

Efficient Techniques for Unbiasing a Bernoulli Generator

James A. Lechner*

Institute for Basic Standards, National Bureau of Standards, Washington, D.C. 20234

(December 21, 1971)

Consider the problem of operating on a sequence of i.i.d. Bernoulli variables with unknown mean p to produce a sequence of symmetric Bernoulli variables. Define the efficiency of any proposed method to be the average number of binary output digits per input digit. The following results are proved: (A) No method exists having efficiency greater than $-p \log_2 p - q \log_2 q$, where $q = 1 - p$. (B) Methods do exist with efficiency arbitrarily close to the bound just given. Examples are given, and compared with other methods in the literature. A technique for finding the methods of (B) above is given.

Key words: Bernoulli; binomial; coin-tossing; computer; efficiency; generator; random numbers; statistics.

1. Introduction

The problem of deriving unbiased Bernoulli variables from a sequence of independent and identically distributed Bernoulli variables with mean p has received attention repeatedly in the literature. The most recent appearance known to this author is a paper by Simons and Hoeffding [4],¹ wherein are given references to other work. The approach adopted by Simons and Hoeffding concentrates on the sample sequences, and investigates stopping rules and decision rules which specify when a sample sequence is terminated with the issuance of a (binary) output digit, and which digit is output. One rule is "better" than another, in their terminology, if it stops as soon as the other for any sample sequence, and sooner for at least one sequence. They define the class of "even" procedures, and find the best even procedure, which they denote by Q_2 . They then find a better (noneven) procedure Q_3 , a lower bound for the expected length of the sample sequence, a proof that there is no procedure which is as good as or better than all others, and a procedure Q_4 which is better than Q_3 for small p values. Incidentally, they start by introducing the procedure of von Neumann [5], denoted Q_1 , which consists of taking successive pairs of digits until a mixed pair (i.e., a pair containing both a 0 and a 1) is obtained, and outputting the second digit of the pair.

In this paper, the efficiency of a procedure is defined in a long-run sense, as the average number of output digits per input digit. Thus a procedure which does not terminate as quickly as one of the Q_i above, for certain sample sequences, and yet produces output digits at a higher average rate, is considered more efficient. For reference, note that the efficiency of Q_1 is $pq \leq 1/4$ (where $q = 1 - p$).

The efficiency of Q_2 is shown by Simons and Hoeffding to be

$$(1/2) \prod_{i=1}^{\infty} (1 + p^{2^i} + q^{2^i})^{-1},$$

which is less than

$$(1/2)(1 + p^2 + q^2)^{-1}(1 + p^4 + q^4)^{-1} \leq 8/27 < 1/3.$$

AMS Subject Classification: Primary 65C10, Secondary 60-04.

* This paper is based on work largely completed while the author was with Research Analysis Corporation, McLean, Va. 22101.

¹ Figures in brackets indicate the literature references at the end of this paper.

As a matter of fact, the efficiency of Q_3 is also less than $1/3$. The lower bound for the expected size of the sample sequence for any procedure which looks at sample sequences in the way considered by these authors—i.e., which consists of a stopping rule and a rule for specifying an output digit whenever the sequence stops—is given by the authors as $p^{-1}q^{-1} - 1$. This immediately leads to an upper bound for the efficiency of such procedures, namely $pq(1 - pq)^{-1}$. This function in turn takes on its maximum value at $p = 1/2$, for which it is $1/3$.

The procedure Q_2 defined by Simons and Hoeffding is equivalent to one presented by this author [2] at the 1964 Institute of Mathematical Statistics Meeting at Amherst, namely, stop as soon as a block from $(k - 1)2^n + 1$ to $k \cdot 2^n$, for any positive integers k and n , is evenly split between zeros and ones, and output the last digit; then start again.

The procedure Q_3 is an ingenious improvement over Q_2 . Simons and Hoeffding noted that there are circumstances when it is certain that the next digit (whatever it turns out to be) will result in termination—and so there is no need to take that last digit, if one simply decides in the beginning to output the next-to-last digit (whether or not the last digit is needed). An example is in order at this point. The sequences 00001100 and 00000011 are equally likely, as are the sequences 00000010 and 00000001. Under Q_2 , these sequences will lead to termination, with output 0,1,0,1, respectively. If we were to output the next-to-last digit at termination, the output would be 0,1,1,0, respectively. Now note that in all but the last sequence, we know before the eighth digit is taken that it will be the last digit. The procedure Q_3 merely takes advantage of this fact, by stopping as soon as it becomes known that termination will occur on the next digit. (Obviously, the *last digit taken* is the output of Q_3 whenever Q_3 stops sooner than Q_2 , while the next-to-last digit is the output when Q_3 stops at the same point as Q_2 .)

In the next section, an intuitive derivation is given of a different kind of procedure, which grew out of a suggestion by Herman Rubin to the author during the Amherst meeting previously mentioned. Following sections present a rigorous evaluation of the method, examples, and proofs.²

2. A Different Approach

Consider again the von Neumann procedure above. It consists of taking pairs of observations, ignoring any pair consisting of two zeros or of two ones, and putting out one digit if a mixed pair occurs. Now two zeros or two ones can happen in only one way each, but a mixed pair can be either 01 or 10—and one is as likely as the other. Thus another way of looking at the von Neumann procedure is in terms of rearrangements of the result that occurred: since there is only one arrangement possible for the sequence 11, no output can be obtained, but a mixed sequence can happen in either of two equally likely ways which are assigned the (output) values 0 and 1 (in arbitrary order).

Now let us extend this technique. Suppose four input digits are examined together. If all are alike, then no rearrangements are possible—hence no output. If there is exactly one 0, or three 0's, then there are four equally likely rearrangements—1110, 1101, 1011, 0111 in the former case, and the complements in the latter. We can assign output sequences 00, 01, 10, and 11 to these four possibilities, and thus obtain two output digits. But what if there are two 1's and two 0's? Then there are $\binom{4}{2}$ = six equally likely possibilities. We can take four of them, and derive two binary digits as above; but we can derive only one binary digit when one of the remaining two possibilities occurs. Thus we will get 0, 1, or 2 output digits for 4 input digits, with probabilities $(p^4 + q^4)$, $2p^2q^2$, and $4p^3q + 4pq^3 + 4p^2q^2$ respectively, for an average of $2pq - 3p^2q^2/2$ output digits per input digit. This value is at least $(13/8)pq$, incidentally, so that this procedure is considerably more efficient than the von Neumann procedure, even for groups of four; it can be made still more efficient for larger groups, as will be seen.

There is something basically unsatisfying about this procedure as just outlined. Consider again the simple example above with group size 4. There are six arrangements of two zeros and two ones,

² The author has recently learned of a related paper by Prof. Peter Elias, to be published in the *Annals of Mathematical Statistics*.

which is more than the four arrangements of one zero and three ones; yet the number of output digits is (on the average) only $5/3$, which is less than the two digits obtained for the case of one zero and three ones. The reason is obvious: 6 is not a power of 2. (For 7 arrangements, the expected number of digits is less: only $10/7$.) One possible way to improve things is to save the set with six arrangements, and combine it with the next set having six arrangements. We thus have 36 possibilities (for the combined set): we can get 5 ($= \log_2 32$) digits with probability $32/36$, and 2 digits with probability $4/36$, for an average of $7/3$ digits per set. We could go further, and find a higher power of 6 which is closer to a power of 2; we could then get even more digits per set.

This simple example embodies the key idea of the method. In the next section, a more complete discussion of the technique is presented. Two lemmas and a theoretical upper bound for efficiency are given, and used to show the existence of procedures with efficiencies arbitrarily close to the upper bound. This bound is then compared to the efficiencies of the Q^i and of selected examples of the procedures, in the following section.

3. Permutation Methods

Consider sets of n independent and identically distributed Bernoulli variables with mean p . Any such set will have from zero to n 1's. The probability that a set will have x 1's is $\binom{n}{x} p^x q^{n-x}$ and a set with x 1's can have any of $\binom{n}{x}$ equally likely configurations. Let $m = \binom{n}{x}$. If m is a power of 2, one can number all of the m configurations with $(\log_2 m)$ -digit binary numbers, in such a way that each such number is used exactly once. Then when one of the m configurations occurs, one can simply output the $(\log_2 m)$ digits in the number assigned to that configuration. If m is not a power of 2, a record may be kept of which configuration occurred (i.e., a number between 1 and m may be recorded). In future sets of n digits, every value of m will (with probability one) come up repeatedly, so that one obtains sequences of numbers between 1 and m (inclusive)—one sequence for each distinct value of m . In each sequence, each number is equally likely to take on any of the m values, of course. With $j(m)$ numbers, each equally likely to have any value between 1 and m , one has $m^{j(m)}$ equally likely possibilities. It will be shown below that the integer $j(m)$ can be chosen so that $m^{j(m)}$ is only slightly larger than a power of 2. Thus with high probability one obtains very nearly $\log_2 m$ digits for each of the $j(m)$ sets.

A "permutation procedure" is a procedure which operates as just outlined. If n is the size of the set upon which the procedure is based, one needs to specify values $j(m)$ for each $m = \binom{n}{x}$, for $x = 0, 1, 2, \dots, n$. (Note that $\binom{n}{x} = \binom{n}{n-x}$.) The j in Lemma 1 and Theorem 1 below is the $j(m)$ just considered, and the i below is the integer such that 2^i is almost as big as $m^{j(m)}$; i.e., $i = \lceil j(m) \log_2 m \rceil$. Let E_u denote the mathematical expectation of the random variable u . Then we have:

THEOREM 1: *Given $\epsilon > 0$, a sequence of independent and identically distributed Bernoulli variables, and an integer $n > 1$, There exists a permutation procedure based on sets of n , with efficiency at least $n^{-1} E \log_2 \binom{n}{x} - \epsilon$.*

To prove this result, we will make use of the following lemma, proved in the Appendix:

LEMMA 1: *Given a real number k , and an $\epsilon > 0$, there exist integers i and j such that $0 \leq k - (i/j) < (\epsilon/j)$.*

PROOF OF THEOREM 1: For each possible value of $m = \binom{n}{x}$, Lemma 1 implies the existence of integers i and j such that $0 \leq \log_2 m - (i/j) < \epsilon/2j$. A permutation procedure using these integers will have efficiency given by

$$E_n = n^{-1} \sum_{x=1}^{n-1} D_x \binom{n}{x} p^x q^{n-x},$$

where D_x denotes the expected number of output digits from a set of n digits having x 1's. Now D_x is at least (i/j) times the probability of getting i digits from a group of j such sets, where $i = \lceil \log_2 m^j \rceil$. This probability is $2^i/m^j$. From the inequality above,

$$0 \geq i - j(\log_2 m) > -\epsilon/2,$$

or equivalently,

$$1 \cong 2^{i/m^j} > 2^{-\epsilon/2} > e^{-\epsilon/2} > 1 - (\epsilon/2);$$

and also, $(i/j) > \log_2 m - \epsilon/2$. Therefore

$$\begin{aligned} E_n &> n^{-1} \sum_{x=0}^n (1 - \epsilon/2) (\log_2 \binom{n}{x} - \epsilon/2) \binom{n}{x} p^x q^{n-x} \\ &= n^{-1} E \log_2 \binom{n}{x} - (\epsilon/2n) \{E \log_2 \binom{n}{x} + 1\} + \epsilon^2/4n \\ &> n^{-1} E \log_2 \binom{n}{x} - \epsilon. \end{aligned}$$

(The last step is valid because $\binom{n}{x} 2^{-n}$, being a probability, is at most unity, implying that $\log_2 \binom{n}{x} \leq n$.) Q.E.D.

Lemma 2, also proved in the Appendix, asserts that

$$\lim_{n \rightarrow \infty} n^{-1} E \log_2 \binom{n}{x} = -p \log_2 p - q \log_2 q.$$

This then is the limiting efficiency of permutation procedures. One can do no better, as evidenced by the following.

THEOREM 2: *Any procedure for generating symmetric Bernoulli variables from a stream of independent and identically distributed Bernoulli variables has efficiency at most $-(p \log_2 p + q \log_2 q)$, where p is the mean value of the independent and identically distributed variables.*

PROOF: Kullback [1, p. 13] proves that for a random variable X , and a statistic Y (a function of X), the average information in an outcome X is at least as great as that in the derived statistic Y . The "information" referred to in this theorem is the information for discrimination in favor of one hypothesis, H_1 , against another one, H_2 . Let X be an arbitrarily long (finite) sequence of independent and identically distributed Bernoulli variables, and Y the derived sequence of symmetric Bernoulli variables. Let n be the length of the X -sequence, and M (a random variable) the length of the Y -sequence; let $m = E(M)$. Let H_1 be the hypothesis that the mean of the elements of X is p_0 ; in terms of Y , this hypothesis implies that the mean of the elements of Y is $1/2$, regardless of p_0 . Let H_2 be the hypothesis that the mean of the X elements is in the closed unit interval. Then (in Kullback's notation)

$$\begin{aligned} I(1;2; X) &= - \sum_{s=0}^n \binom{n}{s} p^s q^{n-s} (s \log_2 p_0 + (n-s) \log_2 q_0) \\ &= -n(p \log_2 p_0 + q \log_2 q_0) \\ &\leq -n(p \log_2 p + q \log_2 q). \end{aligned}$$

Similarly,

$$I(1;2; Y) = E \left\{ - \sum_{t=0}^M \binom{M}{t} 2^{-M} \{-t - (M-t)\} \right\} = E(M) = m.$$

By Kullback's theorem, $m \leq -n(p \log_2 p + q \log_2 q)$. But the efficiency of the procedure under consideration is simply the limit of the ratio m/n as $n \rightarrow \infty$; thus the efficiency is limited by $-(p \log_2 p + q \log_2 q)$. Q.E.D.

Finally, we note that there is a simpler approach to the problem of attaining high efficiency. As can be seen from table 2, the limit of attainable efficiency for small n is not very high. Since one approaches this limit by taking larger values of j , using more storage space and a more complicated rule for determining the output, it would seem worthwhile to consider direct methods using a larger value of n . In other words, if it is necessary to save (an average of) ten sets of $n=8$ digits to come close to the limit $E \log_2 \binom{8}{x}$ derived above, we ought to compare with one set of $n=80$. How does one use such a set? Suppose x is the number of 1's that occur. Then $m = \binom{80}{x}$. One numbers the m configurations arbitrarily from 0 to $m-1$, and sets off a group of size equal to the largest power of

2 less than (or equal to) m ; then from what is left, a group of size equal to the largest power of 2 less than or equal to the number remaining; then with what is still left, repeat the procedure, until the entire group of m is exhausted. If the outcome should fall in a group of size 2^i , one gets i digits. It is shown at the end of the Appendix that this procedure will attain an efficiency of at least $n^{-1}(E(\log_2 \binom{n}{x}) - 2)$, which approaches the value $n^{-1}E(\log_2 \binom{n}{x})$ obtained above as n gets large.

A few words are in order, at this point, about the practical implications of the techniques described above. Specifically, a method of obtaining the proper binary output stream from the input stream is needed. There are two steps to be accomplished: first, derive an index number for the input sequence, which identifies which of the equally likely configurations has occurred; and second, derive the appropriate output sequence from this index number. They will be considered in turn.

Consider first the simple case of a set of n input digits, and let x be the number of 1's among the n digits. As before, let $m = \binom{n}{x}$. We will derive an index number between 0 and $m - 1$ inclusive which gives the position of the actual configuration in a sequence of the m possible configurations, arranged in order of increasing magnitude (considered as binary numbers). Let z denote the actual sequence (of x 1's and $(n - x)$ 0's). Check the first bit. If it is 1, then z is not one of the $\binom{n-1}{x}$ combinations starting with 0. Since these are precisely the first $\binom{n-1}{x}$ combinations, add the binomial coefficient $\binom{n-1}{x}$ into a counter (which will eventually contain the index number corresponding to z). If the first bit is 0, then z is one of the first $\binom{n-1}{x}$ combinations, so don't add anything into the counter. Now check the next bit. If 1, add in the appropriate number — either $\binom{n-2}{x}$ or $\binom{n-2}{x-1}$, depending

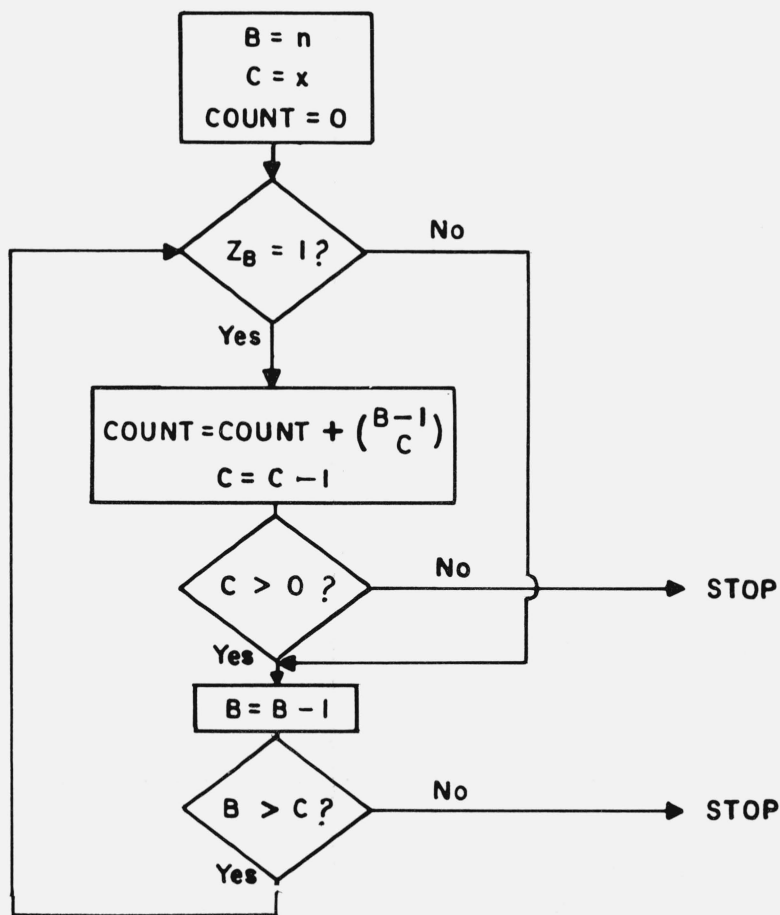


FIGURE 1. Flow chart for procedure assigning an index number between 0 and $m - 1$ inclusive to a sequence $Z_n Z_{n-1} Z_{n-2} \dots Z_1$ containing x ones and $n - x$ zeros; $m = \binom{n}{x}$.

on whether the first bit was a 0 or a 1. Continue in this fashion through all the bits of z , stopping as soon as the remaining bits of z are known to be all alike. This is really a fairly simple procedure to program: a flow chart is shown in figure 1.

For a permutation procedure, one has several (say j) sets of n digits, each set being one of m configurations. One assigns a number between 0 and $m - 1$, as above, to each set of n . Then one assigns the number $n_1 + n_2 \cdot m + n_3 \cdot m^2 + \dots + n_j \cdot m^{j-1}$ to the combination of j sets of n , where n_i is the number assigned to the i th set of n . This is of course a number between 0 and $m^j - 1$, as required.

Now consider the determination of the output sequence, given the index number of the input. Let y denote the index number of the actual result, and M the number of equally likely results (of which this result is one). Define i by $2^i \leq M < 2^{i+1}$. Then there are $i + 1$ bits in the binary representation of M :

$$M = \sum_{j=0}^i a_j 2^j, \quad a_i = 1; \quad a_j = 0 \text{ or } 1, \quad \text{all } j < i$$

If $y < 2^i$, we should obtain i bits; if $2^i \leq y < 2^i + 2^{i_1}$ where i_1 is the largest $j (< i)$ with $a_j = 1$, we should obtain i_1 bits; etc. Now $y < 2^i$ precisely when the i th bit of y is 0 (considering the least significant bit as the 0th bit). And conveniently, if the i th bit of y is 0, then the i less significant bits are symmetric Bernoulli variables—because the possible values of these bits, when y has the i th bit equal to zero, consist of all the binary numbers from 0 to $2^i - 1$. Similarly, $2^i \leq y < 2^i + 2^{i_1}$ precisely when the i th bit of y is 1 and the i_1 st bit is 0; and when this is true, the lower-order i_1 bits are symmetric Bernoulli variables. This process can be continued through all the bits of y . In summary, the procedure is to check in turn all bits of y corresponding to non-zero bits of M , beginning with the most significant; as soon as one of the checked bits is found to be 0, stop, and output the remaining bits of y .

4. Examples

The primary results of this section are contained in table 1, which lists the efficiency of each of several techniques. Among those evaluated are Q_1, Q_2 , and Q_3 , from Simons and Hoeffding; the simple procedures outlined above (permutation procedures with $j = 1$) for $n = 4, 8, 20, 50$, and 1024, denoted by S_4, S_8, S_{20}, S_{50} , and S_{1024} and one permutation procedure denoted by P_1 . Of course, P_1 needs to be specified: we require n , and the appropriate number of j values. These are given in table 2. Note that for $x = 3$ or 5, $m = 56$. The 56 sequences can be (arbitrarily) divided into two groups of 28, and index numbers from 0 to 27 can be assigned within each group. Then when $x = 3$ or 5 occurs, one binary digit can be output, 0 or 1 according to which group of 28 contains the result; the index number can be treated just as if it were a result with $x = 2$ or 6. This of course decreases the amount of storage necessary and the average time between outputs.

5. References

- | | |
|--|---|
| [1] Kullback, Solomon, <i>Information Theory and Statistics</i> , Wiley, New York (1959). | [4] Simons, Gordon, and Wassily Hoeffding, Unbiased coin tossing with a biased coin, <i>Annals of Math. Stat.</i> 41 , 341-352 (1970). |
| [2] Lechner, James A., Unbiased coin tosses with a biased coin., Abstract, <i>Annals of Math, Stat.</i> 35 , 1838 (1964). | [5] von Neumann, John, Various techniques used in connection with random digits, Monte Carlo Method, <i>Applied Mathematics Series</i> , No. 12, 36-38. U.S. National Bureau of Standards, Washington, D.C. |
| [3] Niven, Ivan, <i>Irrational Numbers</i> , Carus Monograph No. 11, Mathematical Association of America (1956). | |

6. Appendix

Contained herein are proofs of the two lemmas, and of the lower bound for the efficiency of the S -procedures.

LEMMA 1: Given a positive number k , and $\epsilon > 0$, there exist positive integers i and j such that $0 \leq k - (i/j) < \epsilon/j$.

PROOF: If k is rational, there exist integers i, j such that

$$k - (i/j) = 0.$$

Otherwise, by Theorem 4.2, page 44 of [3], there exist positive integers i, j such that $|jk - i| < \epsilon$. If $jk - i > 0$, we are finished. Otherwise $-\epsilon < jk - i < 0$; let

$$m = [(i - jk)^{-1}] \leq \frac{1}{i - jk}.$$

Then

$$m(jk - i) \geq -1 > (m + 1)(jk - i) > m(jk - i) - \epsilon,$$

so that

$$0 \leq (mj)k - (mi - 1) < \epsilon. \quad \text{Q.E.D.}$$

LEMMA 2: Let x be the number of successes in n independent and identically distributed Bernoulli trials, with probability p of success. Then

$$\lim_{n \rightarrow \infty} \{n^{-1} E \log_2 \binom{n}{x}\} = -p \log_2 p - q \log_2 q.$$

PROOF: (All logs in the sequel are to the base 2.) By Stirling's approximation,

$$\log \binom{n}{x} = n \log n - x \log x - (n - x) \log(n - x) - (1/2) \log(x - x^2/n) + O(n^{-1/2}),$$

for $1 \leq x \leq n - 1$. Dividing by n , and substituting $np + y$ for x , we have

$$\begin{aligned} n^{-1} \log \binom{n}{x} &= \log n - (p + y/n) \log(np + y) - (q - y/n) \log(nq - y) + O(n^{-1/2}) \\ &= -\log\{(p + y/n)^{p+y/n} (q - y/n)^{q-y/n}\} + O(n^{-1/2}) \\ &= -p \log p - q \log q - (y/n) \log(p/q) \\ &\quad - (p + y/n) \log(1 + y/np) - (q - y/n) \log(1 - y/nq) + O(n^{-1/2}). \end{aligned}$$

Now y is a random variable, with mean zero and variance npq , so that y/n has mean zero and variance pq/n . Since $\log \binom{n}{0} = \log \binom{n}{n} = 0$, we can write

$$\begin{aligned} n^{-1} E \log \binom{n}{x} &= \Sigma' \{ -p \log p - q \log q - (y/n) \log(p/q) \\ &\quad - (p + y/n) \log(1 + y/np) - (q - y/n) \log(1 - y/nq) \\ &\quad + O(n^{-1/2}) \} P_{np+y}, \end{aligned}$$

where the prime on the summation indicates that the terms corresponding to $x = 0$ and to $x = n$ are omitted, and where P_{np+y} is the appropriate binomial probability. Thus

$$n^{-1} E \log \binom{n}{x} = -p \log p - q \log q - E_1 - E_2 + O(n^{-1/2}),$$

where

$$E_1 = \Sigma' (p + y/n) \log(1 + y/np) P_{np+y}$$

and

$$E_2 = \Sigma' (q - y/n) \log(1 - y/nq) P_{np+y}.$$

All that remains is to show that E_1 and E_2 approach 0 as n approaches ∞ . The proof will be given for E_1 , that for E_2 being essentially the same.

Given a small positive quantity ϵ , we want to find a value n_0 such that for any $n > n_0$, $|E_1(n)| < \epsilon$. Because $0 < p + (y/n) < 1$,

$$|E_1| < \Sigma' |\log(1 + y/np)| P_{np+y}.$$

Now y has mean zero and variance npq . Thus by Chebychev's inequality, we have for any $a > 0$ the relation

$$Pr(|y/np| > a\sqrt{q/np}) \leq a^{-2}.$$

Therefore for $a\sqrt{q/np} < 1$,

$$\begin{aligned} |E_1| &< \max \{ \log(1 + a\sqrt{q/np}), |\log(1 - a\sqrt{q/np})| \} + a^{-2} \max |\log(1 + y/np)| \\ &= |\log(1 - a\sqrt{q/np})| + a^{-2} \max \{ |\log 1/np|, |\log(n-1)/np| \}. \end{aligned}$$

Consider only $\epsilon < 1/2$, and $n > p^{-2}q^{-1}$. Then $np > p^{-1}q^{-1} > p^{-1} > (n-1)/np > p^{-1}(1-p^2q) = p^{-1} - pq > 1$, so $|\log 1/np| = \log np > |\log(n-1)/np|$. Take $a = (\epsilon/4)(np/q)^{1/2}$. Then

$$|\log(1 - a\sqrt{q/np})| = |\log(1 - \epsilon/4)| < \epsilon/2,$$

and $a^{-2} \log np < 16(\log np)/np\epsilon^2$. Now pick n_0 large enough that $(\log n_0 p)/n_0 p < \epsilon^3/32$. Then for $n > n_0$, $a^{-2} \log np < \epsilon/2$, so $|E_1| < \epsilon$.

Q.E.D.

(Paper 76B1&2-362)