	1.59	4.17	0.37
all nonzero cells (120)	2.22	8.21	0.37
Information Loss: $1.12 \cdot 10^5$ $(5.05 \cdot 10^4)$			

Table 7.5: Amount of unimodally distributed noise in cells, over 1000 replications of ZES

To evaluate the effects of using a bimodal distribution for generating the

perturbation factors compared to using a unimodal distribution, the ZES method can be applied using a uninormal distribution, with mean at 1 and standard deviation 0.1. Note that the bimodal distribution used earlier also has mean 1 and standard deviation roughly equal to 0.1. As can be seen from Table 7.5, generally a smaller amount of perturbation is added. Because a unimodal distribution was used, a significant number of multipliers is close to 1, which is the mean of the distribution. This means that the added noise is close to zero. If a bimodal distribution is used, the probability of a multiplier being equal to one is very small. Most multipliers are supposed to be close to one of the two modes of the distribution. In that case, the multipliers offer a minimum protection level to individual respondents. The information loss is somewhat larger than that of the standard ZES method, but the difference is not very significant.

A similar table can be constructed for the MARG method. First, we use the MARG method in which the marginal cell totals were provided with some noise included. This noise was with equal probability positive or negative directed. In case it was negative directed, marginal cell M would become $M_N = M(1 - pS_d(M))$. Otherwise, M would become $M(1 + pS_d(M))$. The factor $S_d(M)$ is the sensitivity of cell M according to the dominance rule and this factor is somewhere between 0 and 1. The factor p was chosen to be 0.01, so a marginal cell is at most changed by 0.01*1 = 1 percent. This noise then is spread over the interior cells, giving sensitive cells more noise than non-sensitive cells. This is done by weighting cell total T_{ij} by $z_{ij} = 0.85 + 0.3 * S_d(T_{ij})$. This approach was described in Section 4.2.1. Interior cells that contain more than 1000 contributors are fixed at their real value. For the other cells, the cell perturbation factor was applied directly to all contributors to that cell. Note that this also could have been done using any method described in Chapter 5. This approach results in Table 7.6. Clearly, this method provides control over the amount of noise present in marginal cells. As can be seen in Table 7.6, marginal cells receive a very small amount of perturbation. A drawback of this is the loss of control over the interior cells. On average, interior cells receive a slightly smaller amount of noise than in the ZES method. However, the range of values is larger: the largest perturbation measured is 15 percent, while for the ZES

method this is 10 percent. Also the distinction between sensitive cells and nonsensitive cells is less clear in Table 7.6 than in Table 7.3. In Table 7.6 the nonsensitive cells take values between 0.00 and 12.26 percent, while sensitive cells take values ranging from 0.53 to 15.49 percent. The conclusion is that for the MARG method the variability of the noise added is higher, and that the "power" of the MARG method is lower than that of the ZES method. This means that the MARG method cannot distinguish sensitive cells from nonsensitive cells as good as the ZES method can.

percentage noise in:	average	max	min
marginal cells (28)	0.20	0.88	0.00
interior cells (92)	2.06	15.49	0.00
sensitive cells (18)	3.02	15.49	0.53
nonsensitive cells (102)	1.38	12.26	0.00
all nonzero cells (120)	1.63	15.49	0.00
Information Loss: $4.95 \cdot 10^4$	($3.03 \cdot 10^4$)		

Table 7.6: Amount of perturbation in cells, after MARG

When applying the MARG method *after* the ZES method, the marginals are fixed at their true values. Also cells with more than 1000 contributors are fixed at their true values. Cells are weighted by the same factor z_{ij} as was used in the MARG method, and again the cell perturbation factor was directly applied to all contributors to that cell. This results in Table 7.7.

percentage noise in:	average	max	min
marginal cells (28)	0.00	0.00	0.00
interior cells (92)	2.21	9.25	0.00
sensitive cells (18)	4.84	9.25	1.29
nonsensitive cells (102)	1.14	5.71	0.00
all nonzero cells (120)	1.70	9.25	0.00
Information Loss: $4.14 \cdot 10^4$	($2.44 \cdot 10^4$)		

Table 7.7: Amount of perturbation in cells, after MARG/ZES

For both MARG methods, the information loss is smaller than that of the ZES method. This can be explained by the fact that in the MARG methods the perturbation of the marginal cells and the cells at a high level

of aggregation is kept very small. Therefore, the information loss of those larger-valued cells is very small, and this implies that the total information loss of the table is also smaller. Cells with large values usually have a higher information loss than cells with smaller values, because information loss is measured in terms of (perturbed) cell totals. Therefore, information loss in a cell is dependent on the absolute size of the cell value. As marginal cells are more important than interior cells, it is desirable that their information loss is penalized more severely than the information loss of interior cells. Also for both MARG methods, it is not guaranteed that all cells are safe after perturbation. The exact values for the parameters to the methods have to be found by trial and error to make sure they result in safe tables.

percentage noise in:	average	max	min
marginal cells (28)	2.53	9.14	0.67
interior cells (92)	4.53	16.44	0.98
sensitive cells (18)	11.26	16.44	7.87
nonsensitive cells (102)	2.78	7.77	0.67
all nonzero cells (120)	4.06	16.44	0.67
Information Loss: $2.40 \cdot 10^5$	($5.91 \cdot 10^4$)		

Table 7.8: Amount of perturbation in cells, after ZES(μ)

If the ZES method is applied using a bimodal distribution with mean found as described in Section 5.1, the μ found is very large. This is caused by cells that are not only sensitive according to an $(3, 70)$ rule, but are also sensitive according to an $(2, 70)$ and even an $(1, 70)$ rule. We will refer to this type of cells as *supersensitive* cells. For these cells, μ converges to $\frac{1}{k} - 1 = 43\%$ if k = 70%. This is by far too large. If nonsensitive cells are perturbed by this amount of noise, noise may not cancel out very well across contributors. Therefore, the supersensitive cells are ignored when finding μ. These supersensitive cells can hardly be made safe by applying SDP methods, so they shouldn't occur in the table at all. Supersensitive cells should be removed by redesigning the table, for instance by combining two rows. If the supersensitive cells are ignored, and for $q = 1$, the largest μ found equals $\mu = 16.7\%$. Applying this μ in the ZES method results in Table 7.8. The information loss of this method is roughly equal to that of the standard ZES method, which is logical. Moreover, all sensitive cells are

safe after the perturbation process, excluding the supersensitive cells.

If the LP-problem formulation of Section 5.2 is used, the desired cell totals are found as described in section 4.1. A maximum cell value change of 20 % is permitted.

percentage noise in:	average	max	min
marginal cells (28)	1.57	17.03	0.00
interior cells (92)	2.49	20.00	0.00
sensitive cells (18)	14.32	20.00	2.17
nonsensitive cells (102)	0.15	5.86	0.00
all nonzero cells (120)	2.27	20.00	0.00
Information Loss: $8.1 \cdot 10^4$	($3.56 \cdot 10^4$)		

Table 7.9: Amount of perturbation in cells, after optimizing LP

When using the methods of section 4.1, the amount of perturbation in each cell can be controlled exactly. Therefore, all nonsensitive cells receive no perturbation at all. Only some nonsensitive marginal cells receive a small amount of perturbation, caused by sensitive cells in the corresponding row or column. Sensitive cells receive exactly the amount of perturbation desired, under the condition that the perturbed cell total does not deviate more than 20% from the original cell total. Therefore, cells that require more than 20% of perturbation are not entirely safe after perturbation. However, as was argued earlier in this section, these supersensitive cells should not be published at all because they cannot be made safe by SDP methods. As can be seen in Table 7.9, sensitive cells are generally much more perturbed than nonsensitive cells. Due to the control provided by this approach (nonsensitive cells are fixed at their real values), the information loss is somewhat smaller than in the ZES method.

Similar results apply when the problem is formulated as a nonlinear programming problem (NLP). The disadvantage of the nonlinear formulation is that the Hessian of the objective function needs to be computed (see Appendix A.2.3). If the number of respondents is large, the Hessian may not fit into memory anymore. For datafile A, with its 65,000 respondents, this is the case. The nonlinear formulation can only be tested with the data of datafile B, because the Hessian of 10,000 respondents fits into memory (it

percentage noise in:	average	max	min
marginal cells (19)	2.03	9.42	0.00
interior cells (72)	2.27	17.90	0.01
sensitive cells (2)	15.41	17.90	12.92
nonsensitive cells (89)	1.90	9.42	0.01
all nonzero cells (91)	2.22	17.9	0.01
Information Loss: $3.80 \cdot 10^3$ $(1.71 \cdot 10^3)$			

Table 7.10: Amount of perturbation in cells, after optimizing NLP

has about 1,500,000 entries). The computation of the Hessian takes more time than the optimization in itself. The results are given in Table 7.10. Because of the protection offered by the sample weights, there aren't very much sensitive cells in the table. Again a nonsensitive marginal cell receives a lot of perturbation because its columns only contains one sensitive cell. This problem my be solved by perturbing the other (nonsensitive) cells in the column slightly in the opposite direction. This approach should be subject to further research. Table 7.10 is generated from datafile B, so for evaluation purposes, the same table is perturbed by the ZES method, with $\mu = 10\%$. This results in Table 7.11.

percentage noise in:	average	max	min
marginal cells (19)	0.69	4.72	0.03
interior cells (72)	1.12	8.91	0.06
sensitive cells (2)	6.48	8.91	3.99
nonsensitive cells (89)	0.91	4.72	0.03
all nonzero cells (91)	1.03	8.91	0.03
Information Loss: $1.01 \cdot 10^3$ $(9.80 \cdot 10^2)$			

Table 7.11: Amount of perturbation in cells, after ZES on table of file B

7.2.2 The effects on a set of tables

To evaluate the effects of SDP on a set of tables, 20 tables were defined out of the data of datafile B. For this set of tables, the mean to be applied in the ZES method was found to be 18.8%. The results can be found in Table 7.12. The perturbed cells show similar behaviour as in Table 7.8,

which represents the single table case. Because 20 tables were involved, the extremes are more extreme.

percentage noise in:	average	max	min
marginal cells (345)	1.73	8.92	0.04
interior cells (1260)	2.76	18.85	0.00
sensitive cells (155)	6.80	18.85	0.64
nonsensitive cells (1450)	2.08	11.74	0.00
all nonzero cells (1605)	2.54	18.85	0.00
Information Loss: $3.11 \cdot 10^5$ $(1.22 \cdot 10^5)$			

Table 7.12: Amount of perturbation in cells of 20 tables, after ZES method with $\mu = 18.8\%$

The optimization methods can also be applied to a set of tables. The nonlinear programming approach was used to find multipliers for three tables generated from datafile B. This results in table 7.13. If the optimization approach is applied to several tables at once, a respondent may be in a inflated cell in one table, and in a deflated cell in another table. If this happens on a large scale, which is likely if many tables are in the predefined set, the perturbed cell totals may not fit very well to the desired cell totals. Moreover, nonsensitive marginal cells may end up severely perturbed, as was mentioned before. In Table 7.13, marginals cells are even more perturbed than interior cells, on average. These issues need further research. A solution may be to use methods similar to the IPF method, for instance the top-down approach.

percentage noise in:	average	max	min
marginal cells (63)	6.22	17.65	0.10
interior cells (288)	5.64	19.31	0.00
sensitive cells (7)	17.42	19.31	12.92
nonsensitive cells (344)	5.46	17.65	0.00
all nonzero cells (351)	5.75	19.31	0.00
Information Loss: $2.86 \cdot 10^5$ $(4.97 \cdot 10^4)$			

Table 7.13: Amount of perturbation in cells of 3 tables, after NLP method

Chapter 8

Conclusions and further research

8.1 Conclusions

In this report, a family of Statistical Disclosure Control methods named Source Data Perturbation (SDP) is evaluated. In SDP, each respondent in the microdata file is assigned a weight factor. When tables are generated from the microdata, each quantitative attribute is weighted multiplicatively by the weight factor assigned to the corresponding respondent. Several methods can be used to find appropriate multiplicative weight factors. However, these methods share the following properties:

- Sensitive cells are perturbed significantly
- Nonsensitive cells are not changed significantly, or not changed at all
- Tables generated from the microdata are consistent
- No bias is introduced in the tables
- The original microdata is not changed

The major advantages of SDP methods are that tables generated from a base microdata file are consistent. Moreover, the original microdata is not changed. Furthermore, because the perturbation is done in a controlled fashion, the table data are still useful for processing by data users. These

data users are not able to deduce information concerning individual respondents. If sample weights are provided with the data, these already provide some protection the respondents. This extra protection is accounted for when applying perturbation factors to the data attributes.

One of the disadvantages of SDP is that trends and ratios are not protected. This problem can be solved by considering each respondent's multiplicative weight factor to be a base multiplier. Then to each attribute a multiplier slightly different from the base multiplier is applied. This way, trends and ratios are somewhat protected. However, this approach may create inconsistencies in hierarchical tables, because partial totals may not add up to the respondent total anymore. This should be accounted for when assigning multipliers deviating from the base multiplier. It was shown that this can be done by looking at the dependencies introduced by the hierarchical definitions.

Another disadvantage is that very sensitive cells cannot be handled adequately by SDP methods. For these sensitive cells, too much perturbation has to be added to the respondents. Therefore, these very sensitive cells should be removed from the tables before SDP is applied. This can be done by redesigning the table, for instance. SDP methods can also be combined with other disclosure control methods, especially with other data masking methods such as recoding methods. SDP methods should not be combined with data hiding methods such as suppression methods. Rather, they are supposed to form an alternative for these methods.

To measure the safety of perturbed tables, a sensitivity measure for perturbed cells was deduced. This sensitivity measure is based on the safety requirement on individual respondents, which can be deduced from standard sensitivity measures such as the (n, k)-dominance rule. The safety requirement on individual respondents demands that the contributions of any individual respondent cannot be estimated to within too narrow margins. It is shown that if a (n, k)-dominance rule is satisfied, then a limit is imposed to the accuracy within which a contribution of an individual respondent can be estimated. This limit is equal to $\frac{1}{k} - 1$ percent of the contribution that is estimated.

SDP methods first look at the table level to see how much the tables in a predefined set of tables should be perturbed. Given the perturbation levels found in the first stage, in the second stage this is translated into perturbation factors that are assigned to the respondents at the microdata level. Besides the tables in the predefined set also other tables can be published from the microdata. For these tables, the first stage is skipped, and therefore they are not guaranteed to be safe after the perturbation process.

Several SDP methods are evaluated in this report, each of which has its own strengths and weaknesses. These are compared in Table 8.1.

method	**speed**	**control**	**safety**	**information loss**
ZES	very fast	sufficient	not guaranteed	larger
ZES(μ)	very fast	sufficient	guaranteed	larger
MARG	fast	over marginals	not guaranteed	smaller
OPT	slower	over interior	guaranteed	smaller

Table 8.1: Properties of SDP methods

How each of these methods was applied is discussed in Section 7.2. The ZES and the ZES(μ) method can be applied very quickly, as it takes only a fraction of a second to draw the multipliers. The MARG methods take somewhat more time than the ZES methods, but this is only a matter of seconds. The optimization methods take more time, as the LP formulation needs about one minute to be solved, while the NLP formulation needs 15 to 30 minutes. Naturally these figures depend on the number of respondents and tables to which the perturbation processes are applied, as well as on the implementation and hardware used. The ZES methods also provide control over the effects of the perturbation process: sensitive cells receive receive more noise than nonsensitive cells, and marginal cells are perturbed less severely than interior cells. The MARG methods are designed specifically to control the marginal cells, which they do satisfactory. As a result of this, the control over the interior cells decreases. The average perturbation of sensitive cells compared to the average perturbation of nonsensitive cells is good, but the perturbation of some cells differs significantly from the average values. The optimization methods provide control over the interior cells, but marginal cells may end up severely perturbed. This is caused by

sensitive interior cells that are alone in their row or column and are therefore not countereffected by other, opposite directed sensitive cells. This can be prevented by slightly perturbing the nonsensitive cells in the appropriate row or column, to provide opposite directed noise. Another problem of the optimization methods is that the perturbed cell totals may not fit to the desired cell totals anymore if a large set of tables is used.

The ZES(μ) method and the optimization methods guarantee safety, for all cells excluding the ones that are too sensitive. Because of the fitting problems of the optimization methods, not all sensitive cells are guaranteed to be safe if one of the optimization methods is applied to a set of tables. However, for the ZES(μ) method this is guaranteed.

Finally, the information loss of the MARG methods and the optimization methods is somewhat smaller than that of the ZES methods, because these methods provide more control by fixing important nonsensitive cells at their true values.

The general properties of SDP methods can be evaluated using a more general framework, which makes it easier to compare SDP methods to other Statistical Disclosure Control methods.

8.2 Evaluation of SDP methods

The properties of SDP methods can be evaluated using the evaluation criteria applied in [9]. Using these criteria, it is easier to compare SDP methods to other SDC methods. Also it is a good review to get an impression of the possibilities and limitations of SDP methods. Each evaluation criterion is presented, explained, and evaluated.

1. *Security: partial or exact disclosure of individual respondents is not possible.*

   If SDP is applied, a true contribution X_i cannot be disclosed, so exact disclosure is not possible. However, in some situations a intruder may be able to find $m_i X_i$, or some ratio $\frac{X_i}{Y_i}$ (see Section 6.2). In these situations partial disclosure is possible.

2. *Robustness: additional knowledge of the external user, apart from the published information, does not induce disclosure.*

If a table is published twice, once using a SDP approach and once using another disclosure control method such as cell suppression, partial disclosure may be possible. This is the case if information from both tables can be combined to obtain more information concerning individual respondents. Therefore creating inconsistencies across tables should not be allowed.

3. *Flexibility: the method can handle frequency count tables as well as tables of magnitude data; several confidential attributes can be handled simultaneously; both qualitative and quantitative attributes can be protected.*

   SDP methods can handle quantitative data as well as frequency counts of qualitative data. However, perturbed tables of frequency count data may need to be rounded. SDP methods cannot handle qualitative data, but that is not required when protecting table data. As the multiplier assigned to a respondent is applied to all (quantitative) attributes of that respondent, several confidential attributes are handled simultaneously.

4. *Richness of information: information loss due to the disclosure measure should be as small as possible. The method used should not create inconsistencies across tables derived from the same dataset. The published data still has to be useful for further processing by the external user.*

   The amount of information loss depends on the desired level of protection offered to individual respondents. As the perturbation of nonsensitive cells is kept small, these cells are still useful for further processing. Sensitive cells receive more perturbation than nonsensitive cells, and therefore these are less useful for further processing by data users. However, data users are not supposed to process sensitive cells in a useful way, because this induces disclosure and data users are not supposed to be able to deduce information about individual respondents.

5. *Costs: the costs of the implementation of the method are low. Data users quickly understand the method, so they know how to process the released data correctly. The costs of daily processing of statistics are low.*

   The basic principles of SDP methods are fairly easy to understand. The implementation of the ZES method is very simple. Implementation of the ZES(μ) method requires some additional functionality but is also very

simple. Implementing the MARG and optimization methods is somewhat more difficult, especially the optimization methods. These need help from optimizers such as MOSEK. The costs of creating perturbed tables are very low when using the ZES method. Drawing multipliers from some distribution can be done very efficiently and should take no longer than one second for a large microdata file. Applying MARG to tables should not take very long either, especially if little perturbation is imposed and not many cells are fixed at their true values. Optimization methods take more time, ranging from one minute when linear programming is used to 20 minutes when nonlinear programming is used. When using nonlinear programming methods the number of respondents is a limiting factor. If this number is large, the Hessian matrix may not fit in computer memory. Evidently, methods that provide more control the amounts of perturbation add take more time.

8.3 Further research

Further research in the field of SDP methods should focus on the weaknesses of the methods. Is it desirable to protect trends and ratios, and in case it is, how much additive noise should be provided to make them safe? When accounting for the protection already offered by the sample weights, should only large contributions with small sample weights be perturbed, or should every respondent be protected by a multiplicative weight factor? The checking mechanism that was sketched in Section 6.2 needs to be implemented. Also, the application of MARG to several tables simultaneously and the optimization problem formulation that minimizes information loss should be implemented. Moreover, the SDP methods proposed in this report can be refined and fine-tuned to provide more control over the effects of the perturbation. This is especially the case for the MARG methods and the optimization methods. The distribution of $\sum_{i=1}^{\infty} r_i X_i$ discussed in Section 5.1 should be investigated, accounting for the fact that the r_i's may imply a inflation or a deflation. When perturbing several tables, a top-down IPF approach may improve consistency over all tables. SDP methods should be included in existing SDC software such as ARGUS. This also makes it possible to combine SDP methods with existing SDC methods.

Appendix A

A.1 The standard IPF method

In this section, the standard IPF method is described, for the case of 2-dimensional tables.

Suppose we have a g x h matrix $\mathbf{A}$, with row totals u_i, $i = 1 \ldots g$ and column totals v_j, $j = 1 \ldots h$. Suppose the vector of desired row totals $\overline{\mathbf{u}}$ and the vector of desired column totals $\overline{\mathbf{v}}$ are given, and therefore, each element of $\mathbf{A}$ has to be changed to force the actual totals to equal the desired totals. We would like to find the matrix $\overline{\mathbf{A}}$ that corresponds to $\overline{\mathbf{u}}$ and $\overline{\mathbf{v}}$ ($\overline{\mathbf{A}}$ can be seen as some perturbation of the matrix $\mathbf{A}$). This involves two steps:

1. multiply each row of A by a scalar that will make the row sum equal the row constraint. This gives matrix $\mathbf{A}^1$.

2. multiply each column of matrix $\mathbf{A}^1$ by a scalar that will make the column total equal its constraint. This gives matrix $\mathbf{A}^2$.

This process is repeated until convergence is reached. So, in general,

$$\mathbf{A}^{2t+1} = r^{t+1}\mathbf{A}^{2t},$$

$$\mathbf{A}^{2t+2} = \mathbf{A}^{2t+1}s^{t+1} = r^{t+1}\mathbf{A}^{2t}s^{t+1}$$

where

$$r_i^{t+1} = \frac{\overline{u}_i}{\sum_{j=1}^{h} a_{ij}^{2t}}$$

and

$$s_j^{t+1} = \frac{\overline{v}_j}{\sum_{i=1}^{g} a_{ij}^{2t+1}}$$

A.2 The SoDaP program

In this section, the functional design of a SDP Application is discussed. The functional design describes which steps have to be taken to generate a set of safe tables from a microdata file. Also the implementation of the testing program SoDaP is discussed. Finally, the role of the MOSEK optimizer is looked into.

A.2.1 Functional design

The functional design of the testing program SoDaP is shown in Figure A.1 on page 79. When using the testing program, first the metadata file has to be specified. The metadata is read, and the user can input information about which variables should be used for specifying tables, and which variables should be treated as response variables. Then, using this information, the microdata file is read and response variables are stored as floating points, while the other variables are seen as categorical variables and are stored as integers. It is assumed that the data fits into memory. If this is not the case, the program terminates, and less variables and/or less records should be read from the microdata. After the data is read into memory, tables can be specified, and each specified table will be added to the set of tables. When done specifying tables, the method of adding noise can be specified. These methods are the methods proposed in chapter 3.

The perturbation strategies do not guarantee safe tables, so this has to be checked. If there is no disclosure risk, the set of tables is declared safe and is released. Also the assigned weight factors are saved and a disclosure report is written. However, if the checking mechanism indicates a negative outcome, there are several possibilities:

1. Some "conventional" (table level) safety measures are applied, such as global recoding. This can be done interactively, as the program can indicate in which cells in which tables the problems arise.

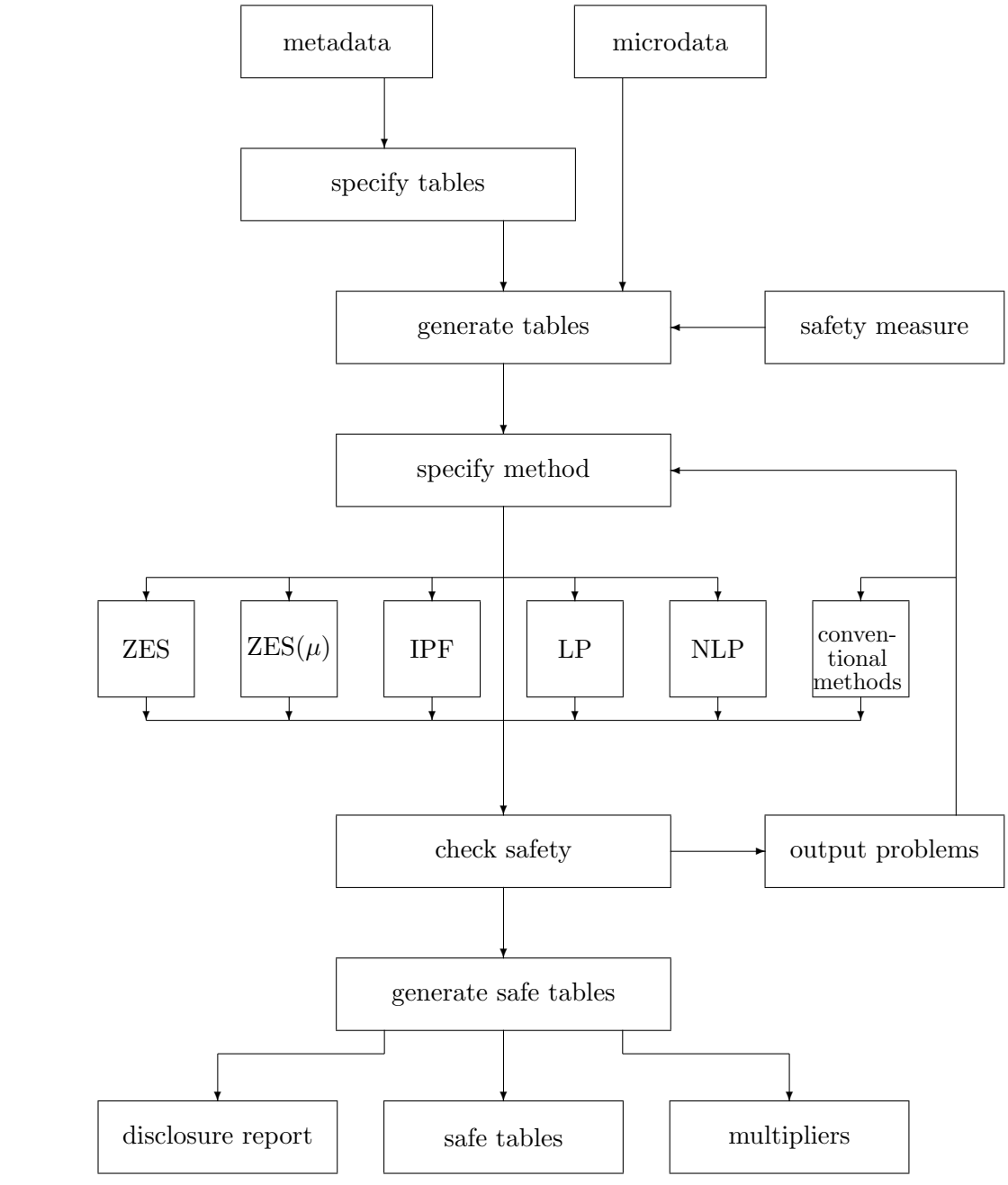


Figure A.1: Functional design of SoDaP

2. Another perturbation strategy can be applied.

3. The set of predefined tables can be adjusted, given the problem indication by the checking mechanism

The checking mechanism is sketched in Section 6.2.

A.2.2 Data structures

The testing program SoDaP was implemented using the C++ programming language. The data structures used in the implementation are chosen in a very straightforward way. Respondents are modeled by the class **Respondent**. A **Respondent** object contains all information found in the corresponding record of the microdata file, the perturbation multiplier, and interface functions for accessing the information of a **Respondent**. Typically, these are used for returning perturbed or real values on some variable such as measure of size, profit, etc.

The **Table** object models tables that are generated from the microdata. **Tables** are constructed of **Cells**, which contain the aggregate value on some response variable for some combination of the table spanning variables. Each **Cell** knows which **Respondents** contribute to it, and the advantage of this construction is that after assigning noise factors the tables do not have to be build all over again. This saves a lot of time, especially when the microdata file is large and many tables are generated from it. Also, **Cells** provide interface functions to access **Cell** attributes, such as original value, perturbed value, desired value, sensitivity and number of contributors. **Cells** also contain member functions to evaluate their sensitivity, compute desired cell totals, and to compute the amount of perturbation after applying multiplicative noise.

Tables contain information concerning **Table** spanning variables, number of rows and columns, and parameters of the safety measures used. These parameters are not the same for each table, as the properties of each specific table imply their own disclosure risks. **Table** interface functions provide access to **Table** member data and to the interface and member functions of individual **Cells**, given row and columns indices.

The entire problem is modeled by the **Problem instance** class, which contains a list of **Respondents**, which is actually the microdata, and con-

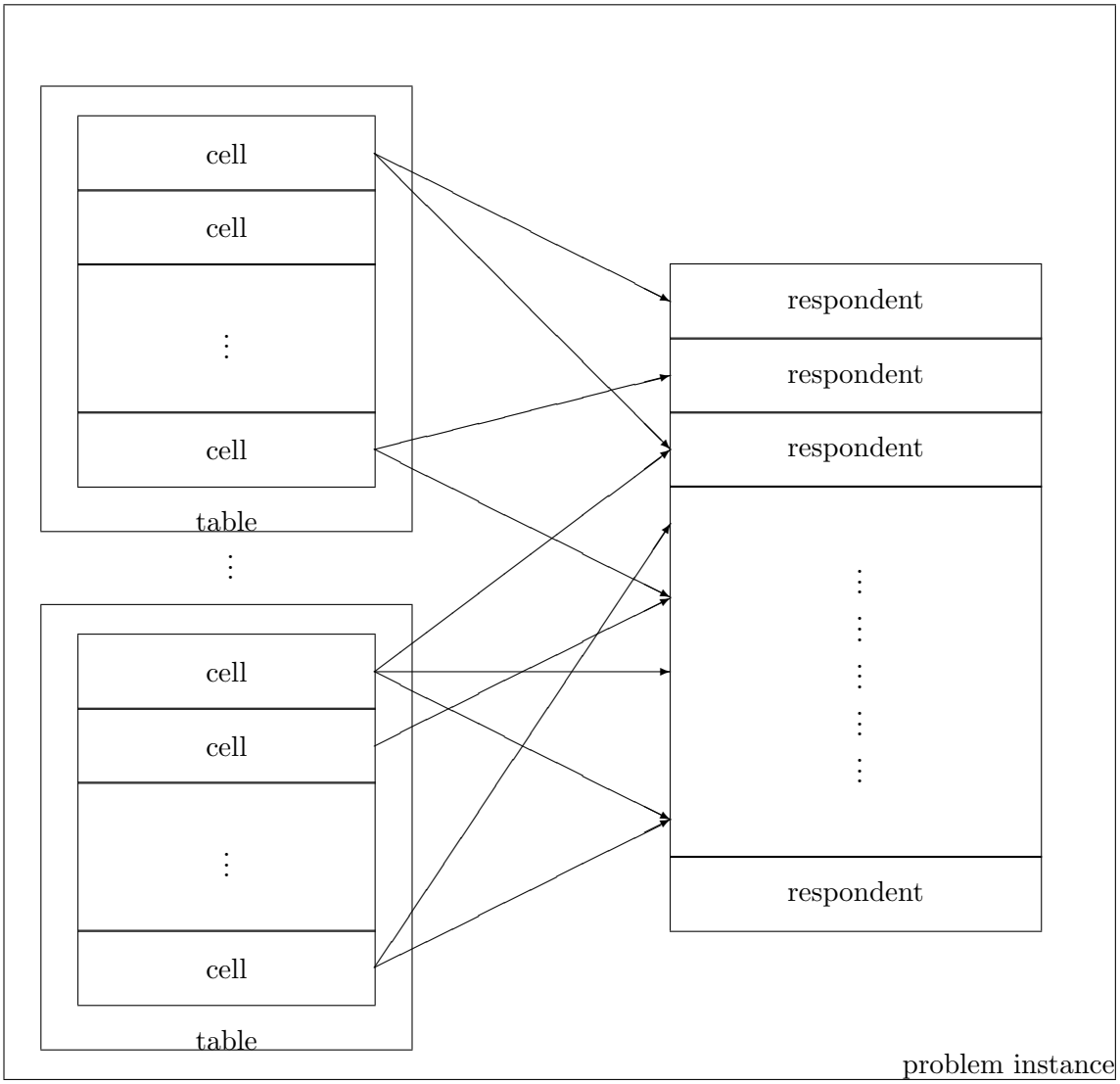


Figure A.2: Data structures

tains a list of **Tables**, which are the tables generated from the microdata that are to be published. The **Problem instance** object provides the perturbation methods of chapter 3, and provides interface functions to let the user specify the datafiles and the tables that are to be generated. Also it contains functions for showing tables on the screen and to save them to file.

Each cell has a list of pointers to respondents contributing to them. Each respondent contributes once to each table, but several cells in one table may have a connection to the same respondent. This is because if a respondent contributes to a cell in some table, it also contributes to the marginal cells of that table. Hence a respondent can be referenced several times from the same table. Also a respondent may be connected to several tables, as he may contribute to more than one table.

A.2.3 The MOSEK optimizer

To solve the optimization problems of Section 5.2 the software package MOSEK is used (see the MOSEK user's manual [2] or www.MOSEK.com). MOSEK solves linear, quadratic, and quadratically convex constrained optimization problems. To this end, MOSEK provides an interior point optimizer and a primal simplex optimizer. Also, MOSEK is designed to handle sparse and large-scaled problems, which is very appropriate for the problems of interest in this report. The interior point method implemented in MOSEK is the homogeneous and self-dual algorithm. For details on the implementation see [1]. Interior points methods are especially appropriate for problems with a large number of variables and constraints, and for this kind of problems they appear to be superior to the simplex method. Most interior point methods have polynomial complexity. Moreover, they can also solve nonlinear programming problems, while the simplex method can only solve LP problems.

The problem can be provided to MOSEK through an input file, or through the MOSEK Application Program Interface, the API. This API can be called directly from C++. In any case, the input parameters have to be inputted according to a model of the form

$$\tfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} q_{ij}^{0} x_i x_j + c_j x_j + c^f$$

subject to the functional constraints

$$l_k^c \leq \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij}^k x_i x_j + a_{kj} x_j \leq u_k^c \qquad k = 0, \ldots, m$$

and the bounds

$$l_j^x \leq x_j \leq u_j^x \qquad j = 0, \ldots, n$$

Also, instead of inputting the predefined quadratic terms $q_{ij} x_i x_j$, it is possible to input some nonlinear, convex function directly into the API. In this case, the API also needs information concerning the gradient and the Hessian of the inputted function. When using the ℓ_2-problem formulation of Section 5.2, the nonlinear function $f(w)$ is

$$f(w) = \sum_{i=1}^{|C|} \left(b_i - \sum_{j=1}^{|R|} m_j X_{ij} \right)^2 = \sum_{i=1}^{|C|} (e_i)^2$$

using residual e_i for shorthand notation. The gradient $\nabla f(w)$ is a $|R|$ x 1 vector:

$$\nabla f(w) = \begin{bmatrix} -2 \sum_{i=1}^{|C|} e_i X_{i1} \\ -2 \sum_{i=1}^{|C|} e_i X_{i2} \\ \vdots \\ -2 \sum_{i=1}^{|C|} e_i X_{i|R|} \end{bmatrix}$$

Finally, the Hessian is a $|R|$ x $|R|$ matrix that does not depend on w, so it only needs to be computed once:

$$\nabla^2 f(w) = \begin{bmatrix} 2 \sum_{i=1}^{|C|} X_{i1} X_{i1} & \cdots & 2 \sum_{i=1}^{|C|} X_{i1} X_{i|R|} \\ \vdots & \ddots & \vdots \\ 2 \sum_{i=1}^{|C|} X_{i|R|} X_{i1} & \cdots & 2 \sum_{i=1}^{|C|} X_{i|R|} X_{i|R|} \end{bmatrix}$$

As the number of respondents in the problem may be in the order of several thousands, n x n may get very large. For reasons of efficiency, MOSEK only stores the lower triangular part of the Hessian, as it assumes it is symmetrical, and furthermore MOSEK only stores the nonzero elements.

Entry k,l of the Hessian represents the cells in which respondent k and respondent l appear together, which means that if respondent k never occurs in a cell where respondent l also occurs, then entry k,l is zero, and doesn't have to be stored.

Bibliography

[1] Andersen, E.D., Y. Ye, 1998, A Computational Study of the Homogeneous Algorithm for Large-scale Convex Optimization, *Computational Optimization and Applications* 10, 243-269, Kluwer Academic Publishers.

[2] Andersen, E.D., 2000, *The MOSEK Base system and Application Program Interface version 1.3*, User's Manual, EKA Consulting ApS.

[3] Bacharach, M., 1970, *Biproportional Matrices & Input-Output Change*, Cambridge University Press.

[4] Cheney, W., D. Kincaid, 1994, *Numerical Mathematics and Computing*, Brooks / Cole Publishing Company.

[5] Cochran, W.G., 1977, *Sampling Techniques*, 3rd ed., John Wiley & Sons.

[6] Cox, Lawrence H., 1981, Linear Sensitivity Measures in Statistical Disclosure Control, *Journal of Statistical Planning and Inference* 5, 153-164.

[7] Evans, T., L. Zayatz, J. Slanta, 1996, *Using Noise for Disclosure Limitation of Establishment Tabular Data*, U.S. Census Bureau.

[8] Fagan, W.T., B. Greenberg, 1985, *Algorithms for making tables additive: raking, maximum likelihood, and minimum chi-square*, US Bureau of the Census, SRD/RR-85/15.

[9] Helmpecht, B., D. Schackis, 1996, *Manual on Disclosure Control Methods*, Eurostat, Office for Official Publications of the European Communities.

[10] Hundepool et al., 1998, *μ-ARGUS user's manual*, version 3.0, Department of Statistical Methods, Statistics Netherlands.

[11] Hundepool et al., 1998, *τ-ARGUS user's manual*, version 2.0, Department of Statistical Methods, Statistics Netherlands.

[12] Kooiman, P., J. Nobel, L. Willenborg, 1999, Statistical data protection at Statistics Netherlands, *Netherlands Official Statistics*, volume 14.

[13] Kooiman, P., L. Willenborg, J. Gouweleeuw, 1997, *PRAM: a method for disclosure limitation of microdata*, Statistics Netherlands.

[14] Law, A.M., W.D. Kelton, 1991, *Simulation Modeling and Analysis*, 2nd edition, McGraw-Hill.

[15] Miller, R.E., P.D. Blair, 1985, *Input-Output Analysis, Foundations and Extensions*, Prentice-Hall.

[16] *Report on Statistical Disclosure and Disclosure-avoidance Techniques*, 1978, Subcommittee on diclosure-avoidance techniques of the Federal committee on statistical methodology, Office of Federal Statistical Policy and Standards, U.S. Department of Commerce.

[17] Ross, S.M., 1997, *Introduction to probability models*, 6th edition, Academic Press.

[18] Willenborg, L., T. de Waal, 1996, *Statistical Disclosure Control in Practice*, Lecture Notes in Statistics, vol 111, Springer - Verlag, New York.

[19] Zayatz, L., B.T. Evans, R. Moore, 1998, *New directions in disclosure limitation at the Census Bureau*, US Bureau of the Census.

[20] Zayatz, L., P. Massell, P. Steel, 1999, Disclosure limitation practices and research at the US Census Bureau, *Netherlands Official Statistics*, volume 14.

[21] Zayatz, L., P. Steel, S. Rowland, 1999, *Disclosure limitation for Census 2000*, US Bureau of the Census.