基于 PA 分布的 INAR(1) 模型分析

李天恺，林瑞洋，董宇坤
(中国科学技术大学，管理学院)

摘　要

离散值自回归模型中通常选取服从泊松分布的随机项 ϵ_t，这类基于泊松分布的 INAR(1) 模型具有方便计算等优点，但泊松分布离散指数等于 1 的特点也使其在分析过离散数据时存在缺陷，因此有必要选择一种单参、过离散分布的随机项作为补充。本文提出了以 PA 分布作为随机项构建的 INAR(1) 模型，讨论了该模型的一些基本性质以及参数估计的方法，最后通过模拟实验和分析真实数据来展示该模型的优越性。

关键词：PA 分布、过离散、参数估计、一阶整值自回归模型

目录

1 前言 **1**

2 PA-INAR(1) 模型 **2**

2.1 PA 分布 2

2.2 PA-INAR(1) 模型的定义与性质 4

3 模型参数估计 **5**

3.1 Yule-Walker 方法 (矩估计法) 5

3.2 条件最小二乘估计 6

3.3 条件极大似然估计 7

4 模拟实验 **8**

5 真实数据案例 **11**

6 结论 **12**

1 前言

近年来，计数时间序列在制药、金融等领域得到了越来越广泛的应用，该方向最常见的模型当属由 McKenzie(1985) 提出的一阶整值自回归模型。对于非负整值随机变量 X 和常数 $\alpha \in (0,1)$，定义细化算子 $\alpha \circ X = \sum_{i=1}^{X} \xi_i$，其中 ξ_i 服从 $P(\xi_i = 1) = \alpha$ 的伯努利分布 (Steutel&van Harn,1979)。考察一种类似于 AR(1) 的模型:

$$X_t = \alpha \circ X_{t-1} + \epsilon_t \tag{1}$$

其中 ϵ_t 是同分布的离散随机变量，记其均值为 μ_ϵ，方差为 σ_ϵ^2。(1) 式即为 INAR(1) 模型的表达式，根据 Alzaid 和 Al-Osh 的文章，我们可以给出其均值和方差的表达式 (Alzaid&Al-Osh,1988)

$$\mu_X = \frac{\mu_\epsilon}{1-\alpha}, \sigma_X^2 = \frac{\sigma_\epsilon^2 + \alpha\mu_\epsilon}{1-\alpha^2} \tag{2}$$

INAR(1) 模型的优劣很大程度上取决于 ϵ_t 服从的分布，比较常见的选择是令 ϵ_t 服从泊松分布，构造 P-INAR(1) 模型。由于泊松分布具有期望等于方差的特点，故 ϵ_t 的离散指数 $I := \frac{Var(\epsilon_t)}{E(\epsilon_t)} = 1$，故泊松分布的随机项在处理过离散数据时存在缺陷。

为了解决这种缺陷，历来的研究大致都遵循两个方向，第一种常见的方向是改变细化算子的计算方式，如 Weiß(2018) 总结了数种不同的细化算子。另一种方式是改变误差项的分布来构建不同的 INAR(1) 模型。Livio(2018) 提出了一种基于 Poisson-Lindley 分布的 INAR(1) 模型，即 PL-INAR(1) 模型。这种分布具有过离散的特性，因此这种模型能够拟合过离散的数据。本文采用第二种方式来提出一个优化的 INAR(1) 模型。

为了更好的处理过离散的数据，本文尝试使用 PA 分布作为随机项构建 PA-INAR(1) 模型。PA 分布的优势在于其只有一个参数，概率函数比较简单，并且属于指数分布族。PA 分布的离散指数大于 1，这使其能够更好的估计过离散的数

据。

为了展示 PA-INAR(1) 的优点，我们将把它与 P-INAR(1) 模型和 PL-INAR(1) 模型进行比较。在 2.1 节中，我们会给出 PA 分布的定义以及性质。在 2.2 节中我们会定义出 PA-INAR(1) 模型及其性质，以便探究 PA-INAR(1) 的参数估计方法。在第 3 小节中，我们将给出三种不同的参数估计方法，分别是 Yule-Walker 估计、条件最小二乘估计以及条件极大似然估计。第 4 小节中，我们将展示三种估计方法的优劣势比较，并探究估计结果的渐近正态性。第 5 小节中，我们将利用真实数据，比较 PA-INAR(1)、P-INAR(1) 和 PL-INAR(1) 三种模型的拟合效果，以此来展示 PA-INAR(1) 的优越性。最终我们将全文的结论放在了第 6 小节。

2 PA-INAR(1) 模型

本小节中，我们先对 PA 分布的定义和性质作介绍 (Hassan et al.,2020)。之后我们引入 PA-INAR(1) 模型并展示其性质。

2.1 PA 分布

首先对于服从泊松分布，参数为 $\lambda > 0$ 的随机变量 X, 有概率质量函数：

$$p_1(x) = \frac{e^{-\lambda}\lambda^x}{x!} \tag{3}$$

其次考察 Ailamujia 随机变量的概率密度函数：

$$f_1(X = x, \alpha) = 4x\alpha^2 e^{-2\alpha x}, x \geq 0, \alpha > 0 \tag{4}$$

Theorem 1. $P(\lambda)$ 和 $AD(\alpha)$ 的复合分布有概率质量函数：

$$f_{PAD}(X = x, \alpha) = \frac{4\alpha^2(1+x)}{(1+2\alpha)^{x+2}}, x = 0, 1, 2, \ldots, \alpha > 0 \tag{5}$$

证明. 由式 (4)，该复合分布的概率质量函数有：

$$
\begin{aligned}
f_{PAD}(X=x,\alpha) &= \int_0^{\infty} p_1(x \mid \lambda) f_1(x,\lambda) d\lambda \\
f_{PAD}(X=x,\alpha) &= \int_0^{\infty} \frac{e^{-\lambda}\lambda^x}{x!} 4\lambda\alpha^2 e^{-2\alpha\lambda} d\lambda \\
f_{PAD}(X=x,\alpha) &= \frac{4\alpha^2}{x!} \int_0^{\infty} e^{-(1+2\alpha)\lambda} \lambda^{(x+2)-1} d\lambda \\
f_{PAD}(X=x,\alpha) &= \frac{4\alpha^2(1+x)}{(1+2\alpha)^{x+2}}, x=0,1,2,\ldots,\alpha>0.
\end{aligned}
$$

□

Definition 1. 称离散变量 *X* 取值于 $\mathrm{N}_0=\{0,1,2,...\}$ 服从参数为 $\lambda>0$ 的 *PA* 分布，若

$$
P(X=x)=\frac{4\lambda^2(1+x)}{(1+2\lambda)^{x+2}} \tag{6}
$$

记为 $X \sim PA(\lambda)$

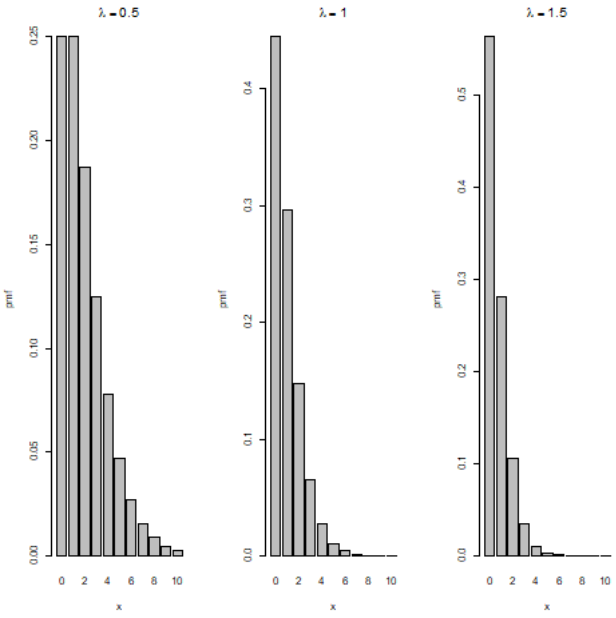

Figure 1: 不同参数值的 PA 分布

可以看到，PA 分布只有一个参数 λ，属于单参指数族分布。若 $X \sim PA(\lambda)$，则有：

$$E(X) = \frac{1}{\lambda}, Var(X) = \frac{1+2\lambda}{2\lambda^2} \tag{7}$$

由此计算出该分布的离散指数：

$$I = \frac{Var(X)}{E(X)} = \frac{2\lambda+1}{2\lambda} \tag{8}$$

可以看到 PA 分布的离散指数恒大于 1，故可以将其用于分析过离散的数据。

2.2 PA-INAR(1) 模型的定义与性质

我们根据 PA 分布可以定义如下的 PA-INAR(1) 模型：

Definition 2. 设 $\{X_t\}_{t\in \mathrm{N}_0}$ 为根据 *(1)* 式定义的 *INAR(1)* 序列，则称其为 *PA-INAR(1)* 过程，若 $\{\epsilon_t\}_{t\in \mathrm{N}_0}$ 为服从 $PA(\lambda)$ 分布的独立同分布序列。此时有：

$$X_t = \alpha \circ X_{t-1} + \epsilon_t, \epsilon_t \sim PA(\lambda) \tag{9}$$

其中 $\alpha \in (0,1), \lambda > 0, t \geq 1$，且 ϵ_t 与 ξ_i 和 X_{t-1} 独立。

由式 (7) 易知 ϵ_t 的均值与方差有限，因此 $\{X_t\}_{t\in \mathrm{N}_0}$ 为具有周期性的平稳 Markov 链 (Du&Li,1991)。转移概率为：

$$\begin{aligned} P_{ij} &= P(X_t = i | X_{t-1} = j) && (10)\\ &= P(\alpha \circ X_{t-1} + \epsilon_t = i | X_{t-1} = j) && (11)\\ &= \sum_{m=0}^{min(i,j)} P(\alpha \circ X_{t-1} = m | X_{t-1} = j) P(\epsilon_t = i - m) && (12)\\ &= \sum_{m=0}^{min(i,j)} \binom{j}{m} \alpha^m (1-\alpha)^{j-m} \frac{4\lambda^2(1+i-m)}{(1+2\lambda)^{i-m+2}} && (13) \end{aligned}$$

由此可以得到联合分布函数:

$$
\begin{aligned}
f\left(i_1, i_2 \ldots i_T\right) &= P\left(X_1 = i_1, X_2 = i_2, \ldots, X_T = i_T\right) \\
&= P\left(X_1 = i_1\right) P\left(X_2 = i_2 \mid X_1 = i_1\right) \ldots P\left(X_T = i_T \mid X_{T-1} = i_{T-1}\right) \\
&= P\left(X_1 = i_1\right) \prod_{k=1}^{T-1} \left[\sum_{m=0}^{\min(i_k, i_{k+1})} \binom{i_k}{m} \alpha^m (1-\alpha)^{i_k - m} P\left(\epsilon_{k+1} = i_{k+1} - m\right) \right].
\end{aligned} \tag{14}
$$

Theorem 2. 设 X_t 为式 *(9)* 定义的 *PA-INAR(1)* 模型，则它满足以下性质:

1. $E\left[X_t \mid X_{t-1}\right] = \alpha X_{t-1} + \mu_\epsilon = \alpha X_{t-1} + 1/\lambda$
2. $\text{Var}\left[X_t \mid X_{t-1}\right] = \alpha(1-\alpha) \cdot X_{t-1} + \sigma_\epsilon^2$
3. $\mu := E\left[X_t\right] = \frac{\mu_\epsilon}{1-\alpha} = \frac{1}{(1-\alpha)\lambda}$
4. $\sigma^2 := \text{Var}\left[X_t\right] = \frac{\sigma_\epsilon^2 + \alpha\mu_\epsilon}{1-\alpha^2} = \frac{2\alpha\lambda + 2\lambda + 1}{2(1-\alpha^2)\lambda^2}$

3 模型参数估计

在实际的模型当中，参数 α 和 λ 是未知的，因此，我们需要采取一定的方法来估计 (α, λ) 的值，在本小节中，我们将介绍三种估计参数的方法，分别是 Yule-Walk 方法，条件最小二乘法和条件极大似然估计法。

3.1 Yule-Walker 方法 (矩估计法)

首先，对任何 INAR(1) 模型，即 $X_t = \alpha \circ X_{t-1} + \epsilon_t$, 考虑他的样本自相关函数，即 $\hat{\gamma}(k) = \sum (X_t - \bar{X})(X_{t-k} - \bar{X})$, 由 INAR(1) 模型的性质，我们可以很容易得到:

$$
Cov(X_t, X_{t-1}) = \alpha \sigma_X^2 \tag{15}
$$

由此可以得到：

$$Corr(X_t, X_{t-1}) = \alpha \tag{16}$$

于是我们可以轻易得到 α 的矩估计，即：

$$\hat{\alpha}_{MM} := \hat{\rho}(1) := \hat{\gamma}(1)/\hat{\gamma}(0), \quad with\,\hat{\gamma}(k) = \sum_{t=k+1}^{T}(X_t - \bar{X})(X_{t-k} - \bar{X}) \tag{17}$$

接下来，我们根据 PA 分布的性质，由序言中提到的 (2)，我们有 $E[X_t] = \frac{\mu_\epsilon}{1-\alpha}$，再由 2.1 节中 (7),$\mu_\epsilon = 1/\lambda$, 因此，我们自然有 λ 的矩估计：

$$\hat{\lambda}_{MM} := \frac{1}{(1 - \hat{\alpha}_{MM}) \cdot \bar{X}} \tag{18}$$

由上，我们得到了 (α, λ) 的 Yule-Walker 估计, 也可以称为矩估计 (MM 估计)。

3.2 条件最小二乘估计

考虑利用 x_{t-1} 来预测 X_t 时的平方误差[1]，我们求出这个条件平方误差的累和 (conditional sum of squares,CSS)，即：

$$CSS(\alpha, \lambda) := \sum(x_t - E[X_t|x_{t-1}])^2 \tag{19}$$

我们很自然地可以将 (α, λ) 的条件最小二乘估计取为：

$$(\hat{\alpha}_{CSS}, \hat{\lambda}_{CSS}) := argmin_{\alpha,\lambda} CSS(\alpha, \lambda) = argmin_{\alpha,\lambda} \sum(x_t - E[X_t|x_{t-1}])^2 \tag{20}$$

由 2.2 节中的定理 1 可知, 在 PA-INAR(1) 模型中，$E[X_t|x_{t-1}] = \alpha \cdot x_{t-1} + \mu_\epsilon$, 于是 (20) 可以求得显式解：

[1]注意，这里大写字母 X 表示随机变量，是未知的，小写字母 x 表示样本，是已知的，之后将不做重复说明

$$\hat{\alpha}_{CLS} = \frac{\sum_{t=2}^{T} X_t X_{t-1} - \frac{1}{T-1} \cdot \sum_{t=2}^{T} X_t \cdot \sum_{t=1}^{T-1} X_t}{\sum_{t=1}^{T-1} X_t^2 - \frac{1}{T-1} \cdot (\sum_{t=1}^{T-1} X_t)^2}, \tag{21}$$

$$\hat{\mu}_{CLS} = \frac{1}{T-1}(\sum_{t=2}^{T} X_t - \hat{\alpha}_{CLS} \cdot \sum_{t=1}^{T-1} X_t), \tag{22}$$

$$where\, E[X_t|x_{t-1}] = \alpha \cdot \ x_{t-1} + \mu_\epsilon \tag{23}$$

再根据 $\mu_\epsilon = 1/\lambda$, 因此，我们可以得到 λ 的 CLS 估计:

$$\hat{\lambda}_{CLS} = \frac{1}{\hat{\mu}_{CLS}} \tag{24}$$

由上，我们得到了 (α, λ) 的条件最小二乘估计。

3.3 条件极大似然估计

和一般的极大似然估计相同，我们需要得到对数似然函数，而在 INAR(1) 模型中，模型可以视为离散的 Markov 链，具有转移概率:

$$p_{k|l} := P(X_t = k|X_{t-1} = l) = \sum_{j=0}^{min(k,l)} \binom{l}{j} \alpha^j (1-\alpha)^{l-j} P(\epsilon_t = k - j) \tag{25}$$

因此，对数似然函数可以展开写成如下形式:

$$l(\boldsymbol{\theta}) = \ln p_{x_1}(\boldsymbol{\theta}) + \sum_{t=2}^{T} \ln p_{x_t|x_{t-1}}(\boldsymbol{\theta}) \tag{26}$$

其中,后一项可以由转移概率矩阵求得,而前一项在简单的分布 (例如 Poisson 分布) 下尚可以求得，而在一些较为复杂的分布 (例如负二项分布) 下并无显式解。此时有两种方法可以解决问题，一种是利用蒙特卡洛方法求得首项的近似分布，另一种方法是在 T 较大时，首项对对数似然函数的影响可以忽略不计，我们可以选择舍弃首项，转而考虑 $l(\boldsymbol{\theta}|x_1) = \sum_{t=2}^{T} \ln p_{x_t|x_{t-1}}(\boldsymbol{\theta})$, 在本文中，我们选择第二种方法，即:

$$\hat{\theta}_{CML} = argmax_{\theta} l(\boldsymbol{\theta}|x_1) \tag{27}$$

$$= argmax_{\theta} \sum_{t=2}^{T} \ln[\sum_{j=0}^{min(x_t, x_{t-1})} \binom{x_{t-1}}{j} \alpha^j (1-\alpha)^{x_{t-1}-j} P(\varepsilon_t = x_t - j)] \tag{28}$$

$$= argmax_{\theta} \sum_{t=2}^{T} \ln[\sum_{j=0}^{min(x_t, x_{t-1})} \binom{x_{t-1}}{j} \alpha^j (1-\alpha)^{x_{t-1}-j} \frac{4(x_t - j + 1)\lambda^2}{(1+2\lambda)^{x_t - j + 2}}] \tag{29}$$

在对给定样本进行计算时，我们可以利用 BFGS 等类似方法求得 $\hat{\theta}_{CML}$。容易验证 PA-INAR(1) 过程满足 1993 年 Franke 和 Seligmann(1993) 所提出的条件 (C1)-(C6)。因此 CML 估计具有渐进正态性。

4 模拟实验

我们将采用蒙特卡洛实验的方式来比较 PA-INAR(1) 模型的三种参数估计方法的表现，其中 CML 估计将由 BFGS 拟牛顿优化算法得到最终解，YW 估计将作为优化算法的初值。模拟实验用 R 语言进行，我们选取样本数量分别为 100,250,500,1000，重复次数定为 200 次。我们将选取 α 的真值分别为 0.25,0.5 和 0.75，λ 的真值分别为 0.5 和 1.5。

在样本量 T=1000，$\alpha = 0.5, \lambda = 0.5$ 的情形下，作出 PA-INAR(1) 模型的 CLS、YW 和 CML 估计的 QQ 图，如图 2 所示。可以看到图 2 中的 6 张 QQ 图都近似为一条直线，这表明对参数的估计都是基本符合正态分布的。

表 1、表 2 展示了参数估计的数值结果，包含估计值和估计的均方误差。比较这些数据可以发现对于相同的 λ 和 T，λ 估计值的均方误差随着 α 的增加而增加，而 α 估计值的均方误差随着 α 的增加而减少。同时，对于相同的 α 和 T,λ 估计值和 α 估计值的均方误差随着 λ 的增大而增大。除此之外，可以观察到 CLS 估计值与 YW 估计值较接近。所有估计值的均方误差都会随着样本量 T 的增大而趋向于 0，其中 CML 估计值的均方误差收敛速度更快，并且在小样本情况下

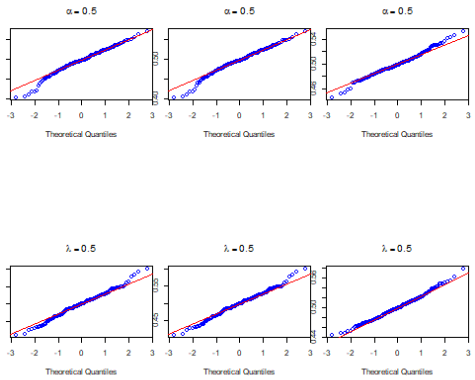

Figure 2: 取样本量为 1000 时，PA-INAR(1) 模型不同估计的 QQ 图

CML 估计值的表现显著得比 YW 估计和 CLS 估计要更好，因此可以认为 CML 估计的表现比 YW 估计、CLS 估计更好。

Table 1: 不同参数 α 和 λ 下三种估计方法对 PA-INAR(1) 模型的估计均值和均方误差

T	$\hat{\alpha}_{CLS}$	$\hat{\lambda}_{CLS}$	$\hat{\alpha}_{YW}$	$\hat{\lambda}_{YW}$	$\hat{\alpha}_{CML}$	$\hat{\lambda}_{CML}$
$(\alpha, \lambda) = (0.25, 0.5)$						
100	0.228296 (0.010816)	0.506491 (0.006358)	0.226367 (0.010744)	0.511850 (0.006646)	0.245283 (0.006025)	0.511850 (0.004573)
250	0.246170 (0.003507)	0.499989 (0.002579)	0.245259 (0.003494)	0.502129 (0.002613)	0.252768 (0.001397)	0.502383 (0.001507)
500	0.245801 (0.002103)	0.504456 (0.001532)	0.245059 (0.002094)	0.505327 (0.001545)	0.248660 (0.000983)	0.505340 (0.000997)
1000	0.249183 (0.001055)	0.501812 (0.000726)	0.248929 (0.001057)	0.502309 (0.000731)	0.248450 (0.000486)	0.500805 (0.000455)
$(\alpha, \lambda) = (0.5, 0.5)$						
100	0.479829 (0.009396)	0.508606 (0.010279)	0.474844 (0.009492)	0.513888 (0.010802)	0.501682 (0.003045)	0.520031 (0.005567)
250	0.493425 (0.003354)	0.499291 (0.004061)	0.491503 (0.003374)	0.501352 (0.004112)	0.498476 (0.001256)	0.500060 (0.001807)
500	0.491623 (0.002010)	0.498121 (0.002380)	0.490795 (0.002021)	0.499223 (0.002384)	0.496584 (0.000638)	0.500387 (0.001016)
1000	0.496039 (0.000859)	0.497456 (0.001102)	0.495631 (0.000861)	0.498055 (0.001102)	0.498862 (0.000306)	0.499186 (0.000560)

表 1 继续 – 从上一页

T	$\hat{\alpha}_{CLS}$	$\hat{\lambda}_{CLS}$	$\hat{\alpha}_{YW}$	$\hat{\lambda}_{YW}$	$\hat{\alpha}_{CML}$	$\hat{\lambda}_{CML}$
$(\alpha, \lambda) = (0.75, 0.5)$						
100	0.715371	0.466290	0.707325	0.469563	0.750470	0.504932
	(0.006135)	(0.016608)	(0.006593)	(0.016163)	(0.000834)	(0.004151)
250	0.737256	0.490166	0.733972	0.491710	0.748896	0.502939
	(0.001985)	(0.006535)	(0.002048)	(0.006539)	(0.000385)	(0.002241)
500	0.743037	0.496439	0.741599	0.497460	0.749615	0.503741
	(0.001000)	(0.003493)	(0.001022)	(0.003349)	(0.000163)	(0.000803)
1000	0.745711	0.496439	0.744944	0.494320	0.751170	0.501754
	(0.000467)	(0.010001)	(0.000475)	(0.001929)	(0.000089)	(0.000498)

Table 2: 不同参数 α 和 λ 下三种估计方法对 PA-INAR(1) 模型的估计均值和均方误差

T	$\hat{\alpha}_{CLS}$	$\hat{\lambda}_{CLS}$	$\hat{\alpha}_{YW}$	$\hat{\lambda}_{YW}$	$\hat{\alpha}_{CML}$	$\hat{\lambda}_{CML}$
$(\alpha, \lambda) = (0.25, 1.5)$						
100	0.225305	1.550517	0.223052	1.566833	0.236190	1.565066
	(0.010970)	(0.099103)	(0.010903)	(0.103753)	(0.007765)	(0.088133)
250	0.240489	1.497236	0.239638	1.503353	0.245776	1.504697
	(0.004075)	(0.030128)	(0.004083)	(0.030340)	(0.002939)	(0.026150)
500	0.247791	1.500556	0.247293	1.503537	0.248586	1.501142
	(0.002268)	(0.018076)	(0.002262)	(0.018079)	(0.001736)	(0.017124)
1000	0.245628	1.499025	0.245364	1.500454	0.248928	1.504677
	(0.001015)	(0.007665)	(0.001015)	(0.007680)	(0.000682)	(0.006202)
$(\alpha, \lambda) = (0.5, 1.5)$						
100	0.474331	1.511941	0.468896	1.524345	0.495333	1.548029
	(0.009895)	(0.106648)	(0.010007)	(0.106630)	(0.004382)	(0.073433)
250	0.491022	1.511842	0.488451	1.516765	0.498670	1.520521
	(0.004366)	(0.048074)	(0.004344)	(0.048250)	(0.001836)	(0.027012)
500	0.497484	1.505766	0.496351	1.508483	0.501531	1.510748
	(0.002207)	(0.026690)	(0.002195)	(0.265495)	(0.000982)	(0.015552)
1000	0.496946	1.498813	0.496325	1.500112	0.499899	1.504558
	(0.001038)	(0.010999)	(0.001033)	(0.010985)	(0.000468)	(0.006855)
$(\alpha, \lambda) = (0.75, 1.5)$						
100	0.718715	1.447453	0.711114	1.462609	0.747694	1.530603
	(0.006507)	(0.192145)	(0.007129)	(0.198229)	(0.001655)	(0.076202)
250	0.736742	1.479655	0.733918	1.487987	0.747998	1.518818
	(0.002154)	(0.064296)	(0.002335)	(0.067228)	(0.000630)	(0.031292)
500	0.741909	1.482987	0.740589	1.487723	0.749565	1.512520
	(0.001060)	(0.035986)	(0.001102)	(0.036661)	(0.000309)	(0.013565)
1000	0.745846	1.499359	0.745046	1.500918	0.749440	1.509439
	(0.000665)	(0.022834)	(0.000685)	(0.023150)	(0.000151)	(0.006341)

5 真实数据案例

在本小节中，将利用 PA-INAR(1) 模型对真实数据进行拟合，并与 P-INAR(1)，PL-INAR(1) 进行比较。真实数据集来自香港地区强制公司清盘案和破产案的每月聆讯数目。数据可在https://www.oro.gov.hk/cht/statistics/compulsory_winding_up_and_bankruptcy/stat.php查到。我们选取了自 1998 年 1 月至 2003 年 12 月的数据，共 72 个数据点。

图3中展示了数据的时序图、ACF 图和 PACF 图。我们可以从 PACF 图中发现，样本基本满足偏相关函数一阶截尾，符合 INAR(1) 的性质。我们对数据运用 Ljung-Box 检验后，p 值小于 2.2×10^{-16}, 远远小于 0.05，这意味着样本存在某些相关性。根据 PACF 图的结果，样本适合用 AR(1) 类型模型进行拟合。

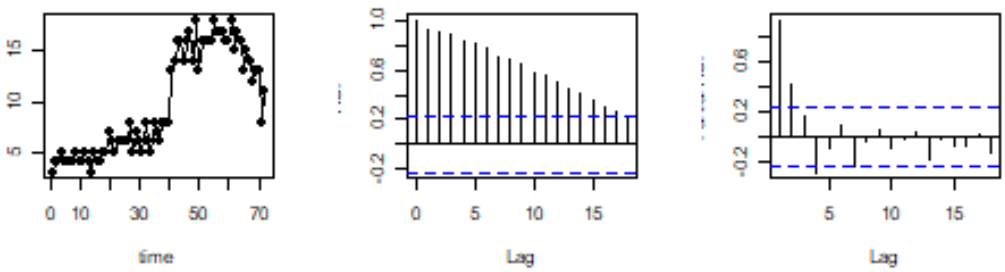

Figure 3: 数据的时序图、ACF 图和 PACF 图

根据计算样本均值 $\bar{X} = 9.6944$, 样本方差 $S_X^2 = 26.4969$，我们可以计算得到离散指数 $\bar{I}_X = S_X^2/\bar{X} = 2.7332$。因此样本是过离散的。

作为比较，我们计算了 PA-INAR(1),P-INAR(1) 和 PL-INAR(1) 的拟合参数的 CML 估计，以及 AIC 和 BIC。其中每个模型的 CML 估计均由 BFGS 拟牛顿优化算法得到最终解，以各自的 YW 估计作为初值。并且我们另外选用了 2004 年 1 月至 12 月的数据来诊断模型的预测误差。预测值取由 $E(X_{t+k}|X_t) = \alpha^k X_t + \mu_\epsilon \frac{1-\alpha^k}{1-\alpha}$ 给出。最终结果呈现在表3中。

从表3的结果可以看出，PA-INAR(1) 在 AIC 准则和 BIC 准则下都优于 P-INAR(1) 和 PL-INAR(1)。另外，我们发现 PL-INAR(1) 的预测结果的样本均值和

Table 3: 不同模型的估计参数与预测效果

模型	参数	CML 估计	AIC	BIC	μ_X	σ_X^2	MSE
PA-INAR(1)	α	0.8764	284.71	289.27	9.3724	0.3203	4.0082
	λ	0.7643					
P-INAR(1)	α	0.8586	286.80	291.35	9.4224	0.3155	4.0172
	λ	1.4806					
PL-INAR(1)	α	0.8827	285.41	289.96	9.3693	0.3287	4.0291
	θ	1.1675					
样本					9.0833	2.9924	

样本方差更接近真实值，不过 PA-INAR(1) 与其的差距也不到 3%，并且三个模型拟合出的样本均值和样本方差都离真实值较远，参考价值不大。值得一提的是，PA-INAR(1) 具有更小的均方误差。综合上述内容，PA-INAR(1) 是此样本的最佳拟合模型。

6 结论

本文提出了一种基于二项细化算子，以 PA 分布作为随机项的 INAR(1) 模型。基于 PA 分布的过离散性质，新提出的 PA-INAR(1) 模型更适合分析过离散数据。我们讨论了 PA-INAR(1) 模型的基础性质，包括转移概率、条件期望、条件方差等等，列举了参数估计的三种方法，包括条件极大似然估计 (CML 估计)、条件最小二乘估计 (CLS 估计)、Yule-Walker 估计法 (YW 估计)。这三种方法的估计结果都基本符合正态分布，且 CML 估计的均方误差收敛更快，这显示 CML 方法是估计参数的最优方法。在最后的真实数据案例中，通过比较 AIC,BIC 以及模型拟合得到的样本均值和样本方差，可以认为 PA-INAR(1) 是该样本的最佳拟合模型。

尽管还有很多过离散的分布可以作为随机项来构造 INAR(1) 模型，但 PA 分布的优势在于其只有一个参数，概率函数比较简单，并且属于指数分布族。这些优点使得 PA-INAR(1) 模型在分析过离散数据时有着较高的应用价值。本文还有许多课题需要进一步研究，例如零截断 PA 分布等 PA 分布的拓展，这为我们今后研究相关的 INAR 模型提供了思路。

参考文献

[1] Alzaid, A., & Al-Osh, M. (1988). First-order integer-valued autoregressive (INAR (1)) process: distributional and regression properties. Statistica Neerlandica, 42(1), 53-61.

[2] Franke, J., & Seligmann, T. H. (1993). Conditional maximum likelihood estimates for INAR (1) processes and their application to modelling epileptic seizure counts. Developments in time series analysis, 310-330.

[3] Hassan, A., Shalbaf, G. A., Bilal, S., & Rashid, A. (2020). A new flexible discrete distribution with applications to count data. Journal of Statistical Theory and Applications, 19(1), 102-108.

[4] Jin-Guan, D., & Yuan, L. (1991). The integer-valued autoregressive (INAR (p)) model. Journal of time series analysis, 12(2), 129-142.

[5] Lívio, T., Khan, N. M., Bourguignon, M., & Bakouch, H. S. (2018). An INAR (1) model with Poisson–Lindley innovations. Econ Bull, 38(3), 1505-1513.

[6] McKenzie, E. (1985). Some simple models for discrete variate time series 1. JAWRA Journal of the American Water Resources Association, 21(4), 645-650.

[7] Steutel, F. W., & van Harn, K. (1979). Discrete analogues of self-decomposability and stability. The Annals of Probability, 893-899.

[8] Weiß, C. H. (2018). An introduction to discrete-valued time series. John Wiley & Sons.

基于 PA 分布的 INAR(1) 模型分析 (代码)

李天恺　　林瑞洋　　董宇坤

目录

1　写在前面 **1**

2　代码展示 **1**

3　分工情况 **9**

1　写在前面

该部分内容为正文的附加内容，主要作用为展示正文中的代码函数，但展示内容不包括作图代码，具体循环结构 (例如第 4 节中模拟实验当中的循环部分) 等复杂的部分。文中会附带对每部分代码的简略解释，所有代码都已经过测试。

2　代码展示

```
## 返回 PA 分布的 cdf 值
pad_cdf <- function(x,lambda){
  return(1-(4*lambda+2*lambda*x+1)/((1+2*lambda)^(x+2)))
}
```

```r
## 生成服从 PA 分布的随机数，该函数主要用于第 4 节中的模拟实验
pad_sim <- function(n,lambda){
  u <- runif(n)
  x <- rep(0,n)
  for (i in 1:n) {
    k <- 0
    repeat{
      if(u[i] < pad_cdf(k,lambda)) break
      k <- k + 1
    }
    x[i] <- k
  }
  return(x)
}

## 细化算子
multi <- function(alpha,x){
  x_new <- sum(sample(0:1,x,prob = c((1-alpha),alpha),replace = T))
  return(x_new)
}

## 生成算法 (可以生成 PA-INAR(1) 和 P-INAR(1))
inar_sim <- function(n = 100, alpha = 0.5, lambda1 = 1
                     , type = "PAD",initial = 0){
  x <- rep(0,n)
  x[1] <- initial
  if(type == "Poisson"){
    eps <- rpois(n,lambda1)
  }else if (type=="PAD") {
    eps <- pad_sim(n,lambda1)
  }
  for (t in 2:n) {
      x[t] <- multi(alpha,x[t-1]) + eps[t]
```

```
  }
  return(x)
}

##Yule-Walker 方法
pad_yw <- function(x){
  x_bar <- mean(x)
  n <- length(x)
  gam0 <- sum((x-x_bar)^2)
  gam1 <- 0
  for (t in 2:n) {
    gam1 <- gam1 + (x[t] - x_bar) * (x[t-1] - x_bar)
  }
  alpha <- gam1/gam0
  lambda <- 1/((1-alpha)*x_bar)
  return(c(alpha,lambda))
}

##Conditional Least Squares Estimation
pad_cls <- function(x){
  n <- length(x)
  alpha <- (sum(x*c(0,x[-n]))-
              sum(x[-1])*sum(x[-n])/(n-1))/(sum((x[-n])^2)-(sum(x[-n]))^2/(n-1))
  mu <- 1/(n-1)*(sum(x[-1])-alpha*sum(x[-n]))
  lambda <- 1/(mu)
  return(c(alpha,lambda))
}

##Conditional Maximum Likelihood Estimation

## 对数条件似然函数
pad_loglh <- function(parameters,data){
  x <- data
```

```
  n <- length(x)
  loglh <- 0
  alpha <- pmin(pmax(parameters[1], .Machine$double.eps),
                1-(.Machine$double.eps))
  lambda <- pmax(parameters[2], .Machine$double.eps)
  for (t in 2:n) {
    u <- 0
    for (j in 0:(min(x[t],x[t-1]))) {
      u <- u + choose(x[t-1],j) * (alpha^j) * ((1-alpha)^(x[t-1]-j))
      * 4 * lambda^2 * (1+x[t]-j)/((1+2*lambda)^(x[t]-j+2))
    }
    loglh <- loglh + log(u)
  }
  return(-loglh)
}

## 条件极大似然估计
pad_ml <- function(x){
  theta_initial <- pad_yw(x)
  result <- optim(par = theta_initial,fn = pad_loglh,data = x,method = "BFGS")
  theta <- result$par
  return(c(pmin(pmax(theta[1], 0), 1), theta[2]))
}

# 模拟实验一次循环展示
#thm_a 和 thm_t 用于控制初值
set.seed(123)
times <- c(100,250,500,1000)
n <- 200
alpha_yw <- rep(0,4)
theta_yw <- rep(0,4)
alpha_cls <- rep(0,4)
theta_cls <- rep(0,4)
```

```
alpha_ml <- rep(0,4)
theta_ml <- rep(0,4)
err <- matrix(0, nrow = 4, ncol = 6)
a_yw <- rep(0,n)
t_yw <- rep(0,n)
a_cls <- rep(0,n)
t_cls <- rep(0,n)
a_ml <- rep(0,n)
t_ml <- rep(0,n)
thm_a <- 0.75
thm_t <- 1.5
for (i in 1:4) {
  for (j in 1:n) {
    x <- inar_sim(n = times[i],alpha = thm_a,lambda1 = thm_t,
                  type = "PAD",initial = 0)
    yw <- pad_yw(x)
    cls <- pad_cls(x)
    ml <- pad_ml(x)
    a_yw[j] <- yw[1]
    t_yw[j] <- yw[2]
    a_cls[j] <- cls[1]
    t_cls[j] <- cls[2]
    a_ml[j] <- ml[1]
    t_ml[j] <- ml[2]
  }
  alpha_yw[i] <- mean(a_yw)
  err[i,1] <- mean((a_yw-thm_a)^2)
  theta_yw[i] <- mean(t_yw)
  err[i,2] <- mean((t_yw-thm_t)^2)
  alpha_cls[i] <- mean(a_cls)
  err[i,3] <- mean((a_cls-thm_a)^2)
  theta_cls[i] <- mean(t_cls)
  err[i,4] <- mean((t_cls-thm_t)^2)
```

```
  alpha_ml[i] <- mean(a_ml)
  err[i,5] <- mean((a_ml-thm_a)^2)
  theta_ml[i] <- mean(t_ml)
  err[i,6] <- mean((t_ml-thm_t)^2)
}


## 真实数据案例代码
#poisson 对数似然函数
poi_loglh <- function(parameters,data){
  x <- data
  n <- length(x)
  loglh <- 0
  alpha <- pmin(pmax(parameters[1], .Machine$double.eps),
                1-(.Machine$double.eps))
  lambda <- pmax(parameters[2], .Machine$double.eps)
  for (t in 2:n) {
    u <- 0
    for (j in 0:(min(x[t],x[t-1]))) {
      u <- u + choose(x[t-1],j) * (alpha^j) * ((1-alpha)^(x[t-1]-j)) *
        exp(-lambda) * (lambda^(x[t]-j))/factorial((x[t]-j))
    }
    loglh <- loglh + log(u)
  }
  return(-loglh)
}

##poisson lindley 对数似然函数
pl_loglh <- function(parameters,data){
  x <- data
  n <- length(x)
  loglh <- 0
  alpha <- pmin(pmax(parameters[1], .Machine$double.eps),
```

```
                1-(.Machine$double.eps))
  theta <- pmax(parameters[2], .Machine$double.eps)
  for (t in 2:n) {
    u <- 0
    for (j in 0:(min(x[t],x[t-1]))) {
      u <- u + choose(x[t-1],j) * (alpha^j) * ((1-alpha)^(x[t-1]-j)) *
        theta^2 * (theta + 2 + x[t]-j)/(theta + 1)^(x[t]-j+3)
    }
    loglh <- loglh + log(u)
  }
  return(-loglh)
}

##poisson Yule-Walker 估计
poi_yw <- function(x){
  x_bar <- mean(x)
  n <- length(x)
  gam0 <- sum((x-x_bar)^2)
  gam1 <- 0
  for (t in 2:n) {
    gam1 <- gam1 + (x[t] - x_bar) * (x[t-1] - x_bar)
  }
  alpha <- gam1/gam0
  lambda <- (1-alpha)*x_bar
  return(c(alpha,lambda))
}

##poisson lindley Yule-Walker 估计
pl_yw <- function(x){
  x_bar <- mean(x)
  n <- length(x)
  gam0 <- sum((x-x_bar)^2)
  gam1 <- 0
```

```
  for (t in 2:n) {
    gam1 <- gam1 + (x[t] - x_bar) * (x[t-1] - x_bar)
  }
  alpha <- gam1/gam0
  theta <- 2/((1-alpha)*x_bar)
  return(c(alpha,theta))
}



##aic 和 bic 计算
ic_cal <- function(x,type = "UPA"){
  if(type=="PAD"){
    theta_initial <- pad_yw(x)
    result <- optim(par = theta_initial,fn = pad_loglh,data = x,method = "BFGS")
  }else if(type=="Poisson"){
    theta_initial <- poi_yw(x)
    result <- optim(par = theta_initial,fn = poi_loglh,data = x,method = "BFGS")
  }else if(type=="PL"){
    theta_initial <- pl_yw(x)
    result <- optim(par = theta_initial,fn = pl_loglh,data = x,method = "BFGS")
  }

  n <- length(x)
  log_value <- - result$value
  par <- result$par
  aic <- -2 * log_value + 4
  bic <- -2 * log_value + log(n) * 2
  return(c(aic,bic,par))
}
```

3 分工情况

董宇坤主要负责数据收集、协助其余二人工作。

林瑞洋主要负责框架构建、论文撰写。

李天恺主要负责代码实现。

三人均参与文献查找等工作。

总的来说三人工作量基本相等。