Program Abstracts/Algorithms

MPROB: Computation of multinomial probabilities

MIGUEL A. GARCÍA-PÉREZ
Universidad Complutense, Madrid, Spain

An algorithm for computing exact multinomial probabilities is presented that uses the fewest number of operations that are possible without symbolic simplification of the multinomial coefficient and performs them in a sequence that minimizes the potential for overflow or underflow errors. Multinomial probabilities computed with this algorithm are accurate to machine precision, unlike those based on logarithmic transformations and/or the evaluation of factorials, using Stirling's approximation.

Statistical inference for multinomial processing tree models (Riefer & Batchelder, 1988) often relies on the asymptotic chi-square distribution (García-Pérez, 1994; Riefer & Batchelder, 1991). The asymptotic approximation has been shown to be poor when sample size and/or the number of multinomial cells is small, yielding results that may differ greatly from those of exact multinomial tests (see Chapman, 1976; Larntz, 1978; Lawal, 1984; Read, 1984; Read & Cressie, 1988, chap. 5; Yarnold, 1972). All of the studies just mentioned have focused on the case of symmetric null hypotheses (i.e., cases in which all the cell probabilities are identical), but it has also been shown that the asymptotic approximation is markedly less accurate when testing skewed null hypotheses (Koehler & Larntz, 1980; Núñez-Antón & García-Pérez, 1995). Furthermore, all of the above applies when the null hypothesis is completely specified (i.e., no parameters need to be estimated from the data), and, in addition, the accuracy of the asymptotic chi-square distribution is suspect when parameters need to be estimated from the data (Read & Cressie, 1988). Since statistical inference with multinomial processing tree models often involves (1) small samples from distributions with few cells, (2) skewed hypotheses, and (3) concurrent parameter estimation, exact tests should be preferred over tests based on the asymptotic chi-square distribution.

Exact multinomial tests require the computation of a large number of multinomial probabilities (see Read & Cressie, 1988, pp. 136-138), each of which is likely to incur overflow or underflow errors. Each of these point probabilities must be computed as accurately as possible;

otherwise, their sum will yield a significance level that risks being as poor an approximation as that obtained with the asymptotic chi-square distribution.

Statistical/numerical packages (e.g., IMSL, NAG) do not include subroutines for the computation of multinomial probabilities. Dunlap, Myers, and Silver (1984) evaluated them, using logarithmic transformations and Stirling's formula, but these approximations strongly compromise the accuracy of the results (see below) and, therefore, cannot yield truly exact probabilities. Also, Berry and Mielke (1995) proposed an algorithm for the evaluation of cumulative multinomial probabilities that is inefficient for the computation of point probabilities.

This paper presents an algorithm (MPROB) that computes exact (to machine precision) multinomial probabilities with the fewest operations that are possible without time-consuming symbolic evaluation of the multinomial coefficient. MPROB is designed so as to minimize the potential for overflow and underflow errors, and, although these may still occur with excessively large sample sizes, MPROB is capable of providing exact results in cases where approximation algorithms (Berry & Mielke, 1995; Dunlap et al., 1984) fail. Nevertheless, these overflow and underflow errors will not occur with small sample sizes.

MPROB ALGORITHM

Let $\mathbf{X} = (X_1, X_2, \dots, X_J)$, with $J > 1$, be a random vector, with

$$\sum_{j=1}^J X_j = N$$

and $X_j \geq 0$ for all $1 \leq j \leq J$, and let $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_J)$ be a discrete probability distribution. Then, under $\boldsymbol{\pi}$, any realization $\mathbf{x}$ of $\mathbf{X}$ has a probability given by

$$Pr(\mathbf{X} = \mathbf{x}) = \frac{N!}{x_1!x_2! \dots x_J!} \pi_1^{x_1} \pi_2^{x_2} \dots \pi_J^{x_J}. \quad (1)$$

Consider $\mathbf{x} = (20, 15, 5)$ and $\boldsymbol{\pi} = (0.5, 0.2, 0.3)$. Substituting in Equation 1, expanding the factorials, and removing identical terms in the numerator and denominator results in Equation 2, shown at the bottom of the next page.

Without additional symbolic simplification, the number of operations reduces to $N - x_1$ products in both the numerator and the denominator and J further products of powers. Since the number of products and divisions is identical, they can be interwoven to avoid large partial results (i.e., $21/1 \times 22/2 \times 23/3 \times \dots \times 40/5$). In addition, the last $J - 1$ powers on the right of Equation 2 also expand into $N - x_1$ products that can also be performed concurrently to help keep partial results low-valued. The x_1 products that the first power,

Correspondence concerning this article should be addressed to M. A. García-Pérez, Departamento de Metodología, Facultad de Psicología, Universidad Complutense, Campus de Somosaguas, 28223 Madrid, Spain (e-mail: miguel@psi.ucm.es).

$$\pi_1^{x_1},$$

expands into can also be interwoven, but note that x_1 can be larger than, smaller than, or equal to $N - x_1$. Application of this scheme results in the following sequence of operations:

$$0.5 \frac{21}{1} 0.2 \times 0.5 \frac{22}{2} 0.2 \times \dots \times 0.5 \frac{35}{15} 0.2 \times 0.5 \frac{36}{1} 0.3 \\ \times 0.5 \frac{37}{2} 0.3 \times \dots \times 0.5 \frac{40}{5} 0.3. \quad (3)$$

The least number of operations under the scheme in Equation 3 occurs when x_1 is the largest value among all x_j , as was the case in the example above. Also, the size of each partial result is also kept to the lowest possible value if x_2 is the largest remaining value in x , and so on. Therefore, an efficient algorithm for computing multinomial probabilities should sort the components of x in decreasing order and rearrange the components of π accordingly. The formal description of the MPROB algorithm is thus as follows:

- step 0. Copy input vector x into workspace vector y , and input vector π into workspace vector p . Arrange the components of y in decreasing order and move the components of p accordingly.
- step 1. Set $p = 1$, $t = \rho_1$, $i = 2$, $n = 0$, and $m = y_1$.
- step 2. Set $l = y_i$. For $k = 1$ to l , (1) update $n = n + 1$, (2) if $n > y_1$, then set $t = 1$, and (3) update $p = p \times t \times \rho_i \times (k + m) / k$.
- step 3. If $i = J$, then go to step 5; otherwise, go to step 4.
- step 4. Set $m = m + y_i$, $i = i + 1$, and go to step 2.
- step 5. If $n < y_1$, then for $k = n + 1$ to y_1 , update $p = p \times \rho_1$.

Note also that this algorithm works for all $J > 1$ and, therefore, can also be used to compute binomial probabilities when $J = 2$.

ACCURACY TESTS

Determining the accuracy of point probabilities computed with MPROB is somewhat hampered by the lack of independently obtained exact values to compare the results with. As an indirect assessment, the probability was computed of each and all vectors x under several multinomial distributions, and tests consisted of checking out whether the sum of these probabilities added to unity. For this purpose, MPROB was implemented as a Fortran 77 subroutine using double precision, and the driver programs ran on a DEC Alpha Server 8200.

Tests ran for the factorial combination of 4 values for J (2, 3, 4, and 5), 12 values for N (from 20 to 570, in 50-unit steps), and 20 probability distributions π for each J (as obtained by substituting λ in the equations of Table 1 with values from 0 to .95, in .05-unit steps). Across the 960 tests, the largest amount by which the sum of probabilities differed from unity increased with J : The differences were within $\pm 1.1 \times 10^{-15}$, $\pm 2.5 \times 10^{-14}$, $\pm 1.7 \times 10^{-13}$, and $\pm 1.3 \times 10^{-11}$, respectively, for $J = 2, 3, 4$, and 5. Yet, these largest errors do not give a fair picture of the accuracy of the algorithm: Errors were more than an order of magnitude smaller than these maximal errors in 92, 159, 161, and 171 of the 240 conditions tested, respectively, for $J = 2, 3, 4$, and 5.

The source of the increasing error with increasing J seems to lie in the machine representation of the probabilities making up the multinomial distribution, of which there are more as J increases. Indeed, the errors were slightly larger when the same tests were run representing these probabilities as single-precision reals, and they virtually disappeared when probabilities were represented as quadruple-precision reals. In other words, the accuracy of MPROB is only limited by machine precision.

The fact that the sum of probabilities does not add to unity indicates that some of the point probabilities calculated with MPROB must be slightly erroneous. Consider the case $J = 3$, $N = 570$, and $\lambda = 0$, where MPROB produced probabilities whose sum differed from unity by 2.02×10^{-15} . To determine the magnitude of the individual errors that are responsible for this difference and whether they occur within specific regions of the sample space of the implied multinomial distribution, one could obtain exact results with a symbolic computation package (e.g., Mathematica; Wolfram, 1992) and compare them with those produced by MPROB.

Figure 1 shows graphically the outcome of this comparison across the 163,306 simplex lattice points of the multinomial distribution that results when $J = 3$ and $N = 570$. Differences (represented by lightness) between MPROB and exact results occur around the center of the plot—that is, around the point $(N/3, N/3, N/3)$, which corresponds to the most likely outcome under this distribution (given that $\lambda = 0$). The largest difference was 1.63×10^{-8} and occurred at (189, 190, 191). It might, then, seem that MPROB produces errors where the multinomial probabilities are larger, but this conclusion is incorrect, for the reason that is described next.

Point probabilities computed with Mathematica correspond to the exact distribution where $\pi = (1/3, 1/3, 1/3)$, whereas those computed with MPROB pertain to a slightly different distribution that results when the probabilities in π are represented as floating-point reals. It turns out that the floating-point representation of $1/3$ falls slightly short of 0.3, and, therefore, MPROB and Mathematica

$$\frac{21 \cdot 22 \cdot 23 \cdot 24 \cdot 25 \cdot 26 \cdot 27 \cdot 28 \cdot 29 \cdot 30 \cdot 31 \cdot 32 \cdot 33 \cdot 34 \cdot 35 \cdot 36 \cdot 37 \cdot 38 \cdot 39 \cdot 40}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 \cdot 11 \cdot 12 \cdot 13 \cdot 14 \cdot 15 \cdot 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5} 0.5^{20} 0.2^{15} 0.3^5. \quad (2)$$

Table 1
Equations Expressing the Probability Distributions Used in the Accuracy Tests

Probability Distributions	Number of Multinomial Cells (J)			
	2	3	4	5
π_1	$\frac{1}{2}(1 - \lambda)^2$	$\frac{1}{3}(1 - \lambda)^3$	$\frac{1}{4}(1 - \lambda)^4$	$\frac{1}{5}(1 - \lambda)^5$
π_2	$\lambda^2 + 2\lambda(1 - \lambda) + \pi_1$	$\frac{3}{2}\lambda(1 - \lambda)^2 + \pi_1$	$\frac{4}{3}\lambda(1 - \lambda)^3 + \pi_1$	$\frac{5}{4}\lambda(1 - \lambda)^4 + \pi_1$
π_3		$\lambda^3 + 3\lambda^2(1 - \lambda) + \pi_2$	$3\lambda^2(1 - \lambda)^2 + \pi_2$	$\frac{10}{3}\lambda^2(1 - \lambda)^3 + \pi_2$
π_4			$\lambda^4 + 4\lambda^3(1 - \lambda) + \pi_3$	$5\lambda^3(1 - \lambda)^2 + \pi_3$
π_5				$\lambda^5 + 5\lambda^4(1 - \lambda) + \pi_4$

computed point probabilities under slightly different distributions. Then, it should come as no surprise that the differences turn out to be larger when the resulting point probabilities are larger. Since there is no way of obtaining *exact* results with Mathematica when it uses the same (real-valued) vector π that is indeed used by MPROB, results of comparisons such as those in Figure 1 are not dependable. In any case, Figure 1 shows *how close* to the ideal distribution one can get with an algorithm whose implementation is limited by inaccurate machine representation of the exact probability distribution.

Figure 1. Difference between point probabilities obtained with MPROB and exact values obtained with Mathematica (Wolfram, 1992) for the case $J = 3$, $N = 570$, and $\lambda = 0$. The barycentric plot represents the sample space of the multinomial distribution, consisting of 163,306 points arranged in a triangular lattice. Coordinates of the point at each vertex of the lattice are indicated for reference. The magnitude of the difference at each point is represented by lightness, black corresponding to a null difference (which occurred for 152,951 points) and white corresponding to the largest difference (which turned out to be 1.63×10^{-8}). Larger differences occur at the center of the plot, where the probabilities are larger, but the differences are nevertheless small. (See the text for further discussion.)

COMPARISON WITH OTHER ALGORITHMS

With its computational structure, MPROB performs the fewest number of operations that are possible without resorting to approximations, and we have already discussed the fact that the accuracy of the result is only limited by machine precision. Yet, it is worth comparing the efficiency of MPROB with that of four alternative algorithms, which are described next.

Step 0 may seem to be a superfluous expense for the computation of a single point probability. Yet, computation of an entire multinomial distribution (as in the accuracy tests just described) or computation of a substantial number of point probabilities (as in exact multinomial tests) may benefit from this preliminary phase. To determine the efficiency of this preprocessing, our first alternative algorithm (ALT1) skipped step 0 and carried out all computations on the (unsorted) input vectors x and π , rather than on the (sorted) vectors y and ρ .

A second alternative (ALT2) implemented a logarithmic transformation of the operations in steps 2 and 5 and included also the preprocessing in step 0. A third algorithm (ALT3) implemented Equation 1 but, on the basis of Stirling's formula (Cameron, 1994, p. 31),

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n = \exp\left[n \ln(n) - n + \frac{1}{2} \ln(2\pi n)\right], \quad (4)$$

approximated the multinomial coefficient as

$$\frac{N!}{x_1! x_2! \dots x_J!} \approx \exp\left[N \ln(N) - \sum_{x_j \neq 0}^J x_j \ln(x_j) + \frac{1}{2} \left(\ln(2\pi N) - \sum_{x_j \neq 0}^J \ln(2\pi x_j)\right)\right]. \quad (5)$$

Our fourth alternative (DMS) is the implementation of Dunlap et al. (1984), which amounts to evaluating the logarithm of Equation 1 using tabulated log factorials for inte-

Figure 2. CPU time used by alternative algorithms, against that used by MPROB, across 240 conditions with $J = 3$. The dashed line indicates equality of CPU times. Different symbols pertain to different alternative algorithms, as indicated in the inset. Each cluster of identical symbols corresponds to the 20 multinomial distributions evaluated for a given sample size N , and separate clusters pertain to the 12 sample sizes used, from $N = 20$ (at the lower left corner, where CPU times could barely be resolved by the timer) to $N = 570$ (at the upper right corner). Note that ALT1 and ALT2 were slower than MPROB, whereas ALT3 and DMS were faster. Since the vertical scale is logarithmic, the (approximately constant) vertical offset of ALT1 and ALT2 clusters with respect to the dashed line indicates the factor by which the execution of these algorithms was slower: approximately 1.5 and 3.5, respectively, for ALT1 and ALT2. As for ALT3 and DMS, the offsets grow with N , implying that these alternatives required comparatively less time than MPROB as N increased.

gers below 18 and approximations based on Stirling's formula for integers at and above 18.

Figure 2 shows the CPU time used by each of the four alternative algorithms against that used by MPROB in the evaluation of the complete multinomial distributions referred to in the accuracy tests described above. To avoid clutter, results are shown only for $J = 3$, since results for other values of J displayed the same pattern. Except at $N = 20$ (where the CPU time could hardly ever be resolved by the computer timer), ALT1 expended about 50% more time than MPROB, and ALT2 required more than three times as much time as MPROB. On the other hand, ALT3 was faster than MPROB and required comparatively less time as N increased; the same was true for DMS, which indeed incurred the least computational cost.

Figure 3 makes clear that DMS achieves its speed at the expense of accuracy, incurring errors that can be up to 10^{10} times larger than those of MPROB. The second

quickest algorithm, ALT3, incurred even larger errors (ranging between 1.1×10^{-3} and 6.6×10^{-2}). On the other hand, none of the two slower alternatives (ALT1 and ALT2) produced smaller errors: ALT1 reproduced the errors of MPROB almost identically; and ALT2 produced errors that were slightly larger than those of MPROB. Since both ALT1 and ALT2 are slower than MPROB and do not provide any increased accuracy, they are not useful as alternatives.

Although Figure 2 suggests that the computational cost of MPROB may be high, it should be kept in mind that the reported times correspond to the computation of entire multinomial distributions whose sample space increases with N . Among those reported in Figure 2, the largest execution time was 6.674 sec when $N = 570$ and $\lambda = 0.55$. In this case, 163,306 point probabilities were evaluated, resulting in a negligible average cost of 4.09×10^{-2} msec per point probability.

CONCLUDING REMARKS

We have presented an algorithm for the computation of exact multinomial probabilities, whose accuracy is only limited by machine precision, and we have also shown that resultant inaccuracies are negligible with small sample sizes. We have also shown that the algorithm is efficient, as compared with alternative algorithms (Berry & Mielke, 1995; Dunlap et al., 1984) that compute approximations whose accuracy has never been determined.

It is interesting to note that, besides providing the exact result, MPROB is capable of handling cases in which these alternative approximations fail. Consider $x =$

Figure 3. Errors incurred by DMS against those incurred by MPROB across the same 240 conditions of Figure 2. Error is defined as $\sum Pr(X = x) - 1$, where the sum runs across the entire sample space of the corresponding multinomial distribution. Note that the horizontal and vertical scales differ by a factor of 10^{10} . Note also that there appears to be no systematic relationship between the errors incurred by DMS and those incurred by MPROB.

(500,1000,1000) and $\boldsymbol{\pi} = (0.2, 0.4, 0.4)$. The exact multinomial probability of $\mathbf{x}$ under $\boldsymbol{\pi}$ (computed symbolically with Mathematica; Wolfram, 1992) is $3.5577452334995 \times 10^{-4}$. A double-precision Fortran implementation of the algorithm in Berry and Mielke (1995) produces a floating overflow error, and a double-precision implementation of the algorithm in Dunlap et al. (1984) returns a value of 0, whereas MPROB produces a value that differs from the exact result by less than 5×10^{-17} . In any case, MPROB will be most useful with sample sizes much smaller than that in the preceding example, where exact multinomial tests should replace the (inaccurate) tests based on the asymptotic chi-square distribution.

AVAILABILITY

A thoroughly documented listing of the Fortran subroutine (MPROB.FOR) is available from the author (e-mail: miguel@psi.ucm.es). Also available is an executable MS-DOS program for PCs (MULTI.EXE, with documentation) that computes an individual multinomial probability given a probability distribution $\boldsymbol{\pi}$ and a vector of observations $\mathbf{x}$, each with J components.

REFERENCES

- BERRY, K. J., & MIELKE, P. W., JR. (1995). Exact cumulative probabilities for the multinomial distribution. *Educational & Psychological Measurement*, **55**, 769-772.
- CAMERON, P. J. (1994). *Combinatorics: Topics, techniques, algorithms*. Cambridge: Cambridge University Press.
- CHAPMAN, J.-A. W. (1976). A comparison of the X^2 , $-2\log R$, and multinomial probability criteria for significance tests when expected frequencies are small. *Journal of the American Statistical Association*, **71**, 854-863.
- DUNLAP, W. P., MYERS, L., & SILVER, N. C. (1984). Exact multinomial probabilities for one-way contingency tables. *Behavior Research Methods, Instruments, & Computers*, **16**, 54-56.
- GARCÍA-PÉREZ, M. A. (1994). Parameter estimation and goodness-of-fit testing in multinomial models. *British Journal of Mathematical & Statistical Psychology*, **47**, 247-282.
- KOEHLER, K. J., & LARNTZ, K. (1980). An empirical investigation of goodness-of-fit statistics for sparse multinomials. *Journal of the American Statistical Association*, **75**, 336-344.
- LARNTZ, K. (1978). Small-sample comparisons of exact levels for chi-squared goodness-of-fit statistics. *Journal of the American Statistical Association*, **73**, 253-263.
- LAWAL, H. B. (1984). Comparisons of the X^2 , Y^2 , Freeman-Tukey and Williams's improved G^2 test statistics in small samples of one-way multinomials. *Biometrika*, **71**, 415-418.
- NÚÑEZ-ANTÓN, V., & GARCÍA-PÉREZ, M. A. (1995, September). *On the accuracy of the χ^2 distribution as an approximation to the finite-sample distribution of goodness-of-fit statistics in one-way multinomial models*. Paper presented at the 26th European Mathematical Psychology Group Meeting, Regensburg, Germany.
- READ, T. R. C. (1984). Small-sample comparisons for the power divergence goodness-of-fit statistics. *Journal of the American Statistical Association*, **79**, 929-935.
- READ, T. R. C., & CRESSIE, N. A. C. (1988). *Goodness-of-fit statistics for discrete multivariate data*. New York: Springer-Verlag.
- RIEFER, D. M., & BATCHELDER, W. H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, **95**, 318-339.
- RIEFER, D. M., & BATCHELDER, W. H. (1991). Statistical inference for multinomial processing tree models. In J.-P. Doignon & J.-C. Falmagne (Eds.), *Mathematical psychology: Current developments* (pp. 313-335). New York: Springer-Verlag.
- WOLFRAM, S. (1992). *Mathematica: A system for doing mathematics by computer* (2nd ed.). New York: Addison-Wesley.
- YARNOLD, J. K. (1972). Asymptotic approximations for the probability that a sum of lattice random vectors lies in a convex set. *Annals of Mathematical Statistics*, **43**, 1566-1580.

(Manuscript received May 22, 1997;
revision accepted for publication July 13, 1998.)