A Polynomial-time Algorithm to Approximately Count Contingency Tables when the Number of Rows is Constant [⋆]

Mary Cryan and Martin Dyer

School of Computing, University of Leeds, Leeds LS2 9JT, United Kingdom.
(maryc | dyer)@comp.leeds.ac.uk

Abstract

We consider the problem of counting the number of contingency tables with given row and column sums. This problem is known to be $\#P$-complete, even when there are only two rows [11]. In this paper we present the first fully-polynomial randomized approximation scheme for counting contingency tables when the number of rows is constant. A novel feature of our algorithm is that it is a hybrid of an exact counting technique with an approximation algorithm, giving two distinct phases. In the first, the columns are partitioned into "small" and "large". We show that the number of contingency tables can be expressed as the weighted sum of a polynomial number of new instances of the problem, where each instance consists of some new row sums and the original large column sums. In the second phase, we show how to approximately count contingency tables when all the column sums are large. In this case, we show that the solution lies in approximating the volume of a single convex body, a problem which is known to be solvable in polynomial time [7].

Key words: Contingency tables, approximate counting, randomized algorithms.

1 Introduction

Suppose we are given two vectors of positive integers, $r = (r_1, \ldots, r_m)$ and $c = (c_1, \ldots, c_n)$, such that $\sum_{i=1}^{m} r_i = \sum_{j=1}^{n} c_j$. We say that an $m \times n$ matrix $[X_{i,j}]$ of non-negative integers is a *contingency table* with *row* sums r and *column* sums c if $\sum_{j=1}^{n} X_{i,j} = r_i$ for every row i and $\sum_{i=1}^{m} X_{i,j} = c_j$ for every column j. We denote the set of all contingency tables by $\Sigma_{r,c}$.

[⋆] Supported by the EPSRC grant "Sharper Analysis of Randomised Algorithms: a Computational Approach" and by the EC IST project RAND-APX.

It is well-known that for any input satisfying $\sum_{i=1}^{m} r_i = \sum_{j=1}^{n} c_j$, there exists at least one contingency table with row sums r and column sums c (see, for example, Diaconis and Gangolli [5]). It is easy to construct one element of $\Sigma_{r,c}$ using the "North-West corner" rule (see, for example, Hadley [12]).

In this paper we consider the problem of *approximately* counting the set of all contingency tables with specified row and column sums. We present the first *fully polynomial randomized approximation scheme (fpras)* [17] for counting such tables when the number of rows is *constant.* The definition of an fpras has been given elsewhere but we include it here to be precise. An *fpras* for contingency tables is an algorithm that takes a list of row sums r and a list of column sums c as input, together with an error parameter $\epsilon \in (0, 1)$. The algorithm must satisfy two conditions to be an fpras. Firstly, it must output an approximate value that lies within $(1 \pm \epsilon)|\Sigma_{r,c}|$, with high probability. Second, its running time must be polynomial in the size of the input and also in ϵ^{-1}. Here we present an fpras for the case when m is constant.

Our algorithm also implies a polynomial time procedure for the closely related problem of *sampling* such a table almost uniformly at random. See the surveys of Jerrum and Sinclair [14], or Dyer and Greenhill [8], for more definitions and background about approximate counting and sampling.

The counting problem is of considerable interest, both from the theoretical and practical viewpoints. The thesis of Mount provides much useful information on this problem and its relatives [20]. Dyer, Kannan and Mount [11] have shown that the problem of counting contingency tables is $\#P$-complete even if there are only two rows; therefore, we do not expect to be able to exactly count the number of contingency tables in polynomial-time, even for two-rowed tables. The existence of an fpras for counting contingency tables has been an open question for several years. For example, the 1997 survey by Jerrum and Sinclair [14] listed it as an important open problem in the complexity of approximate counting.

Practically, contingency tables play an important role in statistics, where they are used to tabulate the results of surveys. The analysis of such tables provides strong motivation for the problem of efficiently sampling contingency tables with given row and column sums almost uniformly at random. Diaconis and Efron [4] provide many details on the practical motivation for the sampling problem.

Before presenting our algorithm, we summarize previous work on the problem of counting contingency tables. The first polynomial-time algorithm for counting contingency tables was due to Barvinok [1], who proved that the number of contingency tables can be counted *exactly* in polynomial-time, when the number of rows and columns is constant (see also [10]).

Most other early papers on the subject addressed the sampling problem. The paper of Diaconis and Gangolli [5] seems to be the first to describe a Markov chain on the space of contingency tables which converges to the uniform distribution. The convergence rate of this chain was subsequently analyzed by Diaconis and Saloff-Coste [6] for the case when the number of rows and columns is fixed and by Hernek [13] for the case when there are two rows. The analyses for both cases showed that the chain mixed in pseudopolynomial time (the running time is polynomial in the table sum). Chung et al. [3] gave a Markov chain for contingency tables that converges in pseudopolynomial time for any row and column sums which are sufficiently large.

The first polynomial-time algorithm for approximately counting contingency tables with unbounded dimension was the algorithm of Dyer, Kannan and Mount [11]. They (i) gave a sampling algorithm that converges in polynomial time for any input with row sums of size $\Omega(n^2 m)$ and column sums of size $\Omega(nm^2)$; (ii) showed how to use the sampling algorithm to approximately count the number of contingency tables for inputs satisfying the same constraints. This result was later refined by Morris [19], who showed that the result also holds when the row sums are $\Omega(n^{3/2} m \log m)$ and the column sums are $\Omega(m^{3/2} n \log n)$. Dyer and Greenhill [9] gave a polynomial-time algorithm for counting contingency tables when the table has two rows. They first defined a Markov chain for sampling from the set of contingency tables with the given row and column sums, and showed that this chain converges in polynomial-time when the input has two rows. Then they showed how to use their sampling algorithm to obtain an fpras for the corresponding counting problem. The result we prove here is a generalization of Dyer and Greenhill's (from two rows to m rows), but we use an entirely different approach.

A novel feature of our algorithm, which is described in Section 2, is that it is a hybrid of an exact counting algorithm and an approximation algorithm. It can be viewed as having two phases. The input is a list containing a constant number of row sums, a list of column sums, and an error parameter $\epsilon > 0$. In the first phase of the algorithm (Step 1 below) we partition the columns of the table into "small columns" and "large columns". Every contingency table for the given row and column sums can be split into two smaller tables – a table on the small columns (with some list of partial row sums), and a table on the large columns (whose list of row sums is the original list of row sums less the list of partial row sums). We show that the number of different lists of partial row sums that may occur on the table of small columns is polynomial in the number of columns and ϵ^{-1}. By dynamic programming, we can count the number of contingency tables on the small columns for any given list of partial row sums in polynomial time. We then write the number of contingency tables for the original input as the weighted sum (each weight is the count computed for some list of partial row sums) of a polynomial number of terms, where each term is the number of contingency tables for some list of row sums and the large columns.

In the second phase of the algorithm (Step 2), we approximately count contingency tables for each of the new instances of the problem generated in the first phase. Consider any specific instance. We know the number of rows is constant and all the columns are large. We partition the rows using a different method to that used for the columns. We define a “gap factor” which is sufficiently large. Then we partition the rows into small rows and substantially larger rows – each of the large rows must be larger than the product of any small row and the gap factor. Note that the number of contingency tables for our given row and column sums can be written as the sum, over all possible partial column sums for the small rows, of the number of contingency tables for the given row and column sums which have these partial column sums. Our partitioning of the rows ensures that any partial column sums will be small in comparison to the large column sums. In Sections 3 and 4 we show that in this case the number of contingency tables with given partial column sums does not depend much on the specific partial column sums that are considered. Therefore we can estimate the number of contingency tables by choosing a fixed list of partial column sums, and calculating the product of the total number of tables for the small rows (with any partial column sums) and the number of contingency tables for our instance which have the fixed partial column sums. The total number of tables for the small rows can be calculated using binomial coefficients. The second quantity we need to compute is a single instance of the problem of counting contingency tables, where all the columns are large and all the rows are large. In Section 3 we show that, in this case, the number of contingency tables is very close to the volume of a convex polytope. We use the polynomial-time algorithm of Kannan, Lovász and Simonovits [16], for approximating the volume of convex bodies, to estimate the volume of this polytope.

For many combinatorial problems, the problem of *approximately counting* the number of discrete structures satisfying a given property is closely related to the problem of *sampling* one discrete structure with this property almost uniformly at random. In random sampling, we usually want to construct a *(fully) polynomial almost-uniform sampler* (see, for example, Jerrum, Valiant and Vazirani [15] , Sinclair and Jerrum [21]). It is well-known that for a special class of problems known as *self-reducible problems*, the existence of a polynomial-time algorithm for approximate counting implies the existence of a fully polynomial almost-uniform sampler [15,21]. The contingency tables problem is unusual because it is not known to satisfy the condition of self-reducibility (or a more general condition discussed by Dyer and Greenhill [8]). However, in Section 5 we will show that our fpras can be used to obtain a polynomial almost-uniform sampler for sampling almost uniformly at random from the space of contingency tables with given row and column sums, when the number of rows is constant.

2 The Algorithm

Before presenting the algorithm, we introduce some notation. First, for any lists $r = (r_1, \ldots, r_m)$ and $c = (c_1, \ldots, c_n)$ of non-negative integers, we say that a $m \times n$ integer matrix X is a contingency table with row sums r and column sums c iff

$$\begin{aligned} X_{i,j} &\geq 0 \quad \text{for all } i, j, \\ \sum_{j=1}^{n} X_{i,j} &= r_i \quad \text{for all } i, \\ \sum_{i=1}^{m} X_{i,j} &= c_j \quad \text{for all } j \end{aligned}$$

We let $\Sigma_{r,c}$ denote the set of all contingency tables with row sums r and column sums c. The cardinality of this set, denoted $|\Sigma_{r,c}|$, is the number of contingency tables with the given row and column sums. We always assume that $\sum_{i=1}^{m} r_i$ is equal to $\sum_{j=1}^{n} c_j$ (otherwise $\Sigma_{r,c}$ is empty) and denote this total (also called the *table sum*) by N.

Throughout this paper we will assume that $m \geq 2$ is a constant. We assume without loss of generality that $n \geq m$.

Our algorithm takes a list $r = (r_1, \ldots, r_m)$ of row sums and a list $c = (c_1, \ldots, c_n)$ of column sums, an error parameter ϵ satisfying $0 < \epsilon < 1$ and a confidence parameter η satisfying $0 < \eta < 1$. The algorithm runs in time polynomial in n, $\log N$, ϵ^{-1} and $\log \eta^{-1}$ and returns an estimate $S_{r,c}$. In Sections 3 and 4, we will prove that $|S_{r,c} - |\Sigma_{r,c}|| \leq \epsilon |\Sigma_{r,c}|$ with probability at least $1 - \eta$.

The following quantities will be useful in describing the algorithm:

$$\begin{aligned} p_\epsilon &= \log_n(20nm/\epsilon) \\ p &= 2(m-1)(p_\epsilon + 2) + 1 \\ q &= (p-1)/2(m-1) \end{aligned}$$

Note that q is equal to $p_\epsilon + 2$.

We will apply the following Observation (cf. page 63 of Mount [20]):

Observation 1 *Let $r = (r_1, \ldots, r_m)$ and $c = (c_1, \ldots, c_n)$ be two lists of positive integers satisfying $\sum_{i=1}^{m} r_i = \sum_{j=1}^{n} c_j$.*

Let $1 \leq k < n$. Let S be the set of ordered partitions s of $\sum_{j=1}^{k} c_j$ into m parts that satisfy $s_i \leq r_i$ for all $1 \leq i \leq m$. Then

$$|\Sigma_{r,c}| = \sum_{s \in S} |\Sigma_{s,(c_1,\ldots,c_k)}| \times |\Sigma_{r-s,(c_{k+1},\ldots,c_n)}| \tag{1}$$

Let $1 \leq \ell < m$. Let T be the set of ordered partitions t of $\sum_{i=1}^{\ell} r_i$ into n parts that satisfy $t_j \leq c_j$ for all $1 \leq j \leq n$. Then

$$|\Sigma_{r,c}| = \sum_{t \in T} |\Sigma_{(r_1,\ldots,r_\ell),t}| \times |\Sigma_{(r_{\ell+1},\ldots,r_m),c-t}| \tag{2}$$

The following observation will also be useful

Observation 2 *Let $m \geq 2$ be an integer, and let M be another positive integer. Then the number of ordered partitions of M into m parts is*

$$\binom{M+m-1}{m-1} \leq 2M^{m-1}$$

Our algorithm is based on Observation 1.

In Step 1 of the algorithm, we choose an appropriate value for k and calculate $|\Sigma_{s,(c_1,\ldots,c_k)}|$ exactly for all $s \in S$.

In Step 2 we approximate $|\Sigma_{r-s,(c_{k+1},\ldots,c_n)}|$ within $(1 \pm \epsilon)$ of its true value with high probability, for every $s \in S$.

In Step 3 we apply Equation (1) to estimate $\Sigma_{r,c}$ within $(1 \pm \epsilon)$ with high probability.

2.1 Step 1

Assume that $(c_1, \ldots, c_n)$ is sorted in non-decreasing order. Let k be the index such that $c_j \leq n^p$ for all $j \leq k$ and $c_j > n^p$ for all $j \geq k+1$.

Columns $c_1, \ldots, c_k$ are the "small columns" of the table.

Columns $c_{k+1}, \ldots, c_n$ are the "large columns".

In this step of our algorithm, we will use dynamic programming to calculate $|\Sigma_{s,(c_1,\ldots,c_k)}|$ for every partition $s \in S$. In fact, our algorithm will consider each column index h $(1 \leq h \leq k)$ in increasing order, and compute $|\Sigma_{s,(c_1,\ldots,c_h)}|$ for every ordered partition s of $\sum_{j=1}^{h} c_j$ into m parts.

We will let S_h represent the set of ordered partitions of $\sum_{j=1}^{h} c_j$ into m parts, for $1 \leq h \leq k$.

If $h = 1$, then $|\Sigma_{s,(c_1)}| = 1$ for every partition s of c_1 into m parts. Note that because $c_1 \leq n^p$, then by Observation 2, the number of ordered partitions we will consider is at most $2(n^p)^m$.

If $2 \leq h \leq k$, then we apply Equation (1) of Observation 1. Let $s \in S_h$. For us, the values of the parameters n, k and r of Equation (1) are $\widehat{n} = h, \widehat{k} = h-1$ and $\widehat{r} = s$. Then by Equation (1) we have

$$\begin{aligned} |\Sigma_{s,(c_1,\ldots,c_h)}| &= \sum_{q \in S_{h-1}} |\Sigma_{q,(c_1,\ldots,c_{h-1})}| \times |\Sigma_{s-q,c_h}| \\ &= \sum_{\substack{q \in S_{h-1}, \\ q_i \leq s_i \text{ for all } i}} |\Sigma_{q,(c_1,\ldots,c_{h-1})}|, \end{aligned} \tag{3}$$

since $\Sigma_{s-q,c_h} = 1$ if $s_i - q_i \geq 0$ for all $1 \leq i \leq m$ (the single "table" is given by $X_{i,h} = s_i - q_i$ for all i) and $\Sigma_{s-q,c_h} = 0$ otherwise. Therefore we use the $|\Sigma_{q,(c_1,\ldots,c_{h-1})}|$ values (constructed in the previous phase of our algorithm) to obtain $|\Sigma_{s,(c_1,\ldots,c_h)}|$.

Note that because $c_j \leq n^p$ for all $j \leq k$, therefore

$$\sum_{j=1}^{h} c_j \leq hn^p \leq n^{p+1}. \tag{4}$$

for any $1 \leq h \leq k$. By Observation 2 and by Inequality (4), the number of ordered partitions of $\sum_{j=1}^{h} c_j$ into m parts is at most $2(n^{p+1})^m$. Therefore $|S_h| \leq 2n^{2m(p+1)}$, which is polynomial in n and ϵ^{-1}.

Therefore for any particular $h \leq k$, we perform $O(n^{m(p+1)})$ operations to compute $|\Sigma_{s,(c_1,\ldots,c_h)}|$; therefore using $O(n^{2m(p+1)})$ arithmetic operations, we compute a table containing $|\Sigma_{s,(c_1,\ldots,c_h)}|$ for every ordered partition s of $\sum_{j=1}^{h} c_j$ into m parts. Since $k \leq n$, this means that we compute the table of $|\Sigma_{s,(c_1,\ldots,c_k)}|$ using $O(n^{2m(p+1)+1})$ arithmetic operations.

By definition, $p+1 = 2(m-1)(p_\epsilon+2)+2 = 2(m-1)p_\epsilon + 4m - 2$. Therefore

$$n^{p+1} = \left(\frac{20nm}{\epsilon}\right)^{2(m-1)} n^{4m-2}.$$

Therefore Step 1 uses

$$O\left(\frac{n^{12m^2}}{\epsilon^{4m^2}}\right) \tag{5}$$

arithmetic operations to compute the set of all $|\Sigma_{s,(c_1,\ldots,c_k)}|$ values for $s \in S$.

We know that none of the integers we compute is greater than N^{nm}, therefore each addition or comparison performed during Step 1 can be carried out in $O(n \log N)$ time.

We also know $|S| \leq 2(n^{p+1})^m$, and therefore $|S|$ is

$$O\left(\frac{n^{6m^2}}{\epsilon^{2m^2}}\right). \tag{6}$$

2.2 Step 2

In this step we show how to approximate the value of $|\Sigma_{r-s,(c_{k+1},\ldots,c_n)}|$ within a multiplicative factor of $(1 \pm \epsilon)$ of its true value in polynomial time, with high probability, for any given $s \in S$.

First let $\eta' = \eta/|S|$, where η is the original failure probability given as input to the algorithm. By (6) this implies $\eta' = \eta\epsilon^{2m^2}/n^{6m^2}d$, where d is the constant inside the O in (6).

Sort the rows of $r - s$ into non-decreasing order and rename this vector by r'.

Let n' denote $n - k$, and rename the $(c_{k+1}, \ldots, c_n)$ vector by $(c'_1, \ldots, c'_{n'})$.

We will estimate $|\Sigma_{r',c'}|$.

Let $\widehat{N} = \sum_{j=1}^{n'} c'_j$ be the table sum on the large columns.

Now classify the rows of r' as "small rows" or "large rows" as follows: If $r'_1 \geq n^q$, then we classify all the rows as large rows. Otherwise $r'_1 < n^q$. Then let ℓ be the smallest index such that $r'_{\ell+1} > n^q r'_\ell$ (if such an ℓ exists). The rows 1 to ℓ are the "small rows" and the rows greater than ℓ are the "large rows".

Define $R = \sum_{i=1}^{\ell} r'_i$.

We consider three cases.

Case 1: All the rows are large rows ($r'_1 \geq n^q$). In this case, the row sums r' and the column sums c' satisfy the conditions of Theorem 3 (see Section 3). Therefore, by Theorem 3, the value of $|\Sigma_{r',c'}|$ is within $(1 \pm \epsilon/15)$ of the volume of the convex polytope $\mathrm{P}(r', c')$ defined in Section 3. We use the polynomial-time algorithm of Kannan, Lovász and Simonovits [16] for approximating the volume of a convex body, to approximate $\mathrm{vol}(\mathrm{P}(r', c'))$ within a factor of $(1 \pm$

$\epsilon/5$), with probability at least $1-\eta'$. Thus we approximate $|\Sigma_{r',c'}|$ within $(1\pm\epsilon)$ with probability at least $1-\eta'$.

Case 2: All the rows are small rows. We show this case cannot occur. Suppose this is a possibility. Since all the rows are small rows, the table sum $\widehat{N}$ is equal to R. This table sum is bounded above by mn^{qm}. By definition of q,

$$\begin{aligned} mn^{qm} &= mn^{\frac{p-1}{2(m-1)}m} \\ &= mn^{\frac{p-1}{2}(1+1/(m-1))} \\ &\leq mn^{p-1} && \text{because } m \geq 2 \\ &\leq n^{p} && \text{because } m \leq n \end{aligned}$$

Therefore if all the rows were small rows, the table sum on the large columns would be at most n^p. However, since all the large columns were assumed to have $c_j > n^p$, $\widehat{N} \leq n^p$ implies that there are no large columns. This is a contradiction (if there are no large columns, then $|\Sigma_{r,c}|$ would have been computed exactly by Step 1, and Step 2 would not be carried out).

Case 3: There are small rows and large rows. The quantity R plays a central role in the analysis for this case. Before proceeding, note that $R \leq \sum_{i=1}^{\ell} n^{qi}$, which is at most $(m-1)n^{q(m-1)}$ (since $\ell < m$, we have at least one large row). Substituting for q and then for p,

$$R \leq (m-1)n^{(p-1)/2} = (m-1)n^{(m-1)(p_\epsilon+2)} \tag{7}$$

$$n^p/R \geq \quad n^{(p-1)/2} \quad = n^{(m-1)(p_\epsilon+2)} \tag{8}$$

Now we show how to approximate $|\Sigma_{r',c'}|$ for this case. By Equation (2) of Observation 1, we write

$$|\Sigma_{r',c'}| = \sum_{t} |\Sigma_{(r'_1,\ldots,r'_\ell),t}| \times |\Sigma_{(r'_{\ell+1},\ldots,r'_m),c'-t}| \tag{9}$$

where the sum is taken over all partitions t of the value R into a list of n' non-negative integers.

From here on we will denote the large row sums $(r'_{\ell+1},\ldots,r'_m)$ by $(u_1,\ldots,u_{m'})$, and any list of modified large column sums $c'-t$ by $(v_1,\ldots,v_{n'})$. By construction, every u_i is at least n^q. To obtain a lower bound for the v_j values, remember that by construction $c_j \geq n^p$ for every $1 \leq j \leq n'$. Also we know $t_j \leq R$ for every $1 \leq j \leq n'$. Therefore every v_j value is at least as big as $n^p - R$, and by (8), this is at least $n^p/2$.

In Section 3, we will define a convex polytope $\mathrm{P}(u,v)$ in $(m'-1)(n'-1)$-dimensional space for any large row sums u and modified large column sums v. Let $\mathrm{vol}(\mathrm{P}(u,v))$ denote the the volume of the convex polytope $\mathrm{P}(u,v)$. We will prove the following theorems:

Theorem 3 For any list u of large row sums and any list v of modified large column sums, $|\Sigma_{u,v}|$ lies within $(1 \pm \epsilon/15)$ of $\mathrm{vol}(\mathrm{P}(u,v))$ (See Section 3).

Theorem 4 Let u be a list of large row sums and let v and $\widehat{v}$ be two lists of modified large column sums. Then $\mathrm{vol}(\mathrm{P}(u,v)) \le (1+\epsilon/15)\mathrm{vol}(\mathrm{P}(u,\widehat{v}))$ (See Section 4).

Now we show that Theorems 3 and 4 allow us to approximate all of the different $|\Sigma_{u,v}|$ values (there could be exponentially many of these) in a single step. Define some fixed list of modified column sums $\widehat{v}$ by choosing an arbitrary partition $\widehat{t}$ of R, and defining $\widehat{v}$ as $c' - \widehat{t}$. Let v be *any* other list of modified column sums. By Theorem 3 we have

$$\begin{aligned} |\Sigma_{u,v}| &\le (1+\epsilon/15)\mathrm{vol}(\mathrm{P}(u,v)) \\ &\le (1+\epsilon/15)^2\mathrm{vol}(\mathrm{P}(u,\widehat{v})) \\ &\le (1+\epsilon/5)\mathrm{vol}(\mathrm{P}(u,\widehat{v})) \end{aligned}$$

where the second line follows by Theorem 4. Also by Theorems 3 and 4 we have

$$\begin{aligned} |\Sigma_{u,v}| &\ge (1-\epsilon/15)\mathrm{vol}(\mathrm{P}(u,v)) \\ &\ge (1-\epsilon/15)\mathrm{vol}(\mathrm{P}(u,\widehat{v}))/(1+\epsilon/15) \\ &\ge (1-\epsilon/5)\mathrm{vol}(\mathrm{P}(u,\widehat{v})). \end{aligned}$$

By (9), the product of $\mathrm{vol}(\mathrm{P}(u,\widehat{v}))$ and $\sum_t |\Sigma_{(r'_1,\ldots,r'_\ell),t}|$ approximates $|\Sigma_{r',c'}|$ within $(1 \pm \epsilon/5)$.

We calculate $\sum_t |\Sigma_{(r'_1,\ldots,r'_\ell),t}|$ directly as follows: Since we are summing over all possible column sums t, we are simply counting the number of $\ell \times n'$ tables with the row sums $(r'_1,\ldots,r'_\ell)$ (and any column sums). This is equal to the product of the terms $\binom{r'_i+n'-1}{n'-1}$ over all i such that $1 \le i \le \ell$ (the term for i counts the number of ways of partitioning r'_i into an ordered list of n' non-negative integers).

We use the algorithm of Kannan, Lovász and Simonovits [16] to approximate $\mathrm{vol}(\mathrm{P}(u,\widehat{v}))$ within a factor of $(1 \pm \epsilon/5)$ with probability at least $1-\eta'$. Taking the product of this value and $\sum_t |\Sigma_{(r'_1,\ldots,r'_\ell),t}|$, we will approximate $|\Sigma_{r',c'}|$ within a factor of $(1 \pm \epsilon)$, with probability at least $1-\eta'$.

To bound the running time for Step 2 of the algorithm, we use the O^* notation, where we ignore logarithmic factors as well as constant factors.

The algorithm of Kannan, Lovász and Simonovits [16] approximates the volume of a convex body P in d dimensions to within $(1 \pm \epsilon)$ of its true value with high probability by sampling $O^*(d^5/\epsilon^2)$ random d-dimensional points and for each of these points, performing an *oracle call* to test whether the point lies in the convex body. The total number of random bits used to generate all the points that are tested is $O^*(d^6/\epsilon^2)$.

The convex polytopes that we construct (either in Case 1 or Case 3) have dimension less than or equal to nm. Also, for the convex polytopes $\mathrm{P}(u, \widehat{v})$ that we consider (defined in Section 3), we can test a point for membership of $\mathrm{P}(u, \widehat{v})$ using $O(mn)$ arithmetic operations. Therefore we can use the algorithm of Kannan, Lovász and Simonovits [16] to approximate $\mathrm{vol}(\mathrm{P}(u, \widehat{v}))$ (or $\mathrm{vol}(\mathrm{P}(r', c'))$, in Case 1) within $(1 \pm \epsilon/5)$ (with probability at least $1 - \eta'$) using $O^*(n^6/\epsilon^2)$ arithmetic operations.

The number of arithmetic operations used to approximate $|\Sigma_{r-s,(c_{k+1},\ldots,c_n)}|$ is dominated by the number of arithmetic operations of the volume estimation algorithm. Also, we can assume that all the arithmetic operations are carried out on numbers of size $O^*(N^{mn})$, and therefore we can assume that each arithmetic operation takes $O^*(n^2)$ time. Therefore the time to estimate $|\Sigma_{r-s,(c_{k+1},\ldots,c_n)}|$, for any $s \in S$ is

$$O^*(n^8/\epsilon^2)$$

By (6), we will estimate $|\Sigma_{r-s,(c_{k+1},\ldots,c_n)}|$ for $O(n^{6m^2}/\epsilon^{2m^2})$ different $s \in S$. The total running time to estimate all these values is

$$O^* \left(\frac{n^{6m^2+8}}{\epsilon^{2m^2+2}} \right)$$

2.3 Step 3

Finally, in Step 3, we use (1) of Observation 1 to construct an estimate $S_{r,c}$ of $|\Sigma_{r,c}|$, using the exact values of $|\Sigma_{s,(c_1,\ldots,c_k)}|$ for $s \in S$ (constructed in Step 1), and the estimates of $|\Sigma_{r-s,(c_{k+1},\ldots,c_n)}|$ for $s \in S$ (constructed in Step 2).

By definition of $\eta' = \eta/|S|$, we know that with probability at least $(1 - \eta)$, all of the estimates constructed in Step 2 lie within $(1 \pm \epsilon)$ of their true values.

Therefore

$$||\Sigma_{r,c}| - S_{r,c}| \leq \epsilon|\Sigma_{r,c}|$$

with probability at least $(1-\eta)$.

Combining the running times of Step 1 and Step 2, the running time of our entire algorithm is

$$O^*\left(\frac{n^{12m^2}}{\epsilon^{4m^2}}\right).$$

3 Approximating $|\Sigma_{u,v}|$ by the volume of a convex body

In this section we prove the claim that the number of contingency tables with given row and column sums can be closely approximated by the volume of a convex polytope, if the row and column sums are large enough. We begin by introducing some notation. Let $u = (u_1, \ldots, u_{m'})$ be a list of row sums and $v = (v_1, \ldots, v_{n'})$ be a list of column sums. Let N' be the table sum. Then $\Sigma_{u,v}$ is equivalent to the set of non-negative integer solutions for the following system of inequalities (see, for example, Dyer, Kannan and Mount [11]):

$$\sum_{j=1}^{n'-1} X_{i,j} \leq u_i \qquad \text{for } 1 \leq i \leq m'-1 \tag{10}$$

$$\sum_{i=1}^{m'-1} X_{i,j} \leq v_j \qquad \text{for } 1 \leq j \leq n'-1 \tag{11}$$

$$\sum_{i=1}^{m'-1} \sum_{j=1}^{n'-1} X_{i,j} \geq N' - u_{m'} - v_{n'} \tag{12}$$

In this setting we assume:

$X_{i,n'} = u_i - \sum_{j=1}^{n'-1} X_{i,j}$ for $i \leq m'-1$;
$X_{m',j} = v_j - \sum_{i=1}^{m'-1} X_{i,j}$ for $j \leq n'-1$, and
$X_{n',m'} = \sum_{i=1}^{m'-1} \sum_{j=1}^{n'-1} X_{i,j} - (N' - v_{n'} - u_{m'})$.

In this section and the next one, we work in the $(m'-1)(n'-1)$-dimensional space and assume that i ranges over $1 \leq i \leq m'-1$ and j ranges over $1 \leq j \leq n'-1$.

We define $\mathrm{P}(u,v)$ as the convex polytope consisting of the set of non-negative real solutions for (10), (11) and (12).

For any convex body P and any $\alpha > 0$, we define the dilation of P by α to be the set $\alpha \mathrm{P} = \{\alpha X : X \in \mathrm{P}\}$. It is well-known that for any d-dimensional convex body P, $\mathrm{vol}(\alpha \mathrm{P}) = \alpha^d \mathrm{vol}(\mathrm{P})$ (see Corollary 15, page 101 of Kelley and Srinivasan [18]).

Theorem 3 *Let n be an integer and p and q be defined as in Section 2. Let $u = (u_1, \ldots, u_{m'})$ be a list of row sums such that $u_i \geq n^q$ for every i, and $v = (v_1, \ldots, v_{n'})$ be a list of column sums such that $v_j \geq n^p/2$ for every j (by construction $m' \leq m$ and $n' \leq n$). Then*

$$(1 - \frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,v)) \leq |\Sigma_{u,v}| \leq (1 + \frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,v)).$$

Proof: We assume without loss of generality that $u_{m'}$ is the largest row sum among the u_i, and that $v_{n'}$ is the largest column sum among the v_j. Therefore $u_{m'} \geq N'/m'$ and $v_{n'} \geq N'/n'$.

The following interpretation of $|\Sigma_{u,v}|$ will be useful: for each $Z \in \Sigma_{u,v}$, we define a hypercube $H(Z)$ such that $X \in H(Z)$ iff $0 \leq X_{i,j} - Z_{i,j} < 1$ for all $1 \leq i \leq m'-1$ and $1 \leq j \leq n'-1$. Then every point in $\mathrm{P}(u,v)$ is associated with at most one integer point $Z \in \Sigma_{u,v}$. Also, for every $Z \in \Sigma_{u,v}$, the volume of the hypercube associated with Z, denoted $\mathrm{vol}(H(Z))$, is exactly 1 (though some of the hypercube $H(Z)$ may lie outside $\mathrm{P}(u,v)$).

In part (i) of this proof we will define two extra convex polytopes called $\mathrm{P}^-(u,v)$ and $\mathrm{P}^+(u,v)$. We will show that

$$\mathrm{P}^-(u,v) \subseteq \cup_{Z \in \Sigma_{u,v}} H(Z) \quad \text{and} \quad \cup_{Z \in \Sigma_{u,v}} H(Z) \subseteq \mathrm{P}^+(u,v).$$

As $\mathrm{vol}(\cup_{Z \in \Sigma_{u,v}} H(Z)) = |\Sigma_{u,v}|$, this shows

$$\mathrm{vol}(\mathrm{P}^-(u,v)) \leq |\Sigma_{u,v}| \leq \mathrm{vol}(\mathrm{P}^+(u,v)). \tag{13}$$

In Part (ii) we will show that

$$(1 - \frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,v)) \leq \mathrm{vol}(\mathrm{P}^-(u,v))$$

and

$$\mathrm{vol}(\mathrm{P}^+(u,v)) \leq (1 + \frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,v)).$$

Putting this together with (13), we will have

$$(1 - \frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,v)) \leq |\Sigma_{u,v}| \leq (1 + \frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,v))$$

as required.

(i): Let $\mathrm{P}^{-}(u,v)$ be the set of all real $(m'-1)(n'-1)$-dimensional points X with non-negative entries that satisfy the following three sets of inequalities:

$$\sum_{j=1}^{n'-1} X_{i,j} \le u_i \qquad \text{for } 1 \le i \le m'-1 \tag{14}$$

$$\sum_{i=1}^{m'-1} X_{i,j} \le v_j \qquad \text{for } 1 \le j \le n'-1 \tag{15}$$

$$\sum_{i=1}^{m'-1} \sum_{j=1}^{n'-1} X_{i,j} \ge N' - u_{m'} - v_{n'} + (m'-1)(n'-1) \tag{16}$$

It should be obvious that $\mathrm{P}^{-}(u,v) \subseteq \mathrm{P}(u,v)$. We will show something stronger. Let $X \in \mathrm{P}^{-}(u,v)$, and let Z be the unique point with integer entries such that $X \in H(Z)$. We will show $Z \in \mathrm{P}(u,v)$. Then since Z is an integer point by definition, we will have $Z \in \Sigma_{u,v}$.

By definition of $H(Z)$ and the fact that the $X_{i,j}$ values are non-negative, we know $Z_{i,j} \ge 0$ for all $1 \le i \le m'-1$, $1 \le j \le n'-1$.

Also, because $Z_{i,j} \le X_{i,j}$ for all $1 \le i \le m'-1$, $1 \le j \le n'-1$, therefore (14) and (15) imply that Z satisfies (10) and (11) for $\mathrm{P}(u,v)$.

Finally,

$$\sum_{i=1}^{m'-1} \sum_{j=1}^{n'-1} Z_{i,j} \ge (\sum_{i=1}^{m'-1} \sum_{j=1}^{n'-1} X_{i,j}) - (m'-1)(n'-1),$$

and combining this with (16), we have

$$\sum_{i=1}^{m'-1} \sum_{j=1}^{n'-1} Z_{i,j} \ge N' - u_{m'} - v_{n'},$$

which is (12).

So $Z \in \Sigma_{u,v}$. Therefore $\mathrm{P}^{-}(u,v) \subseteq \cup_{Z \in \Sigma_{u,v}} H(Z)$.

Define $\mathrm{P}^{+}(u,v)$ to be the set of all real $(m'-1)(n'-1)$-dimensional points X with non-negative entries that satisfy the following inequalities:

$$\sum_{j=1}^{n'-1} X_{i,j} \le u_i + (n'-1) \qquad \text{for } 1 \le i \le m'-1 \tag{17}$$

$$\sum_{i=1}^{m'-1} X_{i,j} \le v_j + (m'-1) \qquad \text{for } 1 \le j \le n'-1 \tag{18}$$

$$\sum_{i=1}^{m'-1} \sum_{j=1}^{n'-1} X_{i,j} \geq N' - u_{m'} - v_{n'} \tag{19}$$

Clearly $\mathrm{P}(u,v) \subseteq \mathrm{P}^+(u,v)$. Now let $Z \in \Sigma_{u,v}$. Then Z is also in $\mathrm{P}(u,v)$ and satisfies (10), (11), and (12). We will show that $H(Z) \subseteq \mathrm{P}^+(u,v)$.

Let $X \in H(Z)$, so therefore $X_{i,j} \geq Z_{i,j}$ for all $1 \leq i \leq m'-1$, $1 \leq j \leq n'-1$. Therefore all of the entries of X are non-negative.

By (12) and by $X_{i,j} \geq Z_{i,j}$, we have $\sum_{i=1}^{m'-1} \sum_{j=1}^{n'-1} X_{i,j} \geq N' - u_{m'} - v_{n'}$, which is (19).

By definition of $H(Z)$,

$$\sum_{j=1}^{n'-1} X_{i,j} \leq (\sum_{j=1}^{n'-1} Z_{i,j}) + (n'-1),$$

and combining this with (10), we obtain (17). By a similar argument, X satisfies (18).

Therefore $\cup_{Z \in \Sigma_{u,v}} H(Z) \subseteq \mathrm{P}^+(u,v)$.

Therefore we have shown that

$$\begin{aligned} \mathrm{P}^-(u,v) \subseteq \cup_{Z \in \Sigma_{u,v}} H(Z) \quad & \text{and} \\ \cup_{Z \in \Sigma_{u,v}} H(Z) \subseteq \mathrm{P}^+(u,v), & \end{aligned}$$

and therefore we have proved (13), as required.

(ii): We define $\delta = \epsilon/20m'n'$. Note that $n^{-p_\epsilon} = \epsilon/20mn$, which is at most δ. Thus $n^{p_\epsilon} \geq 1/\delta$.

For this section of the proof, it will be useful to move the origin to a point lying inside $\mathrm{P}(u,v)$. Let p' be the real $(m'-1)(n'-1)$-dimensional point defined by $p'_{i,j} =_{def} u_i v_j / N'$. We move the origin of $\mathrm{P}(u,v)$ to p' as follows: substituting $Y + p'$ for X in (10), (11) and (12), we find that the point X lies in $\mathrm{P}(u,v)$ iff the point $Y = X - p'$ satisfies $Y_{i,j} \geq -u_i v_j / N'$ for all $1 \leq i \leq m'-1$, $1 \leq j \leq n'-1$ and also satisfies the following system of inequalities:

$$\sum_{j=1}^{n'-1} Y_{i,j} \leq \frac{u_i v_{n'}}{N'} \qquad \text{for } 1 \leq i \leq m'-1 \tag{20}$$

$$\sum_{i=1}^{m'-1} Y_{i,j} \leq \frac{u_{m'} v_j}{N'} \qquad \text{for } 1 \leq j \leq n'-1 \tag{21}$$

$$\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j} \geq -\frac{u_{m'}v_{n'}}{N'} \tag{22}$$

Let $\mathrm{P}'(u,v)$ be the set of real $(m'-1)(n'-1)$-dimensional points Y that satisfy (20)-(22) and satisfy $Y_{i,j} \geq -u_i v_j/N'$ for all $1 \leq i \leq m'-1$, $1 \leq j \leq n'-1$. Clearly

$$\mathrm{vol}(\mathrm{P}'(u,v)) = \mathrm{vol}(\mathrm{P}(u,v)).$$

We now move the origin for the polytopes $\mathrm{P}^-(u,v)$ and $\mathrm{P}^+(u,v)$, using the same point p'. We define two more transformed convex polytopes $\mathrm{Q}^-(u,v)$ and $\mathrm{Q}^+(u,v)$, where

$$\begin{aligned}&\mathrm{vol}(\mathrm{P}^-(u,v)) = \mathrm{vol}(\mathrm{Q}^-(u,v)) \qquad \text{and}\\ &\mathrm{vol}(\mathrm{P}^+(u,v)) = \mathrm{vol}(\mathrm{Q}^+(u,v)).\end{aligned}$$

$\mathrm{Q}^-(u,v)$ is the set of points Y satisfying $Y_{i,j} \geq -u_i v_j/N'$ for all $1 \leq i \leq m'-1$, $1 \leq j \leq n'-1$ and satisfying

$$\sum_{j=1}^{n'-1} Y_{i,j} \leq \frac{u_i v_{n'}}{N'} \qquad \text{for } 1 \leq i \leq m'-1 \tag{23}$$

$$\sum_{i=1}^{m'-1} Y_{i,j} \leq \frac{u_{m'} v_j}{N'} \qquad \text{for } 1 \leq j \leq n'-1 \tag{24}$$

$$\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j} \geq -\frac{u_{m'}v_{n'}}{N'} + (m'-1)(n'-1) \tag{25}$$

$\mathrm{Q}^+(u,v)$ is the set of points Y satisfying $Y_{i,j} \geq -u_i v_j/N'$ for all $1 \leq i \leq m'-1$, $1 \leq j \leq n'-1$ and satisfying

$$\sum_{j=1}^{n'-1} Y_{i,j} \leq \frac{u_i v_{n'}}{N'} + (n'-1) \qquad \text{for } 1 \leq i \leq m'-1 \tag{26}$$

$$\sum_{i=1}^{m'-1} Y_{i,j} \leq \frac{u_{m'} v_j}{N'} + (m'-1) \qquad \text{for } 1 \leq j \leq n'-1 \tag{27}$$

$$\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j} \geq -\frac{u_{m'}v_{n'}}{N'} \tag{28}$$

We prove $(1-\delta)\mathrm{P}'(u,v) \subseteq \mathrm{Q}^-(u,v)$. Let $Y \in (1-\delta)\mathrm{P}'(u,v)$, so $Y/(1-$

$\delta) \in \mathrm{P}'(u,v)$. We show that Y satisfies the lower bounds for $\mathrm{Q}^-(u,v)$ and Inequalities (23)-(25).

Lower bounds: The lower bounds for $\mathrm{P}'(u,v)$ ensure that $Y_{i,j} \geq -(1-\delta)u_iv_j/N'$ for all $1 \leq i \leq m'-1$, $1 \leq j \leq n'-1$; therefore $Y_{i,j} \geq -u_iv_j/N'$ holds trivially.

Inequality (23): By (20), $\sum_{j=1}^{n'-1} Y_{i,j} \leq (1-\delta)u_iv_{n'}/N'$, which is less than $u_iv_{n'}/N'$.

Inequality (24) follows by an similar argument.

Inequality (25): By (22), we have

$$\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j} \geq -\frac{u_{m'}v_{n'}}{N'} + \delta\frac{u_{m'}v_{n'}}{N'}.$$

By definition, $\delta u_{m'}v_{n'}/N' \geq \delta v_{n'}/m' \geq \delta n^{p-1}/2$ (using $n \geq m \geq m'$). Therefore by definition of p and by $n^{p_\epsilon} \geq 1/\delta$, we find

$$\begin{aligned}\delta\frac{u_{m'}v_{n'}}{N'} \geq \quad \delta\frac{n^{p-1}}{2} = & \quad \delta\frac{n^{2(m-1)(p_\epsilon+2)}}{2} \\ \geq & \quad \frac{n^{4(m-1)}}{2} \\ \geq & \quad (m'-1)(n'-1),\end{aligned}$$

where the second last step follows by $m \geq 2$ and $n^{p_\epsilon} \geq 1/\delta$, and the last step follows by $m-1 \geq 1$ and $n \geq m \geq 2$. Then

$$\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j} \geq -\frac{u_{m'}v_{n'}}{N'} + (m'-1)(n'-1),$$

which is (25).

Now we show $\mathrm{Q}^+(u,v) \subseteq (1+\delta)\mathrm{P}'(u,v)$. Let $Y \in \mathrm{Q}^+(u,v)$. We show that $Y/(1+\delta)$ satisfies the lower bounds for $\mathrm{P}'(u,v)$ and Inequalities (20)-(22).

Lower bounds: By definition of $\mathrm{Q}^+(u,v)$, we know $Y_{i,j} \geq -u_iv_j/N'$ for all $1 \leq i \leq m'-1$, $1 \leq j \leq n'-1$. Then $Y_{i,j} \geq -(1+\delta)u_iv_j/N'$ holds trivially, so $Y/(1+\delta)$ satisfies the lower bounds for $\mathrm{P}'(u,v)$.

Inequality (20): By (26),

$$\sum_{j=1}^{n'-1} Y_{i,j} \leq \frac{u_iv_{n'}}{N'} + (n'-1).$$

Define $\delta' = (n'-1)N'/u_i v_{n'}$, so we have

$$\sum_{j=1}^{n'-1} Y_{i,j} \leq (1+\delta')\frac{u_i v_{n'}}{N'}.$$

Then by $N'/v_{n'} \leq n'$ and $u_i \geq n^q$, we have $\delta' \leq 1/n^{q-2}$. By definition $q-2 = p_\epsilon$, so we have $\delta' \leq \delta$. Therefore $Y/(1+\delta)$ satisfies (20).

Inequality (21): By (27),

$$\sum_{i=1}^{m'-1} Y_{i,j} \leq \frac{u_{m'} v_j}{N'} + (m'-1).$$

Define $\delta'' = (m'-1)N'/u_{m'} v_j$, and write

$$\sum_{i=1}^{m'-1} Y_{i,j} \leq (1+\delta'')\frac{u_{m'} v_j}{N'}.$$

Applying $N'/u_{m'} \leq m'$ and $v_j \geq n^p/2$, and using our assumptions that $m' \leq m$ and $m \leq n$, we have $\delta'' \leq 2/n^{p-2}$. By definition of p and by $n^{-p_\epsilon} \leq \delta$, we have $\delta'' \leq \delta$, and $Y/(1+\delta)$ satisfies (21).

Inequality (22): By (28), $\sum_{i=1}^{m'} \sum_{j=1}^{n'-1} Y_{i,j} \geq -u_{m'} v_{n'}/N'$. But $-u_{m'} v_{n'}/N' \geq -(1+\delta) u_{m'} v_{n'}/N'$, so $Y/(1+\delta)$ satisfies (22).

Now we have $(1-\delta)\mathrm{P}'(u,v) \subseteq \mathrm{Q}^-(u,v)$ and $\mathrm{Q}^+(u,v) \subseteq (1+\delta)\mathrm{P}'(u,v)$, and this gives

$$\begin{aligned} \mathrm{vol}((1-\delta)\mathrm{P}'(u,v)) &\leq \mathrm{vol}(\mathrm{Q}^-(u,v)) \quad \text{and} \\ \mathrm{vol}(\mathrm{Q}^+(u,v)) &\leq \mathrm{vol}((1+\delta)\mathrm{P}'(u,v)). \end{aligned}$$

Also

$$\begin{aligned} \mathrm{vol}((1-\delta)\mathrm{P}'(u,v)) &= (1-\delta)^{(m'-1)(n'-1)}\mathrm{vol}(\mathrm{P}'(u,v)), \\ \mathrm{vol}((1+\delta)\mathrm{P}'(u,v)) &= (1+\delta)^{(m'-1)(n'-1)}\mathrm{vol}(\mathrm{P}'(u,v)). \end{aligned}$$

But $(1-\delta)^{(m'-1)(n'-1)} \geq (1-(m'-1)(n'-1)\delta)$, and by the definition of δ, this is at least $(1-\epsilon/20)$. Therefore

$$(1-\frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}'(u,v)) \leq \mathrm{vol}(\mathrm{Q}^-(u,v)).$$

Then by $\mathrm{vol}(\mathrm{P}'(u,v)) = \mathrm{vol}(\mathrm{P}(u,v))$ and $\mathrm{vol}(\mathrm{Q}^{-}(u,v)) = \mathrm{vol}(\mathrm{P}^{-}(u,v))$, we have

$$(1-\frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,v)) \leq \mathrm{vol}(\mathrm{P}^{-}(u,v)). \tag{29}$$

Also $(1+\delta)^{(m'-1)(n'-1)} \leq e^{\epsilon/20}$ (using $(1+x/n)^n \leq e^x$), and since $\epsilon < 1$, this is at most $(1+\epsilon/15)$. Therefore

$$\mathrm{vol}(\mathrm{Q}^{+}(u,v)) \leq (1+\frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}'(u,v)),$$

and by $\mathrm{vol}(\mathrm{P}'(u,v)) = \mathrm{vol}(\mathrm{P}(u,v))$ and $\mathrm{vol}(\mathrm{Q}^{+}(u,v)) = \mathrm{vol}(\mathrm{P}^{+}(u,v))$,

$$\mathrm{vol}(\mathrm{P}^{+}(u,v)) \leq (1+\frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,v)). \tag{30}$$

Combining (30) and (29) with (13), we have our result. □

4 Approximating the volume of a convex body by another convex body

In this section we prove the second claim made in Case 3 of Step 2 of our algorithm. We will use notation from Sections 2 and 3 and some of the ideas from Section 3.

Theorem 4 *Let $(u_1,\ldots,u_{m'})$ and $(v_1,\ldots,v_{n'})$ be lists of row and column sums such that $m' \leq m-1$, $n' \leq n$, $u_i \geq n^q$ for all i and $v_j \geq n^p/2$ for all j. Suppose that $(\widehat{v}_1,\ldots,\widehat{v}_{n'})$ is another list of column sums satisfying $\widehat{v}_j \geq n^p/2$ for all j, and also satisfying $|v_j - \widehat{v}_j| \leq R$ for all j. Then*

$$\mathrm{vol}(\mathrm{P}(u,v)) \leq (1+\frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,\widehat{v})).$$

Proof: Again, let $\delta = \epsilon/20m'n'$.

Assume without loss of generality that $v_{n'}$ is the largest column sum among the v_j.

Let p' be the real $(m'-1)(n'-1)$-dimensional point defined by $p'_{i,j} =_{def} u_i v_j/N'$. We will use the same trick that we used in part (ii) of Theorem 3, and consider the convex polytope $\mathrm{P}'(u,v)$ centred at this point.

Remember that $\mathrm{vol}(\mathrm{P}'(u,v)) = \mathrm{vol}(\mathrm{P}(u,v))$.

We now construct $\mathrm{P}'(u,\widehat{v})$ by taking the identical point p' that we used for $\mathrm{P}'(u,v)$ and letting $Y \in \mathrm{P}'(u,\widehat{v})$ iff $Y + p' \in \mathrm{P}(u,\widehat{v})$ (remember that this center point p' is defined in terms of the u_i and v_j values, rather than the u_i and $\widehat{v}_j$ values). Then we consider $(1+\delta)\mathrm{P}'(u,\widehat{v})$. Then Y is an element of $(1+\delta)\mathrm{P}'(u,\widehat{v})$ iff $Y_{i,j} \geq -(1+\delta)u_iv_j/N'$ for all i,j and

$$\sum_{j=1}^{n'-1} Y_{i,j} \leq (1+\delta)\frac{u_iv_{n'}}{N'} \qquad \text{for } 1 \leq i \leq m'-1 \tag{31}$$

$$\sum_{i=1}^{m'-1} Y_{i,j} \leq (1+\delta)((\widehat{v}_j - v_j) + \frac{u_{m'}v_j}{N'}) \qquad \text{for } 1 \leq j \leq n'-1 \tag{32}$$

$$\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j} \geq (1+\delta)((v_{n'} - \widehat{v}_{n'}) - \frac{u_{m'}v_{n'}}{N'}) \tag{33}$$

We will show $\mathrm{P}'(u,v) \subseteq (1+\delta)\mathrm{P}'(u,\widehat{v})$. Within this proof we will show that the quantity $(v_{n'} - \widehat{v}_{n'}) - u_{m'}v_{n'}/N'$ (lower bound on $\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j}$ for $\mathrm{P}'(u,\widehat{v})$) is negative and that each of the $(\widehat{v}_j - v_j) + u_{m'}v_j/N'$ values (upper bounds on $\sum_{i=1}^{m'-1} Y_{i,j}$ for $\mathrm{P}'(u,\widehat{v})$) is positive.

Let Y be any element of $\mathrm{P}'(u,v)$, so Y satisfies (20)-(22) and $Y_{i,j} \geq -u_iv_j/N'$. We prove $Y \in (1+\delta)\mathrm{P}'(u,\widehat{v})$ by checking that it satisfies the four types of constraints for $(1+\delta)\mathrm{P}'(u,\widehat{v})$: Inequalities (31)-(33), and the lower bounds on the entries of Y.

Lower bounds: We know $Y_{i,j} \geq -u_iv_j/N'$ for all $1 \leq i \leq m'-1$, $1 \leq j \leq n'-1$. Then $Y_{i,j} \geq -(1+\delta)u_iv_j/N'$, as required.

Inequality (31): By (20) we know $\sum_{j=1}^{n'-1} Y_{i,j} \leq u_iv_{n'}/N'$, and by $\delta > 0$, this trivially implies (31).

Inequality (32): Consider the quantity $(1+\delta)(\widehat{v}_j - v_j) + \delta u_{m'}v_j/N'$. We know that $\widehat{v}_j - v_j \geq -R$ and that

$$u_{m'}v_j/N' \geq v_j/m' \geq n^p/2m'.$$

Therefore $(1+\delta)(\widehat{v}_j - v_j) + \delta u_{m'}v_j/N'$ is at least as big as $\delta n^p/2m' - 2R$. By (8) and by $n^{p_\epsilon} \geq 1/\delta$,

$$\begin{aligned}\delta n^p/2m' - 2R &= R(\delta n^p/2m'R - 2)\\ &\geq R(\delta n^{(p-1)/2}/2m' - 2)\\ &= R(\delta n^{(m-1)(p_\epsilon+2)}/2m' - 2)\end{aligned}$$

$$\begin{aligned}&\geq R(\delta n^{p_\epsilon} n^{2(m-1)}/2m' - 2)\\&\geq R(n^{2(m-1)}/2m' - 2)\\&\geq 0,\end{aligned}$$

where the last step follows by $n \geq m \geq 2$ and $m' \leq m-1$. By (21), we know $\sum_{i=1}^{m'-1} Y_{i,j}$ is bounded above by $u_{m'}v_j/N'$. Therefore we have

$$\begin{aligned}\sum_{i=1}^{m'-1} Y_{i,j} &\leq \frac{u_{m'}v_j}{N'} + (1+\delta)(\widehat{v}_j - v_j) + \delta\frac{u_{m'}v_j}{N'}\\&= (1+\delta)((\widehat{v}_j - v_j) + \frac{u_{m'}v_j}{N'}),\end{aligned}$$

so (32) is satisfied.

Inequality (33): Consider $(1+\delta)(v_{n'} - \widehat{v}_{n'}) - \delta u_{m'}v_{n'}/N'$. Using $v_{n'} - \widehat{v}_{n'} \leq R$ and $u_{m'}v_{n'}/N' \geq v_{n'}/m' \geq n^p/2m'$, we have

$$\begin{aligned}(1+\delta)(v_{n'} - \widehat{v}_{n'}) - \delta u_{m'}v_{n'}/N' &\leq 2R - \delta n^p/2m'\\&\leq 0\end{aligned}$$

because (8) and $n^{p_\epsilon} \geq 1/\delta$ imply that $2R - \delta n^p/2m'$ is negative. By (22), the double sum $\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j}$ is bounded below by $-u_{m'}v_{n'}/N'$. Therefore

$$\begin{aligned}\sum_{i=1}^{m'-1}\sum_{j=1}^{n'-1} Y_{i,j} &\geq -\frac{u_{m'}v_{n'}}{N'} + (1+\delta)(v_{n'} - \widehat{v}_{n'}) - \delta\frac{u_{m'}v_{n'}}{N'}\\&= (1+\delta)((v_{n'} - \widehat{v}_{n'}) - \frac{u_{m'}v_{n'}}{N'})\end{aligned}$$

so (33) is satisfied.

Then $\mathrm{P}'(u,v) \subseteq (1+\delta)\mathrm{P}'(u,\widehat{v})$ and therefore

$$\mathrm{vol}(\mathrm{P}'(u,v)) \leq (1+\delta)^{(m'-1)(n'-1)}\mathrm{vol}(\mathrm{P}'(u,\widehat{v})).$$

By the same argument given at the end of Theorem 3, we obtain

$$\mathrm{vol}(\mathrm{P}'(u,v)) \leq (1+\frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}'(u,\widehat{v})),$$

or equivalently,

$$\mathrm{vol}(\mathrm{P}(u,v)) \leq (1+\frac{\epsilon}{15})\mathrm{vol}(\mathrm{P}(u,\widehat{v})).$$

□

5 Generating a contingency table almost uniformly at random

An *almost-uniform sampler* for contingency tables is an algorithm that takes a list of row sums r, a list of column sums c and an error parameter $\varepsilon \in (0,1)$, and returns an element $X \in \Sigma_{r,c}$ with probability $\sigma(X)$, such that

$$\sum_{X \in \Sigma_{r,c}} |\sigma(X) - |\Sigma_{r,c}|^{-1}| \leq \varepsilon.$$

The sampler is a *polynomial almost-uniform sampler (paus)* if it runs in time polynomial in the number of rows and columns, the table sum, and ε^{-1}. The sampler is a *fully polynomial almost-uniform sampler (fpaus)* if the dependence on the error parameter is polynomial in $(\log \varepsilon^{-1})$.

The error term $\sum_{X \in \Sigma_{r,c}} |\sigma(X) - |\Sigma_{r,c}|^{-1}|$ is the *variation distance* between the output distribution of our sampler and the uniform distribution on $\Sigma_{r,c}$.

We now describe how to convert our fpras into a paus for the set of contingency tables with row sums r and column sums c, when the number of rows is constant. If $\varepsilon < 1$, we show how to generate a point with probabilities within $1 \pm \varepsilon$ of the uniform distribution on the set of contingency tables. We are currently unable to improve this to an fpaus, since the contingency table problem is not self-reducible, as required by the methods of [15], nor does it apparently even satisfy the weaker condition of [8]. This is a somewhat surprising technical difficulty, given that it has recently been shown that a fpaus does in fact exist for this problem [2].

Let $\epsilon = \varepsilon/5$. We first perform Step 1 from Section 2 and partition the columns into small columns and large columns.

S is the set of ordered partitions s of $\sum_{j=1}^{k} c_j$ into m parts such that $s_i \leq r_i$ for all $1 \leq i \leq m$.

For any $1 \leq h \leq k$, S_h is the set of ordered partitions q of $\sum_{j=1}^{h} c_j$ into m parts.

The dynamic programming algorithm constructs $|\Sigma_{s,(c_1,\ldots,c_k)}|$ for all $s \in S$. It also constructs $|\Sigma_{q,(c_1,\ldots,c_h)}|$, for every $q \in S_h$ and $1 \leq h \leq k$.

Carrying out Step 2 of our original algorithm, we obtain an approximation to $S_{r-s,(c_{k+1},\ldots,c_n)}$, for every $s \in S$, leading to an approximation of $S_{r,c}$.

Let s be any ordered partition of $\sum_{j=1}^{k} c_j$ into m parts such that $s_i \leq r_i$ for all $1 \leq i \leq m$. Then Equation (1) of Observation 1 implies that if we choose a contingency table X according to the uniform distribution on $\Sigma_{r,c}$, the probability $\rho(s)$ that X has the partial row sums s is

$$\rho(s) = \frac{|\Sigma_{s,(c_1,\ldots,c_k)}| \times |\Sigma_{r-s,(c_{k+1},\ldots,c_n)}|}{|\Sigma_{r,c}|}.$$

Define $\widehat{\rho}(s)$ by

$$\widehat{\rho}(s) = \frac{|\Sigma_{s,(c_1,\ldots,c_k)}| \times S_{r-s,(c_{k+1},\ldots,c_n)}}{S_{r,c}}.$$

Since we have an *fpras*, we can ensure that $|\widehat{\rho}(s)/\rho(s) - 1| \leq \epsilon$ for all $s \in S$, with arbitrarily high probability. Therefore if we can

(i) choose $s \in S$ according to the probabilities $\widehat{\rho}(s)$,

(ii) choose an element of $\Sigma_{s,(c_1,\ldots,c_k)}$ within $1 \pm \epsilon$ of the uniform probability,

(iii) choose an element of $\Sigma_{r-s,(c_{k+1},\ldots,c_n)}$ uniformly within $1 \pm \epsilon$ of the uniform probability,

we will generate from a distribution σ with probabilities within $(1 \pm \epsilon)^3$ of the uniform distribution. Therefore the probabilities of our distribution σ will all lie within $(1 \pm 4\epsilon)$ of $|\Sigma_{r,c}|^{-1}$ (using the fact that $\epsilon = \varepsilon/5 \leq 1/5$).

Clearly (i) can be accomplished, since we have explicitly computed the numerators and denominator of all the $\widehat{\rho}(s)$ values.

We now show that we can generate a sample uniformly at random from $\Sigma_{s,(c_1,\ldots,c_k)}$. We construct the values for the hth column of X in decreasing order. Suppose we have already constructed columns $h+2, \ldots, k$ of the table and that s is the current partial row sum for the first $h+1$ rows. From equation (3), we choose $q \in S_h$ $(0 \leq q_i \leq s_i,\ i \in [m])$ with probability $|\Sigma_{q,(c_1,\ldots,c_h)}|/|\Sigma_{s,(c_1,\ldots,c_{h+1})}|$, and set column h to $(s-q)$. We iterate this until all the entries in the small columns have been assigned.

We now complete the ℓ small rows. These are chosen independently to be any ordered partition of r'_i into n' parts $(i \in [\ell])$. This can be done as follows. Choose a sample of size $(n'-1)$ uniformly without replacement from $[r'_i+n'-1]$, and sort to give $k_1 < k_2 \cdots < k_{n'-1}$. Let $k_0 = 0$, $k_{n'} = r'_i+n'$. Then the elements of the partition are $(k_j - k_{j-1} - 1)$ $(j \in [n'])$.

The departure from uniform of the points in the small rows and columns will be very small. (It arises only from the precision of our random number generation.) We can certainly ensure that all probabilities are within $1 \pm \epsilon$ of their target values.

We now subtract the partial column totals over the small columns from the large column totals. We now have to generate an integer point uniformly in a polytope of the form given in (10)–(12). Since all row and column totals are sufficiently large, we can do this by the method given in [11]. Hence we can obtain a sample point with probabilities within $1 \pm \epsilon$ of the uniform distribution on this set.

Finally, to show that the variation distance between the uniform distribution and our output distribution σ is bounded, note that by (i), (ii) and (iii) we have $||\Sigma_{r,c}|^{-1} - \sigma(X)| \leq 4\epsilon|\Sigma_{r,c}|^{-1}$ for all $X \in \Sigma_{r,c}$. Therefore the variation distance satisfies

$$\sum_{X \in \Sigma_{r,c}} ||\Sigma_{r,c}|^{-1} - \sigma(X)| \leq \sum_{X \in \Sigma_{r,c}} 4\epsilon|\Sigma_{r,c}|^{-1} = 4\epsilon < \varepsilon,$$

as required.

Acknowledgments

We are grateful to Leslie Goldberg, Catherine Greenhill and Mark Jerrum for useful discussions concerning the results presented in this paper, and for reading a first draft.

References

[1] A.I. Barvinok, A polynomial-time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, **19**(4), 1994, pp. 769–779.

[2] M. Cryan, M. Dyer, L. Goldberg, M. Jerrum and R. Martin, Rapidly mixing Markov chains for sampling contingency tables with a constant number of rows. *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, 2002, pp. 711–720.

[3] F.K.R Chung, R.L. Graham and S.-T. Yau, On sampling with Markov chains. *Random Structures & Algorithms*, **9**(1–2), 1996, pp. 55–77.

[4] P. Diaconis and B. Efron, Testing for independence in a two-way table: new interpretations of the chi-square statistic (with discussion). *Annals of Statistics*, **13**, 1995, pp. 845–913.

[5] P. Diaconis and A. Gangolli, Rectangular arrays with fixed margins, in: D. Aldous, P.P. Varaiya, J. Spencer and J.M. Steele (Eds.), *Discrete Probability and Algorithms*, IMA Volumes on Mathematics and its Applications, **72**, Springer, New York, 1995, pp. 15–41.

[6] P. Diaconis and L. Saloff-Coste, Random walk on contingency tables with fixed row and column sums. Technical Report, Department of Mathematics, Harvard University, 1995.

[7] M. Dyer, A. Frieze and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM*, **38**(1), 1991, pp. 1–17.

[8] M. Dyer and C. Greenhill, Random walks on combinatorial objects. *Surveys in Combinatorics 1999* (J. D. Lamb and D A. Preece, eds.), London Mathematical Society Lecture Note Series **267**, Cambridge University Press, 1999, pp. 101–136.

[9] M. Dyer and C. Greenhill, Polynomial-time counting and sampling of two-rowed contingency tables. *Theoretical Computer Science*, **246**, 2000, pp. 265–278.

[10] M. Dyer and R. Kannan, On Barvinok's algorithm for counting lattice points in fixed dimension. *Mathematics of Operations Research*, **22**(3), 1997, pp. 545–549.

[11] M. Dyer, R. Kannan and J. Mount, Sampling contingency tables. *Random Structures & Algorithms*, **10**(4), 1997, pp. 487–506.

[12] G. Hadley, Transportation Problems (Chapter 9). *Linear Programming*, Addison-Wesley, Massachusetts, 1962, pp. 273–330.

[13] D. Hernek, Random generation of $2 \times n$ contingency tables. *Random Structures & Algorithms*, **13**(1), 1998, pp. 71–79.

[14] M. Jerrum and A. Sinclair, Markov chain Monte Carlo method: an approach to approximate counting and integration. D.S. Hochbaum (Ed), *Approximation Algorithms for NP-Hard Problems*, PWS, Boston, 1997, pp. 482–520.

[15] M. Jerrum, L.G. Valiant and V.V. Vazirani, Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, **43**, 1986, pp. 169–188.

[16] R. Kannan, L. Lovász and M. Simonovits, Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Structures & Algorithms*, **11**(1), 1997, pp. 1–50.

[17] R. Karp and M. Luby, Monte-Carlo algorithms for enumeration and reliability problems. in *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, ACM Press, 1983, pp. 56–64.

[18] J.L. Kelley and T.P. Srinivasan, *Measure and Integral* (Volume 1), Springer-Verlag Graduate Texts in Mathematics, **116**, Springer, Berlin, 1988.

[19] B. Morris, Improved bounds for sampling contingency tables. *3rd International Workshop on Randomization and Approximation Techniques in Computer Science*, volume 1671 of *Lecture Notes in Computer Science*, 1999, pp. 121–129.

[20] J. Mount, *Application of Convex Sampling to Optimization and Contingency Table Generation.* PhD thesis, Technical report CMU-CS-95-152, Computer Science Department, Carnegie Mellon University, 1995.

[21] A. Sinclair and M. Jerrum, Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, **82**, 1989, pp. 93–133.