Computational Statistics and Data Analysis xx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

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

A simple and fast method for computing the Poisson binomial distribution function

William Biscarri [a,*], Sihai Dave Zhao [a], Robert Brunner [a,b,c,d]

[a] Department of Statistics, University of Illinois at Urbana-Champaign, IL, United States
[b] Department of Accountancy, University of Illinois at Urbana-Champaign, IL, United States
[c] Department of Information Science, University of Illinois at Urbana-Champaign, IL, United States
[d] National Center for Supercomputing Applications, Urbana, IL, United States

ARTICLE INFO

Article history:
Received 26 April 2017
Received in revised form 9 January 2018
Accepted 12 January 2018
Available online xxxx

Keywords:
Poisson binomial
Convolution
Fourier transform
Independent Bernoulli sum

ABSTRACT

It is shown that the Poisson binomial distribution function can be efficiently calculated using simple convolution based methods. The Poisson binomial distribution describes how the sum of independent but not identically distributed Bernoulli random variables is distributed. Due to the intractability of the Poisson binomial distribution function, efficient methods for computing it have been of particular interest in past Statistical literature. First, it is demonstrated that simply and directly using the definition of the distribution function of a sum of random variables can calculate the Poisson binomial distribution function efficiently. A modified, tree structured Fourier transform convolution scheme is then presented, which provides even greater gains in efficiency. Both approaches are shown to outperform the current state of the art methods in terms of accuracy and speed. The methods are then evaluated on a real data image processing example in order to demonstrate the efficiency advantages of the proposed methods in practical cases. Finally, possible extensions for using convolution based methods to calculate other distribution functions are discussed.

1. Introduction

Suppose that $\{X_i\}_{i=1}^n$ is a collection of independent, but not necessarily identically distributed *Bernoulli*(p_i) random variables, and we are interested in calculating the distribution function of their sum, $Y = \sum X_i$. Known as the Poisson binomial distribution function, this quantity is of particular interest to a variety of fields and applications, and has seen wide use in recent years. For example, it has been used in genetics (Melton et al., 2015), device failure time analysis (Hong and Meeker, 2013), reliability analysis in meteorological forecasting (DeChant and Moradkhani, 2015), insurance (Pitacco, 2007), item response theory (González et al., 2016), ecology (Calabrese et al., 2014), survey sampling (Chen and Liu, 1997), and even analysis of golf (Elmore and Urbaczewski, 2016).

The distribution itself has received significant attention in past statistical literature, for example, Hoeffding (1956), where the focus has primarily been on calculating the cumulative distribution function

$$P(Y \leq y) = \sum_{s=0}^{y} \sum_{A \in F_s} \prod_{i \in A} p_i \prod_{i \in A^c} (1 - p_i), \tag{1}$$

* Correspondence to: 725 S Wright St #101, Champaign, IL 61820, United States.
E-mail address: wbiscar2@illinois.edu (W. Biscarri).

https://doi.org/10.1016/j.csda.2018.01.007
0167-9473/

Please cite this article in press as: Biscarri W., et al., A simple and fast method for computing the Poisson binomial distribution function. Computational Statistics and Data Analysis (2018), https://doi.org/10.1016/j.csda.2018.01.007.

where F_s is defined as the set of all subsets of size s that can be chosen from the set $\{1, 2, \ldots, n\}$. It is clear that attempting to compute this quantity directly is intractable for even small sample sizes as each F_s contains $\binom{n}{s}$ elements, and this issue has motivated many interesting works.

A variety of approximation methods have been proposed, including Poisson approximations (Le Cam et al., 1960; Chen, 1974; Deheuvels et al., 1986; Wang, 1993; Steele, 1994) and generally more accurate binomial approximations (Choi et al., 2002; Ehm, 1991; Soon, 1996; Roos, 2001). Normal approximations have also been used (Berry, 1941; Mikhailov, 1994), of which perhaps the most effective is the refined normal approximation of Volkova (1996).

Recursive algorithms for exact calculation have also been developed (Wadycki et al., 1973; Barlow and Heidtmann, 1984; Radke and Evanoff, 1994; Chen et al., 1994; Belfore, 1995). While recursive algorithms are capable of computing the exact distribution function, they can have expensive memory costs, and some can experience issues of numerical stability. A more in depth treatment of some recursive algorithms is provided in Hong (2013).

Recently, two other exact methods have been proposed by Fernández and Williams (2010) and Hong (2013). In Fernández and Williams (2010) a closed form expression for the Poisson binomial distribution function is derived using polynomial interpolation and discrete Fourier transforms. Hong (2013) provides a simpler derivation using characteristic functions, and also develops an efficient algorithm to evaluate the distribution function, which is called DFT-CF. A comparison of DFT-CF and past methods is also provided, which concludes that DFT-CF is recommended for general computation of the Poisson binomial distribution function. In this sense, the DFT-CF algorithm is the current state of the art approach for efficient and exact computation.

Curiously, we have observed that directly calculating the Poisson binomial distribution function following only the convolution definition of the distribution function of the sum of random variables is more efficient than the DFT-CF algorithm. In this paper we demonstrate this fact and provide intuition explaining why direct computation in this way could be efficient. This result motivates an even faster, binary tree structured, fast Fourier transform (FFT) algorithm based on the convolution definition, which can be seen as a specialized version of the algorithm in Ruckdeschel and Kohl (2014). The algorithms are then compared in a real data image processing example, which further demonstrates the effectiveness of the proposed methods.

2. Calculating the Poisson binomial distribution function

2.1. Direct convolution

Suppose that we have two random variables, X_1 and X_2, and denote their sum by Y. The density function of Y is calculated by the convolution of the individual densities of X_1 and X_2 using the well known formula

$$P(X_1 + X_2 = k) = (f_1 * f_2)(k) = \sum_{i=0}^{k} f_1(i) \cdot f_2(k - i). \tag{2}$$

Of course, this can easily be extended to any collection of random variables, $\{X_i\}_{i=1}^{n}$, so that the density function of their sum is $f_1 * \ldots * f_n$. For two *Bernoulli*(p_i) random variables, calculating (2) is equivalent to calculating the linear convolution of two sequences, P_1 and P_2, where each P_i is a vector so that $P_i = [1 - p_i, p_i]$. It is easy to see that this can similarly be extended to any number of *Bernoulli*(p_i) random variables so that the density function of their sum is given by $P_1 * \ldots * P_n$. Thus, the density function of Y can be found by first calculating $P_1 * P_2$, and then sequentially convolving the resulting sequence with each remaining P_i until P_n has been reached. Finally, the distribution function of Y can be quickly calculated by taking a cumulative summation over P_n. Algorithm 1 describes the process in more detail, and $[j]$ is used to denote the *jth* element of a vector.

To gain some intuition as to why this naive method of computation could be efficient, consider first calculating the density function when $n = 2$, which would require computing

$$P_1 * P_2 = [(1 - p_1)(1 - p_2), (1 - p_1)p_2 + (1 - p_2)p_1, \ p_1 p_2]. \tag{3}$$

For larger n, we would then convolve P_3 with (3) and continue until all P_i have been incorporated. The *jth* step of the process involves convolving a length 2 sequence with a sequence of length $j + 1$. This, coupled with the fact that $1 - p_i$ must be calculated for each i, makes it clear that $\frac{3}{2}(n^2 + n - 1)$ arithmetic operations will be required and the procedure will have a time complexity of $O(n^2)$. The key, however, is that all of the required operations will be very fast as only addition and multiplication are needed. Finally, we note that due to its simplicity, directly calculating the convolution is easily implemented in any programming language. We will refer to this method of computation as DC.

2.2. Divide and conquer FFT tree convolution

One obvious way to try to improve on the direct convolution approach is to use the well known convolution theorem which states that

$$\mathcal{F}(f_1 * f_2) = \mathcal{F}(f_1) \cdot \mathcal{F}(f_2) \qquad \text{and thus} \qquad f_1 * f_2 = \mathcal{F}^{-1}(\mathcal{F}(f_1) \cdot \mathcal{F}(f_2)), \tag{4}$$

Please cite this article in press as: Biscarri W., et al., A simple and fast method for computing the Poisson binomial distribution function. Computational Statistics and Data Analysis (2018), https://doi.org/10.1016/j.csda.2018.01.007.