Likelihood Inference for Type I Bivariate Pólya-Aeppli Distribution

LIKELIHOOD INFERENCE FOR TYPE I BIVARIATE PÓLYA-AEPPLI DISTRIBUTION

BY

YANG YE, B.Sc.

A THESIS
SUBMITTED TO THE DEPARTMENT OF MATHEMATICS AND STATISTICS
AND THE SCHOOL OF GRADUATE STUDIES
OF MCMASTER UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Master of Science (2015)	McMaster University
(Mathematics and Statistics)	Hamilton, Ontario, Canada

TITLE:	Likelihood Inference for Type I Bivariate Pólya-Aeppli Distribution
AUTHOR:	Yang Ye B.Sc. (Mathematics) McGill University, Montréal, Canada
SUPERVISOR:	Prof. Narayanaswamy Balakrishnan
NUMBER OF PAGES:	xi, 90

Abstract

The Poisson distribution is commonly used in analyzing count data, and many insurance companies are interested in studying the related risk models and ruin probability theory. Over the past century, many different bivariate models have been developed in the literature. The bivariate Poisson distribution was first introduced by Campbell (1934) for modelling bivariate accident data. However, in some situations, a given dataset may possess over-dispersion compared to Poisson distribution which motivated researchers to develop alternative models to handle such situations. In this regard, Minkova and Balakrishnan (2014a) developed the Type I bivariate Pólya-Aeppli distribution by using compounding with Geometric random variables and the trivariate reduction method. Inference for this Type I bivariate Pólya-Aeppli distribution is the topic of this thesis.

The parameters in a model are used to describe and summarize a given sample within a specific distribution. So, their estimation becomes important and the goal of estimation theory is to seek a method to find estimators for the parameters of interest that have some good properties. There exist many methods of finding estimators such as Method of Moments, Bayesian estimators, Least Squares, and Maximum Likelihood Estimators (MLEs). Each method of estimation has its own strength and

weakness (Casella and Berger (2008)). Minkova and Balakrishnan (2014a) discussed the moment estimation of the parameters of the Type I bivariate Pólya-Aeppli distribution. In this thesis, we develop the likelihood inference for this model.

A simulation study is carried out with various parameter settings. The obtained results show that the MLEs require more computational time compared to Moment estimation. However, Method of Moments (MoM) did not result in good estimates for all the simulation settings. In terms of mean squared error and bias, we observed that MLEs performed, in most of the settings, better than MoM.

Finally, we apply the Type I bivariate Pólya-Aeppli model to a real dataset containing the frequencies of railway accidents in two subsequent six year periods. We also carry out some hypothesis tests using the Wald test statistic. From these results, we conclude that the two variables belong to the same univariate Pólya-Aeppli distribution but are correlated.

Acknowledgements

I would like to express my deepest gratitude to Prof. Balakrishnan for his help, guidance, and constructive advice during this thesis work. He always put himself at the position of student, and the time and energy he spent on students are invaluable.

My warm thanks go to my friends Feng Tian, Liu Kai, So HonYiu, Zhu XiaoJun, and Cao Yu for their generous support and to all those who provided me help and support to complete this thesis.

My final acknowledgement is to my parents, for their love, understanding and moral over the years. They gave me the chance and confidence to explore various possibilities in my academic and personal life.

A special thanks to all kind and sympathetic people in this Department and City, the friendly bus drivers, and the supermarket in the next street. They all helped in giving me strength.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Problem

The Poisson distribution is often used as a risk insurance model. One of the most important characteristics of a Poisson distribution is that it is equi-dispersed, that is, mean and variance are equal. When this equality is violated, we will have either an underdispersed or overdispersed data. An overdispersion in the count data is a problem. This similar issue was brought by Fisher *et al.* (1922) during the interpretation of a bacterial count data, and thus developed the Fisher Index of Dispersion test. Given n independent random variables $X_1, X_2, \ldots, X_n \sim Poi(\lambda)$, according to Fisher *et al.* (1922),

$$\begin{aligned} I &= \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{\bar{X}} \qquad (1.1) \\ &= (n-1)\frac{S_X^2}{\bar{X}}, \end{aligned}$$

where $I \sim \chi^2$ with $(n-1)$ degrees of freedom. This Index of Dispersion aims to test the null hypothesis that all the random variables X_i's come from the Poisson distribution against the alternative hypothesis that the variance is greater than the mean.

Over the last century, researchers have developed many different bivariate distributions. The bivariate Poisson distribution is the least complex model and is therefore widely used in many applications. Loukas and Kemp (1986) extended the univariate Index of Dispersion proposed by Fisher *et al.* (1922) to test the bivariate Poisson distribution. This test, called the Bivariate Dispersion Test, has the form

$$I_B = \frac{N(\bar{X}_2 S_1^2 - 2S_{12}^2 + \bar{X}_1 S_2^2)}{\bar{X}_1 \bar{X}_2 - S_{12}^2}, \tag{1.2}$$

where N is the sample size, $\bar{X}_1, \bar{X}_2$ are the sample means, S_1^2, S_2^2 are the sample variances and S_{12} is the sample covariance. The test statistic I_B follows by a χ^2 distribution with $2N$ degrees of freedom.

An evidence towards an alternative distribution means that there is more variability present in the data than what would be expected with a bivariate Poisson distribution. For this reason, many researchers sought alternative models. Lindén and Mäntyniemi (2011) proposed a parametrization of Negative Binomial distribution to solve the over-dispersion problem for ecological count data. Later, in order to describe the heterogeneous insurance type data, Minkova (2004) showed a generalization of the count model by adding a new parameter to number of counts into the Negative Binomial distribution. This resulted in the well known Pólya-Aeppli distribution.

The Pólya-Aeppli distribution is a model that counts the objects that occur in clusters. The number of objects per cluster come from the geometric distribution with probability of success $1-\rho$, where $0 \leq \rho \leq 1$, and the number of clusters follows the Poisson distribution with mean λ (Johnson *et al.* (2005)). This distribution was first studied by A. Aeppli in a thesis, followed by G. Pólya five years later. Anscombe (1950) subsequently finalized the derivation of this distribution and gave the name Pólya-Aeppli distribution (Minkova (2012)). The probability mass function (PMF) of a Pólya-Aeppli random variable N is

$$P(N=n) \quad = \quad e^{-\lambda} \sum_{i=0}^{n} \frac{1}{i!} \binom{n-1}{i-1} (\lambda(1-\rho))^{i} \rho^{n-i}. \tag{1.3}$$

This expression will be formally derived in Chapter 2.

This distribution is widely applied in risk theory. It has many applications including in estimating ruin probability (Minkova (2004)), in queueing theory of income process (Dragieva (2011)), and in the study of times in dynamical systems (Haydn and Valenti (2009)). For example, Haydn and Valenti (2009) showed that the limiting distribution of the behaviour of return times at periodic points in a mixing dynamical system is a compound Poisson distribution. All these works, however, focused on the univariate Pólya-Aeppli distribution. Minkova and Balakrishnan (2014a) were the first to introduce a bivariate form of Pólya-Aeppli distribution.

To extend the univariate distribution to a bivariate form, Minkova and Balakrishnan (2014a) used the trivariate reduction method (Balakrishnan and Lai (2009)). This begins with the bivariate Poisson distribution and then compounds it with geometric distribution to obtain a bivariate Pólya-Aeppli distribution along the same lines of a univariate Pólya-Aeppli distribution. More specifically, let us define Y_1 and Y_2 as

$$Y_1 = Z_1 + Z_3,$$
$$Y_2 = Z_2 + Z_3,$$

where Z_i are independent random variables and $Z_i \sim Poi(\lambda_i)$. Since Y_1 and Y_2 are both sums of two independent Poisson random variables, it is evident that

$$Y_1 \sim Poi(\lambda_1 + \lambda_3),$$
$$Y_2 \sim Poi(\lambda_2 + \lambda_3).$$

Campbell (1934) found the joint PGF of (Y_1, Y_2) as

$$\begin{aligned}\psi_{Y_1,Y_2}(s_1, s_2) &= E(s_1^{Y_1} s_2^{Y_2}) \\ &= e^{-\lambda_1(1-s_1)-\lambda_2(1-s_2)-\lambda_3(1-s_1 s_2)}, \qquad (1.4)\end{aligned}$$

and referred to it as the bivariate Poisson distribution. This method of constructing bivariate distributions is commonly referred to as trivariate reduction method; see, for example, Johnson *et al.* (1997) and Balakrishnan and Lai (2009).

By using such a trivariate reduction method, the PGF of the bivariate random vector (N_1, N_2), to be introduced as bivariate compounding later in Section 2.2, can be derived as

$$\psi(s_1, s_2) = e^{-(\lambda_1+\lambda_2+\lambda_3)} e^{\lambda_1\psi_1(s_1)+\lambda_2\psi_1(s_2)+\lambda_3\psi_1(s_1)\psi_1(s_2)}, \tag{1.5}$$

where $\psi_1(s) = \frac{(1-\rho)s}{1-\rho s}$ is the PGF of a Geometric random variable. Minkova and Balakrishnan (2014a) gave the name Type I bivariate Pólya-Aeppli distribution for the above distribution. Subsequently, Minkova and Balakrishnan (2014b) introduced a Type II bivariate Pólya-Aeppli distribution with joint PGF

$$\psi(s_1, s_2) = e^{-\lambda(1-\psi_1(s_1,s_2))}, \tag{1.6}$$

derived by compounding a bivariate geometric distribution with an univariate Poisson distribution, where $\psi_1(s_1, s_2) = \frac{\theta}{1-\theta_1 s_1-\theta_2 s_2}$ with $\theta = 1 - \theta_1 - \theta_2$.

For the Type I bivariate Pólya-Aeppli distribution, Minkova and Balakrishnan (2014a) discussed the Method of Moments (MoM) estimation for the parameters λ_1, λ_2, λ_3 and ρ. Here, we develop the Maximum Likelihood Estimation (MLE) method and evaluate its performance through Monte Carlo simulations. We then compare this method of estimation with MoM in terms of bias and mean squared error. Interval estimation of parameters and tests of hypotheses based on these methods are also discussed.

1.2 Scope of the Thesis

This thesis focuses on the numerical determination of the method of MLEs and its relative performance compared to the Method of Moment (MoM) estimates. For this purpose, an extensive Monte Carlo simulation study is carried out, and some conclusions are drawn from this study.

In Chapter 2, the basic theory about the univariate and bivariate Pólya-Aeppli distributions is provided along with the corresponding derivations. Chapter 3 presents the point estimation and interval estimation methods in detail as well as the required derivations of the recursive probability mass functions. A short discussion about the Grid Search method is also provided for the purpose of finding the initial values for the Newton-Raphson algorithm in the case when MoM fails. In Chapter 4, a simulation study is conducted using various parameter settings. Comparisons between the performance of MoM and MLE is made in terms of different performance measures. Chapter 5 considers a real data application and some hypotheses tests for the model parameters. Finally, Chapter 6 provides a discussion and concludes with some possible directions for future work.

Chapter 2

Basic Model and Probability Calculations

2.1 Univariate Pólya-Aeppli Distribution

In this section, a basic formulation of univariate Pólya-Aeppli distribution is given. Suppose there are Y independent random variables of the form X, and N denotes the sum of these random variables, namely,

$$N = X_1 + X_2 + ... + X_Y. \tag{2.1}$$

Then, the Pólya-Aeppli model is derived by supposing that

(i) X denotes the number of objects within a cluster, where $X \sim Geo(1-\rho)$,

(ii) Y denotes the number of clusters, where $Y \sim Poi(\lambda)$.

This random variable, N, formed by compounding in this fashion gives rise to the univariate Pólya-Aeppli distribution, and its probability generating function (PGF) can be derived easily. First, we have the probability mass function (PMF) of X as

$$P(X = i) = (1 - \rho)\rho^{i-1}, \tag{2.2}$$

for $i = 1, 2, \ldots$, and its PGF is

$$\begin{aligned} E(s^X) = \psi_1(s) &= \sum_{x=1}^{\infty} s^x (1 - \rho)\rho^{x-1} \\ &= (1 - \rho)s \sum_{x=0}^{\infty} (s\rho)^x \\ &= \frac{(1 - \rho)s}{1 - s\rho}. \end{aligned} \tag{2.3}$$

Also, the PGF of Y is known to be

$$\psi_Y(s) = e^{\lambda(s-1)}. \tag{2.4}$$

Since X_i's are *iid* and independent of Y, the PGF of the random variable N can then be readily found as follows:

$$\begin{aligned}
\psi_N(s) = E(s^N) &= E(s^{X_1+\cdots+X_y}) \\
&= \sum_{y=0}^{\infty} E(s^{X_1+\cdots+X_y}|Y=y)P(Y=y) \\
&= \sum_{y=0}^{\infty} [E(s^X)]^y P(Y=y) \\
&= \sum_{y=0}^{\infty} [\psi_1(s)]^y P(Y=y) \\
&= \psi_Y(\psi_1(s)) \\
&= e^{\lambda(\psi_1(s)-1)} \\
&= e^{-\lambda\left(1-\frac{(1-\rho)s}{1-\rho s}\right)}. \qquad (2.5)
\end{aligned}$$

Since the PGF is

$$\psi_N(s) = \sum_{m=0}^{\infty} P(N=m)s^m,$$

an expression for $P(N=m)$, for all $m \geq 0$, can be obtained from the expression of the PGF by using the inverse binomial expansion:

$$(1-p)^{-r} = \sum_{k=0}^{\infty} \binom{k+r-1}{k} p^k. \qquad (2.6)$$

Now, since the PGF of N in (2.5) can be expressed as

$$\begin{aligned}
e^{-\lambda\left(1-\frac{(1-\rho)s}{1-\rho s}\right)} &= e^{-\lambda}e^{\lambda\left(\frac{(1-\rho)s}{1-\rho s}\right)} \\
&= e^{-\lambda}\sum_{m=0}^{\infty}\left[\frac{\lambda(1-\rho)s}{1-\rho s}\right]^m\frac{1}{m!} \\
&= e^{-\lambda}\sum_{m=0}^{\infty}(\lambda(1-\rho)s)^m(1-\rho s)^{-m}\frac{1}{m!} \\
&= e^{-\lambda}\sum_{m=0}^{\infty}\frac{1}{m!}(\lambda(1-\rho)s)^m\sum_{i=0}^{\infty}\binom{m+i-1}{i}(\rho s)^i,
\end{aligned}$$

upon collecting the coefficient of s^m in the above series, we find an explicit expression for the PMF of N as

$$P(N=m) = e^{-\lambda}\sum_{i=0}^{m}\frac{1}{i!}\binom{m-1}{i-1}(\lambda(1-\rho))^i\rho^{m-i}. \tag{2.7}$$

This is the PMF of the univariate Pólya-Aeppli distribution, and we shall denote it by $N \sim PA(\lambda,\rho)$.

The mean and variance can be calculated from the first and second derivatives of the PGF in (2.5) and then setting $s=1$. Thus, we find

$$\begin{aligned}
E(N) &= \frac{\lambda}{1-\rho}, \\
Var(N) &= \frac{\lambda(1+\rho)}{(1-\rho)^2}.
\end{aligned}$$

From these, we find the ratio between variance and mean to be

$$\frac{Var(N)}{E(N)} = \frac{1+\rho}{1-\rho}.$$

Since the numerator is always greater than the denominator, it is evident that the variance is greater than the mean. Thus, this model becomes overdispersed with respect to the Poisson distribution.

Moreover, if there are two independent Pólya-Aeppli random variables N_a and N_b, $N_a \sim PA(\lambda_a, \rho)$ and $N_b \sim PA(\lambda_b, \rho)$ with the same probability of success ρ, then the sum of them also has the Pólya-Aeppli distribution with parameter $\lambda_a + \lambda_b$ and ρ, that is,

$$N_a + N_b \sim PA(\lambda_a + \lambda_b, \rho). \tag{2.8}$$

This is an important property of the Pólya-Aeppli distribution that will be useful in the construction of the bivariate form in the next section, and this property can be easily shown from the PGF in (2.5).

2.2 Bivariate Pólya-Aeppli Distribution

We now describe the construction of Minkova and Balakrishnan (2014a) of the bivariate Pólya-Aeppli distribution through the trivariate reduction method. Consider four sets of convolutions of independent and identically distributed (iid) $Geo(1-\rho)$

random variables U, V, W, and R defined as follows:

$$\begin{aligned}
U &= U_1 + \cdots + U_{Z_1}, \\
V &= V_1 + \cdots + V_{Z_2}, \\
W &= W_1 + \cdots + W_{Z_3}, \\
R &= R_1 + \cdots + R_{Z_3}, \qquad (2.9)
\end{aligned}$$

where Z_i's are assumed to be independent of the compounding $Geo(1-\rho)$ random variables with $Z_i \sim Poi(\lambda_i)$.

Then, the joint PGF of (W, R) is derived as follows:

$$\begin{aligned}
E(s_1^W s_2^R) &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} P(W=i, R=j) s_1^i s_2^j \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} P(W_1+\cdots+W_{Z_3}=i, R_1+\cdots+R_{Z_3}=j) s_1^i s_2^j \\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\sum_{n=0}^{\infty} P(W_1+\cdots+W_n=i, R_1+\cdots+R_n=j) \frac{\lambda_3^n e^{-\lambda_3}}{n!} s_1^i s_2^j \\
&= \sum_{n=0}^{\infty} \frac{\lambda_3^n e^{-\lambda_3}}{n!} [\psi_1(s_1)\psi_1(s_2)]^n \\
&= e^{-\lambda_3} \sum_{n=0}^{\infty} \frac{[\lambda_3 \psi_1(s_1)\psi_1(s_2)]^n}{n!} \\
&= e^{-\lambda_3(1-\psi_1(s_1)\psi_1(s_2))}, \qquad (2.10)
\end{aligned}$$

where $\psi_1(s)$ is the PGF of the compounding geometric distribution in (2.3). Now, let us define a pair of independent random variables N_1 and N_2 as follows:

$$\begin{aligned} N_1 &= U + W, \\ N_2 &= V + R. \end{aligned} \tag{2.11}$$

Then, the joint PGF of the bivariate random variable (N_1, N_2) is given by

$$\begin{aligned} \psi(s_1, s_2) &= E(s_1^{N_1} s_2^{N_2}) = E(s_1^U s_2^V s_1^W s_2^R) \\ &= E(s_1^U)E(s_2^V)E(s_1^W s_2^R) \\ &= e^{-\lambda_1(1-\psi_1(s_1))} e^{-\lambda_2(1-\psi_1(s_2))} e^{-\lambda_3(1-\psi_1(s_1)\psi_1(s_2))} \\ &= e^{-(\lambda_1+\lambda_2+\lambda_3)} e^{\lambda_1\psi_1(s_1)+\lambda_2\psi_1(s_2)+\lambda_3\psi_1(s_1)\psi_1(s_2)}. \end{aligned} \tag{2.12}$$

This is the PGF of the Type I bivariate Pólya-Aeppli distribution, and in this case we will use the notation $(N_1, N_2) \sim BivPA(\lambda_1, \lambda_2, \lambda_3, \rho)$.

2.3 Joint Probability Mass Function of the Bivariate Pólya-Aeppli Distribution

The following proposition presents recursion formulas for the joint probability mass function of the bivariate random variable (N_1, N_2) given by

$$f(i, j) = P(N_1 = i, N_2 = j), \ i, j = 0, 1, 2, \ldots.$$

Proposition 2.1: *The PMF of the bivariate Pólya-Aeppli distribution satisfies the following recursions:*

$$f(i,0) = \left(2\rho + \frac{(1-\rho)\lambda_1 - 2\rho}{i}\right) f(i-1,0) - \left(1 - \frac{2}{i}\right)\rho^2 f(i-2,0),\ i = 1,2,\dots, \tag{2.13}$$

$$f(0,j) = \left(2\rho + \frac{(1-\rho)\lambda_2 - 2\rho}{j}\right) f(0,j-1) - \left(1 - \frac{2}{j}\right)\rho^2 f(0,j-2),\ j = 1,2,\dots, \tag{2.14}$$

where $f(-1,0) = 0,\ f(0,-1) = 0,$

$$\begin{aligned} f(i+1,j) - \rho f(i+1,j-1) \quad = \quad & \left(2\rho + \frac{(1-\rho)\lambda_1 - 2\rho}{i+1}\right) f(i,j) \qquad (2.15) \\ & -\rho^2 \left(1 - \frac{2}{i+1}\right)(f(i-1,j) - \rho f(i-1,j-1)) \\ & -\left(2\rho^2 - \frac{(1-\rho)(\lambda_3 - \rho(\lambda_1+\lambda_3)) + 2\rho^2}{i+1}\right) f(i,j-1) \end{aligned}$$

for $i = 2,3,\dots, j = 1,2,\dots,$

$$\begin{aligned} f(i,j+1) - \rho f(i-1,j+1) \quad = \quad & \left(2\rho + \frac{(1-\rho)\lambda_2 - 2\rho}{j+1}\right) f(i,j) \qquad (2.16) \\ & -\rho^2 (1 - \frac{2}{j+1})(f(i,j-1) - \rho f(i-1,j-1)) \\ & -\left(2\rho^2 - \frac{(1-\rho)(\lambda_3 - \rho(\lambda_2+\lambda_3)) + 2\rho^2}{j+1}\right) f(i-1,j) \end{aligned}$$

for $i = 1,2,\dots, j = 2,3,\dots$.

Notice that $f(0,0) = e^{-(\lambda_1+\lambda_2+\lambda_3)}$. This follows readily from the construction that $N_1 = N_2 = 0$ if and only if $Z_1 = Z_2 = Z_3 = 0$ with $P(Z_i = 0) = e^{-\lambda_i}$, for $i = 1,2,3$ (Minkova and Balakrishnan (2014a)).

To establish *Proposition* 2.1, we first take the derivative of Eq. (2.12) with respect to s_1 and s_2 and obtain the recursive equations

$$(1-\rho s_1)^2(1-\rho s_2)\frac{\partial\psi(s_1,s_2)}{\partial s_1} = (1-\rho)(\lambda_1+(\lambda_3-\rho(\lambda_1+\lambda_3))s_2)\psi(s_1,s_2), \tag{2.17}$$

$$(1-\rho s_2)^2(1-\rho s_1)\frac{\partial\psi(s_1,s_2)}{\partial s_2} = (1-\rho)(\lambda_2+(\lambda_3-\rho(\lambda_2+\lambda_3))s_1)\psi(s_1,s_2), \tag{2.18}$$

where

$$\psi(s_1,s_2) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} s_1^i s_2^j f(i,j), \tag{2.19}$$

$$\frac{\partial\psi(s_1,s_2)}{\partial s_1} = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} i f(i,j) s_1^{i-1} s_2^j, \tag{2.20}$$

$$\frac{\partial\psi(s_1,s_2)}{\partial s_2} = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} j f(i,j) s_1^i s_2^{j-1}. \tag{2.21}$$

To obtain Eqs. (2.13) and (2.15), we expand Eq. (2.17) upon substituting the expressions in (2.19), (2.20), and (2.21) and equating the coefficients of $s_1^i s_2^0$ and of $s_1^i s_2^j$. Specifically, we get this way

$$\begin{aligned}(1-2\rho s_1+\rho^2 s_1^2)(1-\rho s_2)\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} i f(i,j) s_1^{i-1} s_2^j &= (1-\rho)(\lambda_1+(\lambda_3-\rho(\lambda_1+\lambda_3))s_2) \\ &\quad \times \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} s_1^i s_2^j f(i,j)\end{aligned}$$

yielding the equation

$$
\begin{aligned}
&(1-2\rho s_1+\rho^2 s_1^2-\rho s_2+2\rho^2 s_1 s_2-\rho^3 s_1^2 s_2)\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} i f(i,j) s_1^{i-1} s_2^j \\
&= ((1-\rho)\lambda_1+(1-\rho)(\lambda_3-\rho(\lambda_1+\lambda_3))s_2)\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} s_1^i s_2^j f(i,j). \qquad (2.22)
\end{aligned}
$$

Upong equating the coefficients of $s_1^i s_2^0$ on both sides of (2.22), we obtain

$$(i+1)f(i+1,0)-2i\rho f(i,0)+\rho^2(i-1)f(i-1,0) = (1-\rho)\lambda_1 f(i,0)$$

for $i=0,1,2,\ldots,$

$$if(i,0)-2(i-1)\rho f(i-1,0)+\rho^2(i-2)f(i-2,0) = (1-\rho)\lambda_1 f(i-1,0)$$

for $i=1,2,\ldots,$ which yields

$$if(i,0) = (2\rho(i-1)+(1-\rho)\lambda_1)f(i-1,0)-\rho^2(i-2)f(i-2,0),$$

which yields

$$
\begin{aligned}
f(i,0) &= \left(2\rho-\frac{2\rho}{i}+\frac{(1-\rho)\lambda_1}{i}\right)f(i-1,0)-\rho^2(1-\frac{2}{i})f(i-2,0), \\
&= \left(2\rho+\frac{(1-\rho)\lambda_1-2\rho}{i}\right)f(i-1,0)-\rho^2(1-\frac{2}{i})f(i-2,0)
\end{aligned}
$$

for $i=1,2,\ldots.$ This is precisely Eq. (2.13).

Similarly, upon equating the coefficients of $s_1^i s_2^j$ on the LHS and RHS of Eq. (2.22), we obtain

$$\begin{aligned}
f(i+1,j) \quad - \quad & \rho f(i+1,j-1) \\
= \quad & \left(\frac{2\rho i+(1-\rho)\lambda_1}{i+1}\right) f(i,j) + \left(\frac{(1-\rho)(\lambda_3-\rho(\lambda_1+\lambda_3))-2\rho^2 i}{i+1}\right) f(i,j-1) \\
& -\left(\frac{\rho^2(i-1)}{i+1}\right) f(i-1,j) + \left(\frac{\rho^3(i-1)}{i+1}\right) f(i-1,j-1) \\
= \quad & \left(2\rho + \frac{(1-\rho)\lambda_1-2\rho}{i+1}\right) f(i,j) \\
& -\left(2\rho^2 - \frac{(1-\rho)(\lambda_3-\rho(\lambda_1+\lambda_3))+2\rho^2}{i+1}\right) f(i,j-1) \\
& -\rho^2\left(1-\frac{2}{i+1}\right)(f(i-1,j)-\rho f(i-1,j-1))
\end{aligned}$$

for $i = 2, 3, \ldots, j = 1, 2, \ldots$. This is precisely Eq. (2.15).

Eqs. (2.14) and (2.16) in *Proposition* 4.1 can be provided in a similar manner by equating the coefficients of $s_1^0 s_2^j$ and $s_1^i s_2^j$ on both sides of Eq. (2.18) in the following way:

$$(1-2\rho s_2+\rho^2 s_2^2-\rho s_1+2\rho^2 s_1 s_2-\rho^3 s_2^2 s_1)\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} j f(i,j) s_1^i s_2^{j-1}$$

$$= ((1-\rho)\lambda_2+(1-\rho)(\lambda_3-\rho(\lambda_2+\lambda_3))s_1)\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} s_1^i s_2^j f(i,j). \tag{2.23}$$

$$\tag{2.24}$$

Then, equating the coefficients of $s_1^0 s_2^j$ in Eq. (2.23), we obtain

$$(j+1)f(0,j+1)-2j\rho f(0,j)+\rho^2(j-1)f(0,j-1) \quad = \quad (1-\rho)\lambda_2 f(0,j)$$

for $j = 0, 1, 2, \ldots,$

$$jf(0,j) - 2(j-1)\rho f(0,j-1) + \rho^2(j-2)f(0,j-2) \quad = \quad (1-\rho)\lambda_2 f(0,j-1)$$

for $j = 1, 2, \ldots,$ which yields

$$\begin{aligned} f(0,j) &= \left(2\rho - \frac{2\rho}{j} + \frac{(1-\rho)\lambda_2}{j}\right) f(0,j-1) - \rho^2\left(1 - \frac{2}{j}\right) f(0,j-2), \\ &= \left(2\rho + \frac{(1-\rho)\lambda_2 - 2\rho}{j}\right) f(0,j-1) - \rho^2\left(1 - \frac{2}{j}\right) f(0,j-2) \end{aligned}$$

for $j = 1, 2, \ldots.$ This is precisely Eq. (2.14).

Similarly, a comparison of coefficients of $s_1^i s_2^j$ in Eq. (2.23) yields

$$\begin{aligned} f(i,j+1) &- \rho f(i-1,j+1) \\ &= \left(\frac{2\rho j + (1-\rho)\lambda_2}{j+1}\right) f(i,j) + \left(\frac{(1-\rho)(\lambda_3 - \rho(\lambda_2+\lambda_3)) - 2\rho^2 j}{j+1}\right) f(i-1,j) \\ &\quad - \left(\frac{\rho^2(j-1)}{j+1}\right) f(i,j-1) + \left(\frac{\rho^3(j-1)}{j+1}\right) f(i-1,j-1) \\ &= \left(2\rho + \frac{(1-\rho)\lambda_2 - 2\rho}{j+1}\right) f(i,j) \\ &\quad - \left(2\rho^2 - \frac{(1-\rho)(\lambda_3 - \rho(\lambda_2+\lambda_3)) + 2\rho^2}{j+1}\right) f(i-1,j) \\ &\quad -\rho^2\left(1 - \frac{2}{j+1}\right)\left(f(i,j-1) - \rho f(i-1,j-1)\right) \end{aligned}$$

for $i = 1, 2, \ldots, j = 2, 3, \ldots.$This is precisely Eq. (2.16).

Now, we decrease i by 1 in (2.15) and decrease j by 1 in (2.16) to obtain the following recursions:

$$\begin{aligned} f(i,j) &= \rho f(i,j-1) + \left(2\rho + \frac{(1-\rho)\lambda_1 - 2\rho}{i}\right) f(i-1,j) \\ &\quad -\rho^2\left(1-\frac{2}{i}\right)\left(f(i-2,j) - \rho f(i-2,j-1)\right) \\ &\quad -\left(2\rho^2 - \frac{(1-\rho)(\lambda_3 - \rho(\lambda_1+\lambda_3)) + 2\rho^2}{i}\right) f(i-1,j-1) \end{aligned} \tag{2.25}$$

for $i = 2, 3, \ldots, j = 1, 2, \ldots,$ and

$$\begin{aligned} f(i,j) &= \rho f(i-1,j) + \left(2\rho + \frac{(1-\rho)\lambda_2 - 2\rho}{j}\right) f(i,j-1) \\ &\quad -\rho^2\left(1-\frac{2}{j}\right)\left(f(i,j-2) - \rho f(i-1,j-2)\right) \\ &\quad -\left(2\rho^2 - \frac{(1-\rho)(\lambda_3 - \rho(\lambda_2+\lambda_3)) + 2\rho^2}{j}\right) f(i-1,j-1) \end{aligned} \tag{2.26}$$

for $i = 1, 2, \ldots, j = 2, 3, \ldots.$

Let us now introduce the notation

$$\begin{aligned} a(i) &= 2\rho + \frac{(1-\rho)\lambda_1 - 2\rho}{i}, \\ b(i) &= \rho^2\left(1-\frac{2}{i}\right), \\ c(j) &= 2\rho + \frac{(1-\rho)\lambda_2 - 2\rho}{j}, \\ v(i) &= 2\rho^2 - \frac{(1-\rho)(\lambda_3 - \rho(\lambda_1+\lambda_3)) + 2\rho^2}{i}, \\ w(j) &= 2\rho^2 - \frac{(1-\rho)(\lambda_3 - \rho(\lambda_2+\lambda_3)) + 2\rho^2}{j}. \end{aligned}$$

Then, the recursive form of the PMF of Type I bivariate Pólya-Aeppli distribution given in Eqs. (2.13), (2.14), (2.25) and (2.26) can be expressed as follows:

$$\begin{aligned}
f(i,0) &= a(i)f(i-1,0) - b(i)f(i-2,0), && (2.27)\\
f(0,j) &= c(j)f(0,j-1) - b(j)f(0,j-2), && (2.28)\\
f(i,j) &= \rho f(i,j-1) + a(i)f(i-1,j) - b(i)(f(i-2,j) - \rho f(i-2,j-1))\\
&\quad -v(i)f(i-1,j-1), && (2.29)\\
f(i,j) &= \rho f(i-1,j) + c(j)f(i,j-1) - b(j)(f(i,j-2) - \rho f(i-1,j-2))\\
&\quad -w(j)f(i-1,j-1). && (2.30)
\end{aligned}$$

These recursive expressions will be used in over subsequent derivations.

Chapter 3

Methods of Estimation

3.1 Method of Moments

This section describes the parameter estimation using the Method of Moments (MoM); see Minkova and Balakrishnan (2014a). With the joint PGF of (N_1, N_2) as in (2.12), the marginal PGF of N_1 and N_2 are obtained readily as

$$\begin{aligned}
\psi_{N_1}(s_1) = \psi(s_1, 1) &= e^{-(\lambda_1+\lambda_3)(1-\psi_1(s_1))}, \\
\psi_{N_2}(s_2) = \psi(1, s_2) &= e^{-(\lambda_2+\lambda_3)(1-\psi_1(s_2))}.
\end{aligned}$$

These readily imply that $N_1 \sim PA(\lambda_1+\lambda_2, \rho)$ and $N_2 \sim PA(\lambda_2+\lambda_3, \rho)$, as stated earlier from (2.8). Then, the mean and variance of N_1 and N_2 are obtained immediately as follows:

$$\begin{aligned}
E(N_1) &= \frac{\lambda_1+\lambda_3}{1-\rho}, \\
E(N_2) &= \frac{\lambda_2+\lambda_3}{1-\rho}, \\
Var(N_1) &= \frac{(\lambda_1+\lambda_3)(1+\rho)}{(1-\rho)^2}, \\
Var(N_2) &= \frac{(\lambda_2+\lambda_3)(1+\rho)}{(1-\rho)^2}. \qquad (3.1)
\end{aligned}$$

The cross-derivative with respect to s_1 and s_2 from Eq. (2.12) gives

$$\frac{\partial^2 \psi(s_1,s_2)}{\partial s_1 \partial s_2} = \psi(s_1,s_2)[(\lambda_1+\lambda_3\psi_1(s_1))(\lambda_2+\lambda_3\psi_1(s_2))+\lambda_3]\psi_1'(s_1)\psi_1'(s_2).$$

Now, upon setting $s_1 = s_2 = 1$ and using the fact that $\psi_1(1) = 1$, $\psi'(1) = E(X) = \frac{1}{1-\rho}$, the product moment of N_1 and N_2 is obtained as

$$\begin{aligned}
E(N_1N_2) &= \left.\frac{\partial^2 \psi(s_1,s_2)}{\partial s_1 \partial s_2}\right|_{s_1=s_2=1} \\
&= \psi(1,1)[(\lambda_1+\lambda_3\psi_1(1))(\lambda_2+\lambda_3\psi_1(1))+\lambda_3]\psi_1'(1)^2 \\
&= \frac{(\lambda_1+\lambda_3)(\lambda_2+\lambda_3)+\lambda_3}{(1-\rho)^2}.
\end{aligned}$$

Thus, the covariance between N_1 and N_2 is obtained as

$$\begin{aligned}
Cov(N_1,N_2) &= E(N_1N_2) - E(N_1)E(N_2) \\
&= \frac{(\lambda_1+\lambda_3)(\lambda_2+\lambda_3)+\lambda_3}{(1-\rho)^2} - \left(\frac{\lambda_1+\lambda_3}{1-\rho}\right)\left(\frac{\lambda_2+\lambda_3}{1-\rho}\right) \\
&= \frac{\lambda_3}{(1-\rho)^2},
\end{aligned}$$

from which the correlation coefficient is obtained as

$$\begin{aligned} Corr(N_1, N_2) &= \frac{Cov(N_1, N_2)}{\sqrt{Var(N_1)Var(N_2)}} \\ &= \frac{\frac{\lambda_3}{(1-\rho)^2}}{\sqrt{\frac{(\lambda_1+\lambda_3)(1+\rho)}{(1-\rho)^2}\frac{(\lambda_2+\lambda_3)(1+\rho)}{(1-\rho)^2}}} \\ &= \frac{\lambda_3}{(1+\rho)\sqrt{(\lambda_1+\lambda_3)(\lambda_2+\lambda_3)}}. \end{aligned}$$

Now, let (n_{1i}, n_{2i}) be a random sample from a bivariate Pólya-Aeppli distribution, for $i = 1, 2, \ldots, m$, that is,

$$(n_{1i}, n_{2i}) \sim BivPA(\lambda_1, \lambda_2, \lambda_3, \rho)$$

for $i = 1, \ldots, m$. Then, set

$$\bar{n}_1 = \frac{1}{m}\sum_{i=1}^{m} n_{1i}, \tag{3.2}$$

$$\bar{n}_2 = \frac{1}{m}\sum_{i=1}^{m} n_{2i}, \tag{3.3}$$

$$s_1^2 = \frac{1}{m-1}\sum_{i=1}^{m}(n_{1i} - \bar{n}_1)^2, \tag{3.4}$$

$$s_2^2 = \frac{1}{m-1}\sum_{i=1}^{m}(n_{2i} - \bar{n}_2)^2, \tag{3.5}$$

$$s_{12} = \frac{1}{m-1}\sum_{i=1}^{m}(n_{1i} - \bar{n}_1)(n_{2i} - \bar{n}_2). \tag{3.6}$$

Then, Minkova and Balakrishnan (2014a) mentioned two ways of estimating the parameters using the method of moments. First method uses strictly the equations

$$\begin{aligned}
\hat{\rho} &= \frac{s_1^2 + s_2^2 - (\bar{n}_1 + \bar{n}_2)}{s_1^2 + s_2^2 + \bar{n}_1 + \bar{n}_2}, && (3.7)\\
\hat{\lambda}_1 &= (1-\hat{\rho})\bar{n}_1 - (1-\hat{\rho})^2 s_{12}, && (3.8)\\
\hat{\lambda}_2 &= (1-\hat{\rho})\bar{n}_2 - (1-\hat{\rho})^2 s_{12}, && (3.9)\\
\hat{\lambda}_3 &= (1-\hat{\rho})^2 s_{12}. && (3.10)
\end{aligned}$$

The second method sets

$$\begin{aligned}
\theta_1 &= \frac{\lambda_1}{1-\rho},\\
\theta_2 &= \frac{\lambda_2}{1-\rho},\\
\theta_3 &= \frac{\lambda_3}{1-\rho},\\
\phi &= \frac{1+\rho}{1-\rho},
\end{aligned}$$

and uses the facts that

$$\begin{aligned}
E(N_1) &= \theta_1 + \theta_3,\\
E(N_2) &= \theta_2 + \theta_3,\\
Var(N_1) &= (\theta_1 + \theta_3)\phi,\\
Var(N_2) &= (\theta_2 + \theta_3)\phi.
\end{aligned}$$

It then uses the estimate of ϕ as

$$\tilde{\phi} = \frac{1}{2}\left(\frac{s_1^2}{\bar{n}_1} + \frac{s_2^2}{\bar{n}_2}\right), \tag{3.11}$$

to come up with the moment estimates

$$\tilde{\theta}_1 = \bar{n}_1 - \tilde{\theta}_3, \tag{3.12}$$

$$\tilde{\theta}_2 = \bar{n}_2 - \tilde{\theta}_3, \tag{3.13}$$

$$\tilde{\theta}_3 = \frac{s_{12}}{\frac{1+\tilde{\phi}}{2}}, \tag{3.14}$$

$$\tilde{\rho} = \frac{\tilde{\phi} - 1}{\tilde{\phi} + 1} = \frac{\bar{n}_2(s_1^2 - \bar{n}_1) + \bar{n}_1(s_2^2 - \bar{n}_2)}{\bar{n}_2(s_1^2 + \bar{n}_1) + \bar{n}_1(s_2^2 + \bar{n}_2)}. \tag{3.15}$$

Notice that the MoM may not provide admissible estimates in some cases as it can result in negative parameter estimates. In this case, the MoM will be considered to have failed.

3.2 Method of Maximum Likelihood Estimation

This section describes the Maximum Likelihood Estimation (MLE) using the Newton-Raphson (N-R) algorithm, which is a root finding algorithm to solve a nonlinear system of equations. Since the PMF given in Eqs. (2.13), (2.14), (2.25), and (2.26) are recursive functions and so will be the likelihood function, it becomes necessary to use an iterative method to find the parameter estimates. For this purpose, a multivariable nonlinear system of equations will be solved using the N-R method for the likelihood estimation (Press and Vetterling (1989)).

For facilitating the implementation of this algorithm, we present all the first and second order partial derivatives with respect to the parameters.

3.2.1 First and Second Order Partial Derivatives

We find the partial derivatives of the recursive PMF in Eqs. (2.27), (2.28), (2.29), and (2.30), for which we need the following partial derivatives of the involved coefficients:

$$
\begin{array}{llll}
\frac{\partial a(i)}{\partial \lambda_1} = \frac{1-\rho}{i}, & \frac{\partial a(i)}{\partial \lambda_2} = 0, & \frac{\partial a(i)}{\partial \lambda_3} = 0, & \frac{\partial a(i)}{\partial \lambda_\rho} = 2 + \left(\frac{-\lambda_1-2}{i}\right), \\
\frac{\partial^2 a(i)}{\partial \lambda_1^2} = 0, & \frac{\partial^2 a(i)}{\partial \lambda_1 \lambda_2} = 0, & \frac{\partial^2 a(i)}{\partial \lambda_1 \lambda_3} = 0, & \frac{\partial^2 a(i)}{\partial \lambda_1 \lambda_\rho} = \frac{-1}{i}, \\
 & \frac{\partial^2 a(i)}{\partial \lambda_2^2} = 0, & \frac{\partial^2 a(i)}{\partial \lambda_2 \lambda_3} = 0, & \frac{\partial^2 a(i)}{\partial \lambda_2 \lambda_\rho} = 0, \\
 & & \frac{\partial^2 a(i)}{\partial \lambda_3^2} = 0, & \frac{\partial^2 a(i)}{\partial \lambda_3 \lambda_\rho} = 0, \\
 & & & \frac{\partial^2 a(i)}{\partial \lambda_\rho^2} = 0;
\end{array}
$$

$$
\begin{array}{llll}
\frac{\partial c(j)}{\partial \lambda_1} = 0, & \frac{\partial c(j)}{\partial \lambda_2} = \frac{1-\rho}{j}, & \frac{\partial c(j)}{\partial \lambda_3} = 0, & \frac{\partial c(j)}{\partial \lambda_\rho} = 2 + \left(\frac{-\lambda_2-2}{j}\right), \\
\frac{\partial^2 c(j)}{\partial \lambda_1^2} = 0, & \frac{\partial^2 c(j)}{\partial \lambda_1 \lambda_2} = 0, & \frac{\partial^2 c(j)}{\partial \lambda_1 \lambda_3} = 0, & \frac{\partial^2 c(j)}{\partial \lambda_1 \lambda_\rho} = 0, \\
 & \frac{\partial^2 c(j)}{\partial \lambda_2^2} = 0, & \frac{\partial^2 c(j)}{\partial \lambda_2 \lambda_3} = 0, & \frac{\partial^2 c(j)}{\partial \lambda_2 \lambda_\rho} = \frac{-1}{j}, \\
 & & \frac{\partial^2 c(j)}{\partial \lambda_3^2} = 0, & \frac{\partial^2 c(j)}{\partial \lambda_3 \lambda_\rho} = 0, \\
 & & & \frac{\partial^2 c(j)}{\partial \lambda_\rho^2} = 0;
\end{array}
$$

$$
\begin{array}{llll}
\frac{\partial b(i)}{\partial \lambda_1} = 0, & \frac{\partial b(i)}{\partial \lambda_2} = 0, & \frac{\partial b(i)}{\partial \lambda_3} = 0, & \frac{\partial b(i)}{\partial \lambda_\rho} = 2\rho\left(1 - \frac{2}{i}\right), \\
\frac{\partial^2 b(i)}{\partial \lambda_1^2} = 0, & \frac{\partial^2 b(i)}{\partial \lambda_1 \lambda_2} = 0, & \frac{\partial^2 b(i)}{\partial \lambda_1 \lambda_3} = 0, & \frac{\partial^2 b(i)}{\partial \lambda_1 \lambda_\rho} = 0, \\
 & \frac{\partial^2 b(i)}{\partial \lambda_2^2} = 0, & \frac{\partial^2 b(i)}{\partial \lambda_2 \lambda_3} = 0, & \frac{\partial^2 b(i)}{\partial \lambda_2 \lambda_\rho} = 0, \\
 & & \frac{\partial^2 b(i)}{\partial \lambda_3^2} = 0, & \frac{\partial^2 b(i)}{\partial \lambda_3 \lambda_\rho} = 0, \\
 & & & \frac{\partial^2 b(i)}{\partial \lambda_\rho^2} = 2\left(1 - \frac{2}{i}\right);
\end{array}
$$

$$\frac{\partial v(i)}{\partial \lambda_1} = \frac{\rho(1-\rho)}{i}, \quad \frac{\partial v(i)}{\partial \lambda_2} = 0, \quad \frac{\partial v(i)}{\partial \lambda_3} = \frac{-(1-\rho)^2}{i}, \quad \frac{\partial v(i)}{\partial \lambda_\rho} = 4\rho - \frac{\lambda_1(2\rho-1)+2\lambda_3(\rho-1)+4\rho}{i},$$

$$\frac{\partial^2 v(i)}{\partial \lambda_1^2} = 0, \quad \frac{\partial^2 v(i)}{\partial \lambda_1 \lambda_2} = 0, \quad \frac{\partial^2 v(i)}{\partial \lambda_1 \lambda_3} = 0, \quad \frac{\partial^2 v(i)}{\partial \lambda_1 \lambda_\rho} = \frac{1-2\rho}{i},$$

$$\frac{\partial^2 v(i)}{\partial \lambda_2^2} = 0, \quad \frac{\partial^2 v(i)}{\partial \lambda_2 \lambda_3} = 0, \quad \frac{\partial^2 v(i)}{\partial \lambda_2 \lambda_\rho} = 0,$$

$$\frac{\partial^2 v(i)}{\partial \lambda_3^2} = 0, \quad \frac{\partial^2 v(i)}{\partial \lambda_3 \lambda_\rho} = \frac{2(1-\rho)}{i},$$

$$\frac{\partial^2 v(i)}{\partial \lambda_\rho^2} = 4 - \frac{2(\lambda_1+\lambda_3)}{i};$$

$$\frac{\partial w(j)}{\partial \lambda_1} = 0, \quad \frac{\partial w(j)}{\partial \lambda_2} = \frac{\rho(1-\rho)}{j}, \quad \frac{\partial w(j)}{\partial \lambda_3} = \frac{-(\rho-1)^2}{j}, \quad \frac{\partial w(j)}{\partial \lambda_\rho} = 4\rho - \frac{\lambda_2(2\rho-1)+2\lambda_3(\rho-1)+4\rho}{i},$$

$$\frac{\partial^2 w(j)}{\partial \lambda_1^2} = 0, \quad \frac{\partial^2 w(j)}{\partial \lambda_1 \lambda_2} = 0, \quad \frac{\partial^2 w(j)}{\partial \lambda_1 \lambda_3} = 0, \quad \frac{\partial^2 w(j)}{\partial \lambda_1 \lambda_\rho} = 0,$$

$$\frac{\partial^2 w(j)}{\partial \lambda_2^2} = 0, \quad \frac{\partial^2 w(j)}{\partial \lambda_2 \lambda_3} = 0, \quad \frac{\partial^2 w(j)}{\partial \lambda_2 \lambda_\rho} = \frac{1-2\rho}{j},$$

$$\frac{\partial^2 w(j)}{\partial \lambda_3^2} = 0, \quad \frac{\partial^2 w(j)}{\partial \lambda_3 \lambda_\rho} = \frac{2(1-\rho)}{j},$$

$$\frac{\partial^2 w(j)}{\partial \lambda_\rho^2} = 4 - \frac{2(\lambda_2+\lambda_3)}{j}.$$

Using the above expressions, the first order partial derivatives of the recursive PMF function in Eqs. (2.27), (2.28), (2.29), and (2.30) are then obtained as follows:

$$\begin{aligned}
\frac{\partial f(i,0)}{\partial \lambda_1} &= \frac{\partial a(i)}{\partial \lambda_1} f(i-1,0) + a(i)\frac{\partial f(i-1,0)}{\partial \lambda_1} - b(i)\frac{\partial f(i-2,0)}{\partial \lambda_1}, \\
\frac{\partial f(0,j)}{\partial \lambda_1} &= c(j)\frac{\partial f(0,j-1)}{\partial \lambda_1} - b(j)\frac{\partial f(0,j-2)}{\partial \lambda_1}, \\
\frac{\partial f(i,j)}{\partial \lambda_1} &= \rho\frac{\partial f(i,j-1)}{\partial \lambda_1} + \frac{\partial a(i)}{\partial \lambda_1} f(i-1,j) + a(i)\frac{\partial f(i-1,j)}{\partial \lambda_1} \\
&\quad -b(i)\left[\frac{\partial f(i-2,j)}{\partial \lambda_1} - \rho\frac{\partial f(i-2,j-1)}{\partial \lambda_1}\right] - \frac{\partial v(i)}{\partial \lambda_1} f(i-1,j-1) \\
&\quad -v(i)\frac{\partial f(i-1,j-1)}{\partial \lambda_1}, \\
\frac{\partial f(i,j)}{\partial \lambda_1} &= \rho\frac{\partial f(i-1,j)}{\partial \lambda_1} + c(j)\frac{\partial f(i,j-1)}{\partial \lambda_1} - b(j)\left[\frac{\partial f(i,j-2)}{\partial \lambda_1} - \rho\frac{\partial f(i-1,j-2)}{\partial \lambda_1}\right] \\
&\quad -w(j)\frac{\partial f(i-1,j-1)}{\partial \lambda_1};
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial f(i,0)}{\partial \lambda_2} &= a(i)\frac{\partial f(i-1,0)}{\partial \lambda_2} - b(i)\frac{\partial f(i-2,0)}{\partial \lambda_2},\\
\frac{\partial f(0,j)}{\partial \lambda_2} &= \frac{\partial c(j)}{\partial \lambda_2} f(0,j-1) + c(j)\frac{\partial f(0,j-1)}{\partial \lambda_2} - b(j)\frac{\partial f(0,j-2)}{\partial \lambda_2},\\
\frac{\partial f(i,j)}{\partial \lambda_2} &= \rho\frac{\partial f(i,j-1)}{\partial \lambda_2} + a(i)\frac{\partial f(i-1,j)}{\partial \lambda_2} - b(i)\left[\frac{\partial f(i-2,j)}{\partial \lambda_2} - \rho\frac{\partial f(i-2,j-1)}{\partial \lambda_2}\right]\\
&\quad -v(i)\frac{\partial f(i-1,j-1)}{\partial \lambda_2},\\
\frac{\partial f(i,j)}{\partial \lambda_2} &= \rho\frac{\partial f(i-1,j)}{\partial \lambda_2} + \frac{\partial c(j)}{\partial \lambda_2} f(i,j-1) + c(j)\frac{\partial f(i,j-1)}{\partial \lambda_2}\\
&\quad -b(j)\left[\frac{\partial f(i,j-2)}{\partial \lambda_2} - \rho\frac{\partial f(i-1,j-2)}{\partial \lambda_2}\right]\\
&\quad -\frac{\partial w(j)}{\partial \lambda_2} f(i-1,j-1) - w(j)\frac{\partial f(i-1,j-1)}{\partial \lambda_2};
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial f(i,0)}{\partial \lambda_3} &= a(i)\frac{\partial f(i-1,0)}{\partial \lambda_3} - b(i)\frac{\partial f(i-2,0)}{\partial \lambda_3},\\
\frac{\partial f(0,j)}{\partial \lambda_3} &= c(j)\frac{\partial f(0,j-1)}{\partial \lambda_3} - b(j)\frac{\partial f(0,j-2)}{\partial \lambda_3},\\
\frac{\partial f(i,j)}{\partial \lambda_3} &= \rho\frac{\partial f(i,j-1)}{\partial \lambda_3} + a(i)\frac{\partial f(i-1,j)}{\partial \lambda_3} - b(i)\left[\frac{\partial f(i-2,j)}{\partial \lambda_3} - \rho\frac{\partial f(i-2,j-1)}{\partial \lambda_3}\right]\\
&\quad -\frac{\partial v(i)}{\partial \lambda_3} f(i-1,j-1) - v(i)\frac{\partial f(i-1,j-1)}{\partial \lambda_3},\\
\frac{\partial f(i,j)}{\partial \lambda_3} &= \rho\frac{\partial f(i-1,j)}{\partial \lambda_3} + c(j)\frac{\partial f(i,j-1)}{\partial \lambda_3} - b(j)\left[\frac{\partial f(i,j-2)}{\partial \lambda_3} - \rho\frac{\partial f(i-1,j-2)}{\partial \lambda_3}\right]\\
&\quad -\frac{\partial w(j)}{\partial \lambda_3} f(i-1,j-1) - w(j)\frac{\partial f(i-1,j-1)}{\partial \lambda_3};
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial f(i,0)}{\partial \rho} &= \frac{\partial a(i)}{\partial \rho} f(i-1,0) + a(i)\frac{\partial f(i-1,0)}{\partial \rho} - \frac{\partial b(i)}{\partial \rho} f(i-2,0) - b(i)\frac{\partial f(i-2,0)}{\partial \rho}, \\
\frac{\partial f(0,j)}{\partial \rho} &= \frac{\partial c(j)}{\partial \rho} f(0,j-1) + c(j)\frac{\partial f(0,j-1)}{\partial \rho} - \frac{\partial b(i)}{\partial \rho} f(0,j-2) - b(j)\frac{\partial f(0,j-2)}{\partial \rho}, \\
\frac{\partial f(i,j)}{\partial \rho} &= f(i,j-1) + \rho\frac{\partial f(i,j-1)}{\partial \rho} + \frac{\partial a(i)}{\partial \rho} f(i-1,j) + a(i)\frac{\partial f(i-1,j)}{\partial \rho} \\
&\quad - \frac{\partial b(i)}{\partial \rho}\left[f(i-2,j) - \rho f(i-2,j-1)\right] \\
&\quad - b(i)\left[\frac{\partial f(i-2,j)}{\partial \rho} - f(i-2,j-1) - \rho\frac{\partial f(i-2,j-1)}{\partial \rho}\right] \\
&\quad - \frac{\partial v(i)}{\partial \rho} f(i-1,j-1) - v(i)\frac{\partial f(i-1,j-1)}{\partial \rho}, \\
\frac{\partial f(i,j)}{\partial \rho} &= f(i-1,j) + \rho\frac{\partial f(i-1,j)}{\partial \rho} + \frac{\partial c(j)}{\partial \rho} f(i,j-1) + c(j)\frac{\partial f(i,j-1)}{\partial \rho} \\
&\quad - \frac{\partial b(j)}{\partial \rho}\left[f(i,j-2) - \rho f(i-1,j-2)\right] \\
&\quad - b(j)\left[\frac{\partial f(i,j-2)}{\partial \rho} - f(i-1,j-2) - \rho\frac{\partial f(i-1,j-2)}{\partial \rho}\right] \\
&\quad - \frac{\partial w(j)}{\partial \rho} f(i-1,j-1) - w(j)\frac{\partial f(i-1,j-1)}{\partial \rho}.
\end{aligned}
$$

Similarly, the second order partial derivatives of the recursive PMF function in Eqs. (2.27), (2.28), (2.29), and (2.30) are obtained as follows:

$$
\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial \lambda_1^2} &= \frac{2\partial a(i)}{\partial \lambda_1}\frac{\partial f(i-1,0)}{\partial \lambda_1} + a(i)\frac{\partial^2 f(i-1,0)}{\partial \lambda_1^2} - b(i)\frac{\partial^2 f(i-2,0)}{\partial \lambda_1^2}, \\
\frac{\partial^2 f(0,j)}{\partial \lambda_1^2} &= c(j)\frac{\partial^2 f(0,j-1)}{\partial \lambda_1^2} - b(j)\frac{\partial^2 f(0,j-2)}{\partial \lambda_1^2}, \\
\frac{\partial^2 f(i,j)}{\partial \lambda_1^2} &= \rho\frac{\partial^2 f(i,j-1)}{\partial \lambda_1^2} + \frac{2\partial a(i)}{\partial \lambda_1}\frac{\partial f(i-1,j)}{\partial \lambda_1} + a(i)\frac{\partial^2 f(i-1,j)}{\partial \lambda_1^2} \\
&\quad -b(i)\left[\frac{\partial^2 f(i-2,j)}{\partial \lambda_1^2} - \rho\frac{\partial^2 f(i-2,j-1)}{\partial \lambda_1^2}\right] \\
&\quad -\frac{2\partial v(i)}{\partial \lambda_1}\frac{\partial f(i-1,j-1)}{\partial \lambda_1} - v(i)\frac{\partial^2 f(i-1,j-1)}{\partial \lambda_1^2}, \\
\frac{\partial^2 f(i,j)}{\partial \lambda_1^2} &= \rho\frac{\partial^2 f(i-1,j)}{\partial \lambda_1^2} + c(j)\frac{\partial^2 f(i,j-1)}{\partial \lambda_1^2} - b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial \lambda_1^2} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial \lambda_1^2}\right] \\
&\quad -w(j)\frac{\partial^2 f(i-1,j-1)}{\partial \lambda_1^2};
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial \lambda_1 \partial \lambda_2} &= \frac{\partial a(i)}{\partial \lambda_1}\frac{\partial f(i-1,0)}{\partial \lambda_2} + a(i)\frac{\partial^2 f(i-1,0)}{\partial \lambda_1 \partial \lambda_2} - b(i)\frac{\partial^2 f(i-2,0)}{\partial \lambda_1 \partial \lambda_2}, \\
\frac{\partial^2 f(0,j)}{\partial \lambda_1 \partial \lambda_2} &= \frac{\partial c(j)}{\partial \lambda_2}\frac{\partial f(0,j-1)}{\partial \lambda_1} + c(j)\frac{\partial^2 f(0,j-1)}{\partial \lambda_1 \lambda_2} - b(j)\frac{\partial^2 f(0,j-2)}{\partial \lambda_1 \partial \lambda_2}, \\
\frac{\partial^2 f(i,j)}{\partial \lambda_1 \partial \lambda_2} &= \rho\frac{\partial^2 f(i,j-1)}{\partial \lambda_1 \partial \lambda_2} + \frac{\partial a(i)}{\partial \lambda_1}\frac{\partial f(i-1,j)}{\partial \lambda_2} + a(i)\frac{\partial^2 f(i-1,j)}{\partial \lambda_1 \partial \lambda_2} \\
&\quad -b(i)\left[\frac{\partial^2 f(i-2,j)}{\partial \lambda_1 \partial \lambda_2} - \rho\frac{\partial^2 f(i-2,j-1)}{\partial \lambda_1 \partial \lambda_2}\right] \\
&\quad -\frac{\partial v(i)}{\partial \lambda_1}\frac{\partial f(i-1,j-1)}{\partial \lambda_2} - v(i)\frac{\partial^2 f(i-1,j-1)}{\partial \lambda_1 \partial \lambda_2}, \\
\frac{\partial^2 f(i,j)}{\partial \lambda_1 \partial \lambda_2} &= \rho\frac{\partial^2 f(i-1,j)}{\partial \lambda_1 \partial \lambda_2} + \frac{\partial c(j)}{\partial \lambda_2}\frac{\partial f(i,j-1)}{\partial \lambda_1} + c(j)\frac{\partial^2 f(i,j-1)}{\partial \lambda_1 \partial \lambda_2} \\
&\quad -b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial \lambda_1 \partial \lambda_2} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial \lambda_1 \partial \lambda_2}\right] \\
&\quad -\frac{\partial w(j)}{\partial \lambda_2}\frac{\partial f(i-1,j-1)}{\partial \lambda_1} - w(j)\frac{\partial^2 f(i-1,j-1)}{\partial \lambda_1 \partial \lambda_2};
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial\lambda_1\partial\lambda_3} &= \frac{\partial a(i)}{\partial\lambda_1}\frac{\partial f(i-1,0)}{\partial\lambda_3} + a(i)\frac{\partial^2 f(i-1,0)}{\partial\lambda_1\partial\lambda_3} - b(i)\frac{\partial^2 f(i-2,0)}{\partial\lambda_1\partial\lambda_3},\\
\frac{\partial^2 f(0,j)}{\partial\lambda_1\partial\lambda_3} &= c(j)\frac{\partial^2 f(0,j-1)}{\partial\lambda_1\partial\lambda_3} - b(j)\frac{\partial^2 f(0,j-2)}{\partial\lambda_1\partial\lambda_3},\\
\frac{\partial^2 f(i,j)}{\partial\lambda_1\partial\lambda_3} &= \rho\frac{\partial^2 f(i,j-1)}{\partial\lambda_1\partial\lambda_3} + \frac{2\partial a(i)}{\partial\lambda_1}\frac{\partial f(i-1,j)}{\partial\lambda_3} + a(i)\frac{\partial^2 f(i-1,j)}{\partial\lambda_1\partial\lambda_3}\\
&\quad -b(i)\left[\frac{\partial^2 f(i-2,j)}{\partial\lambda_1\partial\lambda_3} - \rho\frac{\partial^2 f(i-2,j-1)}{\partial\lambda_1\partial\lambda_3}\right]\\
&\quad -\frac{\partial v(i)}{\partial\lambda_3}\frac{\partial f(i-1,j-1)}{\partial\lambda_1} - \frac{\partial v(i)}{\partial\lambda_1}\frac{\partial f(i-1,j-1)}{\partial\lambda_3} - v(i)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_1\partial\lambda_3},\\
\frac{\partial^2 f(i,j)}{\partial\lambda_1\partial\lambda_3} &= \rho\frac{\partial^2 f(i-1,j)}{\partial\lambda_1\partial\lambda_3} + c(j)\frac{\partial^2 f(i,j-1)}{\partial\lambda_1\partial\lambda_3} - b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial\lambda_1\partial\lambda_3} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial\lambda_1\partial\lambda_3}\right]\\
&\quad -\frac{\partial w(j)}{\partial\lambda_3}\frac{\partial f(i-1,j-1)}{\partial\lambda_1} - w(j)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_1\partial\lambda_3};
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial\lambda_2^2} &= a(i)\frac{\partial^2 f(i-1,0)}{\partial\lambda_2^2} - b(i)\frac{\partial^2 f(i-2,0)}{\partial\lambda_2^2},\\
\frac{\partial^2 f(0,j)}{\partial\lambda_2^2} &= \frac{2\partial c(j)}{\partial\lambda_2}\frac{\partial f(0,j-1)}{\partial\lambda_2} + c(j)\frac{\partial^2 f(0,j-1)}{\partial\lambda_2^2} - b(j)\frac{\partial^2 f(0,j-2)}{\partial\lambda_2^2},\\
\frac{\partial^2 f(i,j)}{\partial\lambda_2^2} &= \rho\frac{\partial^2 f(i,j-1)}{\partial\lambda_2^2} + a(i)\frac{\partial^2 f(i-1,j)}{\partial\lambda_2^2} - b(i)\left[\frac{\partial^2 f(i-2,j)}{\partial\lambda_2^2} - \rho\frac{\partial^2 f(i-2,j-1)}{\partial\lambda_2^2}\right]\\
&\quad -v(i)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_2^2},\\
\frac{\partial^2 f(i,j)}{\partial\lambda_2^2} &= \rho\frac{\partial^2 f(i-1,j)}{\partial\lambda_2^2} + \frac{2\partial c(j)}{\partial\lambda_2}\frac{\partial f(i,j-1)}{\partial\lambda_2} + c(j)\frac{\partial^2 f(i,j-1)}{\partial\lambda_2^2}\\
&\quad -b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial\lambda_2^2} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial\lambda_2^2}\right]\\
&\quad -\frac{2\partial w(j)}{\partial\lambda_2}\frac{\partial f(i-1,j-1)}{\partial\lambda_2} - w(j)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_2^2};
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial\lambda_2\partial\lambda_3} &= a(i)\frac{\partial^2 f(i-1,0)}{\partial\lambda_2\partial\lambda_3} - b(i)\frac{\partial^2 f(i-2,0)}{\partial\lambda_2\partial\lambda_3}, \\
\frac{\partial^2 f(0,j)}{\partial\lambda_2\partial\lambda_3} &= \frac{\partial c(j)}{\partial\lambda_2}\frac{\partial f(0,j-1)}{\partial\lambda_3} + c(j)\frac{\partial^2 f(0,j-1)}{\partial\lambda_2\partial\lambda_3} - b(j)\frac{\partial^2 f(0,j-2)}{\partial\lambda_2\partial\lambda_3}, \\
\frac{\partial^2 f(i,j)}{\partial\lambda_2\partial\lambda_3} &= \rho\frac{\partial^2 f(i,j-1)}{\partial\lambda_2\partial\lambda_3} + a(i)\frac{\partial^2 f(i-1,j)}{\partial\lambda_2\partial\lambda_3} \\
&\quad -b(i)\left[\frac{\partial^2 f(i-2,j)}{\partial\lambda_2\partial\lambda_3} - \rho\frac{\partial^2 f(i-2,j-1)}{\partial\lambda_2\partial\lambda_3}\right] \\
&\quad -\frac{\partial v(i)}{\partial\lambda_3}\frac{\partial f(i-1,j-1)}{\partial\lambda_2} - v(i)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_2\partial\lambda_3}, \\
\frac{\partial^2 f(i,j)}{\partial\lambda_2\partial\lambda_3} &= \rho\frac{\partial^2 f(i-1,j)}{\partial\lambda_2\partial\lambda_3} + \frac{\partial c(j)}{\partial\lambda_2}\frac{\partial f(i,j-1)}{\partial\lambda_3} + c(j)\frac{\partial^2 f(i,j-1)}{\partial\lambda_2\partial\lambda_3} \\
&\quad -b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial\lambda_2\partial\lambda_3} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial\lambda_2\partial\lambda_3}\right] \\
&\quad -\frac{\partial w(j)}{\partial\lambda_3}\frac{\partial f(i-1,j-1)}{\partial\lambda_2} - \frac{\partial w(j)}{\partial\lambda_2}\frac{\partial f(i-1,j-1)}{\partial\lambda_3} - w(j)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_2\partial\lambda_3};
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial\lambda_2\partial\rho} &= \frac{\partial a(i)}{\partial\rho}\frac{\partial f(i-1,0)}{\partial\lambda_2} + a(i)\frac{\partial^2 f(i-1,0)}{\partial\lambda_2\partial\rho} - \frac{\partial b(i)}{\partial\rho}\frac{\partial f(i-2,0)}{\partial\lambda_2} - b(i)\frac{\partial^2 f(i-2,0)}{\partial\lambda_2\partial\rho}, \\
\frac{\partial^2 f(0,j)}{\partial\lambda_2\partial\rho} &= \frac{\partial^2 c(j)}{\partial\lambda_2\partial\rho}f(0,j-1) + \frac{\partial c(j)}{\partial\rho}\frac{\partial f(0,j-1)}{\partial\lambda_2} + \frac{\partial c(j)}{\partial\lambda_2}\frac{\partial f(0,j-1)}{\partial\rho} \\
&\quad + c(j)\frac{\partial^2 f(0,j-1)}{\partial\lambda_2\partial\rho} - \frac{\partial b(j)}{\partial\rho}\frac{\partial f(0,j-2)}{\partial\lambda_2} - b(j)\frac{\partial^2 f(0,j-2)}{\partial\lambda_2\partial\rho}, \\
\frac{\partial^2 f(i,j)}{\partial\lambda_2\partial\rho} &= \frac{\partial f(i,j-1)}{\partial\lambda_2} + \rho\frac{\partial^2 f(i,j-1)}{\partial\lambda_2\partial\rho} + \frac{\partial a(i)}{\partial\rho}\frac{\partial f(i-1,j)}{\partial\lambda_2} + a(i)\frac{\partial^2 f(i-1,j)}{\partial\lambda_2\partial\rho} \\
&\quad - \frac{\partial b(i)}{\partial\rho}\left[\frac{\partial f(i-2,j)}{\partial\lambda_2} - \rho\frac{\partial f(i-2,j-1)}{\partial\lambda_2}\right] \\
&\quad - b(i)\left[\frac{\partial^2 f(i-2,j)}{\partial\lambda_2\partial\rho} - \frac{\partial f(i-2,j)}{\partial\lambda_2} - \rho\frac{\partial^2 f(i-2,j-1)}{\partial\lambda_2\partial\rho}\right] \\
&\quad - \frac{\partial v(i)}{\partial\rho}\frac{\partial f(i-1,j-1)}{\partial\lambda_2} - v(i)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_2\partial\rho}, \\
\frac{\partial^2 f(i,j)}{\partial\lambda_2\partial\rho} &= \frac{\partial f(i-1,j)}{\partial\lambda_2} + \rho\frac{\partial^2 f(i-1,j)}{\partial\lambda_2\partial\rho} + \frac{\partial^2 c(j)}{\partial\lambda_2\partial\rho}f(i,j-1) + \frac{\partial c(j)}{\partial\rho}\frac{\partial f(i,j-1)}{\partial\lambda_2} \\
&\quad + \frac{\partial c(j)}{\partial\lambda_2}\frac{\partial f(i,j-1)}{\partial\rho} + c(j)\frac{\partial^2 f(i,j-1)}{\partial\lambda_2\partial\rho} \\
&\quad - \frac{b(j)}{\partial\rho}\left[\frac{\partial f(i,j-2)}{\partial\lambda_2} - \rho\frac{\partial f(i-1,j-2)}{\partial\lambda_2}\right] \\
&\quad - b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial\lambda_2\partial\rho} - \frac{\partial f(i,j-2)}{\partial\lambda_2} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial\lambda_2\partial\rho}\right] \\
&\quad - \frac{\partial^2 w(j)}{\partial\lambda_2\partial\rho}f(i-1,j-1) - \frac{\partial w(j)}{\partial\rho}\frac{\partial f(i-1,j-1)}{\partial\lambda_2} \\
&\quad - \frac{\partial w(j)}{\partial\lambda_2}\frac{\partial f(i-1,j-1)}{\partial\rho} - w(j)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_2\partial\rho};
\end{aligned}
$$

$$\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial\lambda_3^2} &= a(i)\frac{\partial^2 f(i-1,0)}{\partial\lambda_3^2} - b(i)\frac{\partial^2 f(i-2,0)}{\partial\lambda_3^2}, \\
\frac{\partial^2 f(0,j)}{\partial\lambda_3^2} &= c(j)\frac{\partial^2 f(0,j-1)}{\partial\lambda_3^2} - b(j)\frac{\partial^2 f(0,j-2)}{\partial\lambda_3^2}, \\
\frac{\partial^2 f(i,j)}{\partial\lambda_3^2} &= \rho\frac{\partial^2 f(i,j-1)}{\partial\lambda_3^2} + a(i)\frac{\partial^2 f(i-1,j)}{\partial\lambda_3^2} - b(i)\left[\frac{\partial^2 f(i-2,j)}{\partial\lambda_3^2} - \rho\frac{\partial^2 f(i-2,j-1)}{\partial\lambda_3^2}\right] \\
&\quad -\frac{2\partial v(i)}{\partial\lambda_3}\frac{\partial f(i-1,j-1)}{\partial\lambda_3} - v(i)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_3^2}, \\
\frac{\partial^2 f(i,j)}{\partial\lambda_3^2} &= \rho\frac{\partial^2 f(i-1,j)}{\partial\lambda_3^2} + c(j)\frac{\partial^2 f(i,j-1)}{\partial\lambda_3^2} - b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial\lambda_3^2} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial\lambda_3^2}\right] \\
&\quad -\frac{2\partial w(j)}{\partial\lambda_3}\frac{\partial f(i-1,j-1)}{\partial\lambda_3} - w(j)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_3^2};
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial\lambda_3\partial\rho} &= \frac{\partial a(i)}{\partial\rho}\frac{\partial f(i-1,0)}{\partial\lambda_3} + a(i)\frac{\partial^2 f(i-1,0)}{\partial\lambda_3\partial\rho} - \frac{\partial b(i)}{\partial\rho}\frac{\partial f(i-2,0)}{\partial\lambda_3} - b(i)\frac{\partial^2 f(i-2,0)}{\partial\lambda_3\partial\rho}, \\
\frac{\partial^2 f(0,j)}{\partial\lambda_3\partial\rho} &= \frac{\partial c(j)}{\partial\rho}\frac{\partial f(0,j-1)}{\partial\lambda_3} + c(j)\frac{\partial^2 f(0,j-1)}{\partial\lambda_3\partial\rho} - \frac{\partial b(j)}{\partial\rho}\frac{\partial f(0,j-2)}{\partial\lambda_3} - b(j)\frac{\partial^2 f(0,j-2)}{\partial\lambda_3\partial\rho}, \\
\frac{\partial^2 f(i,j)}{\partial\lambda_3\partial\rho} &= \frac{\partial f(i,j-1)}{\partial\lambda_3} + \rho\frac{\partial^2 f(i,j-1)}{\partial\lambda_3\partial\rho} + \frac{\partial a(i)}{\partial\rho}\frac{\partial f(i-1,j)}{\partial\lambda_3} + a(i)\frac{\partial^2 f(i-1,j)}{\partial\lambda_3\partial\rho} \\
&\quad -\frac{\partial b(i)}{\partial\rho}\left[\frac{\partial f(i-2,j)}{\partial\lambda_3} - \rho\frac{\partial f(i-2,j-1)}{\partial\lambda_3}\right] \\
&\quad -b(i)\left[\frac{\partial^2 f(i-2,j)}{\partial\lambda_3\partial\rho} - \frac{\partial f(i-2,j)}{\partial\lambda_3} - \rho\frac{\partial^2 f(i-2,j-1)}{\partial\lambda_3\partial\rho}\right] \\
&\quad -\frac{\partial^2 v(i)}{\partial\lambda_3\partial\rho}f(i-1,j-1) - \frac{\partial v(i)}{\partial\rho}\frac{\partial f(i-1,j-1)}{\partial\lambda_3} \\
&\quad -\frac{\partial v(i)}{\partial\lambda_3}\frac{\partial f(i-1,j-1)}{\partial\rho} - v(i)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_3\partial\rho}, \\
\frac{\partial^2 f(i,j)}{\partial\lambda_3\partial\rho} &= \frac{\partial f(i-1,j)}{\partial\lambda_3} + \rho\frac{\partial^2 f(i-1,j)}{\partial\lambda_3\partial\rho} + \frac{\partial c(j)}{\partial\rho}\frac{\partial f(i,j-1)}{\partial\lambda_3} \\
&\quad +c(j)\frac{\partial^2 f(i,j-1)}{\partial\lambda_3\partial\rho} - \frac{b(j)}{\partial\rho}\left[\frac{\partial f(i,j-2)}{\partial\lambda_3} - \rho\frac{\partial f(i-1,j-2)}{\partial\lambda_3}\right] \\
&\quad -b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial\lambda_3\partial\rho} - \frac{\partial f(i,j-2)}{\partial\lambda_3} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial\lambda_3\partial\rho}\right] \\
&\quad -\frac{\partial^2 w(j)}{\partial\lambda_3\partial\rho}f(i-1,j-1) - \frac{\partial w(j)}{\partial\rho}\frac{\partial f(i-1,j-1)}{\partial\lambda_3} \\
&\quad -\frac{\partial w(j)}{\partial\lambda_3}\frac{\partial f(i-1,j-1)}{\partial\rho} - w(j)\frac{\partial^2 f(i-1,j-1)}{\partial\lambda_3\partial\rho};
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial^2 f(i,0)}{\partial\rho^2} &= \frac{2\partial a(i)}{\partial\rho}\frac{\partial f(i-1,0)}{\partial\rho} + a(i)\frac{\partial^2 f(i-1,0)}{\partial\rho^2} - \frac{\partial^2 b(i)}{\partial\rho^2}f(i-2,0) \\
&\quad -\frac{2\partial b(i)}{\partial\rho}\frac{\partial f(i-2,0)}{\partial\rho} - b(i)\frac{\partial^2 f(i-2,0)}{\partial\rho^2}, \\
\frac{\partial^2 f(0,j)}{\partial\rho^2} &= \frac{2\partial c(j)}{\partial\rho}\frac{\partial f(0,j-1)}{\partial\rho} + c(j)\frac{\partial^2 f(0,j-1)}{\partial\rho^2} - \frac{\partial^2 b(i)}{\partial\rho^2}f(0,j-2) \\
&\quad -\frac{2\partial b(i)}{\partial\rho}\frac{\partial f(0,j-2)}{\partial\rho} - b(j)\frac{\partial^2 f(0,j-2)}{\partial\rho^2}, \\
\frac{\partial^2 f(i,j)}{\partial\rho^2} &= \frac{2\partial f(i,j-1)}{\partial\rho} + \rho\frac{\partial f(i,j-1)}{\partial\rho} + \frac{2\partial a(i)}{\partial\rho}\frac{\partial f(i-1,j)}{\partial\rho} + a(i)\frac{\partial^2 f(i-1,j)}{\partial\rho^2} \\
&\quad -\frac{\partial^2 b(i)}{\partial\rho^2}\left[f(i-2,j) - \rho f(i-2,j-1)\right] \\
&\quad -\frac{2\partial b(i)}{\partial\rho}\left[\frac{\partial f(i-2,j)}{\partial\rho} - f(i-2,j-1) - \rho\frac{\partial f(i-2,j-1)}{\partial\rho}\right] \\
&\quad -\frac{\partial^2 v(i)}{\partial\rho^2}f(i-1,j-1) - \frac{2\partial v(i)}{\partial\rho}\frac{\partial f(i-1,j-1)}{\partial\rho} - v(i)\frac{\partial^2 f(i-1,j-1)}{\partial\rho^2}, \\
\frac{\partial^2 f(i,j)}{\partial\rho^2} &= \frac{2\partial f(i-1,j)}{\partial\rho} + \rho\frac{\partial^2 f(i-1,j)}{\partial\rho^2} + \frac{2\partial c(j)}{\partial\rho}\frac{\partial f(i,j-1)}{\partial\rho} + c(j)\frac{\partial^2 f(i,j-1)}{\partial\rho^2} \\
&\quad -\frac{\partial^2 b(j)}{\partial\rho^2}\left[f(i,j-2) - \rho f(i-1,j-2)\right] \\
&\quad -\frac{2\partial b(j)}{\partial\rho}\left[\frac{\partial f(i,j-2)}{\partial\rho} - f(i-1,j-2) - \rho\frac{\partial f(i-1,j-2)}{\partial\rho}\right] \\
&\quad -b(j)\left[\frac{\partial^2 f(i,j-2)}{\partial\rho^2} - \frac{2\partial f(i-1,j-2)}{\partial\rho} - \rho\frac{\partial^2 f(i-1,j-2)}{\partial\rho^2}\right] \\
&\quad -\frac{\partial^2 w(j)}{\partial\rho^2}f(i-1,j-1) - \frac{2\partial w(j)}{\partial\rho}\frac{\partial f(i-1,j-1)}{\partial\rho} - w(j)\frac{\partial^2 f(i-1,j-1)}{\partial\rho^2}.
\end{aligned}
$$

3.2.2 Finding the Root Using Newton-Raphson Algorithm

Given the probability mass function of the bivariate Pólya-Aeppli distribution in *Proposition* 4.1, the likelihood and log-likelihood functions can be written as

$$L(\lambda_1, \lambda_2, \lambda_3, \rho|\mathbf{i}, \mathbf{j}) = \prod_{k=1}^{m} f(N_{1k}, N_{2k}), \tag{3.16}$$

$$\begin{aligned} l(\lambda_1, \lambda_2, \lambda_3, \rho|\mathbf{i}, \mathbf{j}) &= \log(L(\lambda_1, \lambda_2, \lambda_3, \rho|\mathbf{i}, \mathbf{j})) \\ &= \sum_{k=1}^{m} \log(f(N_{1k}, N_{2k})), \end{aligned} \tag{3.17}$$

respectively, where m is the sample size and $\{N_{1k} = i_k, N_{2k} = j_k\}$ are each pair of observed values of the bivariate random variable (N_1, N_2).

In order to determine the MLEs of the model parameters, the derivatives of the log-likelihood function with respect to each of the parameters $\lambda_1, \lambda_2, \lambda_3, \rho$ have to be taken. This can be represented as a vector of length four, $\boldsymbol{F}(\lambda_1, \lambda_2, \lambda_3, \rho)$, and the components in the vector are functions of $\lambda_1, \lambda_2, \lambda_3$, and ρ:

$$\boldsymbol{F}(\lambda_1, \lambda_2, \lambda_3, \rho) = \begin{bmatrix} f_{\lambda_1}(\lambda_1, \lambda_2, \lambda_3, \rho) \\ f_{\lambda_2}(\lambda_1, \lambda_2, \lambda_3, \rho) \\ f_{\lambda_3}(\lambda_1, \lambda_2, \lambda_3, \rho) \\ f_{\rho}(\lambda_1, \lambda_2, \lambda_3, \rho) \end{bmatrix}. \tag{3.18}$$

More precisely,

$$\boldsymbol{F}(\lambda_1,\lambda_2,\lambda_3,\rho) = \begin{bmatrix} \frac{\partial l}{\partial \lambda_1}(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j}) \\ \frac{\partial l}{\partial \lambda_2}(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j}) \\ \frac{\partial l}{\partial \lambda_3}(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j}) \\ \frac{\partial l}{\partial \rho}(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j}) \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^m f(N_{1k},N_{2k})^{-1}\frac{\partial f}{\partial \lambda_1}(N_{1k},N_{2k}) \\ \sum_{k=1}^m f(N_{1k},N_{2k})^{-1}\frac{\partial f}{\partial \lambda_2}(N_{1k},N_{2k}) \\ \sum_{k=1}^m f(N_{1k},N_{2k})^{-1}\frac{\partial f}{\partial \lambda_3}(N_{1k},N_{2k}) \\ \sum_{k=1}^m f(N_{1k},N_{2k})^{-1}\frac{\partial f}{\partial \rho}(N_{1k},N_{2k}) \end{bmatrix}. \tag{3.19}$$

We then need to solve the system of equations $\boldsymbol{F}(\lambda_1,\lambda_2,\lambda_3,\rho) = \mathbf{0}$. Since the log-likelihood functions are recursive in nature as well as the corresponding derivatives, a multivariate Newton-Raphson recursive algorithm (Press and Vetterling (1989)) will be used to find the root of $\boldsymbol{F}$.

Let us consider the *Jacobian* matrix $\boldsymbol{J}(\lambda_1,\lambda_2,\lambda_3,\rho)$ of the vector equation $\boldsymbol{F}$:

$$\boldsymbol{J}(\lambda_1,\lambda_2,\lambda_3,\rho) = \begin{bmatrix} \frac{\partial f_{\lambda_1}}{\partial \lambda_1} & \frac{\partial f_{\lambda_1}}{\partial \lambda_2} & \frac{\partial f_{\lambda_1}}{\partial \lambda_3} & \frac{\partial f_{\lambda_1}}{\partial \rho} \\ \frac{\partial f_{\lambda_2}}{\partial \lambda_1} & \frac{\partial f_{\lambda_2}}{\partial \lambda_2} & \frac{\partial f_{\lambda_2}}{\partial \lambda_3} & \frac{\partial f_{\lambda_2}}{\partial \rho} \\ \frac{\partial f_{\lambda_3}}{\partial \lambda_1} & \frac{\partial f_{\lambda_3}}{\partial \lambda_2} & \frac{\partial f_{\lambda_3}}{\partial \lambda_3} & \frac{\partial f_{\lambda_3}}{\partial \rho} \\ \frac{\partial f_{\rho}}{\partial \lambda_1} & \frac{\partial f_{\rho}}{\partial \lambda_2} & \frac{\partial f_{\rho}}{\partial \lambda_3} & \frac{\partial f_{\rho}}{\partial \rho} \end{bmatrix} \tag{3.20}$$

$$= \begin{bmatrix} \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_1^2} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_1 \partial \lambda_2} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_1 \partial \lambda_3} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_1 \partial \rho} \\ \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_2 \partial \lambda_1} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_2^2} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_2 \partial \lambda_3} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_2 \partial \rho} \\ \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_3 \partial \lambda_1} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_3 \partial \lambda_2} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_3^2} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \lambda_3 \partial \rho} \\ \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \rho \partial \lambda_1} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \rho \partial \lambda_2} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \rho \partial \lambda_3} & \frac{\partial^2 l(\lambda_1,\lambda_2,\lambda_3,\rho|\mathbf{i},\mathbf{j})}{\partial \rho^2} \end{bmatrix}.$$

With some further calculations, the *Jacobian* matrix in (3.20) can be shown to be:

$$\begin{bmatrix} \sum_{k=1}^{m} \frac{\frac{\partial^2 f}{\partial \lambda_1^2}(N_{1k},N_{2k})}{f(N_{1k},N_{2k})} - \frac{\frac{\partial f^2}{\partial \lambda_1}(N_{1k},N_{2k})}{f(N_{1k},N_{2k})^2} & \cdots & \sum_{k=1}^{m} \frac{\frac{\partial^2 f}{\partial \lambda_1 \partial \rho}(N_{1k},N_{2k})}{f(N_{1k},N_{2k})} - \frac{\frac{\partial f}{\partial \lambda_1}(N_{1k},N_{2k})\frac{\partial f}{\partial \rho}(N_{1k},N_{2k})}{f(N_{1k},N_{2k})^2} \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{m} \frac{\frac{\partial^2 f}{\partial \lambda_1 \partial \rho}(N_{1k},N_{2k})}{f(N_{1k},N_{2k})} - \frac{\frac{\partial f}{\partial \lambda_1}(N_{1k},N_{2k})\frac{\partial f}{\partial \rho}(N_{1k},N_{2k})}{f(N_{1k},N_{2k})^2} & \cdots & \sum_{k=1}^{m} \frac{\frac{\partial^2 f}{\partial \rho^2}(N_{1k},N_{2k})}{f(N_{1k},N_{2k})} - \frac{\frac{\partial f^2}{\partial \rho}(N_{1k},N_{2k})}{f(N_{1k},N_{2k})^2} \end{bmatrix},$$

where the first and second partial derivatives of the recursive PMF function with respect to λ_1, λ_2, λ_3, and ρ are as given earlier in Section 3.2.1.

The Newton-Raphson method for nonlinear systems will then be given by the following iterative procedure:

$$\begin{bmatrix} \lambda_1^{(n)} \\ \lambda_2^{(n)} \\ \lambda_3^{(n)} \\ \rho^{(n)} \end{bmatrix} = \begin{bmatrix} \lambda_1^{(n-1)} \\ \lambda_2^{(n-1)} \\ \lambda_3^{(n-1)} \\ \rho^{(n-1)} \end{bmatrix} - \boldsymbol{J}^{-1} \begin{bmatrix} \lambda_1^{(n-1)} \\ \lambda_2^{(n-1)} \\ \lambda_3^{(n-1)} \\ \rho^{(n-1)} \end{bmatrix} \boldsymbol{F} \begin{bmatrix} \lambda_1^{(n-1)} \\ \lambda_2^{(n-1)} \\ \lambda_3^{(n-1)} \\ \rho^{(n-1)} \end{bmatrix}, \qquad (3.21)$$

for $n \geq 1$, where the initial parameters λ_1^0, λ_2^0, λ_3^0, and ρ^0 will be given from the estimate values obtained from the Method of Moments (MoM). $\boldsymbol{J}^{-1}$ is the inverse *Jacobian* matrix with entries $\lambda_1^{(n-1)}$, $\lambda_2^{(n-1)}$, $\lambda_3^{(n-1)}$, and $\rho^{(n-1)}$. The iterative procedure will be continued until a given tolerance error, ϵ, between the n^{th} and $n+1^{th}$ iterate values is attained or a specified maximum number of iterations is reached. The latter stopping condition will be considered as a failure of convergence for the method of MLE (Press and Vetterling (1989)). If a failure occurs in MoM, it will not be able to provide initial values in which case we can use some grid method as described in the following section. Further comparison and analysis based on a simulation study are provided in Chapter 4.

3.2.3 Simple Grid Search for Parameter Optimization

Woodford and Phillips (2012) mentioned a simple Grid Search method for functions. Given an interval $[a, b]$ which contains the optimal point, the method divides the interval $[a, b]$ into smaller sub-intervals, and finds the optimal values at the end-point of each sub-interval by comparing the function values.

In the context of parameter estimation for the Type I bivariate Pólya-Aeppli distribution, the reference for comparison is the value obtained from the maximized log-likelihood function. The parameter setting which possesses the greater maximized log-likelihood value is the optimal estimate. For each of the parameters from $\boldsymbol{\theta} = (\lambda_1, \lambda_2, \lambda_3, \rho)$, we give a number c to add up and subtract from the true parameter values, θ, in order to obtain a region $[\theta - c, \theta + c]$ that contains the optimal value of the estimate. Then, divide the interval $[\theta - c, \theta + c]$ into k sub-intervals. Each of the parameters from $\boldsymbol{\theta} = (\lambda_1, \lambda_2, \lambda_3, \rho)$ will have k points to check with the MLE function. We need to perform a cross-validation check for all combinations of $\lambda_1, \lambda_2, \lambda_3, \rho$ using the MLE function, and find the optimal combination which has the largest MLE value. In the case where the MoM failed, the optimal values obtained by such a Grid method for the estimates will be used for the N-R algorithm as the initial values.

3.2.4 Confidence Intervals using Fisher Information Matrix

Let θ be the scalar parameter of interest. Then, the Fisher Information is defined as (Casella and Berger (2008))

$$I_m(\theta) = -E_\theta\left(\frac{\partial^2 l(\theta|\mathbf{i},\mathbf{j})}{\partial\theta^2}\right),$$

where m is the sample size and $\{N_1 = i, N_2 = j\}$ are each pair of observed values of the bivariate random variable (N_1, N_2). To extend from one-parameter to multiple-parameter case, the *Jacobian* matrix (3.20) defined in the previous subsection will be used to construct the Fisher Information matrix. Let $\boldsymbol{\theta}$ be the parameter vector for λ_1, λ_2, λ_3 and ρ. Then, the Fisher Information matrix is defined as

$$\mathbf{I}_m(\boldsymbol{\theta}) = -\mathbf{J}(\boldsymbol{\theta}). \tag{3.22}$$

As presented in Casella and Berger (2008), the asymptotic distribution of the MLEs of θ is known to be

$$\sqrt{\mathbf{I}_m(\boldsymbol{\theta})}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \to N(0,1). \tag{3.23}$$

Thus, the confidence interval of the j^{th} component in the parameter vector is simply

$$\left[\hat{\theta}_j - c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{jj}},\ \hat{\theta}_j + c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{jj}}\right], \tag{3.24}$$

where c is the quantile from the standard normal distribution, and the subscript jj stands for the element from the j^{th} column and j^{th} row of the inverse of the Fisher

Information matrix $\mathbf{I}_m(\boldsymbol{\theta})$. More precisely, the confidence interval of λ_1, λ_2, λ_3 and ρ as follows:

$$\left[\hat{\lambda}_1 - c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{11}},\ \hat{\lambda}_1 + c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{11}}\right],$$
$$\left[\hat{\lambda}_2 - c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{22}},\ \hat{\lambda}_2 + c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{22}}\right],$$
$$\left[\hat{\lambda}_3 - c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{33}},\ \hat{\lambda}_3 + c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{33}}\right],$$
$$\left[\hat{\rho} - c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{44}},\ \hat{\rho} + c\sqrt{(\mathbf{I}_m(\hat{\boldsymbol{\theta}})^{-1})_{44}}\right].$$

3.2.5 Confidence Intervals by Using Bootstrap Method

Bootstrap is a simple and powerful method for the interval estimation of parameters. It uses *resampling* to produce many bootstrap samples based on an observed sample and then obtain the necessary information required from the population. Let us denote B for the number of bootstrap samples, and $\boldsymbol{\theta}$ for the parameter vector (λ_1, λ_2, λ_3, ρ). Suppose we are given a random sample $(n_{11}, n_{21}), (n_{12}, n_{22}), ..., (n_{1m}, n_{2m}) \sim BivPA(\boldsymbol{\theta})$, where m is the sample size. The algorithm is to first obtain the estimate $\hat{\boldsymbol{\theta}}$ from the original sample, then randomly produce B independent bootstrap samples using $\hat{\boldsymbol{\theta}}$ as the true $\boldsymbol{\theta}$. Denote $\{\boldsymbol{n}^*_{1.b}, \boldsymbol{n}^*_{2.b}\}$ as the bivariate vector of size m from the b^{th} bootstrap sample, where $\{\boldsymbol{n}^*_{1.b}, \boldsymbol{n}^*_{2.b}\} = \{(n^*_{11b}, n^*_{21b}), (n^*_{12b}, n^*_{22b}), ..., (n^*_{1mb}, n^*_{2mb}) \sim BivPA(\hat{\boldsymbol{\theta}})\}$. We thus have B bootstrap samples $\{\boldsymbol{n}^*_{1.1}, \boldsymbol{n}^*_{2.1}\}, \{\boldsymbol{n}^*_{1.2}, \boldsymbol{n}^*_{2.2}\}, ..., \{\boldsymbol{n}^*_{1.B}, \boldsymbol{n}^*_{2.B}\}$. The b^{th} sample, $\{\boldsymbol{n}^*_{1.b}, \boldsymbol{n}^*_{2.b}\}$, will provide an estimated parameter vector $\boldsymbol{\theta}^*_b$, and so we will obtain the set of estimates $\boldsymbol{\theta}^*_1, \boldsymbol{\theta}^*_2, \ldots, \boldsymbol{\theta}^*_B$ from the B bootstrap samples. There are several ways of using these bootstrap estimates to construct confidence intervals for the parameters of interest, and the percentile bootstrap confidence interval will be used in this thesis. The percentile approach simply uses the percentiles from the bootstrap distribution of $\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_B^*$, based on a reasonably large number of B replications. The components of these B estimates vectors will then be sorted in an increasing order

$$\boldsymbol{\theta}_1^{*'}, \boldsymbol{\theta}_2^{*'}, \ldots, \boldsymbol{\theta}_B^{*'},$$

which is equivalent to the matrix form

$$\begin{bmatrix} \theta^{*'}_{1,\lambda_1} & \theta^{*'}_{2,\lambda_1} & \ldots & \theta^{*'}_{B,\lambda_1} \\ \theta^{*'}_{1,\lambda_2} & \theta^{*'}_{2,\lambda_2} & \ldots & \theta^{*'}_{B,\lambda_2} \\ \theta^{*'}_{1,\lambda_3} & \theta^{*'}_{2,\lambda_3} & \ldots & \theta^{*'}_{B,\lambda_3} \\ \theta^{*'}_{1,\rho} & \theta^{*'}_{2,\rho} & \ldots & \theta^{*'}_{B,\rho} \end{bmatrix},$$

and then the $100(1-\alpha)\%$ confidence intervals will be

$$\begin{bmatrix} \theta^{*'}_{B\alpha,\lambda_1} & , & \theta^{*'}_{B(1-\alpha),\lambda_1} \\ \theta^{*'}_{B\alpha,\lambda_2} & , & \theta^{*'}_{B(1-\alpha),\lambda_2} \\ \theta^{*'}_{B\alpha,\lambda_3} & , & \theta^{*'}_{B(1-\alpha),\lambda_3} \\ \theta^{*'}_{B\alpha,\rho} & , & \theta^{*'}_{B(1-\alpha),\rho} \end{bmatrix}.$$

In other words, for a $B = 1000$ replications bootstrap results at a 95% confidence level, the 25^{th} and 976^{th} component from the sorted estimates will provide the lower and upper bounds of the percentile bootstrap confidence interval (Davison and Hinkley (1997)).

Chapter 4

Simulation Study

This chapter contains a simulation study carried out to evaluate the performance of the MLE and the MoM. It aims to compare the performance of these two methods under different settings. The simulation will have 1000 replications of sample series for each of the parameter setting, and a bootstrap with 1000 replications will be performed for each sample in the simulation. The simulation will consider two sample sizes: $m = 50$ and $m = 100$, and various parameter settings.

In the Type I bivariate Pólya-Aeppli distribution, the parameter λ_3 defined in (2.9) is associated with the number of objects in the second part of the compounding $Geo(1-\rho)$ random variables for both random variables N_1 and N_2 in (2.11). Since the correlation between N_1 and N_2 will depend on λ_3, it is necessary to vary λ_3 in this simulation setting, and we chose to vary it between 0.3 and 0.9. λ_1 and λ_2 were kept unchanged with $\lambda_1 = \lambda_2 = 0.6$. The success probability $1-\rho$ from the Geometric distribution is quite important since it decides how the data will be spread out, in other words, it has a big effect on the variance of the data. So, three values were used for

Parameter	**Setting**
sample size(m)	50, 100
λ_1, λ_2	0.6
λ_3	0.3, 0.9
ρ	0.1, 0.25, 0.5

Table 4.1: Parameter settings for the simulation study

ρ as 0.1, 0.25, and 0.5. Table 4.1 lists the parameter settings for this simulation study.

Several measures of performance were computed for both methods for comparative purposes such as rate of success, time required (second) for the calculation, bias, mean squared error (MSE), average width of confidence interval (CI) (with both Fisher Information matrix and bootstrap method) and their coverage probability (CP). In the case where MoM failed to produce the estimates, we performed a Grid Search to find the initial values for the N-R algorithm. For λ_1, λ_2 and λ_3, we choose 0.3 to add and substract from the true values, and obtain the intervals $[\lambda_1 - 0.3, \lambda_1 + 0.3]$, $[\lambda_2 - 0.3, \lambda_2 + 0.3]$, and $[\lambda_3 - 0.3, \lambda_3 + 0.3]$. For ρ, we choose 0.15 to obtain the interval $[\rho - 0.15, \rho + 0.15]$. Then, divide each of the intervals into 12 sub-intervals for the cross-validation check. The optimal combination of the estimates found in this manner were used as the initial values for the N-R algorithm.

The bivariate data were simulated based on the trivariate reduction method described in Section 2.2, and it is quite straight forward. Given a vector of parameter values of λ_1, λ_2, λ_3 and ρ, four sets (U, V, W, R) of compounding $Geo(1-\rho)$ were simulated as given in (2.9). Then, the random variable N_1 and N_2 were obtained as the sums $U + W$ and $V + R$.

Parameters Settings ($\lambda_1, \lambda_2, \lambda_3, \rho$)	Time (s)		Success Rate (%)			
sample size: 50	MLE	MoM	MLE	Grid	MLE(Grid)	MoM
(0.6, 0.6, 0.3, 0.1)	146.27	0.126	82.1	1.6	83.7	86.1
(0.6, 0.6, 0.9, 0.1)	262.73	0.151	81.2	4.6	85.8	86.9
(0.6, 0.6, 0.3, 0.25)	402.46	0.259	95.8	1.5	**97.3**	**97.1**
(0.6, 0.6, 0.9, 0.25)	488.58	0.213	93	2.6	95.6	97.3
(0.6, 0.6, 0.3, 0.5)	761.18	0.298	89.6	5.8	**95.4**	**92.4**
(0.6, 0.6, 0.9, 0.5)	1766.78	0.645	84.7	5.7	89.9	93.8
sample size: 100	MLE	MoM	MLE	Grid	MLE(Grid)	MoM
(0.6, 0.6, 0.3, 0.1)	290.91	0.227	95	0.5	95.5	95.9
(0.6, 0.6, 0.9, 0.1)	449.34	0.292	94.3	2.1	**96.4**	**95.2**
(0.6, 0.6, 0.3, 0.25)	535.31	0.322	99.5	0.3	**99.8**	**99.6**
(0.6, 0.6, 0.9, 0.25)	920.79	0.453	99.4	0	99.4	100
(0.6, 0.6, 0.3, 0.5)	1520.02	0.588	98.6	1	**99.6**	**98.9**
(0.6, 0.6, 0.9, 0.5)	2541.30	0.82	96.8	0.8	97.6	99.2

Table 4.2: Simulation results for sample sizes 50 and 100. The first column contains different parameters settings, the second and third column have the running time required for N-R and MoM, respectively, and the fourth to seventh give the rates of success with and without using the Grid method.

Table 4.2 presents the time required to estimate the parameter values for 1000 sample replicates using MLE and MoM as well as their rates of success with 1000 sample replications. There is a dramatic difference in the running time between MLE and MoM, with the MLE taking over hundred and thousand seconds to complete the calculations while the MoM only needing less than one second to complete 1000 sample calculations. Notice that the running time gets increased with respect to the change of ρ from 0.1 to 0.5 for both methods and both sample sizes. The increasing in time effect is reasonable since the number of objects within cluster is largely depend on the success probability $1 - \rho$ from the Geometric distribution. A large value of $1 - \rho$ will make the cluster small, and a small probability of success $1 - \rho$ will cause a relatively larger cluster. The time required for the calculation will be increased while the number of objects within cluster gets increased. The last two columns in Table 4.2 give the success rates of MLE using the Grid Search method when MoM failed to provide the estimates. In some cases, the success rate of MLE exceeds the success rate of MoM. Notice that three settings with sample size of $n = 100$ has a greater success rate using MLE over MoM.

We have presented two cases for which the MoM failed to estimate the parameters, but the MLE succeeded. Tables 4.3 and 4.4 contain these datasets chosen from the simulation study under the settings ($\lambda_1 = 0.6, \lambda_2 = 0.6, \lambda_3 = 0.3, \rho = 0.5, n = 50$) and ($\lambda_1 = 0.6, \lambda_2 = 0.6, \lambda_3 = 0.9, \rho = 0.1, n = 100$). We have also presented the estimates of the parameters from both MoM and MLE.

Figure 4.1 provides plots of the rates of success for both methods. The rate of

(i,j)	0	1	2	3	4	5	6	7	8	estimates	MoM	MLE
0	8	1	4	1	2	1			1	λ_1	1.21822531	0.9743466
1	4	1	1	1	1	1				λ_2	1.10136477	0.9084642
2	3	3			1					λ_3	-0.01066636	0.1221384
3	1		2	3						ρ	0.35077476	0.3991278
4	2		1	1								
5		2		1								
6			2									
7												
8												
9			1									

Table 4.3: A random sample from the simulation study under the setting $\lambda_1 = 0.6, \lambda_2 = 0.6, \lambda_3 = 0.3, \rho = 0.5, n = 50$ and the estimates from MoM and MLE.

(i,j)	0	1	2	3	4	5	6	estimates	MoM	MLE
0	8	3	1					λ_1	0.82047748	0.58551576
1	8	15	8	1				λ_2	0.62506438	0.39828974
2		11	14	2				λ_3	1.13365357	1.25389098
3		1	6	4	3	2	1	ρ	-0.02849003	0.03280126
4		1	1	3	2	1				
5		2			1					
6				1						

Table 4.4: A random sample from the simulation study under the setting $\lambda_1 = 0.6, \lambda_2 = 0.6, \lambda_3 = 0.9, \rho = 0.1, n = 100$ and the estimates from MoM and MLE.

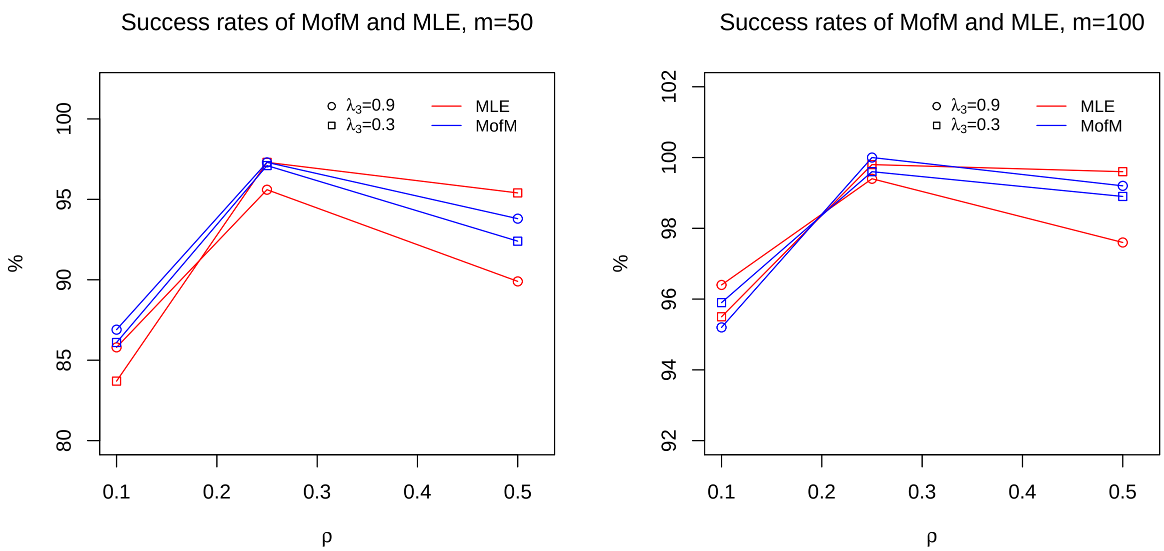

Figure 4.1: Plots of the success rates (%) changing along $\rho = 0.1, 0.25, 0.5$ for MoM and method of MLE (including Grid Search) with sample size m=50 and 100. The left side plot is for $\lambda_3 = 0.9$, while the right side plot is for $\lambda_3 = 0.3$, with $\lambda_1 = \lambda_2 = 0.6$ for both cases.

Parameters Settings	λ_1	λ_2	λ_3	ρ	λ_1	λ_2	λ_3	ρ
sample size: 50	MLE				MoM			
(0.6, 0.6, 0.3, 0.1)	0.0270	0.0294	0.0161	0.0039	0.0322	0.0340	0.0205	0.0045
(0.6, 0.6, 0.9, 0.1)	0.0409	0.0418	0.0343	0.0035	0.0619	0.0620	0.0440	0.0046
(0.6, 0.6, 0.3, 0.25)	0.0317	0.0330	0.0169	0.0058	0.0419	0.0429	0.0245	0.0067
(0.6, 0.6, 0.9, 0.25)	0.0590	0.0597	0.0385	0.0055	0.0940	0.0901	0.0588	0.0060
(0.6, 0.6, 0.3, 0.5)	0.0314	0.0322	0.0166	0.0038	0.0502	0.0508	0.0319	0.0053
(0.6, 0.6, 0.9, 0.5)	0.0677	0.0686	0.0449	0.0034	0.1186	0.1155	0.0823	0.0044
sample size: 100	MLE				MoM			
(0.6, 0.6, 0.3, 0.1)	0.0137	0.0149	0.0086	0.0024	0.0171	0.0186	0.0114	0.0026
(0.6, 0.6, 0.9, 0.1)	0.0225	0.0223	0.0170	0.0022	0.0365	0.0343	0.0227	0.0028
(0.6, 0.6, 0.3, 0.25)	0.0158	0.0172	0.0094	0.0027	0.0226	0.0247	0.0146	0.0032
(0.6, 0.6, 0.9, 0.25)	0.0278	0.0272	0.0194	0.0026	0.0497	0.0456	0.0307	0.0033
(0.6, 0.6, 0.3, 0.5)	0.0162	0.0173	0.0096	0.0015	0.0303	0.0328	0.0215	0.0024
(0.6, 0.6, 0.9, 0.5)	0.0303	0.0294	0.0218	0.0015	0.0641	0.0597	0.0448	0.0021

Table 4.5: Simulation results for sample size 50 and 100. The first four columns contain the MSE values for all four parameter estimates using the MLE method, and the last four columns contain the MSE values for all four parameter estimates using the MoM.

success of MLE with sample size $m = 100$ is usually between $95\% - 100\%$ whereas with sample size $m = 50$ falls between $80\% - 95\%$. A larger sample size will have a higher rate of success for both methods, of course. Left plot shows that MoM has a significantly greater rate of success than method of MLE, but as the sample size increases to 100, both methods achieve quite similar range of success rates, with MLE having a higher rate than MoM in some settings. There is an interesting aspect in the plots in what the rate of success decreases at $\rho = 0.5$ and achieves an optimal success rate at $\rho = 0.25$. A smaller probability of success $1 - \rho$ in Geometric distribution will result in a big spread in the data, thus resulting in a greater chance of providing a negative parameter estimate by MoM. This is an interesting issue that needs further investigation.

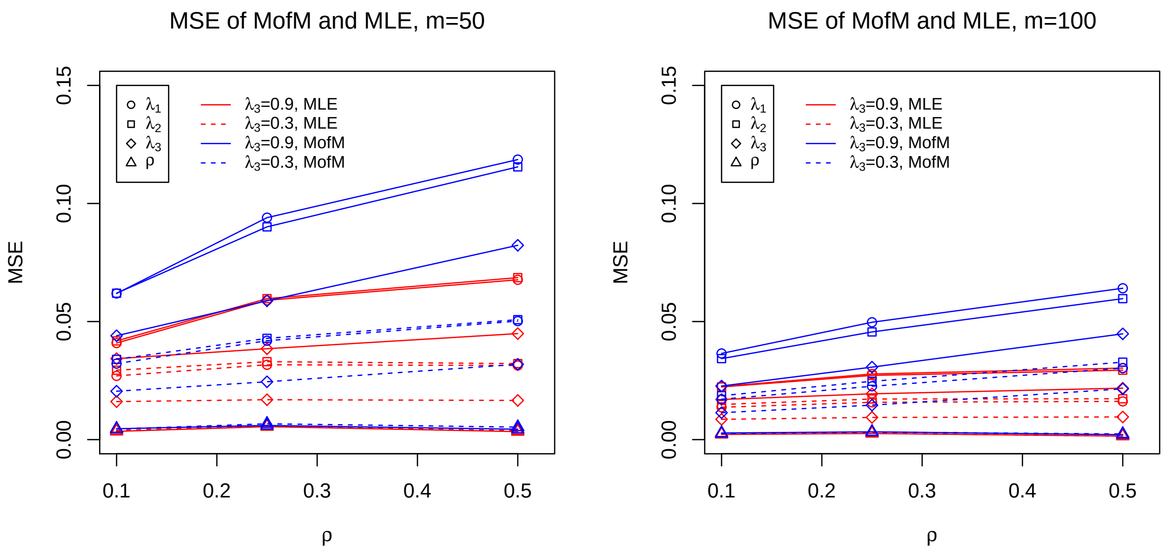

Figure 4.2: Plots of MSE changing along $\rho = 0.1, 0.25, 0.5$ for MoM and MLE with $\lambda_3 = 0.3, 0.9$. The left side plot is for sample size $m = 50$, and the right side plot is for sample size $m = 100$.

Mean squared error (MSE) measures the average squared error between the estimate ($\hat{\theta}$) and its true value (θ). The formula to obtain the MSE is

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{\theta}_i - \theta)^2, \tag{4.1}$$

where n is the number of repeated samples in the simulation study. Naturally, a good estimation method will have smaller values of MSE.

Table 4.5 provides the MSE values, and the MSE values for λ_1 and λ_2 are always greater than those for λ_3 and ρ, and the MSEs obtained from MoM are significantly greater than the MSEs obtained from MLE. The MSEs of ρ are always small for both estimation methods.

Parameters Settings	λ_1	λ_2	λ_3	ρ	λ_1	λ_2	λ_3	ρ
sample size: 50	MLE				MoM			
(0.6, 0.6, 0.3, 0.1)	-0.005	-0.0074	0.0047	0.0077	-0.0122	-0.0179	0.0076	0.0119
(0.6, 0.6, 0.9, 0.1)	0.0039	0.0098	-0.0144	0.0037	-0.0137	-0.0090	-0.0087	0.0111
(0.6, 0.6, 0.3, 0.25)	0.0162	0.0070	0.0016	-0.0130	0.0113	0.0002	0.0013	-0.0081
(0.6, 0.6, 0.9, 0.25)	0.0531	0.0563	-0.0174	-0.0174	0.0395	0.0461	-0.0187	-0.0112
(0.6, 0.6, 0.3, 0.5)	0.0109	0.0060	0.0075	-0.0115	0.0013	-0.0102	0.0198	-0.0117
(0.6, 0.6, 0.9, 0.5)	0.0622	0.0697	-0.0240	-0.0157	0.0543	0.0661	-0.0285	-0.0125
sample size: 100	MLE				MoM			
(0.6, 0.6, 0.3, 0.1)	0.0005	-0.0034	0.0021	-0.0009	-0.0020	-0.0072	0.0021	0.0018
(0.6, 0.6, 0.9, 0.1)	0.0103	0.0103	-0.0080	-0.0017	0.0057	0.0062	-0.0087	0.0015
(0.6, 0.6, 0.3, 0.25)	0.0068	0.0019	0.0013	-0.0059	0.0045	-0.0010	0.0015	-0.0041
(0.6, 0.6, 0.9, 0.25)	0.0199	0.0211	-0.0041	-0.0082	0.0161	0.0182	-0.0065	-0.0055
(0.6, 0.6, 0.3, 0.5)	0.0032	0.0001	0.0034	-0.0037	-0.0013	-0.0047	0.0078	-0.0036
(0.6, 0.6, 0.9, 0.5)	0.0219	0.0226	-0.0056	-0.0067	0.0185	0.0214	-0.0074	-0.0054

Table 4.6: Simulation results for sample sizes 50 and 100. The first four columns contain the Bias for all four parameter estimates using the MLE, and last four columns contain the bias for all four parameter estimates using the MoM.

Figure 4.2 plots the MSE along $\rho = 0.1, 0.25, 0.5$. The two plots also show the greater MSEs of λ_1 and λ_2. The blue lines outline the change of MSE in MoM, and the red lines for MLE. One can see that the red lines are closer to the horizontal axis which means that the MSEs for MLE are smaller than those for MoM. A smaller sample size $m = 50$ results in greater MSEs compared to when $m = 100$. As the sample size increases to $m = 100$, all the MSEs get decreased.

The bias of an estimator is the difference between the expected value of the estimator and the true value of the parameter, and is estimated by:

$$bias(\theta) = \frac{1}{n}\sum_{i=1}^{n}\hat{\theta}_i - \theta. \tag{4.2}$$

An estimator with a bias value that is close to zero is naturally a good estimator as it is

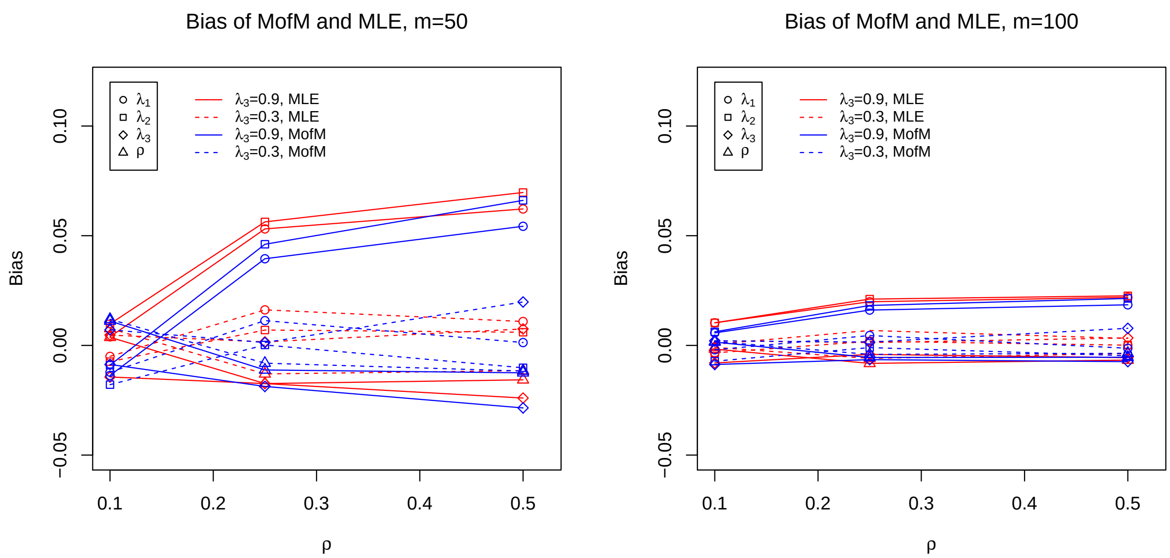

Figure 4.3: Plots of Bias changing along $\rho = 0.1, 0.25, 0.5$ for MoM and MLE method with $\lambda_3 = 0.3, 0.9$. The left side plot is for sample size $m = 50$, and the right side plot is for sample size $m = 100$.

nearly unbiased in this case. The bias value can be either positive or negative. A large negative value shows that the estimator underestimates the parameter considerably, while a large positive value shows that the estimator overestimates the parameter.

Table 4.6 presents the bias values for all the estimates. There are some interesting observations in the bias values changing along with different values of ρ. In Table 4.6, MLE have a significant smaller bias than MoM for all the parameters. The left plot (sample size $m = 50$) from Figure 4.3 shows a significant trend of the bias. Both MoM and MLE tend to have smaller bias when ρ is small, and have bigger bias when ρ increases to 0.5. The right plot from Figure 4.3 with sample size $m = 100$ shows a consistent accuracy for both methods, and all the parameter settings with $\lambda_3 = 0.3$ have better estimates with small bias values. Since almost all bias values of ρ are negative, both methods tend to underestimate the parameter ρ. Also, both methods tend to overestimate λ_3 for all parameter settings when $\lambda_3 = 0.9$, and underestimate λ_3 when $\lambda_3 = 0.3$.

Parameters Settings	λ_1	λ_2	λ_3	ρ	λ_1	λ_2	λ_3	ρ	λ_1	λ_2	λ_3	ρ
sample size: 50	MLE (Fisher)				MLE (Bootstrap)				MoM			
(0.6, 0.6, 0.3, 0.1)	0.6745	0.6721	0.5146	0.2946	0.5888	0.5875	0.4455	0.2237	0.7573	0.7560	0.5686	0.3096
(0.6, 0.6, 0.9, 0.1)	**0.8133**	**0.8195**	0.7142	0.2844	**0.7035**	**0.7050**	0.6874	0.2126	**1.0386**	**1.0400**	0.8155	0.3233
(0.6, 0.6, 0.3, 0.25)	0.7089	0.7031	0.5430	0.2844	0.6474	0.6426	0.4675	0.2728	0.8412	0.8368	0.6425	0.3121
(0.6, 0.6, 0.9, 0.25)	**0.9100**	**0.9141**	0.7867	0.2779	**0.8725**	**0.8736**	0.7663	0.2666	**1.1887**	**1.1925**	0.9549	0.3135
(0.6, 0.6, 0.3, 0.5)	0.7291	0.7255	0.5700	0.2235	0.6688	0.6657	0.4861	0.2281	0.9843	0.9781	0.7830	0.2759
(0.6, 0.6, 0.9, 0.5)	**0.9738**	**0.9815**	0.8647	0.2175	**0.9711**	**0.9760**	0.8441	0.2556	**1.4251**	**1.4297**	1.2044	0.2637
sample size: 100	MLE (Fisher)				MLE (Bootstrap)				MoM			
(0.6, 0.6, 0.3, 0.1)	0.4812	0.4799	0.3651	0.2069	0.4617	0.4598	0.3503	0.1718	0.5428	0.5413	0.4085	0.2190
(0.6, 0.6, 0.9, 0.1)	**0.5836**	**0.5844**	0.5054	0.1994	**0.5462**	**0.5458**	0.5006	0.1661	**0.7483**	**0.7481**	0.5816	0.2295
(0.6, 0.6, 0.3, 0.25)	0.5006	0.4985	0.3836	0.1999	0.4941	0.4918	0.3674	0.2008	0.6027	0.5994	0.4632	0.2237
(0.6, 0.6, 0.9, 0.25)	**0.6251**	**0.6259**	0.5442	0.1995	**0.6260**	**0.6261**	0.5455	0.1958	**0.8415**	**0.8442**	0.6773	0.2233
(0.6, 0.6, 0.3, 0.5)	0.5152	0.5141	0.4014	0.1558	0.5065	0.5052	0.3834	0.1581	0.7011	0.7002	0.5636	0.1950
(0.6, 0.6, 0.9, 0.5)	**0.6622**	**0.6638**	0.5919	0.1514	**0.6648**	**0.6634**	0.5921	0.1543	**1.0027**	**1.0041**	0.8539	0.1859

Table 4.7: Average widths of CIs from simulation results for sample sizes 50 and 100. The first four sets of column contain the CI widths for all four parameters found using the Fisher Information matrix and the MLE, second four sets of column contain the CI widths found using the Bootstrap method with the MLE, and the last four columns contain the CI widths found using the Bootstrap method with the MoM.

The width of confidence intervals (CIs) and coverage probabilities (CP) are also important aspects to evaluate and to compare the performance of different estimation methods. The average width of CI is obtained simply by subtracting the average lower bound from the average upper bound of the CI. The coverage probability of a CI is the number of counts that the true value of the parameter fall within the CI found over the total number of trials. A relatively narrow width of CI with a relatively high coverage probability (close to the nominal level) will be considered as a good method of interval estimation. Tables 4.7 and 4.8 present the average widths of CIs and CPs corresponding to both MLE and MoM, and Figures 4.4 and 4.5 provide the corresponding plots.

The average widths for λ_1 and λ_2 when $\lambda_3 = 0.9$ among the three ways of calculated are significantly wider and has outlined in Table 4.7 with bold font. The bootstrap method with MLE performed slightly better than the one using Fisher Information matrix. But, both these methods based on MLEs produced similar results when the sample size increases to $m = 100$. The average width obtained from Bootstrap with MoM is significantly larger than the methods based on MLE even when the sample size is $m = 100$. This difference can also be observed in Figure 4.4 where in the plots for λ_1 and λ_2 in blue lines are always on the top of other plots for both $m = 50$ and $m = 100$. However, the coverage probabilities in Table 4.8 show a better coverage with MoM except for a few cases outlined in bold. The CIs calculated by using Fisher Information has a better Coverage Probability only under the setting (λ_1=0.6, λ_2=0.6, λ_3=0.3, ρ=0.5) with sample size $m = 100$. From Table 4.8, we see that the MLE using Fisher Information does not have a high coverage probability. Bootstrap method in cases of both MLE and MoM have a relatively better coverage

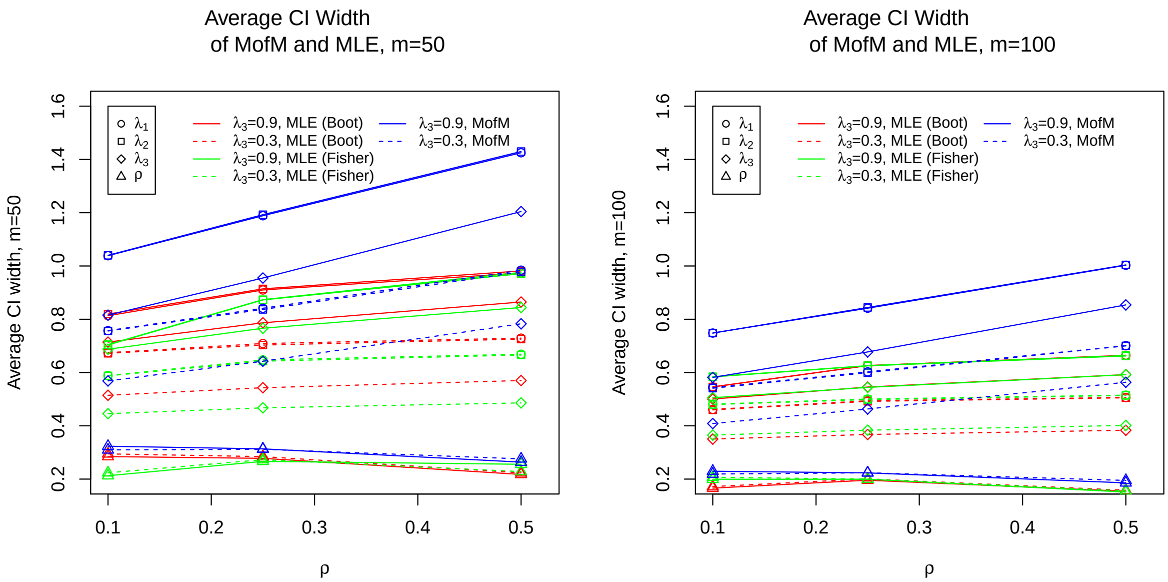

Figure 4.4: Plots of CI widths changing along $\rho = 0.1, 0.25, 0.5$ for MoM and MLE with $\lambda_3 = 0.3, 0.9$. The left side plot is for sample size $m = 50$, and the right side plot is for sample size $m = 100$.

probability. Overall, the methods based on MLE have relatively narrower average width with higher coverage probability than those based on MoM when the sample size is large.

Parameters Settings	λ_1	λ_2	λ_3	ρ	λ_1	λ_2	λ_3	ρ	λ_1	λ_2	λ_3	ρ
sample size: 50	MLE (Fisher)				MLE (Bootstrap)				MoM			
(0.6, 0.6, 0.3, 0.1)	0.9342	0.9245	0.9525	0.9781	0.9568	0.9408	0.9541	0.9951	0.9582	0.9535	0.9605	0.9872
(0.6, 0.6, 0.9, 0.1)	0.9324	0.9373	0.9508	0.9803	0.9536	0.9415	0.9315	0.9952	0.9597	0.9632	0.9425	0.9862
(0.6, 0.6, 0.3, 0.25)	0.9374	0.9280	0.9593	0.9384	0.9577	0.9485	0.9639	0.9217	0.9660	0.9506	0.9701	0.9320
(0.6, 0.6, 0.9, 0.25)	**0.9495**	0.9430	0.9624	0.9430	0.9235	0.9290	0.9419	0.9172	**0.9414**	0.9486	0.9558	0.9496
(0.6, 0.6, 0.3, 0.5)	0.9443	0.9342	0.9710	0.9498	0.9586	0.9565	0.9713	0.9330	0.9600	0.9654	0.9784	0.9372
(0.6, 0.6, 0.9, 0.5)	0.9469	0.9492	0.9575	0.9516	0.9184	0.9162	0.9377	0.9203	0.9701	0.9766	0.9723	0.9435
sample size: 100	MLE (Fisher)				MLE (Bootstrap)				MoM			
(0.6, 0.6, 0.3, 0.1)	0.9568	0.9516	0.9526	0.9705	**0.9634**	**0.9538**	**0.9499**	**0.9906**	**0.9573**	**0.9531**	**0.9489**	**0.9812**
(0.6, 0.6, 0.9, 0.1)	0.9332	0.9495	0.9512	0.9682	0.9379	0.9491	0.9481	0.9866	0.9527	0.9590	0.9401	0.9853
(0.6, 0.6, 0.3, 0.25)	0.9508	0.9387	0.9518	0.9578	**0.9549**	**0.9500**	**0.9489**	**0.9530**	**0.9538**	**0.9408**	**0.9428**	**0.9488**
(0.6, 0.6, 0.9, 0.25)	0.9386	0.9437	0.9557	0.9467	0.9366	0.9407	0.9506	0.9396	0.9410	0.9560	0.9400	0.9560
(0.6, 0.6, 0.3, 0.5)	**0.9523**	**0.9513**	**0.9615**	**0.9645**	**0.9568**	**0.9558**	**0.9539**	**0.9518**	**0.9515**	**0.9484**	**0.9505**	**0.9596**
(0.6, 0.6, 0.9, 0.5)	0.9422	0.9453	0.9639	0.9546	0.9362	0.9410	0.9525	0.9434	0.9546	0.9587	0.9476	0.9486

Table 4.8: CI coverage probabilities from simulation results for sample sizes 50 and 100. The first four sets of column contain the CI coverage probabilities for all four parameters found using the Fisher Information matrix and the MLE, second four sets of column contain the coverage probabilities found using the Bootstrap method with the MLE, and last four columns contain the coverage probabilities found using the Bootstrap method with the MoM.

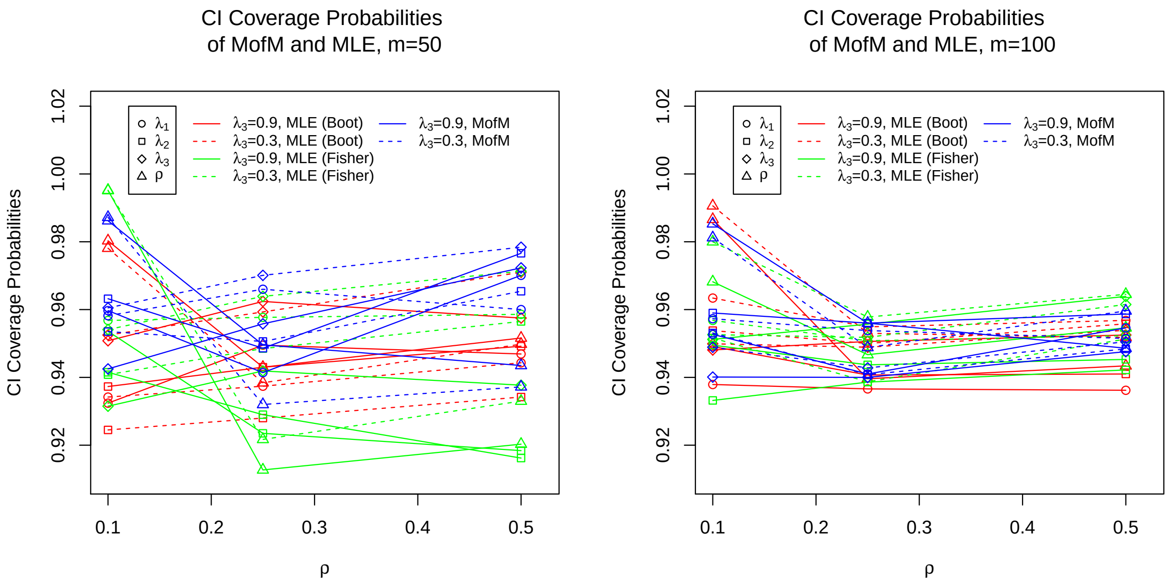

Figure 4.5: Plots of CI coverage probabilities changing along $\rho = 0.1, 0.25, 0.5$ for MoM and MLE with $\lambda_3 = 0.3, 0.9$. The left side plot is for sample size $m = 50$, and the right side plot is for sample size $m = 100$.

Chapter 5

Illustrative Example

We now provide a real data in order to illustrate the methods of inference developed in the preceding chapter. Minkova and Balakrishnan (2014a) used it as a numerical example when they introduced the Type I bivariate Pólya-Aeppli distribution and the corresponding Method of Moments (MoM). Table 5.1 contains the data on a bivariate distribution based on Adelstein's study (unpublished). Specifically, it is about the frequencies of accidents by 122 railway men during two subsequent six year periods. This data was used by Maritz (1950) for fitting a Negative Binomial distribution to detect accident proneness and by Hamdan (1972) for illustrating the fit of a truncated bivariate Poisson distribution.

Based on Eqs. (3.2)-(3.6) in Section 3.1, we found the marginal sample means $\bar{n}_1 = \bar{n}_2 = 1.22951$, the marginal variances $s_1^2 = 1.381452$, $s_2^2 = 1.497155$ and the sample covariance $s_{12} = 0.4145102$. By using the Bivariate Dispersion Test mentioned in (1.2), we obtained the test statistic $I_B = 323.58197$. With 244 degrees of freedom, we clearly reject the null hypothesis at 5% level of significance and conclude that the

Period 1 (Years 1-5)	Period 2 (Years 6-11)								
No. of Accidents	0	1	2	3	4	5	6	7	Total No.
0	21	14	8	1					44
1	17	12	8	3	1			1	42
2	6	9	2	2	2				21
3	1	1	3	3	1				9
4	1	3							4
5				2					2
6									
7									
Total No.	46	39	21	11	4			1	122

Table 5.1: A numerical example of bivariate accident distribution data based on two subsequent periods of six years.

data show an overdispersion.

Minkova and Balakrishnan (2014a) fitted the Type I bivariate Pólya-Aeppli model for the data in Table 5.1, and the results of estimates as well as confidence intervals (CIs) by MoM are presented in Table 5.2. Hence, we use the Fisher Information matrix to find the CIs for all parameters of interest. The N-R method described in Chapter 3 using the MoM estimates as initial values was employed for these data, and the results obtained are presented in Tables 5.3 and 5.4. The Bootstrap methods seen to produce wider CIs than MLE using Fisher Information matrix. Of course, the MLE (using N-R method) took dramatically more time than the MoM.

Parameter	estimate	s.e	CI (Bootstrap)	Length of CI
λ_1	0.665831	0.14114	[0.408757 0.940055]	0.531298
λ_2	0.665831	0.14088	[0.397110 0.963248]	0.566138
λ_3	0.318473	0.10471	[0.104651 0.542470]	0.437819
ρ	0.123467	0.05202	[0.026462 0.216427]	0.189965
Running time (s)	0.02			

Table 5.2: Results for the numerical example using MoM.

Parameter	estimate	s.e	CI (Fisher Information)	Length of CI
λ_1	0.709747	0.12105	[0.472488 0.957007]	0.452902
λ_2	0.703134	0.12189	[0.464241 0.942027]	0.462695
λ_3	0.293456	0.09108	[0.114943 0.471970]	0.351752
ρ	0.101958	0.04674	[0.017973 0.201188]	0.184628
Running time (s)	0.2			

Table 5.3: Results for the numerical example using N-R (CIs using Fisher Information matrix).

Parameter	s.e	CI (Fisher Information)	Length of CI
λ_1	0.11989	[0.473179 0.980239]	0.507060
λ_2	0.12266	[0.470935 0.980977]	0.510042
λ_3	0.08747	[0.099349 0.473961]	0.374612
ρ	0.04544	[0.025124 0.200894]	0.175770

Table 5.4: Results for the numerical example using N-R (CIs using Bootstrap).

	$H_0 : \lambda_1 = \lambda_2$	p-value	$H_0 : \lambda_3 = 0$	p-value
Fisher	0.05628572	0.4922724	3.221963	0.01154437
Bootstrap	0.05785535	0.492335	3.159944	0.01355452

Table 5.5: Wald test statistics for testing $H_0 : \lambda_1 = \lambda_2$ and $H_0 : \rho = 0$ using both Fisher Information matrix and Bootstrap method.

In addition, we may also wish to test the hypotheses

$$H_0 : \lambda_1 = \lambda_2 \quad vs \quad H_1 : \lambda_1 \neq \lambda_2,$$

and

$$H_0 : \lambda_3 = 0 \quad vs \quad H_1 : \lambda_3 > 0.$$

For this, we simply use the Wald test, and the test statistics are found to be as presented in Table 5.5. By using both Fisher Information matrix and Bootstrap method, the result shows that at a 5% level, there is no evidence against $H_0 : \lambda_1 = \lambda_2$. The third and fourth column of Table 5.5 presents the values of the test statistics as well as their p-values for testing $H_0 : \lambda_3 = 0$. At a 5% level, we find enough evidence against the null hypothesis under both methods. Thus, we may conclude that the marginal samples are likely from the same univariate Pólya-Aeppli distribution, and further that the two variables are significantly correlated.

Chapter 6

Discussion and Concluding Remarks

The Pólya-Aeppli distribution is a widely used distribution as a risk insurance model. Minkova and Balakrishnan (2014a) studied the bivariate form of this distribution, and discussed the moment estimates of the parameters. In this work, I have studied the maximum likelihood estimation of the model parameters by using the Newton-Raphson (N-R) algorithm.

A simulation study has been carried to compare the performance of MLE and MoM estimates. Many parameter settings have been considered and the evaluation has been done based on several performance measures such as MSE, bias, width of CIs, and their coverage probabilities. In addition, the running time and convergence success for the methods have also been used for additional comparison.

There is a significant difference between the two methods of estimation in terms of computational time. MoM usually takes a small fraction of time as compared to the MLE. The MLE has a lower rate of success than MoM for small sample sizes, but attains higher rate of success than MoM as the sample size increases. Interestingly, the rate of success reaches an optimal point when $\rho = 0.25$.

The MLE in general has a smaller MSE than MoM. The MSEs for ρ are quite close under both methods of estimation, but MoM possesses a large MSE for λ_1 and λ_2 even with a large sample size $m = 100$. Both methods of estimation possess similar bias though MLE has smaller bias values under a number of settings.

Among the three methods of obtaining Confidence Intervals (Bootstrap with MLE, Bootstrap with MoM, and MLEs with Fisher Information matrix), the MLE with the use of Bootstrap results in CIs with smaller width, but do not always have the highest coverage probabilities. The Bootstrap with MoM, in most cases, have CIs with largest width with a relatively higher coverage probabilities. Considering both width and coverage probability, the MLE methods perform better with larger sample size.

Minkova and Balakrishnan (2014a) also used a real data to illustrate the MoM. Here, we have used the same dataset for parameter estimation by using the MLE. The resulting estimates of the parameters are seen to be different than those obtained by MoM. All the Confidence Intervals obtained from MLE are narrower than the CIs obtained from MoM. The hypothesis tests, carried out by Walt test, show that the two variables are correlated but possibly have the same univariate Pólya-Aeppli distribution.

To conclude, though MLE requires more computational time and effort than MoM, it results in better estimation than MoM in general. As future research, one may develop the Expectation Maximization (EM) algorithm for parameter estimation in the presence of missing data. Recently, Minkova and Balakrishnan (2014b) also introduced the Type II bivariate Pólya-Aeppli distribution by using a different formulation of convolution. It will be of interest to develop the MoM and MLE of model parameters of this Type II bivariate Pólya-Aeppli distribution and then evaluate their relative performance.

Appendix A

R Codes

Partial R codes are provided. Complete R Codes will be provided upon request.

```
#Recursive algorithm from prop 2.1

#This function returns a matrix which contains the entire distribution

#from the joint pmf

pmf.M<-function(x,y,L1,L2,L3,rhoo){

#help functions

a<-function(i){

2*rhoo+{{1-rhoo}*L1-2*rhoo}/i

}

cj<-function(j){

2*rhoo+{{1-rhoo}*L2-2*rhoo}/j

}

b<-function(j){

{1-2/j}*rhoo^2

}
```

```
w<-function(j){
2*rhoo^2-{{1-rhoo}*{L3-rhoo*{L2+L3}}+2*rhoo^2}/j
}
v<-function(i){
2*rhoo^2-{{1-rhoo}*{L3-rhoo*{L1+L3}}+2*rhoo^2}/i
}

A<-matrix(NA,nrow=(y+1),ncol=(x+1))
#initialize f(0,0)
counter<-0
i<-counter;im<-i+1
j<-counter;jm<-j+1
A[im,jm]<-exp(-(L1+L2+L3))
if(x==0 && y==0){return(A)}

#create initial row, f(0,*)
j<-counter+1;jm<-j+1
if(x!=0){A[im,jm]<-cj(j)*A[im,jm-1]}
for(l in ((j+1):x)){
if(x<=1){break}
jm<-l+1
A[im,jm]<-cj(l)*A[im,jm-1]-b(l)*A[im,jm-2]
}
```

```
#create initial column, f(*,0)
i<-counter+1;im<-i+1
j<-counter; jm<-j+1
if(y!=0){A[im,jm]<-a(i)*A[im-1,jm]}
for(k in ((i+1):y)){
if(y<=1){break}
im<-k+1
A[im,jm]<-a(k)*A[im-1,jm]-b(k)*A[im-2,jm]
}


if(min(x,y)<=counter){return(A)}
counter<-counter+1
#initialize rowise
i<-counter;im<-i+1
j<-counter;jm<-j+1
for(l in (j:x)){
jm<-l+1
A[im,jm]<-rhoo*A[im,jm-1]+a(i)*A[im-1, jm]-v(i)*A[im-1,jm-1]
}
#initialize columwise
j<-counter;jm<-j+1
for(k in ((i+1):y)){
if(y<2){break}
im<-k+1
```

```
A[im,jm]<-rhoo*A[im-1,jm]+cj(j)*A[im, jm-1]-w(j)*A[im-1,jm-1]
}

#recursions loop
counter<-counter+1
while(counter<=min(x,y)){
#rowise
i<-counter;im<-i+1
j<-counter;jm<-j+1
for(l in (j:x)){
jm<-l+1
A[im,jm]<-rhoo*A[im,jm-1]+a(i)*A[im-1,jm]-b(i)*(A[im-2,jm]
-rhoo*A[im-2,jm-1])-v(i)*A[im-1,jm-1]
}
if(y<=x && counter==(y)){break;}
#columwise
j<-counter;jm<-j+1
for(k in ((i+1):y)){
im<-k+1
A[im,jm]<-rhoo*A[im-1,jm]+cj(j)*A[im,jm-1]-b(j)*(A[im,jm-2]
-rhoo*A[im-1,jm-2])-w(j)*A[im-1,jm-1]
}
counter<-counter+1
}
```

```
return(A)
}



#This function takes the parameters value with dataset as well as
#the dimension of the data and returns a gradient vector
dL<-function(x,y,P, Data){
f1f<-D.L1(x,y,P[1],P[2],P[3],P[4])/pmf.M(x,y,P[1],P[2],P[3],P[4])
f2f<-D.L2(x,y,P[1],P[2],P[3],P[4])/pmf.M(x,y,P[1],P[2],P[3],P[4])
f3f<-D.L3(x,y,P[1],P[2],P[3],P[4])/pmf.M(x,y,P[1],P[2],P[3],P[4])
frf<-D.Rho(x,y,P[1],P[2],P[3],P[4])/pmf.M(x,y,P[1],P[2],P[3],P[4])
A1<-sum(Data*f1f)
A2<-sum(Data*f2f)
A3<-sum(Data*f3f)
Ar<-sum(Data*frf)
return(c(A1,A2,A3,Ar))
}
#This function takes the parameters value with dataset as well as
#the dimesion of the data and returns a hessian matrix
hL<-function(x,y,P,Data){
ha<-matrix(NA, ncol=4, nrow=4)
df11<-DD.L1(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L1(x,y,P[1],P[2],P[3],P[4])^2
df12<-DD.L12(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
```

```
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L1(x,y,P[1],P[2],P[3],P[4])
*D.L2(x,y,P[1],P[2],P[3],P[4])
df13<-DD.L13(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L1(x,y,P[1],P[2],P[3],P[4])
*D.L3(x,y,P[1],P[2],P[3],P[4])
df1r<-DD.L1R(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L1(x,y,P[1],P[2],P[3],P[4])
*D.Rho(x,y,P[1],P[2],P[3],P[4])
df22<-DD.L2(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L2(x,y,P[1],P[2],P[3],P[4])^2
df23<-DD.L23(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L2(x,y,P[1],P[2],P[3],P[4])
*D.L3(x,y,P[1],P[2],P[3],P[4])
df2r<-DD.L2R(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L2(x,y,P[1],P[2],P[3],P[4])
*D.Rho(x,y,P[1],P[2],P[3],P[4])
df33<-DD.L3(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L3(x,y,P[1],P[2],P[3],P[4])^2
df3r<-DD.L3R(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.L3(x,y,P[1],P[2],P[3],P[4])
*D.Rho(x,y,P[1],P[2],P[3],P[4])
dfrr<-DD.Rho(x,y,P[1],P[2],P[3],P[4])*(1/pmf.M(x,y,P[1],P[2],P[3],P[4]))
-(1/pmf.M(x,y,P[1],P[2],P[3],P[4])^2)*D.Rho(x,y,P[1],P[2],P[3],P[4])^2
D11<-sum(Data*df11)
```

```
D12<-sum(Data*df12)
D13<-sum(Data*df13)
D1r<-sum(Data*df1r)
D22<-sum(Data*df22)
D23<-sum(Data*df23)
D2r<-sum(Data*df2r)
D33<-sum(Data*df33)
D3r<-sum(Data*df3r)
Drr<-sum(Data*dfrr)
ha[1,]<-c(D11,D12,D13,D1r)
ha[2,]<-c(D12,D22,D23,D2r)
ha[3,]<-c(D13,D23,D33,D3r)
ha[4,]<-c(D1r,D2r,D3r,Drr)
return(ha)
}

#This Newton-Raphson algorithm function takes a initial parameters
#vector with dataset as well as the dimesion of the data and returns
#a converged approximation for the parameter to be estimated.
#A failure of convergence will have NAs as return values.
NR.F<-function(x,y,P,Data){
max<-10
err<-0.00001
ii<-0
```

```
er<-rep(10,4)
Pa<-P
result<-matrix(rep(NA, 8), ncol=2, nrow=4)
while((ii<=max) && (er>=err) ){
Pb<-Pa-solve(hL(x,y,Pa,Data))\%*\%dL(x,y,Pa,Data)
e1<-abs(Pb-Pa)
Pa<-Pb
if(min(Pa)<=0 || is.na(Pa[1])){break;}
er<-max(e1)
ii<-ii+1
result[,1]<-Pa
result[,2]<-e1
}
return(result)
}

#This Method of Moment function take the dataset as argument and returns
#a vector of estimated parameters. Any negative result will be consider
#as failure of calculation and will have NAs as return values.
MofM<-function(Data){
i1<-dim(Data)[1]
j1<-dim(Data)[2]
i2<-0
j2<-0
```

```
for(i in 1:i1){
i2<-i2+sum(Data[i,])*(i-1)
}
for(j in 1:j1){
j2<-j2+sum(Data[,j])*(j-1)
}
sa1<-0
sa2<-0
sb<-0
n1.b<-i2/sum(Data)
n2.b<-j2/sum(Data)
for(i in 1:i1){
sa1<-sa1+sum(Data[i,])*((i-1)-n1.b)^2
}
for(j in 1:j1){
sa2<-sa2+sum(Data[,j])*((j-1)-n2.b)^2
}
for(i in 1:i1){
for(j in 1:j1){
sb<-sb+Data[i,j]*((i-1)-n1.b)*((j-1)-n2.b)
}
}
s.1<-sa1/(sum(Data)-1)
s.2<-sa2/(sum(Data)-1)
```

```
s.12<-sb/(sum(Data)-1)
rho.h<-(s.1+s.2-n1.b-n2.b)/(s.1+s.2+n1.b+n2.b)
L1.h<-(1-rho.h)*n1.b-(1-rho.h)^2*s.12
L2.h<-(1-rho.h)*n2.b-(1-rho.h)^2*s.12
L3.h<-(1-rho.h)^2*s.12

rho.t<-(n2.b*(s.1-n1.b)+n1.b*(s.2-n2.b))/(n2.b*(s.1+n1.b)+n1.b*(s.2+n2.b))
phi.t<-(s.1/n1.b+s.2/n2.b)/2
theta3<-(2*s.12)/(1+(s.1/n1.b+s.2/n2.b)/2)
theta1<-n1.b-theta3
theta2<-n2.b-theta3
return(c(L1.h, L2.h, L3.h, rho.h))
}

#This function is an extended of MofM with the Bootstrap CIs as return
MofMF<-function(Data,P){
time<-system.time(t<-MofM(Data))
if(min(t)<=0){return(NA)}
b<-matrix(NA, ncol=4, nrow=1000)
m<-sum(Data)
for(i in 1:1000){
Da<-BivPA(t,m)
b[i,]<-MofM(Da)
}
```

```
s<-apply(b,2,sort)
CIL1<-c(s[25,1],s[975,1])
c1<-(P[1]>=CIL1[1] && P[1]<= CIL1[2])
CIL2<-c(s[25,2],s[975,2])
c2<-(P[2]>=CIL2[1] && P[2]<= CIL2[2])
CIL3<-c(s[25,3],s[975,3])
c3<-(P[3]>=CIL3[1] && P[3]<= CIL3[2])
CIRh<-c(s[25,4],s[975,4])
cr<-(P[4]>=CIRh[1] && P[4]<= CIRh[2])
return(c(t,CIL1,c1,CIL2,c2,CIL3,c3,CIRh,cr,time[1]))
}
#This function takes a vector of parameter values and a sample size m,
#and return a simulated bivariate Polya-Aeppli distribution in a matrix form.
BivPA<-function(P, m){
BivPA<-matrix(0, ncol=2, nrow=m)
BivP<-matrix(0,ncol=m, nrow=m)
for(i in 1:m){
Z1<-rpois(1,P[1])
Z2<-rpois(1,P[2])
Z3<-rpois(1,P[3])
n1<-rgeom(Z1+Z3,1-P[4])
n2<-rgeom(Z2+Z3,1-P[4])
N1<-sum(n1)+Z1+Z3
N2<-sum(n2)+Z2+Z3
```

```
BivPA[i,1]<-N1
BivPA[i,2]<-N2
BivP[N1+1,N2+1]<-BivP[N1+1,N2+1]+1
}
y<-max( BivPA[,1])
x<-max(BivPA[,2])
return(BivP[1:(y+1),1:(x+1)])
}
#This function takes the vector of true parameters value, the vector
#of estimated parameters, and the dataset, and will return the 95\%
#Confidence Intervals using Fisher Information matrix and their
#respected Coverage Probability.
CInt<-function(x,y,t,P,Data){
h<-hL(x,y,t,Data)
s<--solve(h)
CI1<-t[1]+qnorm(0.975)*c(-1,1)*sqrt(s[1,1])
c1<-(P[1]>=CI1[1] && P[1]<= CI1[2])
CI2<-t[2]+qnorm(0.975)*c(-1,1)*sqrt(s[2,2])
c2<-(P[2]>=CI2[1] && P[2]<= CI2[2])
CI3<-t[3]+qnorm(0.975)*c(-1,1)*sqrt(s[3,3])
c3<-(P[3]>=CI3[1] && P[3]<= CI3[2])
CIr<-t[4]+qnorm(0.975)*c(-1,1)*sqrt(s[4,4])
cr<-(P[4]>=CIr[1] && P[4]<= CIr[2])
c(CI1,c1,CI2,c2,CI3,c3,CIr,cr)
```

```
}
#This function takes number of replication N and perform a Bootstrap using
#the Dataset, the estimated vector of parameters, real parameters value,
#and will return a 95\% Confidence Intervals and their
#respected Coverage Probability.
Boots<-function(Data,P,Pa,N){
x<-dim(Data)[2]-1
y<-dim(Data)[1]-1
t<-NR.F(x,y,Pa,Data)
if(is.na(t[1,1])){return(NA)}
p<-t[1:4,1]
ma<-matrix(NA, ncol=4, nrow=N)
i<-1
m<-sum(Data)
while(i<=N){
Da<-BivPA(p,m)
x<-dim(Da)[2]-1
y<-dim(Da)[1]-1
t<-NR.F(x,y,Pa,Da)
if(is.na(t[1,1])){next}
ma[i,]<-t[,1]
i<-i+1
}
r<-apply(na.omit(ma),2,mean)
```

```
s<-apply(na.omit(ma),2,sort)
CIL1<-c(s[ceiling(N*0.025),1],s[round(N*0.975),1])
c1<-(P[1]>=CIL1[1] && P[1]<= CIL1[2])
CIL2<-c(s[ceiling(N*0.025),2],s[round(N*0.975),2])
c2<-(P[2]>=CIL2[1] && P[2]<= CIL2[2])
CIL3<-c(s[ceiling(N*0.025),3],s[round(N*0.975),3])
c3<-(P[3]>=CIL3[1] && P[3]<= CIL3[2])
CIRh<-c(s[ceiling(N*0.025),4],s[round(N*0.975),4])
cr<-(P[4]>=CIRh[1] && P[4]<= CIRh[2])
return(c(CIL1,c1,CIL2,c2,CIL3,c3,CIRh,cr))
}

#This function takes the vector of real parameters value, sample sizes,
#and number of sample replicates for simulation, and will return
#the required results
simT<-function(P,nn,N){
ptm <- proc.time()
ma<-matrix(NA, ncol=17, nrow=N)
mo<-matrix(NA, ncol=17, nrow=N)
Bo<-matrix(NA, ncol=12, nrow=N)
for(i in 1:N){
set.seed(i)
Da<-BivPA(P,nn)
x<-dim(Da)[2]-1
```

```
y<-dim(Da)[1]-1
p<-MofMF(Da,P)
mo[i,1:16]<-p
mo[i,17]<-logL(x,y,p[1:4],Da)
t<-NR.F(x,y,p[1:4],Da)
Bo[i,]<-Boots(Da,P,t[,1],1000)
ma[i,1:4]<-t[,1]
ma[i,5]<-logL(x,y,t[,1],Da)
ma[i,6:17]<-CInt(x,y,t[,1],P,Da)
}
outputB<-apply(Bo,2,mean)

m<-na.omit(ma)
nm<-dim(m)[1]
output<-apply(m,2,mean)
B1<-sum(m[,1]-P[1])/nm
MSE1<-sum((m[,1]-P[1])^2)/nm
B2<-sum(m[,2]-P[2])/nm
MSE2<-sum((m[,2]-P[2])^2)/nm
B3<-sum(m[,3]-P[3])/nm
MSE3<-sum((m[,3]-P[3])^2)/nm
Br<-sum(m[,4]-P[4])/nm
MSEr<-sum((m[,4]-P[4])^2)/nm
```

```
oB1<-sum(mo[,1]-P[1])/nm
oMSE1<-sum((mo[,1]-P[1])^2)/N
oB2<-sum(mo[,1]-P[2])/nm
oMSE2<-sum((mo[,2]-P[2])^2)/N
oB3<-sum(mo[,1]-P[3])/nm
oMSE3<-sum((mo[,3]-P[3])^2)/N
oBr<-sum(mo[,1]-P[4])/nm
oMSEr<-sum((mo[,4]-P[4])^2)/N

outputM<-apply(mo,2,mean)

t<-proc.time() - ptm
return(list(success=nm, setting=P,size=nn, oBias=c(oB1, oB2, oB3, oBr),
oMSE=c(oMSE1,oMSE2,oMSE3,oMSEr), Bias=c(B1, B2, B3, Br),
MSE=c(MSE1, MSE2, MSE3, MSEr), op=output, opM=outputM,opB=outputB, ma, mo, time=t)
}
#This function takes the argument given from the simulation function
#and print out the result in a readable way with labels
output.print<-function(s){
out<-list(success=s$success, setting=s$setting,size=s$size, oBias=s$oBias,
oMSE=s$oMSE,Bias=s$Bias, MSE=s$MSE, op=s$op, opM=s$opM,opB=s$opB, time=s$time)
class(out)<-"sim"
out
}
```

```
 print.sim<-function(out, digits=5){
  #function to print the simulation results
    cat(paste("Simulation result with sample size ",out$size,
  " and parameters: ", out$setting[1],",", out$setting[2],",",out$setting[3],
  ",",out$setting[4],"\n\n"))
     cat(paste("success: ", out$success), "\n")

    cat(paste("N-F estimates", round(out$op[1],digits),round(out$op[2],digits),
  round(out$op[3],digits),round(out$op[4],digits),"\n"))
    cat(paste("N-F estimates Bias", round(out$Bias[1],digits),
  round(out$Bias[2],digits),round(out$Bias[3],digits),
  round(out$Bias[4],digits),"\n"))
    cat(paste("N-F estimates MSE", round(out$MSE[1],digits),
  round(out$MSE[2],digits),round(out$MSE[3],digits),
  round(out$MSE[4],digits),"\n"))
    cat(paste("N-F estimates logL", round(out$op[5],digits),"\n"))
cat(paste("N-F L1 CI & Coverage P", round(out$op[6],digits),
round(out$op[7],digits),round(out$op[8],digits),"\n"))
cat(paste("N-F L2 CI & Coverage P", round(out$op[9],digits),
round(out$op[10],digits),round(out$op[11],digits),"\n"))
cat(paste("N-F L3 CI & Coverage P", round(out$op[12],digits),
round(out$op[13],digits),round(out$op[14],digits),"\n"))
cat(paste("N-F Rho CI & Coverage P", round(out$op[15],digits),
round(out$op[16],digits),round(out$op[17],digits),"\n"))
```

```
cat(paste("N-F Boot L1 CI & Coverage P", round(out$opB[1],digits),
round(out$opB[2],digits),round(out$opB[3],digits),"\n"))
cat(paste("N-F Boot L2 CI & Coverage P", round(out$opB[4],digits),
round(out$opB[5],digits),round(out$opB[6],digits),"\n"))
cat(paste("N-F Boot L3 CI & Coverage P", round(out$opB[7],digits),
round(out$opB[8],digits),round(out$opB[9],digits),"\n"))
cat(paste("N-F Boot Rho CI & Coverage P", round(out$opB[10],digits),
round(out$opB[11],digits),round(out$opB[12],digits),"\n"))

    cat(paste("MofM estimates", round(out$opM[1],digits),
  round(out$opM[2],digits),round(out$opM[3],digits),
  round(out$opM[4],digits),"\n"))
    cat(paste("MofM estimates Bias", round(out$oBias[1],digits),
  round(out$oBias[2],digits),round(out$oBias[3],digits),
  round(out$oBias[4],digits),"\n"))
    cat(paste("MofM estimates MSE", round(out$oMSE[1],digits),
  round(out$oMSE[2],digits),round(out$oMSE[3],digits),
  round(out$oMSE[4],digits),"\n"))
    cat(paste("MofM estimates logL", round(out$opM[17],digits),"\n"))
    cat(paste("MofM L1 CI & Coverage P", round(out$opM[5],digits),
  round(out$opM[6],digits),round(out$opM[7],digits),"\n"))
cat(paste("MofM L2 CI & Coverage P", round(out$opM[8],digits),
round(out$opM[9],digits),round(out$opM[10],digits),"\n"))
```

```
cat(paste("MofM L3 CI & Coverage P", round(out$opM[11],digits),
round(out$opM[12],digits),round(out$opM[13],digits),"\n"))
cat(paste("MofM Rho CI & Coverage P", round(out$opM[14],digits),
round(out$opM[15],digits),round(out$opM[16],digits),"\n"))
    #invisible(out)
 }

#different parameters settings
P1<-c(0.6, 0.6, 0.3, 0.1)
P2<-c(0.6, 0.6, 0.9, 0.1)
P3<-c(0.6, 0.6, 0.3, 0.25)
P4<-c(0.6, 0.6, 0.9, 0.25)
P5<-c(0.6, 0.6, 0.3, 0.5)
P6<-c(0.6, 0.6, 0.9, 0.5)
s1a<-simT(P1,50,1000)
output.print(s1a)
s1b<-simT(P1,100,1000)
output.print(s1b)
s2a<-simT(P2,50,1000)
output.print(s2a)
s2b<-simT(P2,100,1000)
output.print(s2b)
s3a<-simT(P3,50,1000)
output.print(s3a)
```

```
s3b<-simT(P3,100,1000)
output.print(s3b)
s4a<-simT(P4,50,1000)
output.print(s4a)
s4b<-simT(P4,100,1000)
output.print(s4b)
s5a<-simT(P5,50,1000)
output.print(s5a)
s5b<-simT(P5,100,1000)
output.print(s5b)
s6a<-simT(P6,50,1000)
output.print(s6a)
s6b<-simT(P6,100,1000)
output.print(s6b)


############## Real Data Application ###########
#Import dataset
Data <-matrix(c(
21,14,8,1,0,0,0,0,
17,12,8,3,1,0,0,1,
6,9,2,2,2,0,0,0,
1,1,3,3,1,0,0,0,
1,3,0,0,0,0,0,0,
```

```
0,0,0,2,0,0,0,0
), ncol=8, nrow=6, byrow=T)


#MofM
MofM(Data)


#method of MLE
NR.F(7,5,MofM(Data), Data)
```

Bibliography

Anscombe, F. J. (1950). Sampling theory of the negative binomial and logarithmic series distributions. *Biometrika*, **37**, 358–382.

Balakrishnan, N. and Lai, C. D. (2009). *Continuous Bivariate Distributions*. Springer, New York, 2nd edition.

Campbell, J. T. (1934). The Poisson correlation function. *Proceedings of the Edinburgh Mathematical Society*, **4**, 18–26.

Casella, G. and Berger, R. L. (2008). *Statistical Inference*. Thomson Press, Haryana, 2nd edition.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and Their Applications*. Cambridge University Press, England.

Dragieva, V. (2011). Queueing system with Polya-Aeppli input process. *Ann. UACEG*, **43-44**, 7–14.

Fisher, R. A., Thornton, H. G., and Mackenzie, W. A. (1922). The accuracy of the plating method of estimating bacterial populations. *Annals of Applied Biology*, **9**, 325–335.

Hamdan, M. A. (1972). Estimation in the truncated bivariate Poisson distribution. *Technometrics*, **14**, 37–45.

Haydn, N. and Valenti, S. (2009). The compound Poisson distribution and return times in dynamical systems. *Probability Theory and Related Fields*, **144**, 517–542.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1997). *Discrete Multivariate Distributions*. John Wiley & Sons, New York.

Johnson, N. L., Kemp, A. W., and Kotz, S. (2005). *Univariate Discrete Distribution*. John Wiley & Sons, New York, 3rd edition.

Lindén, A. and Mäntyniemi, S. (2011). Using the negative binomial distribution to model overdispersion in ecological count data. *Ecology*, **92**, 1414–1421.

Loukas, S. and Kemp, C. D. (1986). The index of dispersion test for the bivariate Poisson distribution. *Biometrics*, **42**, 941–948.

Maritz, J. S. (1950). On the validity of inferences drawn from the fitting of Poisson and negative binomial distributions to observed accident data. *Psychological Bulletin*, **47**, 434–43.

Minkova, L. D. (2004). The Polya-Aeppli process and ruin problems. *Journal of Applied Mathematics and Stochastic Analysis*, **3**, 221–234.

Minkova, L. D. (2012). *Distributions in Insurance Risk Models*. Ph.D. thesis, Sofia University, Sofia, Bulgaria.

Minkova, L. D. and Balakrishnan, N. (2014a). On a Bivariate Pólya-Aeppli distribution. *Communications in Statistics - Theory and Methods*, **43**, 5026–5038.

Minkova, L. D. and Balakrishnan, N. (2014b). Type II bivariate Pólya-Aeppli distribution. *Statistics & Probability Letters*, **88**, 40–49.

Press, W. H. and Vetterling, W. T. (1989). *Numerical Recipes in Pascal: The Art of Scientific Computing*. Cambridge University Press, England.

Woodford, C. and Phillips, C. (2012). *Numerical Methods with Worked Examples*. Springer, New York.