PERGAMON Computers and Mathematics with Applications 38 (1999) 13–32

An International Journal
computers & mathematics
with applications

www.elsevier.nl/locate/camwa

Resampling for Statistical Confidentiality in Contingency Tables

J. DOMINGO-FERRER* AND J. M. MATEO-SANZ
Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
jdomingo@etse.urv.es

(Received March 1998; revised and accepted June 1999)

Abstract—Resampling schemes, and especially the bootstrap method, were proposed as a subclass of perturbation methods to ensure statistical confidentiality in statistical databases. Later, a method based on bootstrapping was presented to achieve the more specific task of anonymising contingency tables. In this paper, we argue that the latter proposal is either inefficient from a computational point of view or insecure due to a high disclosure risk. For illustration, we show that this bootstrap-based procedure for contingency tables can be emulated and outperformed by a cell-oriented random perturbation method, whose complexity can be theoretically quantified. For a given disclosure risk, our cell-oriented perturbation method is more efficient. For a given computational complexity, our cell-oriented method exhibits a lower disclosure risk. More generally, it can be concluded that the very principle of resampling precludes the design of contingency table anonymisation schemes simultaneously providing security, computational efficiency, and data quality. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords—Statistical disclosure control, Contingency tables, Resampling methods, Random perturbation methods, Statistical databases.

1. INTRODUCTION

When statistical data are published as a printed report or are obtained following a set of queries to a statistical database, statistical confidentiality must be guaranteed. Disclosure control methods attempt to keep individual information anonymous when releasing macrodata (contingency tables or any other statistic) and microdata (individual records). Practical disclosure control methods have followed two basic approaches (see [1]).

QUERY RESTRICTION. This approach consists of five general methods: query-set-size control, query-set overlap control, auditing, cell suppression, and partitioning. The first three methods are intended for on-line statistical databases, whereas the latter two methods are used mainly for off-line statistical data (especially contingency tables).

PERTURBATION. Perturbation methods consist of distorting figures by adding a perturbation to them. Perturbations can be applied directly to data (data perturbation) or just to the answers to user queries while leaving data unchanged (output perturbation). Data perturbation

*Author to whom all correspondence should be addressed at Departament d'Enginyeria Informàtica i Matemàtiques, Escola Tècnica Superior d'Enginyeria, Universitat Rovira i Virgili, Autovia de Salou s/n, E-43006 Tarragona, Catalonia, Spain.
This work is partly supported by the Spanish CICYT under Grant Number TEL98-0699-C02-02.

0898-1221/1999/$ - see front matter © 1999 Elsevier Science Ltd. All rights reserved. Typeset by AMS-TEX
PII: S0898-1221(99)00281-3

methods include probability distribution and fixed-data perturbation methods. Output perturbation methods include varying-output perturbation, rounding, and random-sample methods. Resampling methods are a generalisation of random-sample methods [2].

For on-line statistical databases, output perturbation methods are preferred to data perturbation methods, because the latter suffer from the bias problem [3]. However, here we are going to deal with protection of off-line contingency tables and both approaches are equivalent for off-line disclosure control. One important security evaluation criterion for disclosure control methods is the probability of exact disclosure of an individual attribute; for contingency tables, this means that small frequencies should be especially protected. For a more detailed description of the existing disclosure control methods and their evaluation criteria, see [4–6].

In [2], resampling was shown to be a principle generating a subclass of output-perturbation methods for disclosure control. Specifically, Denning's random-sample method [7] was extended, and the bootstrap and the jack-knife resampling techniques were also used. Using resampling methods is attractive because they are well characterised from a statistical point of view. This allows a pretty straightforward evaluation of their security properties.

In [8], a practical procedure for anonymisation of contingency tables was proposed which relies on the bootstrap method. In this paper, we argue that this resampling method can be outperformed by a cell-oriented random perturbation method. The reason for this lack of performance is the very nature of resampling. In Section 2, we recall Heer's bootstrap procedure and its security properties. In Section 3, a new cell-oriented perturbation method that emulates the bootstrap procedure of Section 2 is presented and its quality and security are discussed. In Section 4, a complexity analysis of both methods is done, which together with their security properties show that the bootstrap procedure is outperformed by the cell-oriented method. For a given disclosure risk, the latter is more efficient and for a given computational complexity, the former exhibits a higher disclosure risk. Section 5 is a conclusion containing some generalisations about resampling methods versus cell-oriented methods. The Appendix contains some auxiliary calculations.

For simplicity of notation, two-way contingency tables will be considered in what follows. However, generalising the methods and concepts below to contingency tables of higher dimension (multiway tables) is not difficult.

2. PREVIOUS WORK ON ANONYMISATION OF CONTINGENCY TABLES BY RESAMPLING

In [8], Heer presented a method for anonymising contingency tables based on resampling. The resampling procedure used is the bootstrap. We next recall the essentials of the proposal and its security properties.

Assume that microdata $z_1, \ldots, z_n$ are aggregated to elaborate macrodata in the form of a contingency table $\mathbf{x}$ with I rows and J columns, which is produced according to certain specifications. Let x_{ij} be the original frequency in the i^{th} row and j^{th} column. In order to produce an anonymised table $\mathbf{x}'$, a bootstrap sample $z'_1, \ldots, z'_n$ is obtained by drawing from the original data $z_1, \ldots, z_n$, n times and with replacement. The bootstrap table $\mathbf{x}'$ thus obtained is an estimate of the original table $\mathbf{x}$ and does not allow anyone to get any precise information of $\mathbf{x}$, due to its random error.

The main features of a bootstrap table are as follows.

- The overall frequency is preserved, since $\sum_{i,j} x_{ij} = \sum_{i,j} x'_{ij} = n$.
- The whole table $\mathbf{x}'$ can be viewed as a sample drawn from a multinomial distribution with parameters n and x_{ij} for all i and j. Each individual bootstrap frequency x'_{ij} can be viewed as a value drawn from a variable X'_{ij} having a binomial distribution where n is the number of trials and $p = x_{ij}/n$ is the success probability per trial. Thus, $E(X'_{ij}) = np = x_{ij}$ and $\mathrm{Var}(X'_{ij}) = np(1-p) = x_{ij}(1 - x_{ij}/n)$. Therefore, $\mathbf{x}'$ is an unbiased estimate of $\mathbf{x}$.

- An original frequency $x_{ij} = 0$ is preserved by default, i.e., $x_{ij} = 0$ implies $x'_{ij} = 0$. If this is undesirable, then a compensated perturbation method could be used on the original table before bootstrapping, in order to replace zero frequencies with small frequencies.

2.1. Ensuring the Quality of a Bootstrap Table

Although, in general, a bootstrap table $\mathbf{x}'$ closely approximates the original table $\mathbf{x}$, it is possible for a given $\mathbf{x}'$ to be very different from $\mathbf{x}$. One way to control the maximum deviation of a bootstrap frequency from the original frequency is to require that the standardised bootstrap frequencies stay below a given boundary $S > 0$. This means that the following quality condition has to be met:

$$-S \leq QC = \frac{x'_{ij} - x_{ij}}{\sqrt{x_{ij}(1 - x_{ij}/n)}} \leq S, \qquad 1 \leq i \leq I, \quad 1 \leq j \leq J, \quad x_{ij} > 0. \tag{1}$$

Equivalently, condition (1) requires x'_{ij} to lie in a closed interval $[l(x_{ij}), u(x_{ij})]$ around x_{ij} whose width depends on x_{ij} and also on S. Notice that x_{ij} being a frequency, $l(\cdot)$ and $u(\cdot)$ can be taken as integer functions

$$l(x_{ij}) = x_{ij} - \left\lfloor S\sqrt{x_{ij}\left(1 - \frac{x_{ij}}{n}\right)} \right\rfloor,$$
$$u(x_{ij}) = x_{ij} + \left\lfloor S\sqrt{x_{ij}\left(1 - \frac{x_{ij}}{n}\right)} \right\rfloor,$$

where $\lfloor z \rfloor$ is the greatest integer less than or equal to z. As a bootstrap table is being generated, if $u(x_{ij})$ is exceeded for some cell x'_{ij}, then the table is discarded and a new table generation is started. Also, when the table has been completely generated, it may be discarded if some x'_{ij} stays below $l(x_{ij})$. Remark that checking the lower limits $l(x_{ij})$ can only be efficiently done once the resampling process is finished: each new draw can cause any table cell to be incremented.

As $n \to \infty$, the binomial distribution of a bootstrap frequency X'_{ij} tends to become a normal distribution. In this case, QC can be viewed as a random variable following a standard normal distribution. Thus, if S is the $\alpha/2$ percentage point of the $N(0,1)$ distribution, then the quality condition specified by inequality (1) is met with a probability of about $1 - \alpha$ for a single cell. However, the probability that all cells of a bootstrap table meet the quality condition with just one table generation is much smaller (see Section 4.1).

It has been suggested that the average of M bootstrap tables is more likely to meet the quality condition. In this case, the (i, j) cell is computed as

$$x_{ij}^M = \frac{1}{M} \sum_{m=1}^{M} x'_{ij}(m).$$

Provided that M is not too large, this approach saves computation by reducing the probability of table regeneration ("wasting" computation is more unlikely). In [8], Heer recommends choosing odd values for M. As discussed in Section 4.1, the value M_{IJ} of M that minimises the expected computation grows logarithmically in the table size IJ: for example, for $IJ = 50$, one has $M_{50} = 3$, but for $IJ = 100$, the value is $M_{100} = 5$.

2.2. Security of the Bootstrap Table

In order to evaluate the security provided by the method, the conditional distribution of the original frequencies given the bootstrap frequencies is examined in [8]. The derivation of such distribution is pretty straightforward from the properties of the bootstrap method. A critical issue to statistical confidentiality is that *small frequencies* (≤ 3) be sufficiently disguised (see [1]). This

prevents inference of individual attributes. The probability that a released bootstrap frequency is identical to the original frequency is approximately given in Table 1 for different values of S and M and for $n \geq 1000$ (the reported probability is practically independent of n when $n \geq 1000$). It can be seen that the probability of exact disclosure $P(X_{ij} = k \mid X_{ij}^M = k)$ increases with the number M of averaged tables. Already for $M = 5$, when an observer sees a 1 in the released table, he knows this is a real 1 with a 76% probability! Thus, the method has a clear limitation.

- If $M = 1$, then the computational complexity is high, as several table generation attempts may be needed to meet the quality condition (see Section 2.1.).
- If $M > 1$, then the level of protection against disclosure of small frequencies is low.

In Section 5, we will conclude that this limitation is inherent to any procedure for statistical confidentiality that relies on resampling.

Table 1. Conditional probabilities $P(X_{ij} = k \mid X_{ij}^M = k)$ for $k = 1, 2, 3$.

	Probability								
	Average of ... Bootstrap Tables								
Frequency	$M = 1$			$M = 3$			$M = 5$		
k	$S = 3$	$S = 2$	$S = 1.5$	$S = 3$	$S = 2$	$S = 1.5$	$S = 3$	$S = 2$	$S = 1.5$
1	.40	.42	.47	.65	.65	.66	.76	.76	.76
2	.27	.28	.28	.46	.46	.46	.58	.58	.58
3	.22	.24	.27	.38	.38	.39	.48	.48	.48

3. A NEW PERTURBATION METHOD

In this section, we present a new perturbation method which has the following features.

1. The probability of exact disclosure can be analytically calculated. For small frequencies, it is similar to the probability of exact disclosure of Heer's method in the best case (no averaging, $M = 1$).
2. The overall frequency of the original table is preserved.
3. By construction, the anonymised final table $\mathbf{x}''$ is an asymptotically unbiased estimate of the original table $\mathbf{x}$.
4. An original frequency $x_{ij} = 0$ is preserved, i.e., $x_{ij} = 0$ implies $x''_{ij} = 0$.
5. Anonymisation of a table does not rely on resampling microdata, but on generating binomial random perturbations.
6. Quality of the disclosure-protected frequencies is ensured by requiring them to meet inequality (1).
7. Already for moderately large tables, the computational complexity is lower than for the method of Section 2 without averaging ($M = 1$). Less computation is wasted when generating the disclosure-protected table because the new procedure is cell-oriented: the quality condition is checked after each cell generation, not on a table basis.
8. Unlike for the bootstrap method and many compensated perturbation methods (see [4,5] for a survey), the computational complexity of our procedure can be theoretically quantified without resorting to simulation.

Thus, the new proposal *emulates* the bootstrap scheme discussed in the previous section by providing the same quality features. However, it will be shown that computational complexity is reduced without degrading security.

3.1. Description of the Method

Given an original contingency table $\mathbf{x}$, all its cells are randomly perturbed to obtain a new table $\mathbf{x}'$ (*perturbation stage*). Then some compensations are performed on $\mathbf{x}'$ and a final anonymised table $\mathbf{x}''$ is obtained (*compensation stage*).

Denote by x_{ij} the value in the cell formed by the i^{th} row and the j^{th} column of table $\mathbf{x}$. Denote by x'_{ij} the corresponding value in table $\mathbf{x}'$. Now x'_{ij} is obtained by *sampling a binomial random variable* X'_{ij} with parameters n and p, where n is the overall frequency count (number of individual microdata) of table $\mathbf{x}$ and p is x_{ij}/n.

In order to preserve data quality, table $\mathbf{x}'$ should not differ too much from table $\mathbf{x}$. We impose on table $\mathbf{x}'$ the same quality condition given by inequality (1). For the sake of simplicity, the x'_{ij} are sampled independently of each other. If the quality condition is superimposed, the values x'_{ij} obtained through simulation should be integers lying in a closed interval $[l(x_{ij}), u(x_{ij})]$ around x_{ij} whose width depends on the value x_{ij} and also on the significance level α. If the sampled x'_{ij} does not lie in the above interval, it is discarded and a new x'_{ij} is generated; the procedure iterates until the quality condition is met.

There is a drawback associated with simulating bootstrap frequencies by repeatedly and independently sampling a binomial random variable: namely, the overall frequency count n of the original table $\mathbf{x}$ *is not preserved* in general by $\mathbf{x}'$. This problem does not arise when a *real* bootstrap sample is drawn (Section 2), because the bootstrap sample size is the same as the original sample size. However, preserving n requires a compensation stage involving little extra computation. It suffices to maintain two additional counters for each cell. The first initially contains $x'_{ij} - l(x_{ij})$, that is the number of integers comprised between x'_{ij} and its lower bound $l(x_{ij})$ resulting from the quality condition; the second initially contains $u(x_{ij}) - x'_{ij}$, that is the number of integers between x'_{ij} and its upper bound $u(x_{ij})$ resulting from the quality condition. Then, the following two cases are considered.

1. If the overall frequency count n' of table $\mathbf{x}'$ is greater than n, then $n' - n$ frequency units should be subtracted from table $\mathbf{x}'$. These units are subtracted from cells in the set C of those having their first counter greater than zero. Specifically, the following is done $n' - n$ times.
   (a) Sample C according to a discrete distribution giving each cell in C a probability proportional to its first counter.
   (b) Subtract one unit from the value of the chosen cell and also from its first counter.
2. If the overall frequency count n' of table $\mathbf{x}'$ is less than n, then $n - n'$ frequency units must be added to $\mathbf{x}'$. These units are added to cells in the set D of those having their second counter greater than zero. Specifically, the following is done $n - n'$ times.
   (a) Sample D according to a discrete distribution giving each cell in the set a probability proportional to its second counter.
   (b) Add one unit to the value of the chosen cell and subtract one unit from its second counter.

Call the table resulting from the above compensation stage $\mathbf{x}''$. In the next two subsections, the data quality and the security offered by our method will be analysed in detail.

NOTE. PRESERVATION OF MARGINAL COUNTS. The new method can be used just in the same way explained above to preserve *either* the row marginals *or* the column marginals (for multidimensional tables, marginals in one of the dimensions can be preserved). For instance, to preserve row marginals, one should use the method independently for each row. In other words, each row of an $I \times J$ table should be dealt with as an $I \times 1$ table to which the method in the paper is to be applied. In this way, the overall frequency is preserved, which for an $I \times 1$ table is the row marginal. By construction, preservation of all row marginals leads to preservation of the overall frequency in the $I \times J$ table. A similar strategy could be used to preserve only column marginals. What cannot be achieved by our method is simultaneous preservation of row *and* column marginals (for multidimensional tables, simultaneous preservation of marginals in two or more dimensions is not possible). It will be shown below that the method is unbiased, so even if marginals are not preserved exactly, the expected values of marginals in $\mathbf{x}''$ are marginals

in $\mathbf{x}$. Tightening the quality condition (1) by decreasing S helps reduce the variance of the nonpreserved marginals, but it also reduces data protection (see Section 3.3).

NOTE. LINKED TABLES. If we have a set of tables sharing *only one* dimension, then our method can be used to protect them consistently. The reason is that we can preserve marginals in *one* of the dimensions (see Note 1), which would be the dimension shared by the tables. If the set of tables share more than one dimension, then it is not possible to protect them consistently with our method. Note that the same problem happens with quite up-to-date statistical disclosure control packages for tables. In the best case and only for some packages, iterative procedures to be used with *cell suppression* (not resampling!) methods are available to protect sets of linked tables (see the comparative study [9]).

3.2. Data Quality Provided by the Method

By construction, table $\mathbf{x}''$ also satisfies quality condition (1) met by $\mathbf{x}'$, because at the compensation stage, no subtraction will be done from a cell having reached $l(x_{ij})$ and no addition will be done to a cell having reached $u(x_{ij})$.

The impact of the perturbation stage on each particular cell can be characterised by the distribution of X'_{ij} for a fixed x_{ij}. The random variable X'_{ij} follows a binomial distribution with parameters n and $p = x_{ij}/n$, but takes values restricted to the closed interval $[l(x_{ij}), u(x_{ij})]$ centred on x_{ij}. Denoting by

$$b(z; n, p) = \binom{n}{z} p^z (1-p)^{n-z},$$

the binomial probability function, we can write

$$P\left(X'_{ij} = x'_{ij} \mid X_{ij} = x_{ij}\right) = \begin{cases} \dfrac{b\left(x'_{ij}; n, x_{ij}/n\right)}{\sum\limits_{l(x_{ij}) \le h \le u(x_{ij})} b\left(h; n, x_{ij}/n\right)}, & \text{if } l\left(x_{ij}\right) \le x'_{ij} \le u\left(x_{ij}\right), \\ 0, & \text{elsewhere.} \end{cases} \tag{2}$$

The random variable $N' = \sum_{i,j} X'_{ij}$ takes values restricted to the interval $[L, U]$, where $L = \sum_{i,j} l(x_{ij})$ and $U = \sum_{i,j} u(x_{ij})$ are parameters of the original table $\mathbf{x}$ (just like n). If the X'_{ij} were unrestricted binomial variables, then by the central limit theorem N' would be approximately distributed as a truncated $N(\sum_{i,j} x_{ij}, \sum_{i,j} x_{ij}(1 - x_{ij}/n))$, or equivalently, $N(n, n - \sum_{i,j} x_{ij}^2/n)$ for a sufficiently large table size IJ. Since the X'_{ij} are restricted to their quality intervals centred on x_{ij}, the central limit theorem can be used to say that N' is approximately distributed inside the interval $[L, U]$ as a truncated $N(n, \sigma^2_{N'})$, where $\sigma^2_{N'} \le n - \sum_{i,j} x_{ij}^2/n$.

The impact of the compensation stage on each particular cell can be characterised by the distribution of X''_{ij} once the values of X_{ij} and X'_{ij} have been fixed. Two cases must be considered for deriving $P(X''_{ij} = x''_{ij} \mid X'_{ij} = x'_{ij}, X_{ij} = x_{ij})$. If $x''_{ij} - x'_{ij} \ge 0$, this means that the overall frequency n' of table $\mathbf{x}'$ is being increased to reach n (the frequency of the original table $\mathbf{x}$ which is to be preserved in $\mathbf{x}''$). In $n - n'$ draws, the probability of drawing exactly $x''_{ij} - x'_{ij}$ times from the cell's second counter (whose initial value is $u(x_{ij}) - x'_{ij}$) can be computed using a hypergeometric distribution. Denoting by

$$h(z; N, n, k) = \frac{\binom{k}{z} \cdot \binom{N-k}{n-z}}{\binom{N}{n}}$$

the hypergeometric probability function, we can write

$$\begin{aligned} &P\left(X''_{ij} = x''_{ij} \mid X'_{ij} = x'_{ij}, X_{ij} = x_{ij}\right) \\ &= \sum_{n'=L}^{n-\left(x''_{ij} - x'_{ij}\right)} P\left(N' = n' \mid X'_{ij} = x'_{ij}, X_{ij} = x_{ij}\right) h\left(x''_{ij} - x'_{ij}; U - n', n - n', u\left(x_{ij}\right) - x'_{ij}\right). \end{aligned} \tag{3}$$

Three comments on equation (3) are in order. First, $U-n'$ is the sum of the second counters for all cells. Second, the fact that n' is a parameter of the hypergeometric probability function explains that this function is multiplied by $P(N'=n' \mid X'_{ij}=x'_{ij}, X_{ij}=x_{ij})$. The distribution of N' has been shown to be $N(n,\sigma^2_{N'})$, and therefore, $N \mid x'_{ij}, x_{ij}$ is distributed as $N(n+x'_{ij}-x_{ij}, \sigma^2_{N'|x'_{ij},x_{ij}})$. Third, the summation extends over all possible values n' given $X'_{ij}=x'_{ij}$ and $X''_{ij}=x''_{ij}$: the lowest possible value is L and the highest one is $n-(x''_{ij}-x'_{ij})$, because if $x''_{ij}-x'_{ij}>0$, this means that n' is to be increased to reach n by at least $x''_{ij}-x'_{ij}$ units (and thus, $n' \leq n-(x''_{ij}-x'_{ij})$).

In the $x''_{ij}-x'_{ij}<0$ case, the overall frequency n' of table $\mathbf{x}'$ is being decreased to reach n. In $n'-n$ draws, the probability of drawing exactly $x'_{ij}-x''_{ij}$ times from the cell's first counter (whose initial value is $x'_{ij}-l(x_{ij})$) can also be computed using a hypergeometric distribution. In a way similar to the $x''_{ij}-x'_{ij}\geq 0$ case and bearing in mind that $n'-L$ is the sum of the first counters for all cells, we obtain

$$P\left(X''_{ij}=x''_{ij} \mid X'_{ij}=x'_{ij}, X_{ij}=x_{ij}\right) \tag{4}$$

$$= \sum_{n'=n+\left(x'_{ij}-x''_{ij}\right)}^{U} P\left(N'=n' \mid X'_{ij}=x'_{ij}, X_{ij}=x_{ij}\right) h\left(x'_{ij}-x''_{ij}; n'-L, n'-n, x'_{ij}-l\left(x_{ij}\right)\right).$$

The next two lemmas give bounds for frequencies in the final table $\mathbf{x}''$.

LEMMA 1. UPPER BOUND FOR CELLS OF $\mathbf{x}''$. *For all cells in table* $\mathbf{x}''$, *it holds that* $x''_{ij}\leq n$.

PROOF. The values resulting from the perturbation stage are x'_{ij} such that $0\leq x'_{ij}\leq n$ because they are sampled from a binomial distribution with parameters n and x_{ij}/n. Getting $x''_{ij}>n$ is only conceivable if compensating means adding (case $n'<n$). In this case, we have the following.

1. Before the compensation stage, all cell values are $0\leq x'_{ij}\leq n$.
2. Only additions are done during the compensation stage (case $n'<n$).
3. After the compensation stage, the overall frequency must be n.

From the above three facts, it follows that no cell value can be $x''_{ij}>n$ after the compensation stage (this would imply the existence of negative cells, but these cannot appear if just additions are done). ∎

It has been shown that $x''_{ij}\leq n$ always holds. To avoid obtaining out-of-range frequencies, one must ensure also that $x''_{ij}\geq 0$. This holds provided that the quality criterion is not too loose.

LEMMA 2. LOWER BOUND FOR CELLS OF $\mathbf{x}''$. *If* $S\leq \max(2,\sqrt{x_{ij}})$, *then* $x''_{ij}\geq 0$.

PROOF. By construction, the compensation procedure always yields x''_{ij} such that $l(x_{ij})\leq x''_{ij}\leq u(x_{ij})$. Thus, it suffices to show that $l(x_{ij})\geq 0$.

Now, if $S\leq\sqrt{x_{ij}}$, then

$$S\sqrt{x_{ij}\left(\frac{1-x_{ij}}{n}\right)}<S\sqrt{x_{ij}}\leq x_{ij}.$$

Since the previous inequality is strict and x_{ij} is an integer, we can infer

$$l\left(x_{ij}\right)=x_{ij}-\left\lfloor S\sqrt{x_{ij}\left(1-\frac{x_{ij}}{n}\right)}\right\rfloor\geq 1.$$

We will finally show that for $x_{ij}<4$, one can take $S=2$ and still $l(x_{ij})\geq 0$. For $x_{ij}=0$, any S will do:

$$l(0)=0-\left\lfloor S\sqrt{0\left(1-\frac{0}{n}\right)}\right\rfloor=0.$$

For $x_{ij}=1$ and $S=2$, one has $2\sqrt{1(1-1/n)}<2$, and therefore,

$$l(1)=1-\left\lfloor 2\sqrt{1\left(1-\frac{1}{n}\right)}\right\rfloor\geq 0.$$

For $x_{ij} = 2$ and $S = 2$, one has $2\sqrt{2(1-2/n)} < 2\sqrt{2} < 3$, and therefore,

$$l(2) = 2 - \left\lfloor 2\sqrt{2\left(1 - \frac{2}{n}\right)} \right\rfloor \geq 0.$$

For $x_{ij} = 3$ and $S = 2$, one has $2\sqrt{3(1-3/n)} < 2\sqrt{3} < 4$, and therefore,

$$l(3) = 3 - \left\lfloor 2\sqrt{3\left(1 - \frac{3}{n}\right)} \right\rfloor \geq 0. \qquad \blacksquare$$

If a table does not contain small frequencies $x_{ij} < 4$, then disclosure control methods are probably not needed, because only tables with small frequencies (usually less than or equal to three) need be protected according to standard disclosure rules in official statistics (see [4]). Thus, in practice, the above proposition means that one should take $S \leq 2$, i.e., the quality criterion should not be too loose. This is not as restrictive as it may sound, because loosening the quality criterion is intrinsically undesirable and has other drawbacks (see Note 3). In fact, $S = 2$ is already pretty loose, since S is the $\alpha/2$ percentage point of the $N(0,1)$ distribution and the area under the $N(0,1)$ curve between $-S = -2$ and $S = 2$ is already 0.9544. *In the rest of the article, we assume that a value for S is taken such that $l(x_{ij}) \geq 0$ for all i, j.*

PROPOSITION 1. PRESERVATION OF ZERO FREQUENCIES. *An original frequency $x_{ij} = 0$ is preserved, i.e., $x_{ij} = 0$ implies $x''_{ij} = 0$.*

PROOF. If an original frequency is $x_{ij} = 0$, then x'_{ij} is obtained by sampling a binomial random variable X'_{ij} with parameters n and p, where n is the overall frequency count of table $\mathbf{x}$ and p is $x_{ij}/n = 0$. Therefore, $x'_{ij} = 0$. On the other hand, if $x_{ij} = 0$, the quality condition yields a quality interval $[l(x_{ij}), u(x_{ij})]$, where

$$l(x_{ij}) = x_{ij} - \left\lfloor S\sqrt{x_{ij}\left(1 - \frac{x_{ij}}{n}\right)} \right\rfloor = x_{ij} = 0,$$

$$u(x_{ij}) = x_{ij} + \left\lfloor S\sqrt{x_{ij}\left(1 - \frac{x_{ij}}{n}\right)} \right\rfloor = x_{ij} = 0.$$

As a consequence, the cell's first and second counters are $x'_{ij} - l(x_{ij}) = 0$ and $u(x_{ij}) - x'_{ij} = 0$, which means that a cell with original frequency $x_{ij} = 0$ will never be modified during the compensation stage (at this stage, cells to be modified are chosen with a probability proportional to either their first or second counter). Therefore, $x''_{ij} = x'_{ij} = x_{ij} = 0$. $\blacksquare$

LEMMA 3. ASYMPTOTICAL UNBIASEDNESS OF THE PERTURBATION STAGE. *For nonextreme original frequencies x_{ij}, it holds that $\lim_{n\to\infty} E(X'_{ij}) = x_{ij}$.*

PROOF. X'_{ij} follows a binomial distribution restricted to the closed interval $[l(x_{ij}), u(x_{ij})]$, i.e., its probability function is zero outside the interval and is proportional to the binomial probability function inside the interval. The interval centre is x_{ij}. On the other hand, if x_{ij} is not too close to 0 or n and $n \to \infty$, then a binomial distribution with parameters n and $p = x_{ij}/n$ is well approximated by a normal distribution with mean x_{ij}, which means that the probability function of X'_{ij} tends to be symmetrical about the point x_{ij}. Thus, $\lim_{n\to\infty} E(X'_{ij}) = x_{ij}$. $\blacksquare$

Next, it will be shown that the impact of the compensation stage on a cell is expected to be proportional to the original cell value x_{ij} if $x_{ij} \leq n/2$. Note that in a table with total frequency n, there can be *at most* one cell such that $x_{ij} > n/2$, which is not much if the table size IJ is assumed to be moderately large. Excluding this cell, the impact for the remaining $IJ - 1$ cells is still expected to be proportional to their original values x_{ij}.

PROPOSITION 2. FAIRNESS OF THE COMPENSATION STAGE. *For original cell values in the range $0 \leq x_{ij} \leq n/2$, the compensation $|x''_{ij} - x'_{ij}|$ is expected to be proportional to x_{ij}.*

PROOF. In the range $0 \leq x_{ij} \leq n/2$, the larger x_{ij}, the wider the quality interval $[l(x_{ij}), u(x_{ij})]$, because

$$l(x_{ij}) = x_{ij} - \left\lfloor S\sqrt{x_{ij}\left(1 - \frac{x_{ij}}{n}\right)} \right\rfloor,$$
$$u(x_{ij}) = x_{ij} + \left\lfloor S\sqrt{x_{ij}\left(1 - \frac{x_{ij}}{n}\right)} \right\rfloor.$$

As a consequence, for $x_{ij} \leq n/2$, the larger x_{ij}, the larger the expected values of the first counter $E(X'_{ij}) - l(x_{ij}) \approx x_{ij} - l(x_{ij})$ and the second counter $u(x_{ij}) - E(X'_{ij}) \approx u(x_{ij}) - x_{ij}$ (by Lemma 3, if x_{ij} is not too small, approximations become equalities as $n \to \infty$). This means that as x_{ij} increases toward $n/2$, the probability of the cell being chosen for a compensation step is also expected to increase (cells are chosen for compensation with a probability proportional to either their first or second counter). ∎

Now it can be proven that the method is unbiased asymptotically in n and IJ.

THEOREM 1. ASYMPTOTICAL UNBIASEDNESS OF THE METHOD. *For nonextreme original frequencies x_{ij} and $S \leq \max(2, \sqrt{x_{ij}})$, then*

$$\lim_{n\to\infty, IJ\to\infty} E\left(X''_{ij}\right) = x_{ij}.$$

PROOF. All expectations and probabilities in the following proof are conditioned to $X_{ij} = x_{ij}$, but to simplify the notation, we will not write it explicitly. By Lemma 2, the condition on S ensures that $l(x_{ij}) \geq 0$, and thus, the quality interval $[l(x_{ij}), u(x_{ij})]$ is centred on x_{ij}. Therefore, $l(x_{ij}) = x_{ij} - w_{ij}$ and $u(x_{ij}) = x_{ij} + w_{ij}$. Call $L = n - W$ and $U = n + W$. We can write

$$\begin{aligned}
E\left(X''_{ij}\right) &= \sum_{k''=-w_{ij}}^{w_{ij}} \left(x_{ij} + k''\right) P\left(X''_{ij} = x_{ij} + k''\right) \\
&= \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{\delta=-W}^{W} \sum_{k'=-w_{ij}}^{w_{ij}} \left(x_{ij} + k''\right) \\
&\quad \times P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij} + k'\right) \\
&\quad \times P\left(N' = n + \delta \mid X'_{ij} = x_{ij} + k'\right) P\left(X'_{ij} = x_{ij} + k'\right) \\
&= \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{\delta=-W}^{W} \sum_{k'=-w_{ij}}^{w_{ij}} \left(x_{ij} + k''\right) \\
&\quad \times P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij} + k'\right) \\
&\quad \times P\left(N' - X'_{ij} = n - x_{ij} + (\delta - k')\right) P\left(X'_{ij} = x_{ij} + k'\right).
\end{aligned} \tag{5}$$

Let us analyse the symmetries in the probabilities appearing in the expression for $E(X''_{ij})$ obtained in the above derivation. In Lemma 3, the distribution of X'_{ij} has been shown to become symmetrical about x_{ij} as $n \to \infty$. Therefore, for $0 \leq k' \leq w_{ij}$,

$$\lim_{n\to\infty} P\left(X'_{ij} = x_{ij} + k'\right) = \lim_{n\to\infty} P\left(X'_{ij} = x_{ij} - k'\right). \tag{6}$$

On the other hand, it has been argued at the beginning of this section that by the central limit theorem, N' tends to follow a normal distribution with mean n as $IJ \to \infty$. Also, from Lemma 3, $\lim_{n\to\infty} E(X'_{ij}) = x_{ij}$. Thus, $N' - X'_{ij}$ tends to follow a normal distribution with mean $n - x_{ij}$ as $IJ \to \infty$ and $n \to \infty$. Therefore, for $0 \leq \delta \leq W$ and $0 \leq k' \leq w_{ij}$,

$$\begin{aligned}
&\lim_{n\to\infty, IJ\to\infty} P\left(N' - X'_{ij} = n - x_{ij} + (\delta - k')\right) \\
&= \lim_{n\to\infty, IJ\to\infty} P\left(N' - X'_{ij} = n - x_{ij} - (\delta - k')\right),
\end{aligned} \tag{7}$$

$$\lim_{n\to\infty,IJ\to\infty} P\left(N' - X'_{ij} = n - x_{ij} + (\delta + k')\right) = \lim_{n\to\infty,IJ\to\infty} P\left(N' - X'_{ij} = n - x_{ij} - (\delta + k')\right). \tag{8}$$

The third set of symmetries comes from the fact that $P(X''_{ij} = x_{ij}+k'' \mid N' = n+\delta, X'_{ij} = x_{ij}+k')$ follows a hypergeometric distribution as argued at the beginning of this section. In what follows, $0 \leq k$, $k'' \leq w_{ij}$, and $0 \leq \delta \leq W$. If $k'' \leq k'$, then by analogy with equation (3) and adapting the notation as indicated at the beginning of this proof, one has

$$P\left(X''_{ij} = x_{ij} - k'' \mid N' = n - \delta, X'_{ij} = x_{ij} - k'\right) = h\left(k' - k''; \delta + W, \delta, k' + w_{ij}\right).$$

If $k'' > k'$, the above probability is zero (when $N' = n - \delta$, the compensation step implies increasing the overall frequency and this cannot result in $X''_{ij} = x_{ij} - k'' < X'_{ij} = x_{ij} - k'$). On the other hand, if $k'' \leq k'$, then by analogy with equation (4), one has

$$P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij} + k'\right) = h\left(k' - k''; \delta + W, \delta, k' + w_{ij}\right).$$

If $k'' > k'$, the above probability is zero (when $N' = n + \delta$, the compensation step implies decreasing the overall frequency and this cannot result in $X''_{ij} = x_{ij} + k'' > X'_{ij} + k'$). Thus, we have the symmetry

$$\begin{aligned} &P\left(X''_{ij} = x_{ij} - k'' \mid N' = n - \delta, X'_{ij} = x_{ij} - k'\right) \\ &\quad = P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij} + k'\right) \\ &\quad = h\left(k' - k''; \delta + W, \delta, k' + w_{ij}\right). \end{aligned} \tag{9}$$

Two more symmetries can similarly be found by analogy with equations (3) and (4),

$$\begin{aligned} &P\left(X''_{ij} = x_{ij} + k'' \mid N' = n - \delta, X'_{ij} = x_{ij} + k'\right) \\ &\quad = P\left(X''_{ij} = x_{ij} - k'' \mid N' = n + \delta, X'_{ij} = x_{ij} - k'\right) \\ &\quad = h\left(k'' - k'; \delta + W, \delta, -k' + w_{ij}\right), \end{aligned} \tag{10}$$

$$\begin{aligned} &P\left(X''_{ij} = x_{ij} + k'' \mid N' = n - \delta, X'_{ij} = x_{ij} - k'\right) \\ &\quad = P\left(X''_{ij} = x_{ij} - k'' \mid N' = n + \delta, X'_{ij} = x_{ij} + k'\right) \\ &\quad = h\left(k'' + k'; \delta + W, \delta, k' + w_{ij}\right). \end{aligned} \tag{11}$$

If $k' > 0$ or $k'' > 0$, then a final symmetry comes from the fact that when $N' = n - \delta$, then the compensation cannot yield $X''_{ij} = x_{ij} - k'' < X'_{ij} = x_{ij} + k'$. Conversely, if $N' = n + \delta$, the compensation cannot yield $X''_{ij} = x_{ij} + k'' > X'_{ij} = x_{ij} - k'$:

$$\begin{aligned} &P\left(X''_{ij} = x_{ij} - k'' \mid N' = n - \delta, X'_{ij} = x_{ij} + k'\right) \\ &= P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij} - k'\right) = 0. \end{aligned} \tag{12}$$

Now, symmetries given by equations (6)–(12) can be exploited to asymptotically simplify the expression of $E(X''_{ij})$ obtained in derivation (5). The algebraic manipulation is lengthy and routine and is given in the Appendix. The idea is to split summations so that only nonnegative indexes appear. Symmetries are used thereafter to pairwise cancel terms multiplied by $-k''$ and k'', so that only probabilities multiplied by x_{ij} remain. Eventually, one gets

$$\lim_{n\to\infty,IJ\to\infty} E\left(X''_{ij}\right) = x_{ij} \cdot 1 = x_{ij},$$

which concludes the proof. ∎

Although Theorem 1 only guarantees asymptotical unbiasedness, numerical simulation shows that the bias of $\mathbf{x}''$ is very small already for $n > 30$ and $IJ > 30$ and nonextreme frequencies x_{ij}. Some negative bias appears for x_{ij} close to 0 and some positive bias appears for x_{ij} close to n (note that only one cell such that $x_{ij} > n/2$ may exist in a table with total frequency n). Bias appears because quality intervals are centred on x_{ij}, whereas for extreme x_{ij}, the restricted binomial distribution of X'_{ij} is no longer bell-shaped about x_{ij}. However, bias in extreme frequencies is far from a problem in disclosure control methods, as long as nonextreme frequencies are approximately preserved. Remember that such methods precisely seek to *disguise* extreme frequencies.

When n is small (say $n < 30$), both our method and Heer's are approximately unbiased only for x_{ij} close to $n/2$. However, it is uncommon to have such small values for n in published contingency tables.

3.3. Security of the Method

In order to assess the security of the method, we need the conditional distribution of the (unknown) original table $\mathbf{x}$ given the anonymised table $\mathbf{x}''$. In what follows, we will assume the prior distribution of an original cell X_{ij} to be discrete uniform, although other distributions (e.g., Poisson) are sometimes preferred as prior for contingency tables. There are two good reasons for our choice: simplicity and fairness when comparing with Heer's procedure, since this author also assumes priors to be discrete uniform. Using Bayes Theorem, we get

$$\begin{aligned} P\left(X_{ij} = x_{ij} \mid X''_{ij} = x''_{ij}\right) &= \frac{P\left(X''_{ij} = x''_{ij} \mid X_{ij} = x_{ij}\right) P\left(X_{ij} = x_{ij}\right)}{\sum\limits_{m:l(m)\le x''_{ij}\le u(m)} P\left(X''_{ij} = x''_{ij} \mid X_{ij} = m\right) P(X_{ij} = m)} \\ &= \frac{P\left(X''_{ij} = x''_{ij} \mid X_{ij} = x_{ij}\right)}{\sum\limits_{m:l(m)\le x''_{ij}\le u(m)} P\left(X''_{ij} = x''_{ij} \mid X_{ij} = m\right)}. \end{aligned} \tag{13}$$

Note that for $X''_{ij} = x''_{ij}$, the summation in the denominator above extends over all values m of X_{ij} such that $X''_{ij} = x''_{ij}$ can be obtained from $X_{ij} = m$, while meeting quality condition (1), i.e., such that $l(m) \le x''_{ij} \le u(m)$. Now using the total probability theorem,

$$\begin{aligned} &P\left(X''_{ij} = x''_{ij} \mid X_{ij} = x_{ij}\right) = \\ &\sum_{l(x_{ij})\le x'_{ij}\le u(x_{ij})} P\left(X''_{ij} = x''_{ij} \mid X'_{ij} = x'_{ij}, X_{ij} = x_{ij}\right) \cdot P\left(X'_{ij} = x'_{ij} \mid X_{ij} = x_{ij}\right), \end{aligned} \tag{14}$$

where $l(x_{ij})$ and $u(x_{ij})$ are the lower and upper bounds for X'_{ij} given by quality condition (1) when $X_{ij} = x_{ij}$. In Section 3.2, it was shown how to compute $P(X''_{ij} = x''_{ij} \mid X'_{ij} = x'_{ij}, X_{ij} = x_{ij})$ and $P(X'_{ij} = x'_{ij} \mid X_{ij} = x_{ij})$.

Once we have shown a way of computing $P(X_{ij} = x_{ij} \mid X''_{ij} = x''_{ij})$, we can find the probability of exact disclosure for small frequencies, that is

$$P\left(X_{ij} = k \mid X''_{ij} = k\right), \qquad \text{for } k = 1, 2, 3.$$

The results for a table with $n \ge 1000$ are summarised in Table 2 (again, the reported probability is practically independent of n when $n \ge 1000$). It can be seen that the disclosure risk for small frequencies is very similar to the one of the method of Section 2 when no averaging is done, i.e., $M = 1$ (best behaviour). In both cases, the highest disclosure probability is around 1/2.

Table 2. Conditional probabilities $P(X_{ij} = k \mid X''_{ij} = k)$ for $k = 1, 2, 3$.

Frequency	Probability		
k	$S = 3$	$S = 2$	$S = 1.5$
1	.43	.45	.51
2	.27	.30	.32
3	.22	.26	.29

So far, it has been shown that methods in Sections 2 and 3 are very similar from the statistical confidentiality point of view, because they provide roughly the same security level against disclosure of small frequencies. Moreover, both methods have the attractive property that their disclosure properties can be quantified mathematically, and not only by simulation. In this section, we will concentrate on analysing the computational complexity of both methods. The first difference is that complexity for the bootstrap method is very hard to establish without simulation, whereas our proposal can be analysed theoretically. But the most important difference is between the complexity functions obtained. The arithmetic in both methods involves additions, subtractions, multiplications, and divisions. The computing time is dominated by the number of multiplications and divisions, so that only these need to be taken into account.

4.1. Complexity of the Bootstrap-Based Method

The first step to assess the complexity of the bootstrap method is to find the probability that a bootstrap table $\mathbf{x}'$ meets the quality condition. Since $\mathbf{x}'$ is a multinomial sample (see Section 2), exact computation of the aforementioned probability involves computing exact multinomial probabilities, but as noted in [10, p. 285], this is "usually prohibitively difficult". Therefore, we will use simulation results quoted by [8] and our own C++ simulations. If α is the significance level being used for the quality condition and $q = 1 - \alpha$, we obtain that the probability that a table with IJ (nonzero) cells passes the quality test is about q^{IJ-1}. If the original table $\mathbf{x}$ contains NZ zero cells, the exponent should be corrected to $IJ - NZ - 1$. However, this correction is not relevant for our analysis, so in what follows, we will assume $NZ = 0$ to keep the notation simple. Thus, the number of bootstrap tables to be generated after the first one is a random variable following a geometric distribution with parameter q^{IJ-1}. The expected number of generated bootstrap tables is

$$\begin{aligned} E(\#\,\text{tables}) &= 1 \cdot q^{IJ-1} + 2 \cdot \left(1 - q^{IJ-1}\right) q^{IJ-1} + 3 \cdot \left(1 - q^{IJ-1}\right)^2 q^{IJ-1} + \cdots \\ &= 1 + \frac{1 - q^{IJ-1}}{q^{IJ-1}} = \frac{1}{q^{IJ-1}}. \end{aligned} \tag{15}$$

If the total frequency of the original table is n, then generation of a whole bootstrap table involves n random number generations. Each random number generation by a congruential method involves one multiplication and one division. Besides, IJ divisions are needed to split the random number range into IJ intervals representing a cell each. From the cost viewpoint, multiplications and divisions are equivalent. Thus, the total number of multiplications or divisions needed is

$$\frac{2n}{q^{IJ-1}} + IJ. \tag{16}$$

Note that expression (16) does not take into account that table generation is aborted as soon as some cell value x'_{ij} exceeds its upper limit $u(x_{ij})$. Thus, the expected amount of random number generations per table is smaller than n, but its exact computation would require computing exact multinomial probabilities. As argued above, this is not practical and can be bypassed using simulation. Our results show that, as IJ increases, the actual number of operations stays slightly below the curve given by expression (16) but still grows exponentially. Therefore, for the sake of clarity, expression (16) can be used to approximately describe the complexity behaviour of Heer's method without averaging.

If the averaging option is chosen, then M_{IJ} must be determined such that it minimises the expected computation.

LEMMA 4. *If M bootstrap tables are averaged, then the probability that a cell X_{ij}^M meets quality condition (1) is*

$$q(M) = F_{N(0,1)}\left(S\sqrt{M}\right) - F_{N(0,1)}\left(-S\sqrt{M}\right), \tag{17}$$

with $F_{N(0,1)}$ being the standardised normal cumulative distribution function.

PROOF. As noted in Section 2.1, the binomial distribution of a bootstrap frequency can be approximated by a normal distribution. Then, when averaging M tables, X_{ij}^M turns out to be distributed as $N(x_{ij}, (x_{ij}(1 - x_{ij}/n))/M)$. Therefore, the probability $q(M)$ of meeting quality condition (1) if x'_{ij} is replaced by a value x_{ij}^M drawn from X_{ij}^M can be computed as shown in equation (17). ∎

Note that, although not explicitly written, $q(M)$ also depends on the significance level α because S is the $\alpha/2$ percentage point of the $N(0,1)$ distribution. Note also that $q(1) = q = 1-\alpha$. Using the simulation results quoted above, if M bootstrap tables with IJ cells are averaged, then the probability that the resulting table meets the quality condition is about $q(M)^{IJ-1}$.

Following [8], if the average of M tables fails to meet the quality condition, a completely new set of M tables has to be generated and averaged. The alternative of just drawing another table and computing an $M+1$-average is not considered because it may be dangerous from the disclosure point of view (disclosure probabilities quickly increase with the number of averaged tables). From the above discussion, the expected number of generated tables can be obtained by multiplying the result of expression (15) by M and replacing q with $q(M)$,

$$E(\#\,\text{tables_with_M-average}) = \frac{M}{q(M)^{IJ-1}}. \tag{18}$$

Now, given a table size IJ, a value M_{IJ} of M must be found such that the right-hand side of equation (18) is minimised. Numerical minimisation shows that $M_{IJ} \approx O(\ln(IJ))$ and $q(M_{IJ})^{IJ-1} \approx 1$. These approximations hold regardless of the significance level used. Let us compute the number of multiplications or divisions required by the averaging option. As above, each table generation involves n random number generations (each needing one multiplication and one division). The IJ divisions to split the random number range can be shared by the M_{IJ} tables, but a supplementary division is needed to average each cell. Thus, the total number of multiplications or divisions can be estimated as

$$O(\ln(IJ)) \cdot 2n + IJ + IJ = 2IJ + 2nO(\ln(IJ)). \tag{19}$$

For the same reasons as that of expression (16), expression (19) slightly overestimates the actual number of operations, which nonetheless grows linearly with IJ.

4.2. Complexity of the New Method

To generate table $\mathbf{x}'$, a binomial random variable with parameters n and x_{ij}/n and restricted to the quality interval $[l(x_{ij}), u(x_{ij})]$ must be sampled for each cell ij. To do so, the probability of each value in the quality interval should be computed. For each cell, $u(x_{ij}) - l(x_{ij})$ probabilities must be computed directly. So, the number of probabilities to be computed for the whole table is

$$\begin{aligned}\sum_{i=1}^{I}\sum_{j=1}^{J} u(x_{ij}) - l(x_{ij}) &= \sum_{i=1}^{I}\sum_{j=1}^{J}\left[x_{ij} + \left\lfloor S\sqrt{x_{ij}\left(1-\frac{x_{ij}}{n}\right)}\right\rfloor - \left(x_{ij} - \left\lfloor S\sqrt{x_{ij}\left(1-\frac{x_{ij}}{n}\right)}\right\rfloor\right)\right] \\ &= \sum_{i=1}^{I}\sum_{j=1}^{J} 2\left\lfloor S\sqrt{x_{ij}\left(1-\frac{x_{ij}}{n}\right)}\right\rfloor.\end{aligned}$$

The above expression can be upper-bounded using that $\sum_{i=1}^{I}\sum_{j=1}^{J}\sqrt{x_{ij}(1-x_{ij}/n)}$ is maximal when $x_{ij} = n/IJ$, for all i, j:

$$\begin{aligned}\sum_{i=1}^{I}\sum_{j=1}^{J} 2\left\lfloor S\sqrt{x_{ij}\left(1-\frac{x_{ij}}{n}\right)}\right\rfloor &\le \sum_{i=1}^{I}\sum_{j=1}^{J} 2\left\lfloor S\sqrt{\frac{n}{IJ}\left(1-\frac{1}{IJ}\right)}\right\rfloor \\ &\approx 2S\sqrt{\frac{n}{IJ}}\sqrt{1-\frac{1}{IJ}}\cdot IJ \approx 2S\sqrt{n}\sqrt{IJ}.\end{aligned} \tag{20}$$

The first approximation in expression (20) is due to suppression of the floor function. The second approximation (20) is based on the fact that usually $IJ \gg 1$, and therefore, $1 - 1/IJ \approx 1$.

To compute the probabilities of values in the quality interval of cell ij, the following recurrence can be used which holds for the binomial distribution [11]:

$$b(z+1; n, p) = b(z; n, p)\frac{n-z}{z+1}\frac{p}{1-p}. \tag{21}$$

Thus, to compute a probability from recurrence (21), three multiplications and one division are needed (four operations altogether). The initial value for the recurrence can be arbitrarily chosen. In this way, the recurrence is used to obtain results proportional to the probabilities of all values in the quality interval of cell ij. To obtain real probabilities between 0 and 1, each result must be divided by the sum of all results (extended over all values in the quality interval for cell ij). Thus, one division should be added to the four operations needed to compute one probability. Even with five operations per probability, this recurrent procedure is more efficient than computing probabilities explicitly.

On the other hand, a random number must be generated for each cell. Using a congruential random number generator, one multiplication and one division are required. From expression (20) and the above considerations, the number of operations required to generate table $\mathbf{x}'$ is at most

$$5 \cdot 2S\sqrt{n}\sqrt{IJ} + 2 \cdot IJ = 10S\sqrt{n}\sqrt{IJ} + 2 \cdot IJ. \tag{22}$$

Finally, the compensation stage yielding table $\mathbf{x}''$ consists of $|n' - n|$ compensation steps, where each step is a random number generation. The discrete distribution from which each random number is drawn changes from step to step, because it depends on the cells first or second counters (see Section 3.1). Therefore, one step needs IJ divisions to split the random number range according to the current discrete distribution and one multiplication and one division to generate a random number congruentially. The following lemma gives a *tight upper bound* for the expected number of compensation steps.

LEMMA 5. *The expected number of compensation steps is upper-bounded by* $\sqrt{2n/\pi}$ *and this bound is tight.*

PROOF. Clearly the number of compensation steps depends on the original table $\mathbf{x}$ and its expected value is given by $E(|N' - n|)$, where $N' = \sum_{i,j} X'_{ij}$ and $n = \sum_{i,j} x_{ij}$. But as argued in Section 3.2, N' approximately follows a $N(n, \sigma^2_{N'})$ distribution with $\sigma^2_{N'} \leq n - (\sum_{i,j} x^2_{ij})/n$ (the bound is tight). So $N' - n$ is distributed as $N(0, \sigma^2_{N'})$. Thus, $E(|N' - n|)$ will be maximal when the variance of $N' - n$ is maximal (worst case). To find the maximal value of the bound on $\sigma^2_{N'}$, solve the following:

$$\text{Min} \qquad f(\{x_{ij}\}) = \sum_{i=1}^{I}\sum_{j=1}^{J} x^2_{ij},$$

$$\text{subject to} \quad \sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij} = n.$$

The minimum can be found by Lagrange's method and is reached when $x_{ij} = (n/IJ)\,\forall i, \forall j$. This gives a maximal bound of $n(1 - 1/IJ) \approx n$ (the last approximation is valid because typically $IJ \gg 1$). Therefore, in the worst case, $N' - n$ is $N(0, n)$. Taking $Y = N' - n$ as a continuous variable, we get the maximal number of compensation steps

$$\begin{aligned} E(|N' - n|) = E(|Y|) &= 2\int_0^\infty y f(y)\, dy \\ &= 2\int_0^\infty \frac{y}{\sqrt{2\pi n}} e^{-y^2/2n}\, dy = \sqrt{\frac{2n}{\pi}}. \end{aligned}$$

∎

Therefore, the expected total number of required multiplications or divisions for the whole procedure is, in the worst case,

$$10S\sqrt{n}\sqrt{IJ} + 2IJ + (IJ+2)\sqrt{\frac{2n}{\pi}}. \tag{23}$$

It can be seen that expression (16) increases exponentially with the number of cells IJ, whereas expression (23) increases linearly. If the total frequency n is taken as a variable, expression (23) grows linearly with $\sqrt{n}$, whereas expression (16) grows linearly with n. For example, if a quality bound $S = 1.5$ is taken (equivalently, $q = 0.866$), already for tables with 30 or more cells, our cell-oriented perturbation method is significantly faster than Heer's method without averaging. The fact that expression (16) is slightly above the real complexity can be neglected when comparing both methods for large IJ. See comparison in Table 3. The only entry for which Heer's method performs fewer operations than ours is $IJ = 50$, $n = 1000$, and $S = 3$; for larger values of IJ and n, our method clearly requires fewer operations.

Table 3. Number of operations with Heer's method without averaging minus number of operations with our method for several values of IJ, n, and S.

IJ	n	$S = 1.5$	$S = 2$	$S = 3$
50	1000	2.3×10^6	1.4×10^4	-5.8×10^3
	25000	5.6×10^7	4.6×10^5	1.7×10^4
	500000	1.1×10^9	9.7×10^6	9.6×10^5
1000	25000	8.4×10^{66}	8.0×10^{24}	4.7×10^5
	500000	1.7×10^{68}	1.6×10^{26}	1.4×10^7
10000	25000	3.4×10^{627}	8.3×10^{206}	2.7×10^{16}
	500000	2.3×10^{629}	1.7×10^{208}	5.5×10^{17}

NOTE 3. If the quality criterion is loosened (by increasing S, and therefore, q), then our method is faster only for large tables. However, note that anonymisation of off-line data requires that the quality criterion remain pretty strict, because the final user cannot repeat a query to get better data quality.

NOTE 4. If averaging is considered, then both methods have a complexity linear in IJ (compare expressions (19) and (23)), but it follows from Tables 1 and 2 that security is better for our proposal.

NOTE 5. Having a growth in complexity that is linear in the number of cells becomes especially interesting in the case of multiway tables. Whereas for two-way tables, the number of cells is usually not very large, it may become really large in higher-dimensional contingency tables.

5. CONCLUSION AND GENERALISATION

A major difference between anonymisation of on-line queries to a statistical database and anonymisation of off-line contingency tables is that, in the latter application, it must be guaranteed that each particular anonymised table being released does not differ too much from the original table (no repair is possible by repeated queries). Keeping this in mind, the following conclusions are in order which can be generalised to contingency tables of arbitrary dimension.

- For resampling methods (bootstrap or jack-knife) to meet the above quality requirement without wasting too much computation, several anonymised versions of the same original table must be averaged, but then too much information on low frequencies is released. This means high disclosure probability, and therefore, poor security.
- Without averaging, the essential problem of resampling is one of computational complexity. By the very principle of resampling, the quality criterion can only be checked on a table basis.

- In cell-oriented perturbation schemes, the anonymised table is constructed one cell after the other and the next cell is not generated until the quality criterion is met by the current cell. While for resampling methods the expected amount of wasted computation grows exponentially with the table size, the growth is linear for cell-oriented methods.
- We have illustrated a way to derive a cell-oriented method that emulates the quality properties of a bootstrap-based method. It is not difficult to generalise this idea of emulating any given resampling scheme for contingency tables by constructing a cell-oriented method. First, the distribution of each cell under the resampling method must be characterised. Then, the cell-oriented method proceeds as follows. For each cell, random values are drawn from the cell's distribution until a certain quality condition is met. Finally, compensations are done to emulate the remaining quality features of the resampling method (preservation of the overall frequency count or marginals in one dimension of the original table).

APPENDIX

We give here the detailed simplification of expression (5) using symmetries given by equations (6)–(12).

$$
\begin{aligned}
E\left(X''_{ij}\right) &= \sum_{k''=-w_{ij}}^{w_{ij}} (x_{ij}+k'') \cdot P\left(X''_{ij} = x_{ij}+k''\right) \\
&= \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{\delta=-W}^{W} \sum_{k'=-w_{ij}}^{w_{ij}} (x_{ij}+k'') \\
&\quad \times P\left(X''_{ij} = x_{ij}+k'' \mid N' = n+\delta, X'_{ij} = x_{ij}+k'\right) \\
&\quad \times P\left(N' = n+\delta \mid X'_{ij} = x_{ij}+k'\right) \cdot P\left(X'_{ij} = x_{ij}+k'\right) \\
&= \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{\delta=-W}^{W} \sum_{k'=-w_{ij}}^{w_{ij}} (x_{ij}+k'') \\
&\quad \times P\left(X''_{ij} = x_{ij}+k'' \mid N' = n+\delta, X'_{ij} = x_{ij}+k'\right) \\
&\quad \times P\left(N' - X'_{ij} = n - x_{ij} + (\delta - k')\right) \cdot P\left(X'_{ij} = x_{ij}+k'\right) \\
&= \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{\delta=-W}^{W} \sum_{k'=1}^{w_{ij}} (x_{ij}+k'') \\
&\quad \times \left[P\left(X''_{ij} = x_{ij}+k'' \mid N' = n+\delta, X'_{ij} = x_{ij}+k'\right)\right. \\
&\quad \times P\left(N' - X'_{ij} = n - x_{ij} + (\delta - k')\right) \cdot P\left(X'_{ij} = x_{ij}+k'\right) \\
&\quad + P\left(X''_{ij} = x_{ij}+k'' \mid N' = n+\delta, X'_{ij} = x_{ij}-k'\right) \\
&\quad \left. \times P\left(N' - X'_{ij} = n - x_{ij} + (\delta + k')\right) \cdot P\left(X'_{ij} = x_{ij}-k'\right)\right] \\
&\quad + \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{\delta=-W}^{W} (x_{ij}+k'') \\
&\quad \times P\left(X''_{ij} = x_{ij}+k'' \mid N' = n+\delta, X'_{ij} = x_{ij}\right) \\
&\quad \times P\left(N' - X'_{ij} = n - x_{ij} + \delta\right) P\left(X'_{ij} = x_{ij}\right) \\
&= \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{\delta=1}^{W} \sum_{k'=1}^{w_{ij}} (x_{ij}+k'') \\
&\quad \times \left[P\left(X''_{ij} = x_{ij}+k'' \mid N' = n+\delta, X'_{ij} = x_{ij}+k'\right)\right. \\
&\quad \left. \times P\left(N' - X'_{ij} = n - x_{ij} + (\delta - k')\right) \cdot P\left(X'_{ij} = x_{ij}+k'\right)\right.
\end{aligned}
$$

$$
\begin{aligned}
&+ P\left(X''_{ij} = x_{ij} + k'' \mid N' = n - \delta, X'_{ij} = x_{ij} + k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} - (\delta + k')\right) \cdot P\left(X'_{ij} = x_{ij} + k'\right) \\
&+ P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij} - k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} + (\delta + k')\right) \cdot P\left(X'_{ij} = x_{ij} - k'\right) \\
&+ P\left(X''_{ij} = x_{ij} + k'' \mid N' = n - \delta, X'_{ij} = x_{ij} - k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} - (\delta - k')\right) \cdot P\left(X'_{ij} = x_{ij} - k'\right)\Big] \\
&+ \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{k'=1}^{w_{ij}} (x_{ij} + k'') \\
&\times \Big[P\left(X''_{ij} = x_{ij} + k'' \mid N' = n, X'_{ij} = x_{ij} + k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} - k'\right) \cdot P\left(X'_{ij} = x_{ij} + k'\right) \\
&+ P\left(X''_{ij} = x_{ij} + k'' \mid N' = n, X'_{ij} = x_{ij} - k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} + k'\right) \cdot P\left(X'_{ij} = x_{ij} - k'\right)\Big] \\
&+ \sum_{k''=-w_{ij}}^{w_{ij}} \sum_{\delta=1}^{W} (x_{ij} + k'') \\
&\times \Big[P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij}\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} + \delta\right) \cdot P\left(X'_{ij} = x_{ij}\right) \\
&+ P\left(X''_{ij} = x_{ij} + k'' \mid N' = n - \delta, X'_{ij} = x_{ij}\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} - \delta\right) \cdot P\left(X'_{ij} = x_{ij}\right)\Big] \\
&+ \sum_{k''=-w_{ij}}^{w_{ij}} (x_{ij} + k'') \\
&\times P\left(X''_{ij} = x_{ij} + k'' \mid N' = n, X'_{ij} = x_{ij}\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij}\right) \cdot P\left(X'_{ij} = x_{ij}\right) \\
= &\sum_{k''=1}^{w_{ij}} \sum_{\delta=1}^{W} \sum_{k'=1}^{w_{ij}} \Big[(x_{ij} + k'') \\
&\times \Big[P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij} + k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} + (\delta - k')\right) P\left(X'_{ij} = x_{ij} + k'\right) && (24) \\
&+ P\left(X''_{ij} = x_{ij} + k'' \mid N' = n - \delta, X'_{ij} = x_{ij} + k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} - (\delta + k')\right) \cdot P\left(X'_{ij} = x_{ij} + k'\right) && (25) \\
&+ P\left(X''_{ij} = x_{ij} + k'' \mid N' = n + \delta, X'_{ij} = x_{ij} - k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} + (\delta + k')\right) \cdot P\left(X'_{ij} = x_{ij} - k'\right) && (26) \\
&+ P\left(X''_{ij} = x_{ij} + k'' \mid N' = n - \delta, X'_{ij} = x_{ij} - k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} - (\delta - k')\right) \cdot P\left(X'_{ij} = x_{ij} - k'\right)\Big] && (27) \\
&+ (x_{ij} - k'') \\
&\times \Big[P\left(X''_{ij} = x_{ij} - k'' \mid N' = n + \delta, X'_{ij} = x_{ij} + k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} + (\delta - k')\right) \cdot P\left(X'_{ij} = x_{ij} + k'\right) && (28) \\
&+ P\left(X''_{ij} = x_{ij} - k'' \mid N' = n - \delta, X'_{ij} = x_{ij} + k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} - (\delta + k')\right) \cdot P\left(X'_{ij} = x_{ij} + k'\right) && (29) \\
&+ P\left(X''_{ij} = x_{ij} - k'' \mid N' = n + \delta, X'_{ij} = x_{ij} - k'\right) \\
&\times P\left(N' - X'_{ij} = n - x_{ij} + (\delta + k')\right) \cdot P\left(X'_{ij} = x_{ij} - k'\right) && (30)
\end{aligned}
$$

$$
\begin{aligned}
&+P\left(X''_{ij}=x_{ij}-k''\mid N'=n-\delta,X'_{ij}=x_{ij}-k'\right)\\
&\times P\left(N'-X'_{ij}=n-x_{ij}-(\delta-k')\right)\cdot P\left(X'_{ij}=x_{ij}-k'\right)]] \qquad (31)\\
&+\sum_{\delta=1}^{W}\sum_{k'=1}^{w_{ij}}x_{ij}\\
&\times\left[P\left(X''_{ij}=x_{ij}\mid N'=n+\delta,X'_{ij}=x_{ij}+k'\right)\right.\\
&\times P\left(N'-X'_{ij}=n-x_{ij}+(\delta-k')\right)\cdot P\left(X'_{ij}=x_{ij}+k'\right)\\
&+P\left(X''_{ij}=x_{ij}\mid N'=n-\delta,X'_{ij}=x_{ij}+k'\right)\\
&\times P\left(N'-X'_{ij}=n-x_{ij}-(\delta+k')\right)\cdot P\left(X'_{ij}=x_{ij}+k'\right)\\
&+P\left(X''_{ij}=x_{ij}\mid N'=n+\delta,X'_{ij}=x_{ij}-k'\right)\\
&\times P\left(N'-X'_{ij}=n-x_{ij}+(\delta+k')\right)\cdot P\left(X'_{ij}=x_{ij}-k'\right)\\
&+P\left(X''_{ij}=x_{ij}\mid N'=n-\delta,X'_{ij}=x_{ij}-k'\right)\\
&\left.\times P\left(N'-X'_{ij}=n-x_{ij}-(\delta-k')\right)\cdot P\left(X'_{ij}=x_{ij}-k'\right)\right]\\
&+\sum_{k''=1}^{w_{ij}}\sum_{k'=1}^{w_{ij}}\left[(x_{ij}+k'')\right.\\
&\times\left[P\left(X''_{ij}=x_{ij}+k''\mid N'=n,X'_{ij}=x_{ij}+k'\right)\right.\\
&\times P\left(N'-X'_{ij}=n-x_{ij}-k'\right)\cdot P\left(X'_{ij}=x_{ij}+k'\right) \qquad (32)\\
&+P\left(X''_{ij}=x_{ij}+k''\mid N'=n,X'_{ij}=x_{ij}-k'\right)\\
&\left.\times P\left(N'-X'_{ij}=n-x_{ij}+k'\right)\cdot P\left(X'_{ij}=x_{ij}-k'\right)\right] \qquad (33)\\
&+(x_{ij}-k'')\\
&\times\left[P\left(X''_{ij}=x_{ij}-k''\mid N'=n,X'_{ij}=x_{ij}+k'\right)\right.\\
&\times P\left(N'-X'_{ij}=n-x_{ij}-k'\right)\cdot P\left(X'_{ij}=x_{ij}+k'\right) \qquad (34)\\
&+P\left(X''_{ij}=x_{ij}-k''\mid N'=n,X'_{ij}=x_{ij}-k'\right)\\
&\left.\times P\left(N'-X'_{ij}=n-x_{ij}+k'\right)\cdot P\left(X'_{ij}=x_{ij}-k'\right)\right]] \qquad (35)\\
&+\sum_{k'=1}^{w_{ij}}x_{ij}\\
&\times\left[P\left(X''_{ij}=x_{ij}\mid N'=n,X'_{ij}=x_{ij}+k'\right)\right.\\
&\times P\left(N'-X'_{ij}=n-x_{ij}-k'\right)\cdot P\left(X'_{ij}=x_{ij}+k'\right)\\
&+P\left(X''_{ij}=x_{ij}\mid N'=n,X'_{ij}=x_{ij}-k'\right)\\
&\left.\times P\left(N'-X'_{ij}=n-x_{ij}+k'\right)\cdot P\left(X'_{ij}=x_{ij}-k'\right)\right]\\
&+\sum_{k''=1}^{w_{ij}}\sum_{\delta=1}^{W}\left[(x_{ij}+k'')\right.\\
&\times\left[P\left(X''_{ij}=x_{ij}+k''\mid N'=n+\delta,X'_{ij}=x_{ij}\right)\right.\\
&\times P\left(N'-X'_{ij}=n-x_{ij}+\delta\right)\cdot P\left(X'_{ij}=x_{ij}\right) \qquad (36)\\
&+P\left(X''_{ij}=x_{ij}+k''\mid N'=n-\delta,X'_{ij}=x_{ij}\right)\\
&\left.\times P\left(N'-X'_{ij}=n-x_{ij}-\delta\right)\cdot P\left(X'_{ij}=x_{ij}\right)\right] \qquad (37)\\
&+(x_{ij}-k'')\\
&\times\left[P\left(X''_{ij}=x_{ij}-k''\mid N'=n+\delta,X'_{ij}=x_{ij}\right)\right.\\
&\times P\left(N'-X'_{ij}=n-x_{ij}+\delta\right)\cdot P\left(X'_{ij}=x_{ij}\right) \qquad (38)\\
&+P\left(X''_{ij}=x_{ij}-k''\mid N'=n-\delta,X'_{ij}=x_{ij}\right)\\
&\left.\times P\left(N'-X'_{ij}=n-x_{ij}-\delta\right)\cdot P\left(X'_{ij}=x_{ij}\right)\right]] \qquad (39)
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{\delta=1}^{W} x_{ij} \\
&\times \left[P\left(X''_{ij}=x_{ij} \mid N'=n+\delta, X'_{ij}=x_{ij}\right)\right. \\
&\times P\left(N'-X'_{ij}=n-x_{ij}+\delta\right)\cdot P\left(X'_{ij}=x_{ij}\right) \\
&+P\left(X''_{ij}=x_{ij} \mid N'=n-\delta, X'_{ij}=x_{ij}\right) \\
&\left.\times P\left(N'-X'_{ij}=n-x_{ij}-\delta\right)\cdot P\left(X'_{ij}=x_{ij}\right)\right] \\
&+\sum_{k''=1}^{w_{ij}}\left[(x_{ij}+k'')\right. \\
&\times P\left(X''_{ij}=x_{ij}+k'' \mid N'=n, X'_{ij}=x_{ij}\right) \\
&\times P\left(N'-X'_{ij}=n-x_{ij}\right)\cdot P\left(X'_{ij}=x_{ij}\right) \qquad (40)\\
&+(x_{ij}-k'') \\
&\times P\left(X''_{ij}=x_{ij}-k'' \mid N'=n, X'_{ij}=x_{ij}\right) \\
&\left.\times P\left(N'-X'_{ij}=n-x_{ij}\right)\cdot P\left(X'_{ij}=x_{ij}\right)\right] \qquad (41)\\
&+x_{ij}P\left(X''_{ij}=x_{ij} \mid N'=n, X'_{ij}=x_{ij}\right) \\
&\times P\left(N'-X'_{ij}=n-x_{ij}\right)P\left(X'_{ij}=x_{ij}\right).
\end{aligned}
$$

Now,

- By symmetries (6), (7), and (9), terms (24) and (31) become identical as $n \to \infty$ and $IJ \to \infty$.
- By symmetries (6), (8), and (10), terms (25) and (30) become identical as $n \to \infty$ and $IJ \to \infty$.
- By symmetry (12), terms (26) and (29) are both zero.
- By symmetries (6), (7), and (11), terms (27) and (28) become identical as $n \to \infty$ and $IJ \to \infty$.
- In terms (32) and (35), probabilities $P(X''_{ij} = x_{ij} + k'' \mid N' = n, X'_{ij} = x_{ij} + k')$ and $P(X''_{ij} = x_{ij} - k'' \mid N' = n, X'_{ij} = x_{ij} - k')$ are 1 if $k'' = k'$ and zero otherwise, because no compensation is done when $N' = n$. This, together with symmetries (6) and (7), explains why terms (32) and (35) become identical as $n \to \infty$ (since $N' = n$, it is not necessary that $IJ \to \infty$).
- By symmetry (12), terms (33) and (34) are both zero.
- By symmetry (12), terms (36) and (39) are both zero.
- By symmetries (7) and (11), terms (37) and (38) become identical as $n \to \infty$ and $IJ \to \infty$.
- By symmetry (12), terms (40) and (41) are both zero.

In all pairs of identical terms, there is a term multiplying $x_{ij} + k''$ and another term multiplying $x_{ij} - k''$. Therefore, after pairwise adding, the terms multiplied by k'' and $-k''$ cancel each other, so every remaining probability is multiplied by x_{ij}. Thus,

$$\lim_{n\to\infty, IJ\to\infty} E\left(X''_{ij}\right) = x_{ij}\cdot 1 = x_{ij}.$$

REFERENCES

1. N.R. Adam and J.C. Wortmann, Security-control methods for statistical databases: A comparative study, *ACM Computing Surveys* **21** (4), 515–556 (1989).
2. D.H. Jones and N.R. Adam, Disclosure avoidance using the bootstrap and other resampling schemes, In *Proceedings of the Fifth Annual Research Conference*, pp. 446–455, U.S. Bureau of the Census, Washington, DC, (1989).
3. N.E. Matloff, Another look at the use of noise addition for database security, In *Proceedings of IEEE Symposium on Security and Privacy*, pp. 173–180, IEEE, New York, (1986).
4. L. Willenborg and T. de Waal, *Statistical Disclosure Control in Practice*, Springer-Verlag, New York, (1996).
5. D. Schackis, *Manual on Disclosure Control Methods*, Eurostat, Luxembourg, (1993).

6. S.E. Fienberg, Confidentiality and disclosure limitation methodology: Challenges for national statistics and statistics research, CMU Department of Statistics, Technical Report Number 668, Pittsburgh, PA, (1997).
7. D.E. Denning, Secure statistical database with random sample queries, *ACM Transactions on Database Systems* **5** (3), 291–315 (1980).
8. G.R. Heer, A bootstrap procedure to preserve statistical confidentiality in contingency tables, In *Proceedings of the International Seminar on Statistical Confidentiality*, (Edited by D. Lievesley), pp. 261–271, Eurostat, Luxembourg, (1993).
9. S. Giessing, Looking for efficient automated secondary cell suppression systems: A software comparison, *Research in Official Statistics* **1** (2), 69–85 (1998).
10. N.L. Johnson and S. Kotz, *Discrete Distributions*, Houghton Mifflin, Boston, MA, (1969).
11. J. Dagpunar, *Principles of Random Variable Generation*, Clarendon Press, Oxford, (1988).