5-1-2013

A Gaming Application of the Negative Hypergeometric Distribution

Steven Norman Jones
University of Nevada, Las Vegas, steve.jones@alumni.unlv.edu

A GAMING APPLICATION OF THE NEGATIVE HYPERGEOMETRIC DISTRIBUTION

by

Steven Norman Jones

Bachelor of Science in Computer Engineering
Massachusetts Institute of Technology
1977

Master of Science in Electrical Engineering and Computer Science
Massachusetts Institute of Technology
1978

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science in Mathematical Sciences

Department of Mathematical Sciences
College of Sciences
The Graduate College

University of Nevada, Las Vegas
May 2013

THE GRADUATE COLLEGE

We recommend the thesis prepared under our supervision by

Steven Jones

entitled

A Gaming Application of the Negative Hypergeometric Distribution

be accepted in partial fulfillment of the requirements for the degree of

Master of Science in Mathematical Sciences
Department of Mathematical Sciences

Rohan Dalpatadu, Ph.D., Committee Chair

Gennady Bachman, Ph.D., Committee Member

Amei Amei, Ph.D., Committee Member

Ashok Singh, Ph.D., Committee Member

Pushkin Kachroo, Ph.D., Graduate College Representative

Tom Piechota, Ph.D., Interim Vice President for Research &
Dean of the Graduate College

May 2013

ABSTRACT

A Gaming Application of the Negative Hypergeometric Distribution

by

Steven Jones

Dr. Rohan Dalpatadu, Advisory Committee Chair
Associate Professor of Mathematical Sciences
University of Nevada, Las Vegas

The Negative Hypergeometric distribution represents waiting times when drawing from a finite sample without replacement. It is analogous to the negative binomial, which models the distribution of waiting times when drawing with replacement. Even though the Negative Hypergeometric has applications it is typically omitted from textbooks on probability and statistics and is not generally known. The main purpose of this thesis is to derive expressions for the mean and variance of a new application of the Negative Hypergeometric to gaming and gambling. Other applications are described as well.

ACKNOWLEDGMENTS

I would like to thank Dr. Rohan Dalpatadu, my thesis adviser and advisory committee chair, for suggesting this fascinating probability distribution as a possible line of research and for his encouragement and support in my pursuit of educational goals. I would also like to thank Dr. Ashok Singh in the Hotel College at UNLV for a stimulating discussion in which he suggested some possible applications of the distribution and related research questions in the gaming field.

Finally I would like to express my thanks to Dr. Pushkin Kachroo in the Electrical and Computer Engineering department for encouraging me to continue my studies in mathematics and also helping me get started with the LaTeX typesetting language.

Contents

List of Tables

List of Figures

Chapter 1

The Negative Hypergeometric Distribution

Suppose a finite population consists of two types of objects, which one can think of as red and white balls in an urn. There are N balls altogether, of which M are red. If balls are drawn randomly, then how many must be drawn to obtain k red ones, where $1 \leq k \leq M$?

This random variable has the Negative Hypergeometric distribution, one that is typically omitted from textbooks on probability and statistics, such as Casella and Berger (2002) or Feller (1957), and is seldom found in books on discrete distributions, either. Feller's classic text (on page 56) contains an exercise related to the distribution but does not name it explicitly. There are several possible reasons for this common omission. First, the distribution does not arise frequently in applications, and when it does, it can often be approximated by the negative binomial. Second, derivation of the moments is typically rather involved (two ways are given in Schuster and Sype 1987). Third, there has been some confusion in the literature as to the correct nomenclature for the distribution (Miller and Fridel 2007).

Nevertheless, the Negative Hypergeometric distribution is considered to be an elementary probabilistic urn model and is usually described in books on that subject, such as Norman and Kotz (1977). It is argued in Miller and Fridel (2007) that the Negative Hypergeometric distribution should be included in probability courses since it rounds out the picture for drawing without replacement: it plays the same role as the negative binomial, which models waiting times when sampling with replacement.

The primary purpose of this thesis is to suggest a new application of the Negative Hypergeometric distribution to gaming and gambling, and to derive results for a compound distribution that arises from this application. Another possible application, to customer satisfaction surveys, is also described. These results are presented in Chapter 2.

First, however, a derivation of the mean and variance using only elementary mathematics is provided, and the formulas for mean and variance are shown to be valid for all degenerate cases. In addition, several ways of generalizing the distribution are discussed.

1.1 Finding the Probability Mass Function

Let X denote the number of balls that must be drawn to obtain k red ones, from an urn containing N balls, of which M are red. First note that $k \leq X \leq k + N - M$, since in a repeated drawing for k of the M red balls one obtains any number of the remaining $N - M$ white ones. Let $f_X(x)$ denote the probability mass function. Then $f_X(x) = \Pr(\text{draw } k-1 \text{ red in } x-1 \text{ draws}) \times \Pr(\text{draw red on the last draw})$; i.e.,

$$f_X(x) = \frac{\binom{N-M}{x-k}\binom{M}{k-1}}{\binom{N}{x-1}} \frac{(M-k+1)}{(N-x+1)}$$

which simplifies to

$$f_X(x) = \frac{\binom{x-1}{k-1}\binom{N-x}{M-k}}{\binom{N}{M}}. \tag{1.1}$$

Hereafter this is referred to as the *Negative Hypergeometric* distribution, and denoted $\mathrm{nhg}(N, M, k)$.

The following Lemma will prove useful in calculating the moments of this distribution. Since probability sums to one, the identity

$$\sum_{x=k}^{k+N-M} \binom{x-1}{k-1}\binom{N-x}{M-k} = \binom{N}{M} \tag{1.2}$$

is valid for any $N > M \geq k > 0$. This leads to a

Lemma. For any natural numbers $a > b > 0$ and $c > 0$,

$$\sum_{y=0}^{a-b} \binom{a-y}{b}\binom{c+y}{c} = \binom{a+c+1}{b+c+1}$$

Proof. Let $k = c+1$, $N = a+k$, $M = b+k$, and $y = x-k$. It readily follows that $N > M > k > 0$ and $k \leq x \leq k+N-M$, so the previous combinatorial argument applies, and the result follows from substitution in Equation 1.2. (A more general form of this identity can be found in Graham, Knuth and Patashnik 1994, 169).

1.2 Computing the Mean and Variance

The mean and variance of the random variable X are now calculated, using the Lemma. The idea is to set $x\binom{x-1}{k-1} = k\binom{x}{k}$ and $x(x+1)\binom{x-1}{k-1} = k(k+1)\binom{x+1}{k+1}$ in the sums for $\mathrm{E}(X)$ and $\mathrm{E}(X(X+1))$, and then re-index:

$$\binom{N}{M} E(X) = \sum_{x=k}^{k+N-M} x\binom{x-1}{k-1}\binom{N-x}{M-k} = \sum_{x=k}^{k+N-M} k\binom{x}{k}\binom{N-x}{M-k}$$
$$= k\sum_{y=0}^{N-M} \binom{k+y}{k}\binom{N-k-y}{M-k} = k\binom{N+1}{M+1}$$

$$E(X) = \frac{k\binom{N+1}{M+1}}{\binom{N}{M}} = \frac{N+1}{M+1} \cdot k \tag{1.3}$$

where the Lemma was applied with $a = N - k$, $b = M - k$ and $c = k$. Similarly,

$$\begin{aligned} \binom{N}{M} E(X(X+1)) &= \sum_{x=k}^{k+N-M} x(x+1)\binom{x-1}{k-1}\binom{N-x}{M-k} \\ &= \sum_{x=k}^{k+N-M} k(k+1)\binom{x+1}{k+1}\binom{N-x}{M-k} \\ &= k(k+1)\sum_{y=0}^{N-M}\binom{k+1+y}{k+1}\binom{N-k-y}{M-k} = k(k+1)\binom{N+2}{M+2} \end{aligned}$$

$$E(X(X+1)) = \frac{k(k+1)\binom{N+2}{M+2}}{\binom{N}{M}} = \frac{k(k+1)(N+1)(N+2)}{(M+1)(M+2)}$$

$$\begin{aligned} \mathrm{Var}(X) &= EX(X+1) - EX - (EX)^2 \\ &= \frac{k(k+1)(N+1)(N+2)}{(M+1)(M+2)} - \frac{k(N+1)}{M+1} - \left(\frac{k(N+1)}{M+1}\right)^2 \end{aligned}$$

With some algebra this simplifies to

$$\mathrm{Var}(X) = \frac{(N+1)(N-M)}{(M+1)^2(M+2)} \cdot k(M+1-k) \tag{1.4}$$

Note that for a fixed N and M, the expected value of X is linear in k and the variance is parabolic. The variance is highest at values of k nearest $(M+1)/2$, and the variance for k and $M+1-k$ are the same. (For example, the variance is the same for $k = 1$ or $k = M$.)

Example. Suppose there are 3 white balls, 4 red, and from the mixture it is desired to draw 2 red balls. Let X be the random variable representing the number of draws. Then $N = 7, M = 4, k = 2$, and $X \sim$ nhg(7,4,2). The mean and variance of X are calculated directly in the table below.

x	$p_X(x)$	$p_X(x) \cdot x$	$p_X(x) \cdot (x - \overline{X})^2$
2	$\binom{1}{1}\binom{5}{2}/\binom{7}{4} = 10/35$	$20/35$	$(10/35)(10-16)^2/25 = 360/875$
3	$\binom{2}{1}\binom{4}{2}/\binom{7}{4} = 12/35$	$36/35$	$(12/35)(15-16)^2/25 = 12/875$
4	$\binom{3}{1}\binom{3}{2}/\binom{7}{4} = 9/35$	$36/35$	$(9/35)(20-16)^2/25 = 144/875$
5	$\binom{4}{1}\binom{2}{2}/\binom{7}{4} = 4/35$	$20/35$	$(4/35)(25-16)^2/25 = 324/875$
Total	1	$\overline{X}$ =16/5	$\text{Var}(X) = 24/25$

Table 1.1: Sample Calculations of Mean and Variance

These quantities are also calculated from Equations 1.3 and 1.4:

$$E(X) = \frac{N+1}{M+1}k = \frac{16}{5}$$

$$\text{Var}(X) = \frac{(N+1)(N-M)}{(M+1)^2(M+2)}k(M-k+1) = \frac{24}{25}$$

1.3 Including Degenerate Cases

In the foregoing discussion it was assumed that the parameters of the distribution satisfy $N > M \geq k > 0$. This can easily be generalized to $N \geq M \geq k \geq 0$ by including a couple of degenerate cases.

First consider the case $k = 0$. Here the random variable X can simply be defined to take the value 0 with probability 1. It is easily checked that Equations 1.3 and 1.4 for the mean and variance of are still valid and give the correct answer of zero.

Assume, then, that $k > 0$. If $N > M$, then the original condition of $N > M \geq k > 0$ is true and all previous results apply. So the only other case to consider is

$N = M$. Appealing again to the model of drawing from a collection of red and white balls, it is clear that if all the balls are red, then the number of balls that must be drawn to obtain k red ones is exactly k. Putting $N = M$ into Equation 1.3 gives E(X) $= k$, and Equation 1.4 gives Var(X) $= 0$, both correct when $N = M$. The previously specified range, $k \leq X \leq k + N - M$, correctly forces $X = k$, and so remains valid. Finally, Equation 1.1 correctly computes the probability mass function at $X = k$:

$$f_X(k) = \frac{\binom{k-1}{k-1}\binom{N-k}{N-k}}{\binom{N}{N}} = 1.$$

By including these two degenerate cases, the following Theorem is obtained:

Theorem. Suppose that from a collection of N objects of which M are specially marked, objects are drawn randomly until exactly k of the marked ones have been obtained, where $N \geq M \geq k \geq 0$. Let X be a random variable representing the number of draws. Then the probability that $X = x$ for any $x \in \mathbb{Z}$ is given by:

$$f_X(x) = \begin{cases} \binom{x-1}{k-1}\binom{N-x}{M-k}/\binom{N}{M} & \text{if } 0 < k \leq x \leq k + N - M \\ 1 & \text{if } k = x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

The mean and variance of X are:

$$E(X) = \frac{N+1}{M+1} \cdot k \tag{1.6}$$

$$\text{Var}(X) = \frac{(N+1)(N-M)}{(M+1)^2(M+2)} \cdot k(M+1-k) \tag{1.7}$$

Proof. When $N > M \geq k > 0$, these results follow directly from Equations 1.1, 1.3, and 1.4. For the case $k = 0$, it is seen that the probability mass function is equal to 1 when $x = 0$ and is zero for all other values of x. The correctness of the other two formulas was checked previously. When $k > 0$ and $N = M$, all three formulas were also validated in the previous discussion.

1.4 Generalizations

The Negative Hypergeometric distribution may be generalized to include non-integer parameters. It is easily checked that Equation 1.1 may be written as:

$$f_X(x) = \binom{N-M}{x-k} \frac{\Gamma(M+1)\ \Gamma(x)\ \Gamma(N-x+1)}{\Gamma(k)\ \Gamma(M-k+1)\ \Gamma(N+1)}$$

With a simple change of variable $Y = X - k$ (so that Y has the range 0, 1, ..., $N - M$), the resulting distribution

$$f_Y(y) = \binom{N-M}{y} \frac{\Gamma(M+1)\ \Gamma(y+k)\ \Gamma(N-k-y+1)}{\Gamma(k)\ \Gamma(M-k+1)\ \Gamma(N+1)}$$

is now valid for real values of k, M and N so long as $k > 0$, $M > k - 1$, and $N - M$ is a natural number, since further substitution of $\alpha = k$, $\beta = M - k + 1$, and $m = N - M$ yields the standard *beta-binomial* distribution (Shuster and Sype 1987, 456; Bowman, Kastenbaum and Shenton 1992):

$$f_Y(y) = \binom{m}{y} \frac{\Gamma(\alpha+\beta)\ \Gamma(y+\alpha)\ \Gamma(\beta+m-y)}{\Gamma(\alpha)\ \Gamma(\beta)\ \Gamma(m+\alpha+\beta)}$$

K. D. Ling suggests another way to generalize the Negative Hypergeometric distribution by defining more sophisticated waiting times for the urn model. In the "later waiting time" scenario, drawing continues until $k1$ white and $k2$ red balls have been drawn from the urn. In the "sooner waiting time" model, $k1$ white *or* $k2$ red are drawn, whichever comes first. Recurrence relations for the means of these random variables are found in Ling's paper (Ling 1993).

To illustrate the "later waiting time" for $k1 = k2 = 1$, the following problem is adapted from Derman (1973, 311):

> A committee is to be drawn randomly from a group consisting of 7 women and 3 men. How many people must be chosen to ensure that the committee has at least one woman and one man? Calculate the mean and standard deviation of this random variable.

To apply the Negative Hypergeometric distribution to this problem, first define a Bernoulli random variable Y which represents the gender of the first person drawn, 0 for female and 1 for male. Then let X be the number of subsequent draws required to select a person of the opposite sex, plus one. Thus,

$$X|Y \sim \begin{cases} \text{nhg}(9,3,1) + 1 & \text{if } Y = 0 \\ \text{nhg}(9,7,1) + 1 & \text{if } Y = 1 \end{cases}$$

$$P(Y = y) = \begin{cases} \frac{7}{10} & \text{if } Y = 0 \\ \frac{3}{10} & \text{if } Y = 1 \end{cases}$$

Combined with Equations 1.6 and refvariance these lead directly to:

$$E(X|Y) = \begin{cases} \frac{10}{4} + 1 = \frac{7}{2} & \text{if } Y = 0 \\ \frac{10}{8} + 1 = \frac{9}{4} & \text{if } Y = 1 \end{cases}$$

$$\text{Var}(X|Y) = \begin{cases} \frac{10 \cdot 6 \cdot 3}{4^2 \cdot 5} = \frac{9}{4} & \text{if } Y = 0 \\ \frac{10 \cdot 2 \cdot 7}{7^2 \cdot 8} = \frac{5}{14} & \text{if } Y = 1 \end{cases}$$

Then applying standard probability calculations,

$$E(E(X|Y)) = \frac{7}{10} \cdot \frac{7}{2} + \frac{3}{10} \cdot \frac{9}{4} = \frac{25}{8}$$

$$E(\{E(X|Y)\}^2) = \frac{7}{10} \cdot \frac{7}{2}^2 + \frac{3}{10} \cdot \frac{9}{4}^2 = \frac{323}{32}$$

$$E(\text{Var}(X|Y)) = \frac{7}{10} \cdot \frac{9}{4} + \frac{3}{10} \cdot \frac{5}{14} = \frac{471}{280},$$

the answer is:

$$E(X) = E(E(X|Y)) = \frac{25}{8} = 3.125$$

$$\begin{aligned} \text{Var}(X) &= \text{Var}(E(X|Y)) + E(\text{Var}(X|Y)) \\ &= E(\{E(X|Y)\}^2) - \{E(E(X|Y))\}^2 + E(\text{Var}(X|Y)) \\ &= \frac{323}{32} - \left(\frac{25}{8}\right)^2 + \frac{471}{280} = \frac{4503}{2240}. \end{aligned}$$

Chapter 2

Some Applications

2.1 Overview

Although largely neglected in probability textbooks, the Negative Hypergeometric distribution does occasionally arise in applications, such as educational testing, linguistics and biostatistics (Miler and Fridel 2007).

One noteworthy application is the distribution of mental test scores. Assuming the distribution of true abilities of test takers is a beta-distributed random variable and each test-taker's chance of a correct score on all test items are independent, identically distributed Bernoulli where the probability of success is linearly dependent on ability, then the resulting distribution of raw test scores has the Negative Hypergeometric distribution (with generalized parameters). For details see Lord and Novick (1968).

Another example involving learning comes from a study of birds' ability to remember the location of food (Ridout 1999). An experiment involving coal tits was conducted as follows. Birds were released individually in a room with four feeders, one of which they had fed from previously, and three empty ones. The one with food was in the same location as before, but with the food now hidden. In searching for the correct feeder, the birds rarely visited an empty feeder twice, so the number of feeders visited is modeled as Negative Hypergeometric. It turned out that the birds' memory of the location of the filled feeder from previous experience is equivalent to replacing the single filled feeder with 3.7 ones and then having the birds find the filled feeder by trial and error; that is, a generalized Negative Hypergeometric with k=1, M=3.7, and N-M=3 (see section 1.4).

The purpose of the rest of this chapter is to suggest a couple new applications of the Negative Hypergeometric distribution; one to gaming, and one to customer satisfaction surveys.

2.2 A Bonus Game for Slot Machines

Many slot machines feature a "bonus game" with almost certain payoffs which is available to the player at high points during regular slot play. The bonus game can be, for example, spins of a wheel of fortune. The Negative Hypergeometric distribution suggests an interesting variation on this.

The bonus game pictured in Figure 2.1 operates similarly to spinning a wheel, except the player selects squares from a rectangular grid until a certain number of "jokers" have been revealed. In the example shown, the play ends with the third joker (compared to, say, three spins of a wheel). This adds interest to the bonus game since the number of prizes will vary and, unlike the wheel of fortune, the player has a chance to win all of the prize squares.

The underlying game board with all squares exposed is shown in Figure 2.2. From this data the expected length of play and winnings can be calculated, as well as the standard deviation of those quantities. Letting k be the number of jokers when play stops, the length of play has the Negative Hypergeometric distribution, in this case with $N = 16$ and $M = 4$. Equations 1.6 and 1.7 give the mean and variance of the

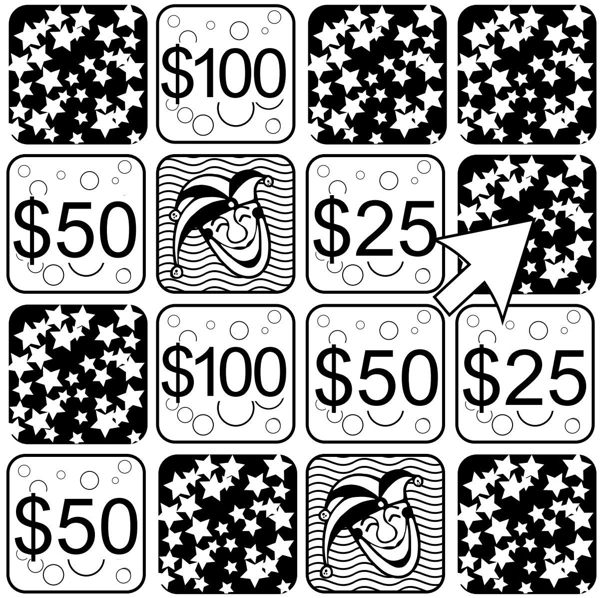

Figure 2.1: Bonus Game In Play. So far the player has collected \$400, but the next selection will end the bonus play round.

length of play X.

Of greater interest to the player is the number of prize squares exposed during play. Denoting this random variable by Y, then clearly $Y = X - k$ (since there are exactly k jokers exposed at the end of play). Thus:

$$\mu_Y = \mathrm{E}Y = \mathrm{E}X - k = \frac{N+1}{M+1}k - k = \frac{N-M}{M+1}k \tag{2.1}$$

and the variance of Y equals the variance of X:

$$\sigma_Y^2 = \mathrm{Var}\ Y = \mathrm{Var}\ X = \frac{(N+1)(N-M)}{(M+1)^2(M+2)}\ k(M+1-k) \tag{2.2}$$

Now consider winnings. Let W be a random variable which is the total of all the amounts shown on prize squares at the end of play. One would expect that

$$\mathrm{E}(W) = \mu_Y \overline{A}$$

Figure 2.2: Bonus Game Board Revealed. All prize squares and jokers are exposed.

where $\overline{A}$ is the average of all prize square amounts. This is indeed the case, and will be justified below. For the game shown in Figure 2.2, with play to stop at the third joker ($k = 3$), $\overline{A} = \$50$ and $\mu_Y = (12/5)\cdot 3 = 7.2$ prize squares, so the expected value of the winnings is $7.2 \cdot \$50 = \$360$. Calculation of the variance of the winnings is more complicated. To derive the formulas for $\mathrm{E}(W)$ and $\mathrm{Var}(W)$, some notation is needed. Let:

N = number of squares on the board

M = number of joker squares, with $N - M > 1$

k = number of jokers to draw before play ends $\quad (1 \le k \le M)$

$N - M$ = number of prize squares

A_j = amount shown on the jth prize square, $j = 1, 2, \ldots, N - M$

(If j is outside this range, take $A_j = 0$)

$\overline{A}$ = average prize amount

$\overline{A^2}$ = average squared prize amount

C_r^n = number of r-combinations of n objects

(This is the notation used in Chen and Koh 1992)

Ω_y = the set of all y-combinations of $\{1, 2, \ldots, N-M\}$,

where $y \in \{0, 1, 2, \ldots, N-M\}$

$\Omega = \cup_{y=0}^{N-M} \Omega_y \quad$ = the space of all game outcomes

$y(\omega)$ = the number of elements in the combination ω, for $\omega \in \Omega$

$$\chi_j(\omega) = \begin{cases} 1 & \text{if } j \text{ is in the combination } \omega \\ 0 & \text{otherwise.} \end{cases}$$

Two combinatorial identities are needed in the derivation:

$$\sum_{\omega \in \Omega_y} \chi_j(\omega) = \text{no. of combinations in which } j \text{ appears}$$

$$= C_{y-1}^{N-M-1} = \frac{y}{N-M}\ C_y^{N-M} = \frac{y}{N-M}\ |\Omega_y|$$

$$\sum_{\omega \in \Omega_y} \chi_i(\omega)\chi_j(\omega) = \text{no. of combinations in which } i \text{ and } j \text{ appear}$$

$$= C_{y-2}^{N-M-2} = \frac{y(y-1)}{(N-M)(N-M-1)}\ C_y^{N-M}$$

$$= \frac{y(y-1)}{(N-M)(N-M-1)}\ |\Omega_y| \qquad (i \neq j)$$

To begin, the probability of each outcome ω is given by

$$P(\omega) = P(\omega | Y = y(\omega))P(Y = y(\omega)) = |\Omega_{y(\omega)}|^{-1} P(Y = y(\omega))$$

and the player's winnings are defined as

$$w(\omega) = \sum_j \chi_j(\omega) \cdot A_j$$

where the summation may be taken over all natural numbers. Thus,

$$\begin{aligned}
\mathrm{E}(W \mid Y = y) &= \frac{1}{P(y)} \sum_{\omega \in \Omega_y} P(\omega) w(\omega) \\
&= \frac{1}{P(y)} \sum_{\omega \in \Omega_y} |\Omega_y|^{-1} P(y) \sum_j \chi_j(\omega) \cdot A_j \;=\; |\Omega_y|^{-1} \sum_j A_j \sum_{\omega \in \Omega_y} \chi_j(\omega) \\
&= |\Omega_y|^{-1} \sum_j A_j \frac{y}{N-M} \; |\Omega_y| \;=\; y\overline{A}
\end{aligned} \tag{2.3}$$

and

$$\mathrm{E}(W) \;=\; \mathrm{E}(y\overline{A}) \;=\; \mu_Y \overline{A}, \tag{2.4}$$

as suggested earlier. To compute $\mathrm{Var}(W)$, one additional identity is needed:

$$\sum_{i \neq j} A_i A_j = (N-M)^2 \, \overline{A}^2 - (N-M) \, \overline{A^2}$$

where, once again, the summation may be taken to be over all natural numbers. (This identity is easily verified by expanding $\overline{A}^2$.)

First calculate, using the assumption that $N > M + 1$,

$$\begin{aligned}
\sum_{\omega \in \Omega_y} w^2(\omega) &= \sum_{\omega \in \Omega_y} \left[\sum_j \chi_j(\omega) A_j^2 + \sum_{i \neq j} \chi_i(\omega) \chi_j(\omega) A_i A_j \right] \\
&= \sum_j A_j^2 \sum_{\omega \in \Omega_y} \chi_j(\omega) + \sum_{i \neq j} A_i A_j \sum_{\omega \in \Omega_y} \chi_i(\omega) \chi_j(\omega) \\
&= \sum_j A_j^2 \frac{y}{N-M} \; |\Omega_y| + \sum_{i \neq j} A_i A_j \frac{y(y-1)}{(N-M)(N-M-1)} \; |\Omega_y| \\
&= |\Omega_y| \left[y \; \overline{A^2} + \frac{y(y-1)[(N-M)\overline{A}^2 - \overline{A^2}]}{N-M-1} \right]
\end{aligned}$$

which gives

$$
\begin{aligned}
E(W^2 \mid Y=y) &= \frac{1}{P(Y=y)} \sum_{\omega \in \Omega_y} P(\omega) w^2(\omega) \\
&= \frac{1}{P(Y=y)} \sum_{\omega \in \Omega_y} |\Omega_y|^{-1} P(Y=y) w^2(\omega) \\
&= y\ \overline{A^2} + \frac{y(y-1)[(N-M)\overline{A}^2 - \overline{A^2}]}{N-M-1}
\end{aligned}
$$

Combining this with Equation 2.3:

$$
\begin{aligned}
\mathrm{Var}(W \mid Y=y) &= E(W^2 \mid Y=y) - (E(W \mid Y=y))^2 \\
&= \left[y\ \overline{A^2} + \frac{y(y-1)[(N-M)\overline{A}^2 - \overline{A^2}]}{N-M-1} \right] - (y\overline{A})^2 \\
&= \frac{\overline{A^2} - \overline{A}^2}{N-M-1}[(N-M)y - y^2],
\end{aligned}
$$

after simplification. So

$$
\mathrm{E}(\mathrm{Var}(W \mid Y)) = \frac{\overline{A^2} - \overline{A}^2}{N-M-1}[(N-M)\mu_Y - (\sigma_Y^2 + \mu_Y^2)] \tag{2.5}
$$

Now from Equation 2.1,

$$
\begin{aligned}
(N-M)\mu_Y - \mu_Y^2 &= \mu(N-M-\mu) \\
&= \frac{N-M}{M+1} k[(N-M)(1 - \frac{k}{M+1}))] = \frac{(N-M)^2}{(M+1)^2} k(M+1-k)
\end{aligned}
$$

and then using Equation 2.2,

$$
\begin{aligned}
&(N-M)\mu_Y - \mu_Y^2 - \sigma_Y^2 \\
&\qquad = \frac{(N-M)^2}{(M+1)^2} k(M+1-k) - \frac{(N+1)(N-M)}{(M+1)^2(M+2)} k(M+1-k) \\
&\qquad\qquad = \frac{(N-M)(N-M-1)}{(M+1)(M+2)} k(M+1-k).
\end{aligned}
$$

Inserting the last result into 2.5 gives

$$
\begin{aligned}
&\mathrm{E}(\mathrm{Var}(W \mid Y)) \\
&\qquad = \frac{\overline{A^2} - \overline{A}^2}{N-M-1} \; \frac{(N-M)(N-M-1)}{(M+1)(M+2)} \; k(M+1-k) \\
&\qquad\qquad = \frac{(\overline{A^2} - \overline{A}^2)(M+1)}{N+1} \; \sigma_Y^2, \qquad (2.6)
\end{aligned}
$$

from 2.2. The variance of W is now easily calculated from Equations 2.3 and 2.6 as

$$
\begin{aligned}
\mathrm{Var}(W) &= \mathrm{E}(\mathrm{Var}(W \mid Y)) \; + \; \mathrm{Var}(\mathrm{E}(W \mid Y)) \\
&= \frac{(\overline{A^2} - \overline{A}^2)(M+1)}{N+1} \; \sigma_Y^2 + \overline{A}^2 \sigma_Y^2 \; = \; \frac{(N-M)\overline{A}^2 + (M+1)\overline{A^2}}{N+1} \; \sigma_Y^2 \qquad (2.7)
\end{aligned}
$$

For the payoffs shown in Figure 2, $\overline{A} = 50.0$ and $\overline{A^2} = 3437.5$. Plugging these values into Equations 1.6, 1.7, 2.4, and 2.7, produces the results shown in the table below. (*Note:* The results in the table and the formulas derived above were checked independently with a computer program, which is reproduced in Appendix A. The program recomputed the values by summing over all possible outcomes of the game.)

Variable	$k = 1$	$k = 2$	$k = 3$	$k = 4$
No. of prize squares (Y)				
expected value (μ_Y)	2.4	4.8	7.2	9.6
standard deviation (σ_Y)	2.3	2.9	2.9	2.3
Winnings (W)				
expected value	\$120.00	\$240.00	\$360.00	\$480.00
standard deviation	\$123.88	\$150.50	\$150.50	\$123.88

Table 2.1: Statistics for the Bonus Game

2.3 Estimating Customer Satisfaction

For this application suppose there is a company named Magic Seminars which puts on motivational programs at different cities in the U.S. and Canada. The seminars are attended by anywhere from 100 to 2000 people. Management has decided to obtain three testimonials from each seminar that had at least 500 in attendance. To do so, professional callers will attempt to reach attendees after the seminar by phone. If a message is left and the call is not returned within 48 hours, the caller will attempt to reach the person one more time. The callers are instructed to keep track of the total number of calls made, including the number of messages left, before three testimonials are obtained. The list of attendees is randomized to avoid any possible bias in the selection of people who are called.

To analyze this situation mathematically, consider a particular seminar event. The exact number of attendees is known, and is denoted N. Attendees can be classified as follows:

> Type 1: Sufficiently satisfied with the seminar that the person will provide a testimonial within 48 hours of being called;
>
> Type 2: All attendees who are not of Type 1.

Let the number of Type 1 attendees be denoted by M. Since the call list is randomized, the total number of calls, including messages left, that must be made to obtain 3 testimonials is a Negative Hypergeometric random variable X with $k = 3$.

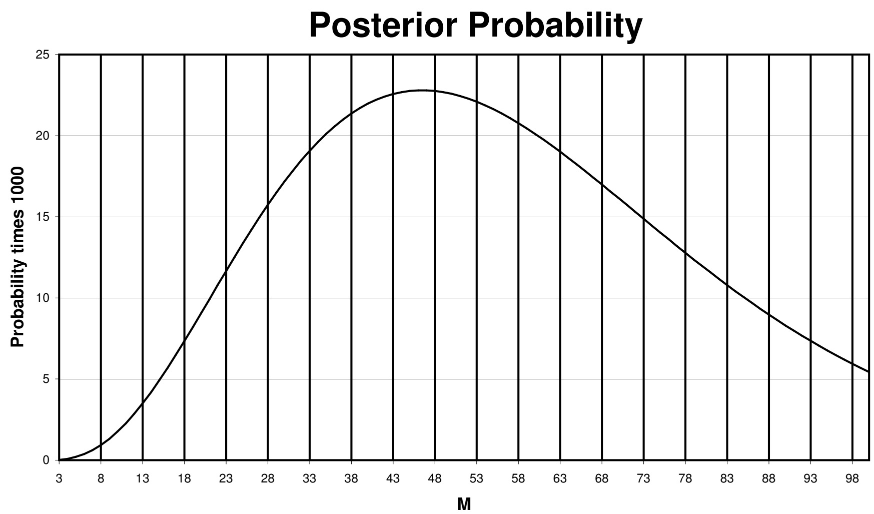

Figure 2.3: Graph of posterior probabilities for various values of M when X=32 when N=500 and k=3. The maximum is at M=46.

Presented this way, the management of Magic Seminars comes up with a novel idea: to use the variable X, which is available at no additional cost once the testimonials are obtained, to estimate overall customer satisfaction. For each seminar, M/N times 100% will be the numerical satisfaction figure. Consultation with a statistician (who is familiar with the Negative Hypergeometric) reveals that this figure may be estimated using the technique of *maximum posterior likelihood*.

To explain how this works, consider a particular seminar event and assume for the moment that M is known. After the callers have obtained three testimonials, X is also known. The value of N is the number of attendees, and $k = 3$. Therefore, it is possible to compute the probability of this value of X using Equation 1.5.

For example, suppose there are 500 attendees and it takes 32 calls to obtain three testimonials. Figure 2.3 displays the The probability that $X = 32$ for the values of M between 3 and 100. The maximum posterior probability occurs at $M = 46$, giving a satisfaction score for this event of $(46/500)\cdot 100\% = 9.2\%$.

It is not necessary to compute the posterior probability for every possible value of M in order to determine the maximum. In fact, the maximum occurs at

$$M_0 = \left\lfloor \frac{k(N+1)}{x} \right\rfloor.$$

To see this, observe that the posterior probability

$$p(M; N, k, x) = \frac{\binom{x-1}{k-1}\binom{N-x}{M-k}}{\binom{N}{M}}$$

increases by the proportion

$$\frac{p(M+1; N, k, x)}{p(M; N, k, x)} = \frac{(M+1)(k+N-x-M)}{(N-M)(M+1-k)}$$

going from M to $M+1$. Denoting the numerator of the last fraction by $f(M)$ and the denominator by $g(M)$, it follows that $p(M+1) > p(M)$ if and only if

$$f(M) - g(M) = k(N+1) - (M+1)x > 0;$$

that is, if and only if $M < \frac{k(N+1)}{x} - 1$. In the case that $\frac{k(N+1)}{x}$ is not an integer, then it is easily checked that

$$\cdots < p(M_0 - 1) < p(M_0) > p(M_0 + 1) > p(M_0 + 2) > \ldots.$$

If $\frac{k(N+1)}{x}$ is an integer, then

$$\cdots < p(M_0 - 2) < p(M_0 - 1) = p(M_0) > p(M_0 + 1) > \ldots.$$

In the latter case the maximum occurs at both $M_0 - 1$ and M_0.

Interestingly, the hypergeometric distribution also works for this application. The interpretation is that x samples are drawn, from which k are of Type 1, and the fact that the last sample drawn was Type 1 is immaterial. (In the usual notation for the hypergeometric distribution, the roles of x and k are reversed from their roles here.) Letting P_H be the probability calculated from the hypergeometric,

$$P_H(K = k \mid N, M, x) = \frac{\binom{M}{k}\binom{N-M}{x-k}}{\binom{N}{x}} = \frac{k}{x}\, p(M; N, k, x)$$

where $p(M; N, k, x)$ is the probability calculated from the Negative Hypergeometric distribution. Since k and x are constants in this application, the value of M that maximizes the posterior probability will be the same in either case.

Appendix A

Computations for the Bonus Game

All the results for the bonus game shown in Table 2.2, as well as equations 2.1, 2.2, 2.4, and 2.7, were all verified independently by having the computer directly sum over all possible combinations. The final printout of the checking program is listed below, followed by the Python source code. (Python a freely available, object-oriented programming language that supports high-level dynamic data types. See Martelli (2006) for more information on the language.)

```
Input data:
   N= 16 ,   M = 4
   A = [100, 25, 25, 50, 25, 25, 100, 50, 25, 50, 25, 100]

Step 1:  compute Ab and A2b
   Ab =  50.0    A2b =  3437.5

Step 2:  verify mean and variance of X for k = 3
   X= 3  prob= 0.00714285714286
   X= 4  prob= 0.0197802197802
   X= 5  prob= 0.0362637362637
   X= 6  prob= 0.0549450549451
   X= 7  prob= 0.0741758241758
   X= 8  prob= 0.0923076923077
   X= 9  prob= 0.107692307692
   X= 10  prob= 0.118681318681
   X= 11  prob= 0.123626373626
   X= 12  prob= 0.120879120879
   X= 13  prob= 0.108791208791
   X= 14  prob= 0.0857142857143
   X= 15  prob= 0.05
By direct summation: p= 1.0   xb= 10.2   VarX= 8.16
From formula:        p= 1.0   xb= 10.2   VarX= 8.16
```

```
Step 3:  verify the formulas for the mean and variance of Y and W
-----------------------------------------------------------------

k =  1    summing 4096 possible outcomes
Direct summation: p= 1.0  mu= 2.4  sig2= 5.44
       EW= 120.0  VarW= 15100.0
From formula:     p= 1.0  mu= 2.4  sig2= 5.44
       EW= 120.0  VarW= 15100.0

k =  2    summing 4096 possible outcomes
Direct summation: p= 1.0  mu= 4.8  sig2= 8.16
        EW= 240.0  VarW= 22650.0
From formula:     p= 1.0  mu= 4.8  sig2= 8.16
        EW= 240.0  VarW= 22650.0

k =  3    summing 4096 possible outcomes
Direct summation: p= 1.0  mu= 7.2  sig2= 8.16
        EW= 360.0  VarW= 22650.0
From formula:     p= 1.0  mu= 7.2  sig2= 8.16
        EW= 360.0  VarW= 22650.0

k =  4    summing 4096 possible outcomes
Direct summation: p= 1.0  mu= 9.6  sig2= 5.44
        EW= 480.0  VarW= 15100.0
From formula:     p= 1.0  mu= 9.6  sig2= 5.44
        EW= 480.0  VarW= 15100.0
```

The source code that generated the above printout follows:

```
N=16;  M=4;  n = N-M;
A = [100,25,25,50,25,25,100,50,25,50,25,100]

print "Input data:"
print '   N=',N, ',   M =',M
print '   A =', A
print; print


print "Step 1:  compute Ab and A2b"
#-----------------------------------------------------------
Ab = 0.0;  A2b = 0.0;
for i in range(n):
```

```
   Ab += A[i];  A2b += A[i]*A[i]
Ab /= n;  A2b /= n
print '   Ab = ',Ab, '   A2b = ',A2b
print; print


# Some extra test code
# s = 0
# for i in range(n):
#    for j in range(n):
#  if i != j:  s += A[i]*A[j]
# print s, n*n*Ab*Ab - n*A2b
# print


# Code for computations involving r.v. X
#-----------------------------------------------------------
def ncomb(n,r):  # number of r-combinations of n objects
   if r == 0:  return 1.0
   m = 1.0
   for i in range(r,0,-1):  m *= (1.0*n-r+i)/i
   return m

def px(x):  return ncomb(x-1,k-1)*ncomb(16-x,4-k)/ncomb(16,4)
#   probability mass function for X



k = 3
print "Step 2:  verify mean and variance of X for k =",k
#--------------------------------------------------------
p = 0;  xb = 0; x2b = 0
for x in range(3,16):
   print '   X=',x,' prob=',px(x)
   p +=  px(x);  xb += x*px(x);  x2b += x*x*px(x)
print 'By direct summation: p=',p, '  xb=',xb, \
   '  VarX=',x2b-xb*xb
print 'From formula:        p=',1.0, '  xb=',(N+1.0)*k/(M+1),\
   '  VarX=',(N+1.0)*(N-M)*k*(M-k+1)/((M+1)*(M+1)*(M+2))
print; print



# Code for generating the outcome space & amounts of winnings
#-------------------------------------------------------------
```

```
def firstc(y):  return range(y)

def nextc(comb,k):
   i = k-1
   comb[i] += 1
   while i>=0 and comb[i]>=n-k+1+i:
         i -= 1
         comb[i] += 1
   # print '    ',i,c
   if comb[0] > n-k:  return 0
   for j in range(i+1,k):
         comb[j] = comb[j-1] + 1
   # print '    ',i,c
   return 1

def win(c):
   w = 0
   if len(c) == 0:  return 0
   for i in range(len(c)):  w += A[c[i]]
   return w

def firstoc():
   oc = [0,[], px(k)]
   return oc

def nextoc(oc):
   y = oc[0];  comb = oc[1]
   # print y,comb
   if y == 0:
         oc[0] = 1;  oc[1] = firstc(1)
         oc[2] = px(k+1)/ncomb(n,1)
         return 1
   if nextc(comb,y):
         oc[1] = comb;  return 1
   y += 1
   if y > n:  return 0
   oc[0] = y;  oc[1] = firstc(y);  oc[2] = px(y+k)/ncomb(n,y)
   return 1


# Some additional test code
#  comb = firstc(2);  print comb, win(comb)
#  while nextc(comb,2):  print comb, win(comb)
#  print ncomb(4,2), ncomb(4,1), ncomb(4,0)
#  print px(3);  print
```

```
print 'Step 3:  verify the formulas for the mean and', \
   'variance of Y and W'
#-----------------------------------------------------------
print '--------------------------------' + \
   '---------------------------------'
for k in range(1,M+1):
   p = 0; yb=0; y2b=0;  wb = 0; w2b=0;  py=0; wby=0;  w2by=0
   oc = firstoc();   # contributes 0 to all means
   p += oc[2];  noc = 1;  # but it contributes to probability
   # print oc, win(oc[1])
   while nextoc(oc):
        y = oc[0];  comb = oc[1];  prob = oc[2];  w = win(comb)
        p += prob;  noc += 1
        yb += prob*y;  y2b +=  prob*y*y
        wb += prob*w;  w2b += prob*w*w
        if y == 5:  # checks values on the condition that y=5
            py += prob;  wby += prob*w;  w2by += prob*w*w
   mu = yb;  sig2 = y2b-yb*yb
   EW = wb;  VarW = w2b-wb*wb
   wby /= py;  w2by /= py
   print; print 'k = ',k,'   summing',noc,'game outcomes'
   print 'Direct summation: p=',p, ' mu=',mu, \
      ' sig2=',sig2,' EW=',EW, ' VarW=',VarW
   print 'From formula:     p=',1.0, \
      ' mu=', (N-M)*k/(M+1.0), \
      ' sig2=', (N+1.0)*(N-M)*k*(M-k+1)/((M+1)*(M+1)*(M+2)),  \
      ' EW=', mu*Ab,  \
      ' VarW=', sig2*((N-M)*Ab*Ab+(M+1.0)*A2b)/(N+1)
   # print 'mu, sig2',mu,sig2, \
         #(N-M)*k/(M+1.0), \
         #(N+1.0)*(N-M)*k*(M+1-k)/((M+1)*(M+1)*(M+2))
   # print 'For y=5: wby, varWy', wby, w2by-wby*wby, \
         #(A2b-Ab*Ab)*((N-M)*5-25)/(N-M-1.0)
   # print '(N-M)*mu-mu*mu',(N-M)*mu-mu*mu, \
         #(N-M)*(N-M)*k*(M+1.0-k)/((M+1.0)*(M+1))
   #print '(N-M)*mu-mu*mu-sig2',(N-M)*mu-mu*mu-sig2, \
         #(N-M)*(N-M-1)*k*(M+1.0-k)/((M+1.0)*(M+2))
   #print 'VarW', \
         #sig2*Ab*Ab+sig2*(A2b-Ab*Ab)*(M+1.0)/(N+1.0), \
         #sig2*((N-M)*Ab*Ab+(M+1.0)*A2b)/(N+1)
print
```

Appendix B

Customer Satisfaction Calculations

The data for the graph of probability as a function of M (in Table 2.3) was calculated with a Python program. A sample print out is shown below and the source code follows.

```
Input paramaters:
   N= 500   k= 3   x= 32

M, p*1000,   p(M+1)/p(M)
-----------------------------
30,17.1836  1.03883
31,17.8508  1.035218
32,18.4795  1.031838
33,19.0678  1.028666
34,19.6144  1.025684
35,20.1182  1.022874
36,20.5784  1.020221
37,20.9945  1.017711
38,21.3663  1.015332
39,21.6939  1.013074
40,21.9775  1.010927
41,22.2177  1.008882
42,22.4150  1.006932
43,22.5704  1.00507
44,22.6848  1.003289
45,22.7594  1.001584
46,22.7955  0.99995
47,22.7944  0.998381
48,22.7575  0.996874
49,22.6863  0.995424
50,22.5825  0.994028
51,22.4476  0.992682
```

```
52,22.2834  0.991384
53,22.0914  0.99013
54,21.8733  0.988919
55,21.6310  0.987746
56,21.3659  0.986612
57,21.0798  0.985512
58,20.7744  0.984446
59,20.4513  0.983411
60,20.1120  0.982406
```

Here is the Python code that generated the above printout:

```
N=500;  k=3;  x=32

# Posterior probability calculation for M
# (output is captured in a TXT file)
#--------------------------------------------------------
def ncomb(n,r):  # number of r-combinations of n objects
   if r == 0:  return 1.0
   m = 1.0
   for i in range(r,0,-1):  m *= (1.0*n-r+i)/i
   return m

def px(M):  return ncomb(x-1,k-1)*ncomb(N-x,M-k)/ncomb(N,M)
#   probability mass function for X

print 'Input paramaters:'
print '   N=',N, '  k=',k, '  x=',x;  print

print "M, p*1000,   p(M+1)/p(M)"
print "----------------------------"
for M in range(k, min(k+N-x+1,101)):
   ratio = round((M+1.0)*(k+N-x-M)/((N-M)*(M+1-k)),6)
   if M>=30 and M<=60:
         print str(M)+","+str(round(1000*px(M),4))+'  '\
            +str(ratio)
```

Bibliography

Bowman, K. O., Marvin Kastenbaum, and L. R. Shenton. "The Negative Hypergeometric Distribution and Estimation by Moments." *Communications in Statistics - Simulation and Computation* 21, no. 2 (1992): 301-332.

Casella, George, and Roger L. Berger. *Statistical Inference.* 2nd ed. Pacific Grove, Calif.: Duxbury / Thompson Learning, 2002.

Chen, Chuan-Chong, and Khee-Meng Koh. *Principles and Techniques of Combinatorics.* Singapore: World Scientific Publishing Co., 1992.

Derman, Cyrus, Leon Gleser, and Ingram Olkin. *A Guide to Probability Theory and Application.* New York: Holt, Rinehart and Winston, 1973.

Feller, William. *An Introduction to Probability Theory and Its Applications.* 2nd ed. New York: John Wiley & Sons, 1957.

Graham, Ronald L., Donald E. Knuth and Oren Patashnik. *Concrete Mathematics.* 2nd ed. Reading, Mass.: Addison-Wesley Publishing Co., 1994.

Miller, Gregory K., and Stephanie L. Fridel. "A Forgotten Discrete Distribution? Reviving the Negative Hypergeometric Model." *The American Statistician* 61 (November 2007): 347-350.

Johnson, Norman L., and Samuel Kotz. *Urn Models and Their Application.* New York: John Wiley & Sons, 1977.

Ling, K. D. "Sooner and Later Waiting Distributions for Frequency Quota Defined on a Pólya-Eggenberger Urn Model." *Soochow Journal of Mathematics* 19, no. 2 (1993): 139-151.

Lord, Frederic M., and Melvin R. Novick. *Statistical Theories of Mental Test Scores.* Reading, Mass.: Addison-Wesley, 1968.

Martelli, Alex. *Python in a Nutshell.* 2nd ed. Sebastapol, Calif.: O'Reilly Media, 2006.

Ridout, Martin S. "Memory in Coal Tits: An Alternative Model." *Biometrics* 55 (June 1999): 660-662.

Schuster, Eugene F., and William R. Sype. "On the Negative Hypergeometric Distribution." *International Journal of Mathematical Education in Science and Technology* 18, no. 3 (1987): 453-459.

VITA

Graduate College
University of Nevada, Las Vegas

Steven N. Jones

Degrees:
Bachelor of Science, Computer Engineering, 1977
Massachusetts Institute of Technology

Master of Science, Electrical Engineering and Computer Science, 1978
Massachusetts Institute of Technology

Special Honors and Awards:
Honorable Mention in National Science Foundation competition (Winter 1977)
Elected to Phi Kappa Phi honor society (Spring 2012)
Received Re-Entry Scholarship award from UNLV Continuing Education
(Fall 2012 and Spring 2013)