Symmetric function algorithm for counting contingency tables with fixed margins

Hashiguchi Hiroki (Science University of Tokyo)

Department of Management Science
Science University of Tokyo
1-3 Kagurazaka Shinjuku-ku Tokyo 162-8601, Japan

Abstract

An exact method is given for counting the number of contingency tables based on the theory of symmetric functions. We systematically calculate each coefficient of complete symmetric function in terms of monomial symmetric functions in order to obtain the number. We give the total number for some $\ell \times m$ contingency tables which have already studied in statistical applications.

1. Introduction

In a variety of combinatorial and statistical applications, one needs to know the number of rectangular arrays of nonnegative integers with given row and column sums. The combinatorial problems include a transportation problem in integer programming. The statistical problems involve goodness of fit test for contingency tables. Algorithms for counting/enumerating contingency tables have been proposed for the purpose of being used in statistics, see Metha and Patel [10], Mount [11] and Saunders [12]. Approximate methods for counting have been given by Gail and Mantel [2] and Diaconis and Efron [13]. An excellent survey can be found in Diaconis and Gangolli [1]. This paper concerns with an exact method for counting based on the theory of symmetric functions ([6], [8]).

Let P_n^k be the set of all partitions of degree k and of length not longer than n, that is,

$$P_n^k = \left\{ \lambda = (\lambda_1, \lambda_2, \dots, \lambda_n) \mid \lambda_1, \dots, \lambda_n \in \mathbb{Z},\ \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0,\ \sum_{i=1}^{n} \lambda_i = k \right\},$$

and let Λ_n denote the ring of symmetric polynomials in $x_1, \dots, x_n$ with rational coefficients. This ring forms a graded ring:

$$\Lambda_n = \bigoplus_{k \geq 0} \Lambda_n^k,$$

where Λ_n^k is a vector space spanned by all homogeneously symmetric polynomials of degree k. Fields of such mathematics as algebra, combinatorics and statistics are interested in several bases for Λ_n^k, which are mentioned in Macdonald [6], Stanley [8], Kendall et al. [5] and Hashiguchi [3].

Given $\lambda \in P_n^k$, define a symmetric polynomial $\mathcal{M}_\lambda(x_1, \ldots, x_n)$

$$\mathcal{M}_\lambda(x_1, \ldots, x_n) = \sum_{(\alpha_1, \alpha_2, \ldots, \alpha_n)} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

where the sum ranges over all distinct permutations $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ of the entries of $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$. The set $\{\mathcal{M}_\lambda(x_1, \ldots, x_n) \mid \lambda \in P_n^k\}$ forms a basis for Λ_n^k. In the theory of symmetric polynomials, the number of variables is usually irrelevant, provided only that it is large enough, and it is often more convenient to work with *symmetric functions* in infinitely many variables. Thus we are ignore the number of variables n, and, in brief, we use notations $\mathcal{M}_\lambda$ and $\lambda \vdash k$ instead of $\mathcal{M}_\lambda(x_1, \ldots, x_n)$ and $\lambda \in P_n^k$, respectively. We call $\mathcal{M}_\lambda$ monomial symmetric function.

Power sum associated with $\lambda \vdash k$ is defined as

$$\mathcal{P}_\lambda = p_{\lambda_1} p_{\lambda_2} \cdots p_{\lambda_n},$$

where $p_i = x_1^i + \cdots + x_n^i$ $(i = 1, 2, \ldots)$, $p_0 = 1$.

Complete symmetric function $\mathcal{H}_\lambda$ is defined as

$$\mathcal{H}_\lambda = h_{\lambda_1} \cdots h_{\lambda_n},$$

where $h_j = \sum_{\lambda \in P_n^j} \mathcal{M}_\lambda$ $(j = 1, 2, \ldots)$, $h_0 = 1$. Both $\{\mathcal{P}_\lambda \mid \lambda \vdash k\}$ and $\{\mathcal{H}_\lambda \mid \lambda \vdash k\}$ form bases for Λ_n^k.

A scalar product can be defined on Λ_n^k by requiring that the basis $\{\mathcal{P}_\lambda \mid \lambda \vdash k\}$ is orthogonal:

$$\langle \mathcal{P}_\lambda, \mathcal{P}_\mu \rangle = z_\lambda \, \delta_{\lambda\mu} \tag{1}$$

where $\delta_{\lambda\mu}$ is the Kronecker delta and

$$z_\lambda = \prod_{i=1}^n i^{a_i} \, a_i!$$

for every partition $\lambda = (1^{a_1}, 2^{a_2}, \ldots, n^{a_n})$ satisfying $\sum_{i=1}^{n} i\, a_i = k$. The orthogonality (1) is equivalent to the duality between $\{\mathcal{M}_\lambda \mid \lambda \vdash k\}$ and $\{\mathcal{H}_\mu \mid \mu \vdash k\}$, i.e.,

$$\langle \mathcal{M}_\lambda,\, \mathcal{H}_\mu \rangle = \delta_{\lambda\mu}.$$

2. Count of contingency tables

Let $\boldsymbol{r} = (r_1, r_2, \ldots, r_\ell)$ and $\boldsymbol{c} = (c_1, c_2, \ldots, c_m)$ be partitions of N. In statistical analysis of contingency tables the number N usually represents a sample size. Let $\Omega_{\boldsymbol{rc}}$ denote the set of all contingency tables in which row i has sum r_i and column j has sum c_j. Thus $\sum_{i,j} t_{i,j} = N$ for every $t = [t_{i,j}]_{1\le i\le \ell, 1\le j\le m} \in \Omega_{\boldsymbol{rc}}$. The following proposition is useful for counting $|\Omega_{\boldsymbol{rc}}|$.

Proposition 1 Let $\boldsymbol{r} = (r_1, \ldots, r_\ell)$ and $\boldsymbol{c} = (c_1, \ldots, c_m) \vdash N$. The following formulas hold.

1. (Macdonald [6]) $\quad h_j = \sum_{\lambda\vdash j} z_\lambda^{-1} \mathcal{P}_\lambda$, i.e., $\mathcal{H}_{\boldsymbol{r}} = \prod_{i=1}^{l} \left(\sum_{\lambda \vdash r_i} z_\lambda^{-1} \mathcal{P}_\lambda \right)$.
2. (Diaconis and Gangolli [1]) $\quad |\Omega_{\boldsymbol{rc}}| = \langle \mathcal{H}_{\boldsymbol{r}},\, \mathcal{H}_{\boldsymbol{c}} \rangle$
3. (Stanley [8]) $\quad \mathcal{H}_{\boldsymbol{r}} = \sum_{\boldsymbol{c}\vdash N} |\Omega_{\boldsymbol{rc}}|\; \mathcal{M}_{\boldsymbol{c}}$

From Proposition 1, the following algorithms are established for computing $|\Omega_{\boldsymbol{rc}}|$ on computer algebra system.

Algorithm 1 (Diaconis and Gangolli [1])

1. Express $\mathcal{H}_{\boldsymbol{r}}$, $\mathcal{H}_{\boldsymbol{c}}$ in terms of $\{\mathcal{P}_\lambda \mid \lambda \vdash N\}$.
2. Compute $\langle \mathcal{H}_{\boldsymbol{r}},\, \mathcal{H}_{\boldsymbol{c}} \rangle$ based on the above results and the orthogonality (1).

On the other hand, Hashiguchi and Niki [4] proposed an algorithm for expanding the product $\mathcal{M}_\lambda \times \mathcal{M}_\mu$ as a linear combination of $\mathcal{M}_\nu$, see Appendix A for details. By using this algorithm, we can calculate the coefficients of $\mathcal{H}_{\boldsymbol{r}}$ in terms of $\{\mathcal{M}_\lambda\}$.

Algorithm 2 (to compute $|\Omega_{\boldsymbol{rc}}|$) Expand repeatedly, term by term in

$$\mathcal{H}_{\boldsymbol{r}} = \prod_{i=1}^{l} h_{r_i} = \prod_{i=1}^{l} \left(\sum_{\lambda\vdash r_i} \mathcal{M}_\lambda \right), \tag{2}$$

by using Hashiguchi and Niki's algorithm until getting the linear combination of $\{\mathcal{M}_\lambda \mid \lambda \vdash N\}$. However, we make no partition both of which length is not longer than m and of which the first element is larger than $c_1 \in \boldsymbol{c}$.

3. Examples

Algorithms 1 and 2 was designed on *Mathematica* which is one of the most famous computer algebra system. A numerical illustration is given as follows.

Example 1 For $\boldsymbol{r} = (4,3,2,2)$ and $\boldsymbol{c} = (6,3,2)$, Algorithm 1 gives

$$\begin{aligned}
&\langle \mathcal{H}_{(4,3,2,2)},\, \mathcal{H}_{(6,3,2)} \rangle \\
&= \left\langle \left(\sum_{\lambda\vdash 4} z_\lambda^{-1}\mathcal{P}_\lambda\right)\left(\sum_{\lambda\vdash 3} z_\lambda^{-1}\mathcal{P}_\lambda\right)\left(\sum_{\lambda\vdash 2} z_\lambda^{-1}\mathcal{P}_\lambda\right)^2,\, \left(\sum_{\lambda\vdash 6} z_\lambda^{-1}\mathcal{P}_\lambda\right)\left(\sum_{\lambda\vdash 3} z_\lambda^{-1}\mathcal{P}_\lambda\right)\left(\sum_{\lambda\vdash 2} z_\lambda^{-1}\mathcal{P}_\lambda\right) \right\rangle \\
&= \left\langle \frac{1}{48}\mathcal{P}_{(4,3,2,2)} + \cdots + \frac{1}{576}\mathcal{P}_{(1^{11})},\, \frac{1}{36}\mathcal{P}_{(6,3,2)} + \cdots + \frac{1}{8640}\mathcal{P}_{(1^{11})} \right\rangle \\
&= \frac{1}{576}\langle \mathcal{P}_{(4,3,2,2)},\, \mathcal{P}_{(4,3,2,2)} \rangle + \frac{1}{2304}\langle \mathcal{P}_{(4,3,2,1,1)},\, \mathcal{P}_{(4,3,2,1,1)} \rangle + \cdots \\
&\quad + \frac{209}{4976640}\langle \mathcal{P}_{(2,1^9)},\, \mathcal{P}_{(2,1^9)} \rangle + \frac{1}{4976640}\langle \mathcal{P}_{(1^{11})},\, \mathcal{P}_{(1^{11})} \rangle \\
&= 136.
\end{aligned}$$

And Algorithm 2 gives

$$\begin{aligned}
\mathcal{H}_{(4,3,2,2)} &\to \left(\mathcal{M}_{(4)} + \cdots + \mathcal{M}_{(2,1,1)}\right)\left(\mathcal{M}_{(3)} + \mathcal{M}_{(2,1)} + \mathcal{M}_{(1,1,1)}\right)\left(\mathcal{M}_{(2)} + \mathcal{M}_{(1,1)}\right)^2 \\
&\to \left(2\mathcal{M}_{(6,1)} + 3\mathcal{M}_{(5,2)} + 6\mathcal{M}_{(4,2,1)} + 4\mathcal{M}_{(5,1,1)}\right)\left(\mathcal{M}_{(2)} + \mathcal{M}_{(1,1)}\right)^2 \\
&\to \left(9\mathcal{M}_{(6,3)} + 17\mathcal{M}_{(6,2,1)} + \cdots + 39\mathcal{M}_{(4,3,2)} + 45\mathcal{M}_{(3,3,3)}\right)\left(\mathcal{M}_{(2)} + \mathcal{M}_{(1,1)}\right) \\
&\to 31\mathcal{M}_{(6,5)} + 88\mathcal{M}_{(6,4,1)} + 136\mathcal{M}_{(6,3,2)} + 97\mathcal{M}_{(5,5,1)} \\
&\qquad + 170\mathcal{M}_{(5,4,2)} + 201\mathcal{M}_{(5,3,3)} + 228\mathcal{M}_{(4,4,3)}
\end{aligned}$$

which shows that $\left|\Omega_{(4,3,2,2),(6,3,2)}\right|$ is 136.

The numbers $|\Omega_{\boldsymbol{rc}}|$ in Table 1 was computed by *Mathematica* program realizing Algorithm 2. Note that contingency tables with $\boldsymbol{r}$ and $\boldsymbol{c}$ in Nos. 1 through 6 can be seen in Metha and Patel [7] and the others in Chapter 5 of Yanagawa [9].

Table 1. Examples for $|\Omega_{\boldsymbol{r}\boldsymbol{c}}|$

No.	$\boldsymbol{r}$	$\boldsymbol{c}$	N	$\lvert\Omega_{\boldsymbol{r}\boldsymbol{c}}\rvert$
1	(5, 5, 5, 5, 4, 4, 4, 4, 4)	(30, 10)	40	35,353
2	(7, 6, 6, 5, 5)	(11, 7, 6, 5)	29	3,187,528
3	(7, 7, 6, 6, 5, 5)	(16, 9, 5, 6)	36	97,080,796
4	(9, 8, 8, 6, 6, 6, 5, 5, 5)	(38, 10, 10)	58	1,326,849,651
5	(7, 7, 6, 5, 5, 4)	(12, 7, 7, 4, 4)	34	2,159,651,513
6	(7, 7, 6, 5, 5, 5, 4)	(15, 8, 7, 5, 4)	39	108,712,356,901
7	(32, 38, 38, 40)	(90, 40, 18)	148	15,272,124
8	(25, 26, 28, 26)	(58, 40, 7)	105	1,106,454
9	(17, 20, 19, 18)	(38, 23, 13)	74	1,107,960
10	(22, 20, 25, 23)	(54, 26, 10)	90	944,944

4. Concluding remarks

Example 1 indicates that Algorithm 1 makes the lowest partition (1^N) in the reverse lexicographic ordering and that Algorithm 2 generates unnecessary terms such as $31\mathcal{M}_{(6,5)}$, $88\mathcal{M}_{(6,4,1)}$. Algorithm 2 is expected to be more efficient than Algorithm 1 when N is large and ℓ and m are small. However, details for their complexities are in future study. Further, there is room for improvement in Algorithm 2 not to make undesirable terms.

A. Algorithm for the product of monomial symmetric functions

Let the symbol '$||$' denote a binary operator of concatenation which means that two sequences are combined. In addition, let 0_k denote the sequence of zeros of length k. For any sequence $T = (t_1, \ldots, t_k)$ of indices, we let $\mu_T = (\mu_{t_1}, \ldots, \mu_{t_k})$ and $\mu_{0_k} \equiv (\overbrace{0 \ldots 0}^{k})$. For any partitions λ, μ, the following algorithm gives the expansion of the $\mathcal{A}_\lambda \cdot \mathcal{A}_\mu$ as the weighted sum of augmented symmetric functions, i.e.,

$$\mathcal{A}_\lambda \cdot \mathcal{A}_\mu = \sum_\nu c[\lambda,\, \mu]_\nu\, \mathcal{A}_\nu, \tag{3}$$

where $\mathcal{A}_\lambda = \left(\prod_{i=1}^{n} a_i!\right) \mathcal{M}_\lambda$ for $\lambda = (1^{a_1}, \ldots, n^{a_n})$.

Algorithm 3 (Hashiguchi and Niki [4])

Input: $\lambda = (\lambda_1 \, \dots \, \lambda_\ell), \mu = (\mu_1 \, \dots \, \mu_m), \, \ell = \ell(\lambda) \geq m = \ell(\mu)$

Output: $r = \sum_\nu c[\lambda, \mu]_\nu \, \mathcal{A}_\nu$ s.t. $r = \mathcal{A}_\lambda \cdot \mathcal{A}_\mu$

1. $r := 0$, $i := 0$.
2. $L := \lambda \, || \, 0_{m-i}$
3. Let M be the set $\{1, 2, \dots, m\}$. $c := \binom{m}{i}$. We take all the subsets of M of size i and write them $I_1, I_2, \dots, I_c$. $j := 1$.
4. $K := 0_\ell \, || \, \mathrm{asort}(M \backslash I_j)$, where $\mathrm{asort}(M \backslash I_j)$ returns the sequence arranged all the elements of $(M \backslash I_j)$ in ascending order.
5. $p := \frac{\ell!}{(\ell-i)!}$. We make all the permutations of $0_{\ell-i} \, || \, \mathrm{asort}(I_j)$ and write them $J_1, J_2, \dots, J_p$. $k := 1$.
6. $J := J_k \, || \, 0_{m-i}$. $N := \mu_J + \mu_K + L$.
7. We sort the elements of N in descending order: $\nu := \mathrm{sort}(N)$.
   $r := r + \mathcal{A}_\nu$.
8. $k := k + 1$. If $k \leq p$ then go back to 6.
   $j := j + 1$. If $j \leq c$ then go back to 4.
   $i := i + 1$. If $i \leq m$ then go back to 2.
   Otherwise return r.

From Algorithm 3, it is easy to derive the algorithm for symbolic product $\mathcal{M}_\lambda \cdot \mathcal{M}_\mu$, namely,

$$\mathcal{M}_\lambda \cdot \mathcal{M}_\mu = \sum_\nu \frac{c[\lambda, \mu]_\nu \pi(\nu)}{\pi(\lambda)\pi(\mu)} \mathcal{M}_\nu \tag{4}$$

where $\pi(\lambda) = \pi_1! \cdots \pi_t!$ for $\lambda = (\lambda_1^{\pi_1} \, \dots \, \lambda_t^{\pi_t})$ and $\lambda_1 > \cdots > \lambda_t$.

If we apply Algorithm 3 to $(2\,1^2)$ and $(3\,2\,1)$, then we have

$$\begin{aligned}\mathcal{A}_{(2\,1^2)} \cdot \mathcal{A}_{(3\,2\,1)} = {} & 2\,\mathcal{A}_{(5\,3\,2)} + 2\,\mathcal{A}_{(4\,2^2)} + 2\,\mathcal{A}_{(4\,3^3)} + 6\,\mathcal{A}_{(4\,3\,2\,1)} + 2\,\mathcal{A}_{(4^2\,1^2)} + 2\,\mathcal{A}_{(5\,3\,1^2)} \\ & + 2\,\mathcal{A}_{(3^3\,1)} + 2\,\mathcal{A}_{(5\,2^2\,1)} + 2\,\mathcal{A}_{(3^2\,2^2)} + 2\,\mathcal{A}_{(4\,2^3)} + 3\,\mathcal{A}_{(3^2\,2\,1^2)} \\ & + 2\,\mathcal{A}_{(4\,2^2\,1^2)} + 2\,\mathcal{A}_{(3\,2^3\,1)} + \mathcal{A}_{(5\,2\,1^3)} + \mathcal{A}_{(4\,3\,1^3)} + \mathcal{A}_{(3\,2^2\,1^3)},\end{aligned}$$

which yields, from (4),

$$\begin{aligned}\mathcal{M}_{(2\,1^2)} \cdot \mathcal{M}_{(3\,2\,1)} = \mathcal{M}_{(5\,3\,2)} &+ 2\,\mathcal{M}_{(4\,2^2)} + 2\,\mathcal{M}_{(4\,3^2)} + 3\,\mathcal{M}_{(4\,3\,2\,1)} + 4\,\mathcal{M}_{(4^2\,1^2)} + 2\,\mathcal{M}_{(5\,3\,1^2)} \\ &+ 6\,\mathcal{M}_{(3^3\,1)} + 2\,\mathcal{M}_{(5\,2^2\,1)} + 4\,\mathcal{M}_{(3^2\,2^2)} + 6\,\mathcal{M}_{(4\,2^3)} + 6\,\mathcal{M}_{(3^2\,2\,1^2)} \\ &+ 4\,\mathcal{M}_{(4\,2^2\,1^2)} + 6\,\mathcal{M}_{(3\,2^3\,1)} + 3\,\mathcal{M}_{(5\,2\,1^3)} + 3\,\mathcal{M}_{(4\,3\,1^3)} + 6\,\mathcal{M}_{(3\,2^2\,1^3)}.\end{aligned}$$

References

[1] Diaconis, P. and Gangolli, A. (1995) Rectangular arrays with fixed margins. In *Discrete Probability and Algorithms* (D. Aldous et al., eds.). 15–41, Springer, New York.

[2] Gail, M. and Mantel, N. (1977). Counting the number of $r \times c$ contingency tables with fixed margins, *J. Amer. Statist. Assoc.*, **77**, 859–862.

[3] Hashiguchi, H. (1999). Computational representations of symmetric polynomials and their applications to distribution theory, Ph.D thesis, Science University of Tokyo.

[4] Hashiguchi, H. and Niki, N. (1995). Symmetric function calculus on computer, *Proc.1st Asian Tech. Conf. Math.*, 626–635.

[5] Kendall, M. G., Stuart, A. and Ord, J. K. (1987). *Kendall's Advanced Theory of Statistics*, **1**, 5th ed., Griffin

[6] Macdonald, I. G. (1995). *Symmetric functions and Hall polynomials, 2nd ed.*, Oxford University Press.

[7] Metha, C. M. and Patel, N. R. (1983). A network algorithm for performing Fisher's exact test in $r \times c$ contingency tables, *J. Amer. Statist. Assoc.*, **78**, 427–434.

[8] Stanley, R. P. (1999). *Enumerative Combinatorics*, vol. 2, Cambridge University Press.

[9] Yanagawa, T. (1986). *Discrete multivariate data analysis*, Kyoritsu, Tokyo (*in Japanese*).

[10] Metha, C. R. and Patel, N. R. (1983). A network algorithm for performing Fisher's exact test in $r \times c$ contingency tables, *J. Amer. Statist. Assoc.*, **78**, 427–434.

[11] Mount, J. (1995). Application of convex sampling to optimization and contingency table generation/counting, Ph. D thesis, Carnegie Mellon University.

[12] Saunders, I. W. (1984). Algorithm AS 205: Enumeration of $R \times C$ tables with repeated row totals, *Appl. Statist.*, 340–352.

[13] Diaconis, P. and Efron, B. (1985). Testing for independence in a two-way table: new interpretations of the chi-square statistics (with discussion), *Ann. Statist.*, **13**, 845–913.