Computational Statistics and Data Analysis 59 (2013) 41–51

Contents lists available at SciVerse ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

On computing the distribution function for the Poisson binomial distribution

Yili Hong

Department of Statistics, Virginia Tech, Blacksburg, VA 24061, USA

ARTICLE INFO

Article history:
Received 29 September 2011
Received in revised form 10 August 2012
Accepted 4 October 2012
Available online 12 October 2012

Keywords:
Characteristic function
k-out-of-n system
Longevity risk
Normal approximation
Sum of independent random indicators
Warranty returns

ABSTRACT

The Poisson binomial distribution is the distribution of the sum of independent and non-identically distributed random indicators. Each indicator follows a Bernoulli distribution and the individual probabilities of success vary. When all success probabilities are equal, the Poisson binomial distribution is a binomial distribution. The Poisson binomial distribution has many applications in different areas such as reliability, actuarial science, survey sampling, econometrics, etc. The computing of the cumulative distribution function (cdf) of the Poisson binomial distribution, however, is not straightforward. Approximation methods such as the Poisson approximation and normal approximations have been used in literature. Recursive formulae also have been used to compute the cdf in some areas. In this paper, we present a simple derivation for an exact formula with a closed-form expression for the cdf of the Poisson binomial distribution. The derivation uses the discrete Fourier transform of the characteristic function of the distribution. We develop an algorithm that efficiently implements the exact formula. Numerical studies were conducted to study the accuracy of the developed algorithm and approximation methods. We also studied the computational efficiency of different methods. The paper is concluded with a discussion on the use of different methods in practice and some suggestions for practitioners.

1. Introduction

1.1. Motivation

The Poisson binomial distribution describes the distribution of the sum of independent and non-identically distributed random indicators. Each indicator is a Bernoulli random variable and the individual probabilities of success vary. A special case of the Poisson binomial distribution is the ordinary binomial distribution, when all success probabilities are equal. The Poisson binomial distribution has many applications in different areas such as reliability, actuarial science, survey sampling, econometrics, and so on. The following gives examples from different areas.

- In some reliability applications, it is often of interest to predict the total number of failures for a fleet of products in the field. Hong et al. (2009) considered the prediction for the total number of field failures for a fleet of high-voltage power transformers. Due to the staggered entry of units into service, individual units in the field have different failure probabilities at a specified future time. Thus the total number of field failures follows a Poisson binomial distribution.
- In actuarial science, the total payout of an insurance company is often related to the Poisson binomial distribution. For example, Pitacco (2007) considered a one-year insurance coverage only providing a death benefit for n insureds. Let C denote the payout due at each death. The individual payout is either 0 or C with probability $1-p_j$ and p_j, respectively, where the death probability p_j varies from individual to individual. Assuming that the individual lifetimes are independent, the total payout for those n insureds is C times the total number of deaths which follows the Poisson binomial distribution.

E-mail address: yilihong@vt.edu.

0167-9473/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2012.10.006

- In econometrics, it is sometimes of interest to predict the number of corporation defaults (e.g., Duffie et al., 2007). The default probabilities differ from corporation to corporation because each corporation has its own unique situation on assets, debts, stock returns and so on. The number of corporation defaults at a future time also follows a Poisson binomial distribution.
- In engineering, Fernández and Williams (2010) provided several interesting examples such as multi-sensor fusion and reliability of k-out-of-n systems, which are related to the Poisson binomial distribution.
- In survey sampling, Chen and Liu (1997) presented an example where the inclusion probabilities of sampling units are different. The total number of units in the sample follows a Poisson binomial distribution.
- The Poisson binomial distribution also has wide applications in areas such as data mining of uncertain databases (Tang and Peterson, 2011), bioinformatics (Niida et al., 2012), and wind energy (Bossavy et al., 2012).

While the Poisson binomial distribution has many applications in different disciplines, the computing of the cumulative distribution function (cdf) of the distribution is not straightforward. Because the individual probabilities of success vary, the naive way of computing the cdf by using enumeration is not practical, even when the number of indicators is small (i.e., around 30). Approximation methods such as the Poisson approximation and normal approximations have been used in literature. There are situations, however, in which approximation methods do not perform well. Thus it is desirable to have a method to compute the exact values of the cdf. It is also useful to know in which situation approximation methods work well. In applications such as predictions for the number of failures and corporation defaults, the number of indicators is usually large. Thus the efficiency of algorithms for computing the exact values of the cdf is also important. This motivates us to provide efficient methods to compute the exact values of the cdf of the Poisson binomial distribution.

1.2. Related literature and this work

The study on the Poisson binomial distribution has a long history. Le Cam (1960) provided an upper bound for the error of the Poisson approximation. Normal approximations are widely used in practice. Volkova (1996) gave a normal approximation with second order correction and provided an upper bound for the error of the approximation. Hong et al. (2009) and Hong and Meeker (2010) applied the approximation in Volkova (1996) to warranty prediction applications. Recursive formulae are available in literature to compute the exact values of the cdf of the Poisson binomial distribution. For example, Barlow and Heidtmann (1984) described a recursive formula for computing the cdf. Chen et al. (1994) provided another recursive formula. Details for these recursive formulae are described in Section 2.5. Fernández and Williams (2010) gave a closed-form expression for the cdf using the technique of polynomial interpolation and the discrete Fourier transform.

The contribution of this paper is summarized as follows.

- We propose a simple derivation for an exact formula for the cdf of the Poisson binomial distribution, which gives the same form as that in Fernández and Williams (2010).
- We develop an algorithm that efficiently implements the exact formula, which outperforms existing methods.
- Numerical studies were conducted to compare the accuracy of the algorithm and approximation methods. We also compared the computational efficiency of different methods.
- Based on the numerical studies, we provide a discussion on the advantages and disadvantages of different methods and some guidelines for practitioners.

The statistical software R (2012) is widely used. There was no package, however, for computing the Poisson binomial distribution function. We developed an R package that efficiently implements both exact methods and approximation methods. The package can be downloaded from the R website, see Section 5 for more details.

1.3. Overview

The rest of the paper is organized as follows. Section 2 describes several exact methods for computing the cdf and algorithms for their efficient implementations. Section 3 describes several approximation methods based on the Poisson and normal approximations. Section 4 conducts a comprehensive numerical study to assess the performance of various methods in terms of accuracy and efficiency. Section 5 discusses software implementation for both the exact and approximation methods. Section 6 provides some concluding remarks and suggestions for practitioners.

2. Exact methods

2.1. Notation

Let $I_j,\ j = 1, \ldots, n$ be a series of n independent and non-identically distributed random indicators. In particular,

$$I_j \sim \text{Bernoulli}(p_j), \quad j = 1, \ldots, n, \tag{1}$$

where $p_j = \Pr(I_j = 1)$ is the success probability of indicator I_j and not all p_j's are equal. The Poisson binomial random variable N is defined as the sum of independent and non-identically distributed random indicators (i.e., $N = \sum_{j=1}^{n} I_j$). Note that N can take values in $\{0, 1, \ldots, n\}$.

Let $\xi_k = \Pr(N = k),\ k = 0, 1, \ldots, n$ be the probability mass function (pmf) for the Poisson binomial random variable N. When all p_j's are identical, the distribution of N is a binomial distribution. The cdf of N, denoted by $F_N(k) = \Pr(N \le k),\ k = 0, 1, \ldots, n$, gives the probability of having at most k successes out of a total of n. The cdf $F_N(k)$ can be expressed by (Wang, 1993)

$$F_N(k) = \sum_{m=0}^{k} \xi_m = \sum_{m=0}^{k} \left\{ \sum_{A \in \mathcal{F}_m} \prod_{j \in A} p_j \prod_{j \in A^c} (1 - p_j) \right\}, \tag{2}$$

where $\mathcal{F}_m$ is the set of all subsets of m integers that can be selected from $\{1, 2, 3, \ldots, n\}$ and A^c is the complement of set A (i.e., $A^c = \{1, 2, 3, \ldots, n\} \setminus A$). Samuels (1965) presented a similar formula as in (2). In order to compute $F_N(k)$ in (2), one needs to enumerate all elements in $\mathcal{F}_m$, which is not practical even when n is small (e.g., $n = 30$). For example, when $n = 30$, $\mathcal{F}_{15}$ contains $30!/[15! \times (30 - 15)!] = 155{,}117{,}520$ elements. Thus efficient methods for computing $F_N(k)$ are desirable.

2.2. Discrete Fourier transform

In this section, we briefly introduce the discrete Fourier transform (DFT). For a sequence of $n + 1$ complex numbers $\{y_0, y_1, \ldots, y_n\}$, the DFT transforms $\{y_0, y_1, \ldots, y_n\}$ into a sequence of $n + 1$ complex numbers $\{z_0, z_1, \ldots, z_n\}$ where $z_k = \sum_{l=0}^{n} y_l \exp(-\boldsymbol{i}\omega kl),\ k = 0, 1, \ldots, n$, and $\omega = 2\pi/(n + 1)$. The inverse discrete Fourier transform (IDFT), which recovers $\{y_0, y_1, \ldots, y_n\}$ from $\{z_0, z_1, \ldots, z_n\}$, is given by

$$y_l = \frac{1}{n+1} \sum_{k=0}^{n} z_k \exp(\boldsymbol{i}\omega lk), \quad l = 0, 1, \ldots, n. \tag{3}$$

Applying the DFT to both sides of Eq. (3), one can also recover $\{z_0, z_1, \ldots, z_n\}$ from $\{y_0, y_1, \ldots, y_n\}$. More details on the DFT can be found in Bracewell (2000, Chapter 11).

There are fast Fourier transform (FFT) algorithms to compute the DFT efficiently. The most commonly-used algorithm is the Cooley–Tukey algorithm (Cooley and Tukey, 1965). There are also subroutines available in C or FORTRAN that implement FFT algorithms. See Bracewell (2000, Chapter 11) for details on FFT algorithms.

2.3. The DFT of the characteristic function of the Poisson binomial distribution

In this section, we provide a derivation for a closed-form expression for the cdf of the Poisson binomial distribution. Our approach is based on the characteristic function (CF) for the Poisson binomial distribution (see, for example, Athreya and Lahiri, 2006, Chapter 10 for details on CF). Fernández and Williams (2010) provided the same closed-form expression for the cdf, which was derived by using the polynomial interpolation technique and the DFT. Our approach, however, is much simpler.

The CF of the Poisson binomial random variable $N = \sum_{j=1}^{n} I_j$ is

$$\begin{aligned} \varphi(t) &= \mathbf{E}[\exp(\boldsymbol{i}tN)] = \sum_{k=0}^{n} \xi_k \exp(\boldsymbol{i}tk) = \mathbf{E}\left[\exp\left(\boldsymbol{i}t \sum_{j=1}^{n} I_j\right)\right] \\ &= \prod_{j=1}^{n} \mathbf{E}[\exp(\boldsymbol{i}tI_j)] = \prod_{j=1}^{n} [1 - p_j + p_j \exp(\boldsymbol{i}t)], \end{aligned} \tag{4}$$

where $\boldsymbol{i} = \sqrt{-1}$. Substituting $t = \omega l,\ l = 0, 1, \ldots, n$ into (4) where $\omega = 2\pi/(n + 1)$, one obtains

$$\frac{1}{n+1} \sum_{k=0}^{n} \xi_k \exp(\boldsymbol{i}\omega lk) = \frac{1}{n+1} \prod_{j=1}^{n} [1 - p_j + p_j \exp(\boldsymbol{i}\omega l)] = \frac{1}{n+1} x_l, \quad l = 0, 1, \ldots, n, \tag{5}$$

where $x_l = \prod_{j=1}^{n} [1 - p_j + p_j \exp(\boldsymbol{i}\omega l)]$. Note that the left hand side of Eq. (5) is the IDFT of the sequence $\{\xi_0, \xi_1, \ldots, \xi_n\}$. Applying the DFT to both sides of Eq. (5), one recovers $\{\xi_0, \xi_1, \ldots, \xi_n\}$. In particular,

$$\xi_k = \frac{1}{n+1} \sum_{l=0}^{n} \exp(-\boldsymbol{i}\omega lk) \prod_{j=1}^{n} [1 - p_j + p_j \exp(\boldsymbol{i}\omega l)] = \frac{1}{n+1} \sum_{l=0}^{n} \exp(-\boldsymbol{i}\omega lk) x_l. \tag{6}$$

The expression in Eq. (6) gives the same closed-form expression as in Fernández and Williams (2010). From (6), the cdf of N can be expressed as

$$F_N(k) = \sum_{m=0}^{k} \xi_m = \frac{1}{n+1} \sum_{l=0}^{n} \sum_{m=0}^{k} \exp(-\boldsymbol{i}\omega lm) x_l = \frac{1}{n+1} \sum_{l=0}^{n} \frac{\{1 - \exp[-\boldsymbol{i}\omega l(k+1)]\} x_l}{1 - \exp(-\boldsymbol{i}\omega l)}. \tag{7}$$

The last equality in (7) follows from the fact that $\exp(-\boldsymbol{i}\omega lm),\ m = 0, 1, \ldots, k$ is a geometric sequence. We refer to the closed-form expression in (7) for $F_N(k)$ as the DFT–CF method.

2.4. *Efficient implementation of the DFT–CF method*

In this section, we develop an efficient algorithm for computing the cdf $F_N(k)$ in (7). To compute $\xi_k,\ k = 0, 1, \ldots, n$, one first needs to compute x_l. Let $x_l = a_l + \boldsymbol{i} b_l,\ l = 0, 1, \ldots, n$, where a_l and b_l are the real and imaginary parts of x_l, respectively. From (5), $x_l = \sum_{k=0}^{n} \xi_k \exp(\boldsymbol{i}\omega lk),\ l = 0, 1, \ldots, n$. Note that $x_0 = \sum_{k=0}^{n} \xi_k = 1$. Because all ξ_k's are real numbers and $\exp[\boldsymbol{i}\omega(n+1)k] = 1$, the conjugate of x_l is

$$\begin{aligned}\overline{x_l} &= a_l - \boldsymbol{i} b_l = \sum_{k=0}^{n} \xi_k \exp(-\boldsymbol{i}\omega lk) = \sum_{k=0}^{n} \xi_k \exp[\boldsymbol{i}\omega(n+1-l)k] \\ &= x_{n+1-l} = a_{n+1-l} + \boldsymbol{i} b_{n+1-l}, \qquad l = 1, \ldots, n.\end{aligned}$$

Thus $a_l = a_{n+1-l}$, and $b_l = -b_{n+1-l}$ for $l = 1, \ldots, n$. Let $z_j(l) = 1 - p_j + p_j \cos(\omega l) + \boldsymbol{i} p_j \sin(\omega l)$, $|z_j(l)|$ be the modulus of $z_j(l)$, and $\mathrm{Arg}[z_j(l)]$ be the principal value of the argument of $z_j(l)$. Note that

$$\begin{aligned}x_l &= \exp\left\{\sum_{j=1}^{n} \log\left[z_j(l)\right]\right\} = \exp\left\{\sum_{j=1}^{n} \log\Big(|z_j(l)| \exp\{\boldsymbol{i}\mathrm{Arg}[z_j(l)]\}\Big)\right\} \\ &= \exp\left\{\sum_{j=1}^{n} \log\left[|z_j(l)|\right]\right\} \exp\left(\boldsymbol{i}\sum_{j=1}^{n} \mathrm{Arg}[z_j(l)]\right) \\ &= \exp\left\{\sum_{j=1}^{n} \log\left[|z_j(l)|\right]\right\} \left(\cos\left\{\sum_{j=1}^{n} \mathrm{Arg}[z_j(l)]\right\} + \boldsymbol{i}\sin\left\{\sum_{j=1}^{n} \mathrm{Arg}[z_j(l)]\right\}\right).\end{aligned}$$

Here $|z_j(l)| = \{[1 - p_j + p_j \cos(\omega l)]^2 + [p_j \sin(\omega l)]^2\}^{1/2}$ and $\mathrm{Arg}[z_j(l)] = \mathrm{atan2}[p_j \sin(\omega l), 1 - p_j + p_j \cos(\omega l)]$. The function $\mathrm{atan2}(y, x)$ is defined as

$$\mathrm{atan2}(y,x) = \begin{cases} \arctan\left(\dfrac{y}{x}\right) & x > 0 \\ \pi + \arctan\left(\dfrac{y}{x}\right) & y \ge 0,\ x < 0 \\ -\pi + \arctan\left(\dfrac{y}{x}\right) & y < 0,\ x < 0 \\ \dfrac{\pi}{2} & y > 0,\ x = 0 \\ -\dfrac{\pi}{2} & y < 0,\ x = 0 \\ 0 & y = 0,\ x = 0. \end{cases}$$

Thus explicit expressions for a_l and b_l are

$$a_l = d_l \cos\left\{\sum_{j=1}^{n} \mathrm{Arg}[z_j(l)]\right\} \quad \text{and} \quad b_l = d_l \sin\left\{\sum_{j=1}^{n} \mathrm{Arg}[z_j(l)]\right\}, \tag{8}$$

where $d_l = \exp\left\{\sum_{j=1}^{n} \log\left[\,|z_j(l)|\,\right]\right\},\ l = 1, \ldots, n$. The following algorithm is used to compute the cdf $F_N(k)$ for $k = 0, 1, \ldots, n$.

Algorithm A.

1. Let $x_0 = 1$. For $l = 1, \ldots [n/2]$, compute the real and imaginary parts of x_l by using the formulae in (8). Here $[\,\cdot\,]$ is the ceiling function.
2. For $l = [n/2] + 1, \ldots, n$, compute the real and imaginary parts of x_l by using the formula $a_l = a_{n+1-l}$, and $b_l = -b_{n+1-l}$.
3. Apply the FFT algorithm to the set $\{x_0/(n+1), x_1/(n+1), \ldots, x_n/(n+1)\}$ to obtain $\{\xi_0, \xi_1, \ldots, \xi_n\}$.
4. Compute the cdf by using $F_N(k) = \sum_{m=0}^{k} \xi_m,\ k = 0, 1, \ldots, n$.

The above algorithm returns the values of the entire cdf by doing FFT once. Because there are C or FORTRAN subroutines available to do the FFT, the implementation of Algorithm A is not difficult. The FFT algorithm that is used for the implementation in this paper is due to Singleton (1969), which is an FFT algorithm based on the Cooley–Tukey algorithm. The original subroutine was written in FORTRAN and it was translated to C, which is available in the R library.

2.5. Recursive formulae

Recursive formulae (RF) are available in literature to compute $F_N(k)$. Barlow and Heidtmann (1984) described the following recursive formula. A better description of the algorithm is available in Kuo and Zuo (2003, Chapter 7). Let $N_j = \sum_{m=1}^{j} I_m$ and $\xi_{k,j} = \Pr(N_j = k)$ where the random indicator I_m is defined in (1). Note that $N = N_n$ and $\xi_k = \xi_{k,n}$. The recursive formula is given by

$$\xi_{k,j} = (1 - p_j)\xi_{k,j-1} + p_j \xi_{k-1,j-1}, \quad 0 \le k \le n,\ 0 \le j \le n. \tag{9}$$

The boundary conditions for (9) are $\xi_{-1,j} = \xi_{j+1,j} = 0$, $j = 0, 1, \ldots, n-1$ and $\xi_{0,0} = 1$. We refer to (9) as the RF1 method. The RF1 method can be computer memory demanding when n is large.

Chen et al. (1994) introduced another recursive formula for computing ξ_k. The algorithm requires all $p_j < 1$. In particular, the formula is given by

$$\xi_0 = \prod_{j=1}^{n}(1 - p_j), \quad \text{and} \quad \xi_k = \frac{1}{k}\sum_{l=1}^{k}(-1)^{l-1} t_l \xi_{k-l}, \quad k = 1, \ldots, n, \tag{10}$$

where $t_l = \sum_{j=1}^{n}[p_j/(1-p_j)]^l$. We refer to (10) as the RF2 method. This formula is sometimes not numerically stable. This is caused by round-off error in ξ_0 and the explosion of the term $[p_j/(1-p_j)]^l$ in t_l, especially when p_j is close to 1 and n is large.

3. Approximation methods

In this section, we describe several commonly-used approximation methods for computing the cdf $F_N(k)$. Approximation methods are still widely used because of their computational efficiency, especially when n is large and the cdf $F_N(k)$ needs to be evaluated many times. For example, in the prediction application in Hong et al. (2009), the cdf needs to be evaluated $B = 10,000$ times in the calibration of prediction intervals for the number of field failures. We will need moments or functions of moments of N in the description of approximation methods. The expectation, standard deviation, and skewness of the distribution of N are

$$\mu = \mathbf{E}(N) = \sum_{j=1}^{n} p_j, \qquad \sigma = [\mathrm{Var}(N)]^{1/2} = \left[\sum_{j=1}^{n} p_j(1 - p_j)\right]^{1/2}, \tag{11}$$

$$\gamma = [\mathrm{Var}(N)]^{-3/2}\,\mathbf{E}\,[N - \mu]^3 = \sigma^{-3}\sum_{j=1}^{n} p_j(1 - p_j)(1 - 2p_j),$$

respectively.

3.1. Poisson approximation

In literature, the Poisson distribution has been used to approximate the distribution of N, which is referred to as the Poisson approximation (PA) method. In particular, the pmf of the Poisson binomial distribution ξ_k is approximated by

$$\xi_k \approx \frac{\mu^k \exp(-\mu)}{k!}, \quad k = 0, 1, \ldots, n, \tag{12}$$

where μ is defined in (11). By Le Cam's theorem (Le Cam, 1960), the approximation error for the PA method is $\sum_{k=0}^{n} \left|\xi_k - \mu^k \exp(-\mu)/(k!)\right| < 2\sum_{j=1}^{n} p_j^2$. Thus the PA method only works well when the expected number of successes μ is small. When μ is large, the performance of the PA method is generally poor. See Section 4.2 for a numerical illustration of the performance of the PA method.

3.2. Normal approximation

The normal approximation (NA) method is based on the central limit theorem. In particular, the NA method with continuous correction approximates the cdf of a Poisson binomial distribution by

$$F_N(k) \approx \Phi\left(\frac{k + 0.5 - \mu}{\sigma}\right), \quad k = 0, 1, \ldots, n, \tag{13}$$

where $\Phi(x)$ is the cdf of the standard normal distribution, and μ and σ are defined in (11). When n is small, the performance of the normal approximation can be poor.

Table 1
Accuracy comparisons for the DFT–CF, RF1 and RF2 methods.

n	n_1	n_2	n_3	p_1	p_2	p_3	TAE		
							DFT–CF	RF1	RF2
30	10	10	10	0.500	0.500	0.500	1.6×10^{-14}	7.4×10^{-15}	7.4×10^{-15}
30	10	5	15	0.500	0.500	0.500	1.3×10^{-14}	5.2×10^{-15}	5.2×10^{-15}
30	10	5	15	0.010	0.500	0.990	1.4×10^{-14}	7.0×10^{-16}	na
300	100	50	150	0.010	0.500	0.990	1.9×10^{-12}	4.7×10^{-14}	na
3000	1000	500	1500	0.010	0.500	0.990	3.6×10^{-10}	1.1×10^{-11}	na
3000	1000	500	1500	0.001	0.010	0.020	3.1×10^{-11}	9.4×10^{-11}	1.6×10^{-10}
3000	1000	500	1500	0.999	0.990	0.980	1.4×10^{-09}	1.1×10^{-14}	na
3000	1000	500	1500	0.001	0.500	0.999	3.4×10^{-10}	7.2×10^{-12}	na
3000	1000	500	1500	0.300	0.500	0.700	3.8×10^{-10}	7.7×10^{-11}	na

3.3. *Refined normal approximation*

Volkova (1996) described a refined normal approximation (RNA) which makes a correction to the skewness of the distribution of N. For the RNA method, the cdf $F_N(k)$ is approximated by

$$F_N(k) \approx G\left(\frac{k+0.5-\mu}{\sigma}\right), \quad k = 0, 1, \ldots, n, \tag{14}$$

where $G(x) = \Phi(x) + \gamma(1-x^2)\phi(x)/6$, $\phi(x)$ is the pdf of the standard normal distribution, and γ is defined in (11). In some situations, the values of the cdf approximated by the RNA method can be outside [0, 1]. Thus those values less than 0 are corrected to 0 and those values larger than 1 are corrected to 1.

4. Numerical studies

4.1. *Accuracy of the implementations of exact methods*

The DFT–CF, RF1 and RF2 methods can provide exact values of the cdf. It is, however, desirable to verify that the software implementations of these methods are correct. In this section, we use the distribution of the sum of binomial random variables to verify the implementations of these methods.

Note that the distribution of the sum of three independent and non-identically distributed binomial distributions is a special case of the Poisson binomial distribution. The pmf in this special case is

$$\xi_k = \sum_{j=0}^{k} b_{k-j,n_3}\left(\sum_{i=0}^{j} b_{i,n_1} b_{j-i,n_2}\right) = \sum_{j=0}^{k}\sum_{i=0}^{j} b_{i,n_1} b_{j-i,n_2} b_{k-j,n_3}, \tag{15}$$

where $n_1 + n_2 + n_3 = n$, and b_{i,n_1}, b_{i,n_2}, and b_{i,n_3} are the pmfs of Binomial(n_1, p_1), Binomial(n_2, p_2), and Binomial(n_3, p_3), respectively. Here p_1, p_2, and p_3 are the success probabilities for these three binomial distributions. In particular,

$$b_{i,n_1} = \binom{n_1}{i} p_1^i (1-p_1)^{n_1-i}, \qquad b_{i,n_2} = \binom{n_2}{i} p_2^i (1-p_2)^{n_2-i}, \quad \text{and} \quad b_{i,n_3} = \binom{n_3}{i} p_3^i (1-p_3)^{n_3-i}.$$

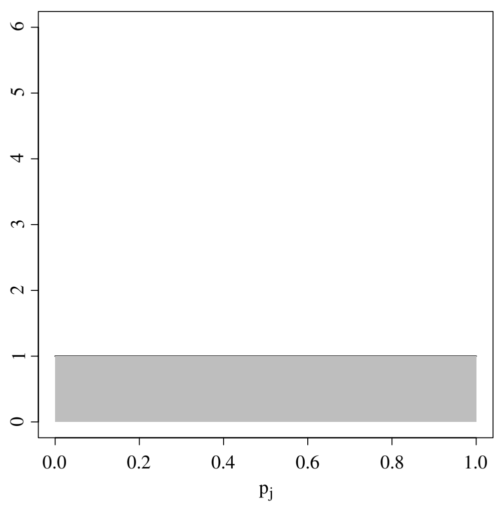
The pmfs b_{i,n_1}, b_{i,n_2} and b_{i,n_3} can be accurately computed by using existing software. With different values of n_1, n_2, n_3 and p_1, p_2, p_3, one can obtain various Poisson binomial distributions. We use the total absolute error (TAE) between two cdfs as a metric for accuracy comparisons. The TAE is defined by

$$\text{TAE} = \sum_{k=0}^{n} |F(k) - F_{\text{bin}}(k)|,$$

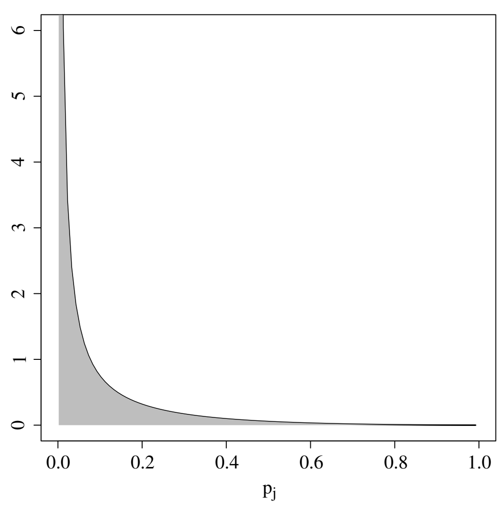
where $F(k)$ is a cdf computed by using one of the exact methods, and $F_{\text{bin}}(k)$ is the cdf computed by using the formula in (15). Table 1 shows the results from the accuracy study for the DFT–CF, RF1 and RF2 methods. Various values of n_1, n_2, n_3 and p_1, p_2, p_3 were chosen to generate different scenarios. The TAE for each scenario was computed. The TAEs are generally less than 1×10^{-10} for the DFT–CF and RF1 methods. Thus the results show that the DFT–CF and RF1 methods can accurately compute the cdf for the Poisson binomial distribution. The RF2 method does not work for most cases because the algorithm is not numerically stable.

4.2. *Accuracy comparisons for approximation methods*

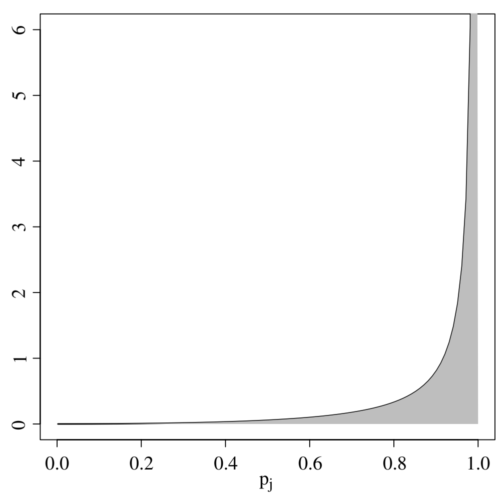
Being able to compute the exact values of the cdf $F_N(k)$ allows us to study the performance of approximation methods. To see the performance of different approximation methods, we simulate success probabilities p_j's from various patterns. Fig. 1 shows the six different patterns in p_j's used in this numerical study. These patterns in the p_j's are generated from



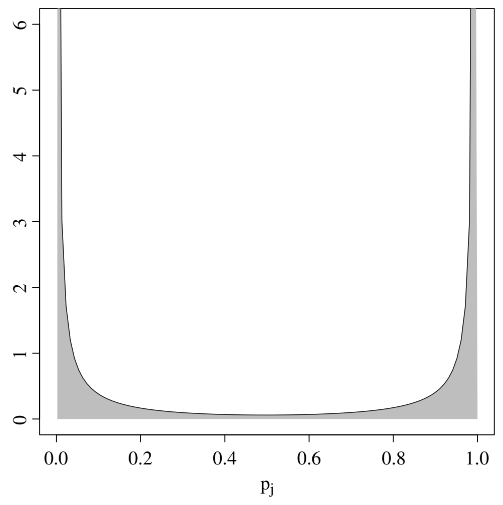
(a) Uniform(0, 1).



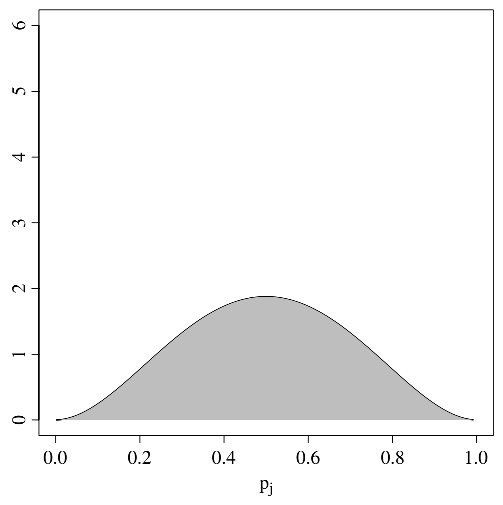
(b) Beta(0.1, 3).



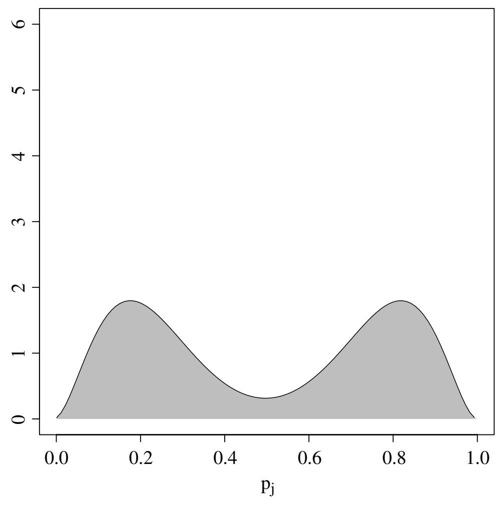
(c) Beta(3, 0.1).



(d) 0.5Beta(3, 0.1) + 0.5Beta(0.1, 3).



(e) Beta(3, 3).



(f) 0.5Beta(3, 10) + 0.5Beta(10, 3).

Fig. 1. Six different patterns in the p_j's used in the numerical study. Here Beta(a, b) is the probability density function of the beta distribution with shape parameters a and b.

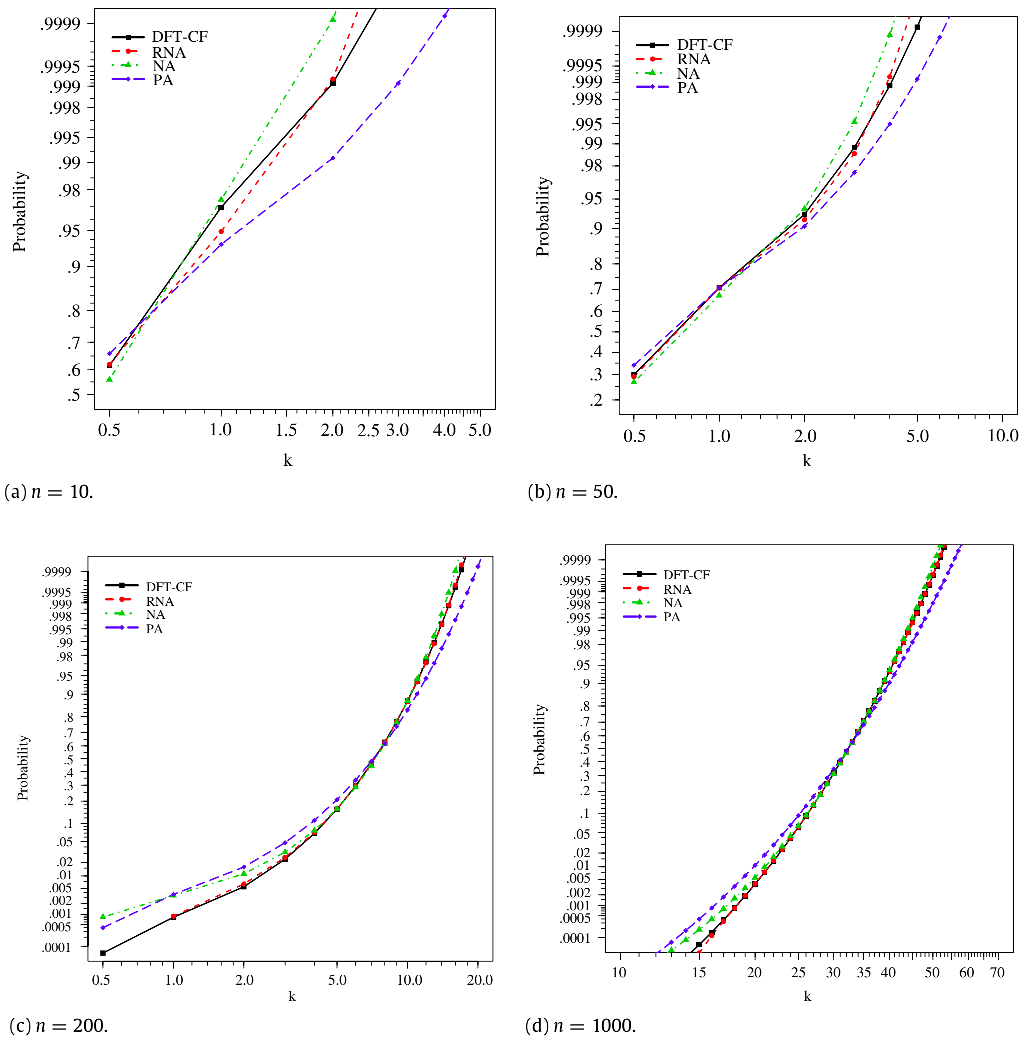


Fig. 2. An illustration of computed cdfs with various methods for Pattern (b) in Fig. 1, when $n = 10, 50, 200$, and 1000. The x-axis is on the logarithm scale (the location where $k = 0.5$ shows the value of the cdf at $k = 0$) and the y-axis is on the scale of the quantile function of the standard normal distribution.

the uniform distribution, beta distribution with various values of shape parameters, and mixtures of beta distributions. For each pattern, various values of n, which is the number of random indicators in N, were chosen to see the effect of n on the accuracy of approximation methods. In particular, the values of n were chosen from $n = 10$, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10,000, and 15,000.

Fig. 2 shows an illustration of computed cdfs by using various methods. Each sub-figure is based on a set of p_j's simulated from Pattern (b) in Fig. 1 when $n = 10, 50, 200$, and 1000, respectively. The x-axis is on the logarithm scale and the y-axis is on the scale of the quantile function of the standard normal distribution (but labeled on the original scales). For the convenience of plotting, the location where $k = 0.5$ shows the value of the cdf at $k = 0$. Note that the Poisson binomial distribution is a discrete distribution. Thus only those points in Fig. 2 show the values of the cdfs. Those segments that connect points are for the convenience of visual comparisons. The RNA method approximates the cdf well and the NA method approximates the cdf moderately well (there are departures in the upper and lower tails of the cdf). The cdf computed by the PA method deviates from the true cdf. Thus the PA method does not perform well. The RF1 method gives exactly the same values (agrees to the ninth decimal places) as the DFT–CF method. The RF2 method does not work because it is not numerically stable. Thus the results for recursive formulae are not shown in Fig. 2.

To make accuracy comparisons for different methods, Table 2 shows the average TAE of 1000 sets of p_j's simulated for each combination of patterns in p_j's and values of n. In particular, the TAE for a set of p_j's is computed by

$$\text{TAE} = \sum_{k=0}^{n} |F(k) - F_N(k)|,$$

Table 2
Average TAE of 1000 sets of p_j's simulated for each combination of patterns in p_j's and values of n for accuracy comparisons of approximation methods.

Pattern	(a)			(b)		
Method	RNA	NA	PA	RNA	NA	PA
$n = 10$	0.0209	0.0281	0.7372	0.0300	0.0466	0.0563
$n = 20$	0.0147	0.0200	1.0728	0.0259	0.0708	0.0897
$n = 50$	0.0092	0.0124	1.6948	0.0216	0.0762	0.1420
$n = 100$	0.0065	0.0086	2.3924	0.0195	0.0873	0.2043
$n = 200$	0.0046	0.0061	3.3763	0.0148	0.0940	0.2912
$n = 500$	0.0029	0.0038	5.3303	0.0092	0.0930	0.4637
$n = 1000$	0.0021	0.0027	7.5429	0.0064	0.0925	0.6521
$n = 2000$	0.0015	0.0019	10.664	0.0045	0.0919	0.9315
$n = 5000$	0.0009	0.0012	16.864	0.0028	0.0919	1.4632
$n = 10{,}000$	0.0007	0.0009	23.844	0.0020	0.0918	2.0727
$n = 15{,}000$	0.0005	0.0007	29.211	0.0016	0.0918	2.5353
Pattern	(c)			(d)		
Method	RNA	NA	PA	RNA	NA	PA
$n = 10$	0.0401	0.0838	1.4915	0.0456	0.0623	1.5046
$n = 20$	0.0459	0.1165	1.9571	0.0574	0.0772	2.0599
$n = 50$	0.0381	0.1086	3.0302	0.0434	0.0535	3.1704
$n = 100$	0.0225	0.0971	4.4510	0.0272	0.0330	4.4456
$n = 200$	0.0149	0.0952	6.7313	0.0185	0.0225	6.2709
$n = 500$	0.0092	0.0932	12.005	0.0114	0.0138	9.8885
$n = 1000$	0.0064	0.0922	18.630	0.0080	0.0095	13.970
$n = 2000$	0.0045	0.0921	28.247	0.0056	0.0067	19.762
$n = 5000$	0.0028	0.0917	46.583	0.0035	0.0043	31.226
$n = 10{,}000$	0.0020	0.0917	66.215	0.0025	0.0030	44.161
$n = 15{,}000$	0.0016	0.0918	81.116	0.0020	0.0024	54.087
Pattern	(e)			(f)		
Method	RNA	NA	PA	RNA	NA	PA
$n = 10$	0.0155	0.0215	0.6144	0.0252	0.0262	0.7726
$n = 20$	0.0109	0.0151	0.8733	0.0169	0.0176	1.0813
$n = 50$	0.0068	0.0095	1.3818	0.0105	0.0109	1.7042
$n = 100$	0.0048	0.0066	1.9534	0.0074	0.0077	2.4101
$n = 200$	0.0034	0.0047	2.7627	0.0052	0.0054	3.4005
$n = 500$	0.0021	0.0030	4.3608	0.0033	0.0034	5.3738
$n = 1000$	0.0015	0.0021	6.1632	0.0023	0.0024	7.6009
$n = 2000$	0.0011	0.0015	8.7117	0.0016	0.0017	10.745
$n = 5000$	0.0007	0.0009	13.771	0.0010	0.0011	16.988
$n = 10{,}000$	0.0005	0.0007	19.485	0.0007	0.0008	24.017
$n = 15{,}000$	0.0004	0.0005	23.869	0.0006	0.0006	29.425

where $F(k)$ is a cdf computed by using one of the approximation methods, and $F_N(k)$ is the cdf computed by using the DFT–CF method. As we can see from the results in Table 2, the PA method does not perform well for most cases. The PA method only works reasonably well when μ is small, for example in Pattern (b) when $n \leq 50$.

For the normal approximation methods, the RNA method performs better than the NA method for almost all cases. For Patterns (b) and (c) where N is highly skewed, the RNA method performs much better than the NA method. When $n \geq 2000$, the TAE for the RNA method is generally less than 0.005. Thus the RNA method is recommended when an approximation method needs to be used.

For all combinations of patterns in the p_j's and values of n considered in Table 2, both the DFT–CF and RF1 methods provide results that agree to the ninth decimal places. The RF2 method, however, does not work in most cases for the same reason mentioned previously. Thus the results for the RF1 and RF2 methods are not shown in Table 2.

4.3. *Efficiency comparisons for exact methods*

The computing time for the exact and approximation methods needs to be considered when n is large. For each combination of patterns in p_j's and values of n as in Section 4.2, 1000 sets of p_j's were simulated. Table 3 gives the average time for computing the entire cdf using the RNA, DFT–CF and RF1 methods based on those 1000 sets of p_j's. The unit of time is the second. The computations were done by using the 64-bit R in a workstation. The workstation has an Intel Xeon CPU (X5570, 2.93 GHz) and 24 GB RAM installed.

The results in Table 3 show that the computing time for the RNA method is generally negligible (less than four milliseconds). The computing time for both the DFT–CF and RF1 methods are generally negligible (less than ten milliseconds) when $n \leq 500$. When $n \geq 1000$, the RF1 method requires more computing time than the DFT–CF method. The RF1 method also requires more RAM. For example, when $n = 15{,}000$, approximately 4 GB memory is needed for computing the entire cdf.

Table 3
Computational efficiency comparisons for the RNA, DFT–CF and RF1 methods, based on 1000 sets of p_j's simulated from each combination of patterns in p_j's and values of n. The unit of time is the second.

Pattern	(a)			(b)		
Method	RNA	DFT–CF	RF1	RNA	DFT–CF	RF1
$n = 10$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 20$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 50$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 100$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 200$	0.000	0.001	0.001	0.000	0.001	0.001
$n = 500$	0.000	0.008	0.005	0.000	0.006	0.006
$n = 1000$	0.000	0.029	0.068	0.000	0.022	0.069
$n = 2000$	0.001	0.111	0.185	0.000	0.084	0.181
$n = 5000$	0.001	0.691	0.825	0.001	0.528	0.814
$n = 10{,}000$	0.002	2.735	3.377	0.002	2.100	3.307
$n = 15{,}000$	0.003	6.176	7.715	0.003	4.736	7.658
Pattern	(c)			(d)		
Method	RNA	DFT–CF	RF1	RNA	DFT–CF	RF1
$n = 10$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 20$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 50$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 100$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 200$	0.000	0.001	0.001	0.000	0.001	0.001
$n = 500$	0.000	0.006	0.005	0.000	0.006	0.005
$n = 1000$	0.000	0.026	0.068	0.000	0.024	0.074
$n = 2000$	0.000	0.098	0.185	0.000	0.094	0.193
$n = 5000$	0.001	0.617	0.809	0.001	0.581	0.827
$n = 10{,}000$	0.002	2.445	3.337	0.002	2.271	3.359
$n = 15{,}000$	0.003	5.517	7.731	0.003	5.141	7.665
Pattern	(e)			(f)		
Method	RNA	DFT–CF	RF1	RNA	DFT–CF	RF1
$n = 10$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 20$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 50$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 100$	0.000	0.000	0.000	0.000	0.000	0.000
$n = 200$	0.000	0.001	0.001	0.000	0.001	0.001
$n = 500$	0.000	0.007	0.005	0.000	0.007	0.004
$n = 1000$	0.000	0.029	0.067	0.000	0.027	0.069
$n = 2000$	0.001	0.109	0.186	0.000	0.104	0.194
$n = 5000$	0.001	0.689	0.859	0.001	0.654	0.912
$n = 10{,}000$	0.002	2.743	3.348	0.002	2.593	3.661
$n = 15{,}000$	0.002	6.181	7.741	0.003	5.847	8.626

The DFT–CF method, however, is less demanding in memory. Thus the DFT–CF method is recommended for computing the exact values for the cdf $F_N(k)$, especially when n is large.

5. Software implementation

The DFT–CF, RF1, RNA and NA methods have been implemented in R. The computationally intensive components such as the FFT are implemented in C and are linked to R. The R functions have been wrapped into an R package `poibin` which can be downloaded from the Comprehensive R Archive Network (http://cran.r-project.org/). The R function in the package for computing the cdf $F_N(k)$ is `ppoibin()`, which has an option that allows users to specify the method for computing.

6. Concluding remarks

In this paper, we focus on the computing of the distribution function for the Poisson binomial distribution. We present a simple derivation for an exact formula with a closed-form expression. We develop an algorithm for efficient implementation of the exact formula and study the advantages and disadvantages of various approximation methods. Numerical studies were conducted to compare the accuracy of the exact and approximation methods. The DFT–CF, RF1, RNA and NA methods have been implemented in an R package.

In practice, the DFT–CF method is generally recommended for computing. The RF1 method can also been used when $n < 1000$, because there is not much difference in computing time from the DFT–CF method. The RNA method is recommended when $n > 2000$ and the cdf needs to be evaluated many times. As shown in the numerical study, the RNA method can approximate the cdf well, when n is large, and is more computationally efficient. The PA method, however, is not recommended because its performance is generally poor. The RF2 method is not recommended either, because the algorithm is not numerically stable.

Acknowledgments

The author would like to thank the editor, an associate editor, and the referees, who provided valuable comments that helped in improving this paper. The author also would like to thank William Q. Meeker and Qingyu Yang for their valuable comments on an earlier version of the paper. The work was supported by funds from the NSF Award CMMI-1068933 and the 2011 DuPont Young Professor Award.

References

Athreya, K.B., Lahiri, S.N., 2006. Measure Theory and Probability Theory. Springer, New York.
Barlow, R.E., Heidtmann, K.D., 1984. Computing k-out-of-n system reliability. IEEE Transactions on Reliability 33, 322–323.
Bossavy, A., Girard, R., Kariniotakis, G., 2012. Forecasting ramps of wind power production with numerical weather prediction ensembles. Wind Energy http://dx.doi.org/10.1002/we.526.
Bracewell, R., 2000. The Fourier Transform & its Applications, third ed. McGraw-Hill, Inc., Singapore.
Chen, X.-H., Dempster, A.P., Liu, J.S., 1994. Weighted finite population sampling to maximize entropy. Biometrika 81, 457–469.
Chen, S.X., Liu, J.S., 1997. Statistical applications of the Poisson-binomial and conditional Bernoulli distributions. Statistica Sinica 7, 875–892.
Cooley, J.W., Tukey, J.W., 1965. An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation 19, 297–301.
Duffie, D., Saita, L., Wang, K., 2007. Multi-period corporate default prediction with stochastic covariates. Journal of Financial Economics 83, 635–665.
Fernández, M., Williams, S., 2010. Closed-form expression for the Poisson-binomial probability density function. IEEE Transactions on Aerospace and Electronic Systems 46, 803–817.
Hong, Y., Meeker, W.Q., 2010. Field-failure and warranty prediction based on auxiliary use-rate information. Technometrics 52, 148–159.
Hong, Y., Meeker, W.Q., McCalley, J.D., 2009. Prediction of remaining life of power transformers based on left truncated and right censored lifetime data. The Annals of Applied Statistics 3, 857–879.
Kuo, W., Zuo, M., 2003. Optimal Reliability Modeling: Principles and Applications. John Wiley & Sons, Inc., Hoboken, NJ.
Le Cam, L., 1960. An approximation theorem for the Poisson binomial distribution. Pacific Journal of Mathematics 10, 1181–1197.
Niida, A., Imoto, S., Shimamura, T., Miyano, S., 2012. Statistical model-based testing to evaluate the recurrence of genomic aberrations. Bioinformatics 28, i115–i120.
Pitacco, E., 2007. Mortality and longevity: a risk management perspective. In: IAA Life Colloquium, Stockholm. Available at: http://www.actuaries.org/LIFE/Events/Stockholm/Pitacco.pdf.
R 2012. The R Project for Statistical Computing. http://www.r-project.org/.
Samuels, S.M., 1965. On the number of successes in independent trials. The Annals of Mathematical Statistics 36, 1272–1278.
Singleton, R.C., 1969. An algorithm for computing the mixed radix fast Fourier transform. IEEE Transactions on Audio and Electroacoustics AU-17, 93–103.
Tang, P., Peterson, E.A., 2011. Mining probabilistic frequent closed itemsets in uncertain databases. In: Proceedings of the 49th ACM Southeast Conference, ACMSE, Kennesaw, GA, pp. 86–91.
Volkova, A.Y., 1996. A refinement of the central limit theorem for sums of independent random indicators. Theory of Probability and its Applications 40, 791–794.
Wang, Y.H., 1993. On the number of successes in independent trials. Statistica Sinica 3, 295–312.