

# Threshold ARMA Model

and its application of Gold price

郑永健, 杨永康, 周文哲, 余潇然

PB20151774, University of Science of Technology of China

PB20151801, University of Science of Technology of China

PB20151807, University of Science of Technology of China

PB20151803, University of Science of Technology of China

January 14, 2023

---

## Abstract

**Abstract** 在本文中我们分析了一种新的非线性时序模型: threshold autoregressive moving-average models (TARMA 模型). TARMA 模型是对我们熟知的门限自回归模型 (TAR) 的推广。

在线性时序分析中, ARMA 作为 AR 的推广, 使得模型具有了 Parsimonious (简洁性)。类似地, TARMA 模型也推广了 TAR 模型, 于是我们也认为模型具有了简洁性。为了运用 TARMA 模型我们需要解决三个问题: 一是 TARMA 模型的概率模型以及参数估计, 对此我们复现了 Greta Goracci 给出的一种鲁棒估计; 二是为什么是 TARMA 而不是 ARMA, 对此我们复现了 Greta Goracci, Kung-Sik Chan, Howell Tong 给出的一种检验门限效应的方法 Supremum Lagrange Multiplier Test; 三是为什么是 TARMA 而不是 TAR, 对此我们对已有的 TAR 实证研究重新应用 TARMA 模型建模表明其确实有简洁性。最后我们在大宗商品黄金价格上应用 TARMA 模型建模预测并解释结果。

---

郑永健: 想法文献综述

杨永康: *SupLM Test* 代码

周文哲: *Robust Parameter Estimation of TARMA* 代码

余潇然: TARMA 基本函数代码, 包括但不限于模拟、预测、误差分析

**任何疑问请联系** 郑永健 E-mail dress: [cokezyj@mail.ustc.edu.cn](mailto:cokezyj@mail.ustc.edu.cn)

**Keywords:** Threshold Autoregressive Moving-Average models; Lagrange multiplier test; Application;

## 1. Introduction

门限模型 (Threshold Model) 是一种著名的非线性时序模型, 用于描述许多领域的复杂现象, 像经济学、金融、生态学、流行病学 (Tong, 1990[1]; Chan, 2009[2]; Giordani 等, 2007[3]; Tong, 2011[4]; Hansen, 2011[5]; Chan 等人, 2017[6])。非线性是由门限机制 (Threshold Mechanism) 引入的, 这使得模型能够描述复杂的非线性特征 (non-linear dynamic features), 例如跳跃 (jump)、极限环 (limit cycle)、时间不可逆性 (time-irreversibility), 同时保持良好的可解释性。自 Tong (1978)[7] 提出以来, 门限模型得到了广泛的研究与实证应用, 尤其是 AR 推广, 即所谓的 TAR 模型。TARMA 模型通过在不同区域包含 MA 部分来扩展 TAR 模型 (Tong, 2015) [8] 因为由于非马尔可夫性质, TARMA 在技术上比 TAR 模型更具挑战性, 对于 TARMA 理论的研究一直到最近由 Chan 和 Goracci (2019) [9] 解决了关于一阶 TARMA 模型的概率结构才取得进展。TARMA 模型拥有许多优越的功能: 它们包括 MA 部分; 它们自然会考虑测量误差; 它们能够描述一类广泛的模型, 从瞬态到遍历性, 甚至几何遍历性的长期概率行为。

此外, 门限模型提供了一种平凡的方式来描述一些看起来本应该是随机游走的序列, 比如著名的 Purchasing Power Parity Puzzle (PPP puzzle, 购买力平价难题)。Li 和 Li (2011)[10] 以及 Goracci(2021)[11] 等人开发了数据序列的门效应检验, 这就是将在后文中介绍的 SupLM Test。此外还可以类比 ARMA-GARCH 模型, 引入 GARCH 成为 TARMA-GARCH, 可处理具有条件异方差性的序列的条件均值。与更简单的 TAR 模型相比, TARMA 模型的估计具有挑战性, 因为缺乏以门限参数为条件的线性参数化 (lack of a linear parameterization conditional on the threshold parameter)。目前的方法大多依赖最小二乘 (LS) 方法 (Li 和 Li, 2011)[10], 该方法严重受异常值和重尾的影响。但是黑天鹅般的异常值在实证数据中经常出现 (Giordani, 2007 年) [3], 无论从理论建模还是实际复现, TARMA 模型的稳健估计和异常值检测问题尚未得到解决。另一方面, ARMA 模型的 Robust Estimation 已得到广泛研究。对于 TAR 模型, Chan 和 Cheung(1994)[12] 研究了 LS 估计中附加异常值的影响并提出 Generalized-M 估计以减轻估计的严重偏差。

于是本文复现的 Robust Estimation of TARMA 也是基于一种 Generalized-M Estimator。G.Goracci 的方法包括用一个有界导数的函数  $\rho(z)$  替换 RSS Criterion of Li et.(2011)[10]。Goracci 证明了, 对于不同类型异常值的估计。参数显示为强一致且渐近正态规律性条件。且与最小二乘估计器类似, 发现  $r$  参数是  $n^{-1}$  的收敛速度, 而 AR 和 MA 参数是根  $n$  一致的并且渐近正态。该方法是 minimized 特殊的目标函数  $\rho$ , 当  $\alpha = 0$  时, 估计变为 LS 估计。通过分析一组商品价格时间序列来展示这种新方法的性能。预测的商品与许多宏观经济变量之间的关系通常是非线性且不对称的, 这取决于价格是上涨还是下跌, 例如参见 Kilian 和 Vigfusson (2011)[13] 和 Kilian 和 Vigfusson(2013)[14]。尽管 TARMA 模型似乎适合此类数据, 但重尾、高杠杆点和非高斯新息使得最小二乘估计不可取。稳健的 TARMA 估计与非稳健估计相比, 通常提供更好的拟合和更小的标准误差。我们稳健的 TARMA 模型确认存在有门限区分的两种模型对应于缓慢、持续的增长和快速收缩。

在之后, 为了检验门效应是否存在, 我们复现了 Supremum Lagrange Multiplier Test(Goracci,2021)[11], 这个检验具有很好的效果: 很低的计算负担; 模型阶数鲁棒性; 小样本也成立 (Goracci,2021)[11]。他以我们常用的 ARMA 模型作为零假设, 构建类似 Score 检验的统计量进行检验, 并给出了拒绝域。

最后我们在黄金价格上使用 TARMA 模型建模, 用 SupLM 检验确实具有门效应, 并且分别用 Robust Estimation 跟 Least Square Estimation 还有 ARMA 模型建模, 进行预测并计算 MAE。

## 2. TARMA Model

门限自回归模型是一类可以描述许多动力学现象的非线性时间序列模型。Tong 和 Lim[15] 提出了门限自回归 (TAR) 模型, 该模型由以下差分方程定义:

$l - TAR(p)$ :

$$X_t = \begin{cases} \phi_{1,0} + \sum_{i=1}^p \phi_{1,i} X_{t-i} + \epsilon_{1,t} & \text{if } X_{t-d} \leq r_1 \\ \phi_{2,0} + \sum_{i=1}^p \phi_{2,i} X_{t-i} + \epsilon_{1,t} & \text{if } r_1 < X_{t-d} \leq r_2 \\ \dots\dots\dots \\ \phi_{l,0} + \sum_{i=1}^p \phi_{l,i} X_{t-i} + \epsilon_{1,t} & \text{if } X_{t-d} > r_{l-1} \end{cases} \quad (1)$$

类似从 AR 到 ARMA, 门限自回归移动平均 (TARMA) 模型由以下差分方程定义:

$l - TARMA(p, q)$ :

$$X_t = \begin{cases} \phi_{1,0} + \sum_{i=1}^p \phi_{1,i} X_{t-i} + \epsilon_{1,t} + \sum_{j=1}^q \theta_{1,j} \epsilon_{1,t-j} & \text{if } X_{t-d} \leq r_1 \\ \phi_{2,0} + \sum_{i=1}^p \phi_{2,i} X_{t-i} + \epsilon_{1,t} + \sum_{j=1}^q \theta_{2,j} \epsilon_{1,t-j} & \text{if } r_1 < X_{t-d} \leq r_2 \\ \dots\dots\dots \\ \phi_{l,0} + \sum_{i=1}^p \phi_{l,i} X_{t-i} + \epsilon_{1,t} + \sum_{j=1}^q \theta_{l,j} \epsilon_{1,t-j} & \text{if } X_{t-d} > r_{l-1} \end{cases} \quad (2)$$

其中: (i)  $l$  是门限区间 (regime) 个数; (ii)  $p$  和  $q$  分别是自回归与滑动平均的阶数; (iii)  $\phi_{k,i}$  和  $\theta_{k,j}$ ,  $k = 1, \dots, l$ ,  $i = 1, \dots, p$ ,  $j = 1, \dots, q$  分别是自回归系数和滑动平均系数; (iv)  $d \in \mathbb{N}$  是滞后步数; (v)  $r_1 < r_2 < \dots < r_{l-1}$  是阈值; (vi)  $\epsilon_{k,t}$ ,  $k = 1, \dots, l$  是误差项 (新息序列)。

TAR, TMA 和 TARMA 是我们熟知的 AR, MA, ARMA 模型的推广, 门限模型允许自回归与滑动平均系数随着门限区间变化。目前 TAR 的研究较为全面, 并被用于开发统计检验, 以检测非线性和非平稳性的存在。如果时间序列受到测量误差的影响, 那么 TAR 过程自然会变成 TARMA 过程。由于测量误差不可避免, 因此 TARMA 比纯粹的 TAR 更符合实际, 对 TARMA 的研究是十分必要的。

TARMA 模型存在阈值, 滑动平均项都是非线性的, 这使得 TAR 的理论不能简单地套用在 TMA 和 TARMA 上。关于 TARMA 参数的估计, 据前人研究可知进行最大似然估计是没有结果的 [10]。DONG Li(2011) 等人 [10] 研究了 TARMA 的最小二乘估计, 并建立了阈值自回归模型和移动平均模型的渐近理论。在此基础上, Goracci G 等 (2022)[16] 于提供了 TARMA 的 robust 估计 (robust M-estimator) 的第一个理论框架, 并通过 Monte Carlo 方法表明 robust 估计在偏差和方差方面都优于最小二乘估计。他们也通过实例表明 robust 估计具有较小的标准误差和更好的预测精度。

TARMA 模型在分析不同领域的非线性动力学方面具有巨大的潜力。例如, Chan 等人 (2020) [17] 使用 TARMA 模型来解决购买力平价难题; Goracc (2020) [18] 分析了太阳黑子

数量和美国男性失业率时间序列，而 Goracci and Giannerini (2021) [11] 将它们用于树突学分析。结果表明，与文献中提出的其他模型相比，TARMA 模型可以提供更好、更简洁的拟合。

### 3. SupLM Test

目前，为了研究特定时间序列是否更符合 TARMA 模型而不是 ARMA 模型，即是否具有门限效应，已经发展出了许多检验方法。Li 和 Li (2011) [10] 发展了一种准似然比统计量来测试 ARMA 过程中的门限效应。Goracci 等人 (2021) [11] 提出一种通过拉格朗日乘子检验测试阈值效应的方法，即 SupLM Test。与准似然比检验相比，SupLM Test 不需要估计备择假设下的模型，为实际操作提供了便利。

为记号方便，令 TARMA 序列  $\{X_t : t = 0, \pm 1, \pm 2, \dots\}$  定义如下：

$$\begin{aligned} X_t = & \phi_{10} + \sum_{k=1}^p \phi_{1k} X_{t-k} + \epsilon_t - \sum_{s=1}^q \theta_{1s} \epsilon_{t-s} \\ & + (\psi_{10} + \sum_{k=1}^p \psi_{1k} X_{t-k}) I(X_{t-d} \leq r) \end{aligned} \quad (3)$$

其中新息  $\{\epsilon_t\}$  为独立同分布，均值为 0、方差有限为  $\sigma^2$  的随机变量，且  $\epsilon_t$  与  $X_{t-1}, X_{t-2}, \dots$  均独立。p,q,d 分别为 AR 阶数，MA 阶数，延迟参数，假设 p,q,d 均已知。I(·) 为示性函数。

令

$$\boldsymbol{\phi} = (\phi_{10}, \phi_{11}, \dots, \phi_{1p})^T; \quad \boldsymbol{\theta} = (\theta_{11}, \dots, \theta_{1q})^T; \quad \boldsymbol{\psi} = (\psi_{10}, \psi_{11}, \dots, \psi_{1p})^T$$

$$\boldsymbol{\zeta} = (\boldsymbol{\phi}^T, \boldsymbol{\theta}^T)^T$$

检验模型中的阈值效应，即对如下假设进行检验：

$$\begin{cases} H_0 & : \boldsymbol{\psi} = \mathbf{0} \\ H_1 & : \boldsymbol{\psi} \neq \mathbf{0} \end{cases}$$

检验模型中 AR 部分的阈值效应的统计量记为 sLM。为了得到检验的渐进性质，需要假设模型在零假设和备择假设下都是遍历和可逆的。假设观测数据为  $X_1, \dots, X_n$ ，基于初值  $X_0, \dots, X_{-p+1}$  的高斯条件似然函数为：

$$l = -\frac{n}{2} \log(\sigma^2 2\pi) - \frac{1}{2\sigma^2} \sum_{t=1}^n \epsilon_t^2$$

形式的，我们有：

$$\begin{aligned} \epsilon_t = & X_t - \left\{ \phi_{10} + \sum_{k=1}^p \phi_{1k} X_{t-k} - \sum_{s=1}^q \theta_{1s} \epsilon_{t-s} \right\} \\ & - \left\{ \psi_{10} + \sum_{k=1}^p \psi_{1k} X_{t-k} \right\} I(X_{t-d} \leq r) \end{aligned}$$

并设  $\epsilon_0, \dots, \epsilon_{-q+1}$  为 0。

令

$$I_n(r) = \begin{pmatrix} I_{n,11} & I_{n,12}(r) \\ I_{n,12}^T(r) & I_{n,22}(r) \end{pmatrix} = \begin{pmatrix} \frac{\partial^2 l}{\partial \xi \partial \xi^T} & \frac{\partial^2 l}{\partial \xi \partial \psi^T} \\ \frac{\partial^2 l}{\partial \psi \partial \xi^T} & \frac{\partial^2 l}{\partial \psi \partial \psi^T} \end{pmatrix}$$

在零假设下, 模型退化为 ARMA 模型, 此时可以通过极大似然估计的方法得到模型参数的估计。我们令  $\frac{\partial \hat{l}}{\partial \psi}(r)$  和  $\hat{I}_n(r)$  分别为  $\frac{\partial l}{\partial \psi}(r)$  和  $I_n(r)$  的估计。由此得到拉格朗日乘子统计量

$$T_n(r) = \left( \frac{\partial \hat{l}}{\partial \psi} \right)^T \{ \hat{I}_{n,22}(r) - \hat{I}_{n,21}(r) \hat{I}_{n,11}^{-1} \hat{I}_{n,12}(r) \}^{-1} \left( \frac{\partial \hat{l}}{\partial \psi} \right)$$

此时,  $T_n(r)$  为阈值参数  $r$  的函数, 为了解决  $r$  缺失的问题, 令  $R = [r_L, r_U]$ ,  $r_L$  和  $r_U$  分别为数据的分位数, 例如: 25% 和 75% 分位数, 对所有  $r \in R$ , 计算  $T_n(r)$ , 取上确界为 sLM 统计量。即 sLM 统计量为

$$T_n = \sup_{r \in [r_L, r_U]} T_n(r)$$

对于 AR 阶数 1-4, MA 阶数 1-2 的 TARMA 模型, Goracci 等人 (2021) [11] 给出了 sLM 统计量渐进零分布分别在 90%, 95%, 99%, 99.9% 水平下的经验分位数表格 1。

**Table 1.** Tabulated quantiles for the asymptotic null distribution of the supLM statistics for the threshold range 25th-75th percentiles. The first two columns denote the AR and the MA orders, respectively.

		sLM statistic			
AR	MA	90%	95%	99%	99.90%
1	1	9.61	11.37	15.19	20.38
2	1	11.53	13.41	17.22	22.17
3	1	13.74	15.71	19.98	25.04
4	1	15.65	17.68	22.25	27.44
1	2	9.64	11.47	15.50	20.25
2	2	11.71	13.48	17.61	22.49
3	2	13.46	15.35	19.33	25.06
4	2	15.55	17.58	21.82	27.80

关于 sLM 的渐进零分布的理论在附录做了简单说明。

#### 4. Estimation for TARMA Model

与此前的 TAR 模型相比, 在尝试对 TARMA 模型进行估计时会更具有挑战性。由于缺乏了在门限参数已知的条件下对 TARMA 各项的线性参数化方法, 对 TARMA 各参数的估计并不容易。为此, 本文将从最小二乘方法开始进行推广, 尝试构造出 TARMA 模型各项参数的 Robust 估计, 从而一定程度上解决 LS 估计受异常值影响较大的问题, 提高 TARMA 参数估计的稳健性。

#### 4.1. LS Estimation for TARMA Model

设  $\{X_t\}_{t \in \mathbb{Z}}$  为一个 TARMA 过程，具体由下述方程定义

$$X_t = \begin{cases} \phi_{1,0} + \sum_{i=1}^p \phi_{1,i} X_{t-i} + \epsilon_t + \sum_{j=1}^q \theta_{1,j} \epsilon_{t-j} & X_{t-d} \leq r \\ \phi_{2,0} + \sum_{i=1}^p \phi_{2,i} X_{t-i} + \epsilon_t + \sum_{j=1}^q \theta_{2,j} \epsilon_{t-j} & X_{t-d} > r \end{cases} \quad (4)$$

其中,  $p \in \mathbb{N}$  和  $q \in \mathbb{N}$ , 分别是自回归和移动平均的阶数;  $\phi$  和  $\theta$  分别为自回归系数和移动平均系数;  $1 \leq d \leq D_0$  为滞后阶数,  $D_0 \in \mathbb{D} \subset \mathbb{N}$ ;  $r \in R \subset \mathbb{R}$  为门限参数, 记为  $\lambda$ ; 记  $\theta = (\lambda, r, d)$ ;  $\{\epsilon_t\}$  为该过程的新息过程, 有  $E(\epsilon_t) = 0$ ,  $E(\epsilon_t^2) = \sigma^2 < \infty$ 。一般而言, 该新息过程通常可被假设为高斯白噪声分布。显然, 当门限参数  $r$  的绝对值趋向于无穷时, (即  $|r| \rightarrow \infty$ ), 原 TARMA 模型将会变成一个线性的 ARMA 模型。

由这一定义方程可看出, 整个 TARMA 模型被分为两个分面, 为方便讨论, 称其为下分面和上分面, 分别对应了  $X_{t-d} \leq r$  和  $X_{t-d} > r$  的两种情况, 而上下分面有着不同的参数。对于 LS 估计而言, 可以取损失函数为  $L_n(\theta) = \sum_{t=1}^n \epsilon_t^2(\theta)$ , 其中  $\epsilon_t^2(\theta)$  为新息函数, 即:

$$\begin{aligned} \epsilon_t(\theta) = X_t - & \left( \phi_{1,0} + \sum_{i=1}^p \phi_{1,i} X_{t-i} + \epsilon_t + \sum_{j=1}^q \theta_{1,j} \epsilon_{t-j} \right) I(X_{t-d} \leq r) \\ & - \left( \phi_{2,0} + \sum_{i=1}^p \phi_{2,i} X_{t-i} + \epsilon_t + \sum_{j=1}^q \theta_{2,j} \epsilon_{t-j} \right) I(X_{t-d} > r) \end{aligned} \quad (5)$$

想要最小化这样的损失函数, 最大的困难在于其关于  $r$  和  $d$  的不连续性。为了解决这样的问题, 尝试使用 plug-in 的方法:

第一步: 对一组给定的  $r$  和  $d$ , 此时损失函数关于  $\lambda$  是连续的, 可求得此时使得损失函数最小的  $\lambda_n$ ;

第二步: 利用第一步中求得的  $\lambda_n$ , 因为  $r$  和  $d$  均只能取有限的值, 故而通过遍历的方式, 不难得到使得损失函数取得最小的  $r$  和  $d$ ;

第三步: 由第二步中获得的最优的  $r$  和  $d$ , 重复第一步, 可求得最优的  $\lambda_n$  和  $\theta_n$ , 至此, 即可最小化目标的损失函数, 并估计出 TARMA 模型的各项参数。

#### 4.2. Robust Estimation for TARMA Model

设  $\{X_t\}_{t \in \mathbb{Z}}$  为一个 TARMA 过程, 定义同 5.1 中, 不难知道, 该模型有着相同的新息函数  $\epsilon_t^2(\theta)$ 。但在 Robust 估计中, 取目标函数为

$$\rho_n(\theta) = \sum_{t=1}^n \rho\left(\frac{\epsilon_t^2(\theta)}{\sigma}\right)$$

其中,  $\rho$  为一个损失函数, 本文仍将新息函数的分布假设为高斯白噪声分布, 而为了保证参数估计的稳健性, 通常对损失函数有如下要求:

- 1、 $\rho(z)$  关于  $|z|$  非降;
- 2、 $\rho(0) = 0$ ;
- 3、当  $z > 0$  时,  $\rho(z)$  关于  $z$  是单调递增的, 从而使得  $\rho(z) < \rho(\infty)$ ;

4、 $\rho(z)$  的导数  $\psi(z)$  是有界的, 即存在某个有限的常数  $c$ , 对任意的  $z$ , 都有  $|\psi(z)| < c$ ; 为满足这些要求, 本文取损失函数

$$\rho = \begin{cases} -\frac{f(z)^\alpha - 1}{\alpha} & \alpha > 0 \\ -\log(f(z)) & \alpha = 0 \end{cases}$$

其中  $f(z)$  为新息函数的概率密度函数。

也即目标函数可写为:

$$\rho_n(\theta) = \begin{cases} -\frac{\sum_{t=1}^n [(2\pi\sigma^2)^{-\alpha/2} \exp\{-\alpha \frac{\epsilon_t^2(\theta)}{2\sigma^2}\} - 1]}{\alpha} & \alpha > 0 \\ -\frac{n}{2} \log(2\pi\sigma^2) + \sum_{t=1}^n \frac{\epsilon_t^2(\theta)}{2\sigma^2} & \alpha = 0 \end{cases}$$

其中  $\alpha$  为该目标函数的调整参数。

注意到当  $\sigma$  已知,  $\alpha = 0$  时, 最小化目标函数即等价于最小化  $\sum_{t=1}^n \epsilon_t^2(\theta)$ , 即为 LS 估计的损失函数。从这个角度看, TARMA 模型的 Robust 估计确为 LS 估计在某种意义上推广。事实上,  $\alpha = 0$  的情况并不能作为 Robust 估计的目标函数, 因为此时  $\rho$  的导数未必有界, 该情况仅作为目标函数的一个极限情况存在。同时也印证了调整参数的作用: 调整参数越大, 估计的稳健性越好; 调整参数越小, 估计的效率越高, 而当  $\alpha$  趋近于 0 时, 原估计趋近于极大似然估计, 自然有着很高的效率。

接下来本文将着重讨论 TARMA 参数的 Robust 方法。

类似于 LS 估计, 进行估计时首要的问题在于目标函数的导数对  $\theta$  的不连续性, 为此, 本文采用与 LS 估计相同的 plug-in 方法: 首先, 对于给定的  $r$  和  $d$ , 求得最优的  $\lambda_n$ , 随后用遍历的方法, 求出在该  $\lambda_n$  下最优的  $r$  和  $d$ , 最后再在新的  $r$  和  $d$  下求出新的  $\lambda_n$ 。如此便可估计出 TARMA 模型的各项参数。

如此, 只需考虑如何在给定  $r$  和  $d$  的条件下去求得最优的  $\lambda_n$ 。观察损失函数, 要想将其最小化, 先对  $\lambda$  求导, 得到:

$$\frac{\partial}{\partial \lambda} \rho_n(\theta) = \sum_{t=1}^n \omega(\epsilon_t(\theta)) \frac{\partial \epsilon_t^2(\theta)}{\partial \lambda} = 0$$

其中  $\omega(\epsilon_t(\theta)) := \psi(|\epsilon_t(\theta)|^{1/2})$ , 其为该估计的权重。为了保证估计的稳定性, 即使新息的实际分布与我们假定分布不相容时权重应当较小。为此, 本文可以为  $\omega$  设置一个上限, 从而保证我们 Robust 估计的稳定性。

由于  $\psi(z)$  并不单调, 原损失函数可能会存在局部最小值, 故而直接求解上式比较困难。为此采用迭代加权最小二乘 (IRLS) 的方法, 步骤如下:

1、利用当前参数计算权重  $\omega_t = \omega(\epsilon_t(\theta))$ 。

2、求解  $\sum_{t=1}^n \omega_t \frac{\partial \epsilon_t^2(\theta)}{\partial \lambda} = 0$  来获得新的参数, 而该式等价于最小化一个加权最小二乘  $\sum_{t=1}^n \omega_t \epsilon_t^2(\theta)$ 。对于这样的一个加权最小二乘问题, 首先按照  $X_{t-d}$  与  $r$  的大小关系将  $X_t$  分为上下两个分面, 再对每个分面采用类似于 ARMA 模型估计的自回归逼近的方法, 构造矩阵:



$$Y = \begin{pmatrix} x_{r1} & x_{r2} & \cdots & x_{rn} \end{pmatrix}^T \quad X = \begin{pmatrix} x_{r1-1} & \cdots & x_{rn-1} \\ \vdots & & \\ x_{r1-p} & \cdots & x_{rn-p} \\ \epsilon_{r1-1} & \cdots & \epsilon_{rn-1} \\ \vdots & & \\ \epsilon_{r1-q} & \cdots & \epsilon_{rn-q} \end{pmatrix}^T$$

对所构造的矩阵进行加权最小二乘估计  $Y \sim X$ ，并用估计的结果计算出残差的标准差用于估计  $\sigma$ ，即可获得在  $r$  和  $d$  给定时所待求的参数  $\lambda$ 。

由此迭代，即可获得在给定  $r$  和  $d$  时最优的  $\lambda$ 。

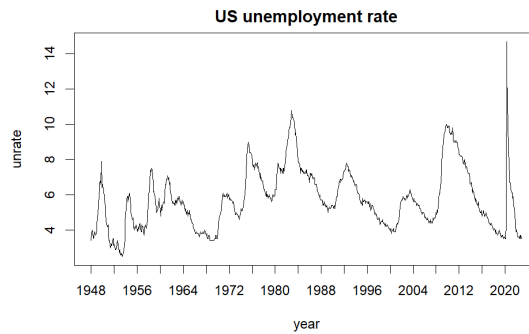
至此，距离完全解决这个问题只剩最后一步：如何选取合适的初值，使得上述过程可以顺利进行。事实上，只需利用 5.1 中所述的 LS 估计即可做到。由于 LS 估计对极端的敏感性，先删去部分极端值，本文选取 10% 的异常值。由此得到所需的初始参数，而后对原数据进行上述 Robust 估计，即可得到 TARMA 模型参数的 Robust 估计。

## 5. Parsimony of TARMA model, evidence from US Unemployment Rate

在线性时间序列分析中，在自回归模型中加入滑动平均项可以在保持模型拟合能力的同时增强模型简约性。本着同样的想法，Gorraci(2020)[18] 等人分析太阳黑子和美国失业率数据，得出 TARMA(1,1) 比 TAR(p) 具有更佳的拟合能力。

本节我们使用美国 1948 至 2022 年失业率数据 (<https://fred.stlouisfed.org/series/UNRATE>)，比较 AR, TAR, ARMA, TARMA 的模型拟合效果，一定程度上复现 Gorraci 的工作。(本文将实际数据的最新若干步作为测试集，其余数据作为训练集，计算测试集部分的预测值与测试集实际值的平均绝对误差 (MAE) 并以此作为衡量模型拟合的效果。)

失业率数据的时序图如下：(Figure 1)。



**Figure 1.** Time series of the monthly US unemployment rate from 1948 to 2022

我们测试不同阶数 AR, TAR, ARMA, TARMA 的 5 步 MAE，综合考虑模型的有效性和简约性，选出四类 MAE 最低同时也是最简约的模型：AR(1), TAR(1), ARMA(1,1), TARMA(1,1)。



它们的参数估计分别如 table2, table3, table4, table5所示。

**Table 2.** Estimate of AR model of unrate

$\hat{\phi}_0$	$\hat{\phi}_1$	$\hat{\sigma}^2$	MAE
5.5825	0.9705	0.1729	0.1694

**Table 3.** Estimate of TAR(1) model of unrate

$\hat{\phi}_{1,0}$	$\hat{\phi}_{1,1}$	$\hat{\phi}_{2,0}$	$\hat{\phi}_{2,1}$	$\hat{\sigma}_1^2$	$\hat{\sigma}_1^2$	$\hat{r}$	MAE
0.2017	0.9654	-0.2883	0.2497	1.1901	0.5000	4.500	0.1814

**Table 4.** Estimate of ARMA model of unrate

$\hat{\phi}_0$	$\hat{\phi}_1$	$\hat{\theta}_1$	$\hat{\sigma}^2$	MAE
5.6167	0.9667	0.0613	0.1724	0.1779

MAE 的大小关系为  $\text{TARMA} < \text{AR} < \text{ARMA} < \text{TAR}$ 。

由于门限的存在,  $\text{TARMA}(1,1)$  优于  $\text{ARMA}$ ; 由于加入滑动平均项,  $\text{TARMA}(1,1)$  优于  $\text{TAR}$ 。TARMA 提供了最好的预测效果。这也与 Gorraci(2020)[18] 的研究相符。

再次回顾数据。偏自相关系数如 (Figure 2) 所示。pacf 仅在一阶显著, 这也对应于我们估计的模型的阶数和滞后阶数  $r$  都估计为 1。(Figure 3) 显示美国失业率曲线多次在阈值 4.8% 附近反向, 我们认为这是阈值效应的体现。建立阈值模型可以更清晰地反应数据非线性的特点。

## 6. Application of TARMA Model

为检验本文所构造 TARMA 模型的实际作用, 尝试对一些实际数据进行建模, 并且与传统所使用的 ARMA 模型等进行比较, 通过计算 MAE (平均绝对误差) 等方式说明该模型的可取之处。

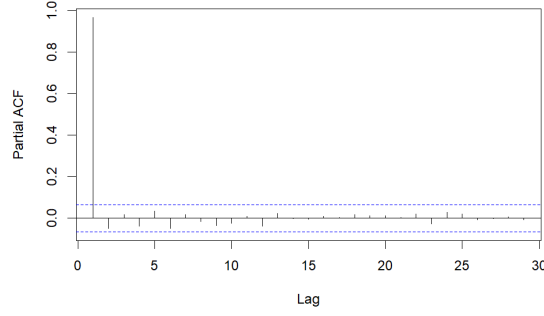
以金 (gold) 从 1960 年 1 月到 2022 年 11 月的价格变化序列作为选用的样例, 数据来源 (<https://www.worldbank.org/en/research/commodity-markets>)。观察该组数据的时序图 (Figure 4)。

观察发现大致在 1960-1970 年间金价保持不变, 这与当时的国际货币协议: 布雷顿森林体系有关, 美元与黄金之间具有固定汇率, 用于战后世界的恢复, 不具有随机性, 故我们舍弃这部分数据。这个体系在美国深陷越战泥潭, 美元超发后崩溃, 在之后至今黄金均为自由交易, 具有极强的时效性, 所以我们选择 1972 年 1 月至 2022 年 11 月的金价数据, 并作出其对数收益率的时序图 (Figure 5)。

注意到从序列可能具有非线性特征, 于是对其进行 SupLM Test, 来验证该情况下是否适用 TARMA 模型。supLM Test 结果如下表6所示:

**Table 5.** LS estimate of TARMA model of unrate

$\hat{\phi}_{1,0}$	$\hat{\phi}_{1,1}$	$\hat{\theta}_{1,1}$	$\hat{\phi}_{2,0}$	$\hat{\phi}_{2,1}$	$\hat{\theta}_{2,1}$	$\hat{\sigma}^2$	$\hat{r}$	MAE
0.1173	0.9795	0.0382	0.185	0.9620	0.1973	0.2526	4.800	0.1409

**Figure 2.** pacf of US unemployment rate

经过 SupLM Test 得到的 sLM 统计量值为 12.91, 与表一对照可知, 在 0.05 水平下可以拒绝原假设, 即金价的对数收益率序列的确存在阈值效益。为简化问题, 本文仅采用只有两个分面的 TARMA 模型, 旨在说明 TARMA 模型在保有对数据估计的高精确度的前提下, 还能够保有形式的简洁性。另外, 基于宏观经济理论, 我们设  $d = 1$ 。

作为对比以及 ACF 与 PACF 图 (Figure 6)(Figure 7), 本文选用 ARMA(1,1) 模型拟合数据, 分别对该模型与 TARMA(1,1) 进行模型参数的估计并计算 MAE, 所得结果如下:

TARMA 的 LS 参数估计结果如表7所示:

模型为:

$$X_t = \begin{cases} 0.2929 - 0.3427X_{t-1} + \epsilon_t + 0.4149\epsilon_{t-1} & X_{t-1} \leq -1.6205 \\ 0.4682 - 0.3220X_{t-1} + \epsilon_t + 0.5377\epsilon_{t-j} & X_{t-1} > -1.6205 \end{cases} \quad (6)$$

其中:  $\epsilon_t \sim i.i.d. \sim N(0, 3.0582)$

TARMA 的 Robust 参数估计结果如表8所示:

模型为:

$$X_t = \begin{cases} -0.1538 - 0.2909X_{t-1} + \epsilon_t + 0.3457\epsilon_{t-1} & X_{t-1} \leq 0.7007 \\ 0.7731 - 0.0904X_{t-1} + \epsilon_t + 0.2742\epsilon_{t-j} & X_{t-1} > 0.7007 \end{cases} \quad (7)$$

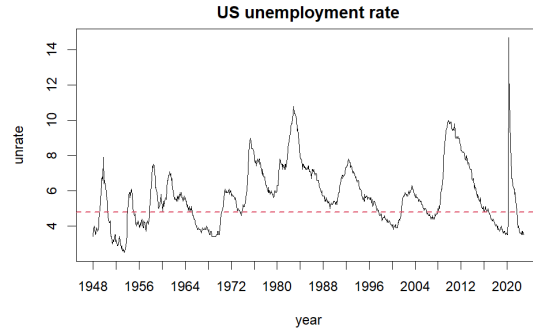
其中:  $\epsilon_t \sim i.i.d. \sim N(0, 1.3956)$

ARMA 参数估计结果如表9所示:

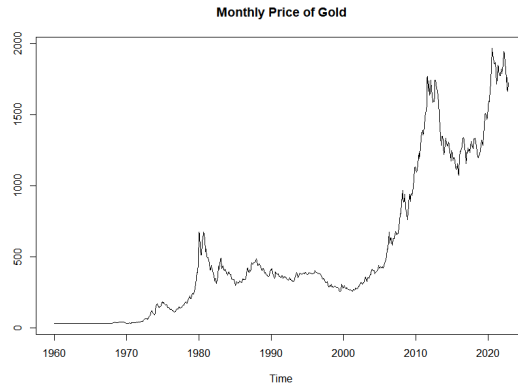
模型为:

$$X_t = 0.5268 - 0.1076X_{t-1} + \epsilon_t + 0.4057\epsilon_{t-1} \quad (8)$$

其中:  $\epsilon_t \sim i.i.d. \sim N(0, 17.99)$



**Figure 3.** US unemployment rate with threshold line



**Figure 4.** monthly prices of gold from 1960

三个模型的 MAE 如表10: 由上述结果, 可得出下述结论:

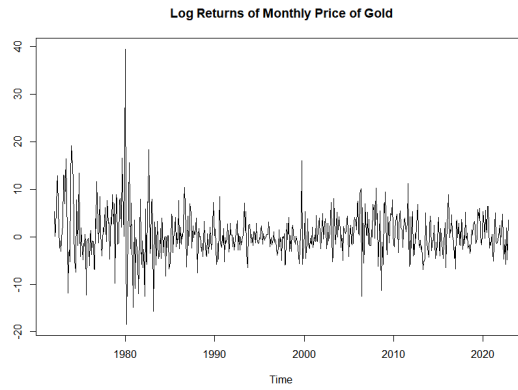
1、根据估计, 当前一天的收益率大于 0.7% 时, 金价出现明显的差异, TARMA 模型以此大致分为两个分面, 也就是所谓的商品价格的非对称性 [13]。同时鲁棒估计得到的  $\hat{\sigma}^2$  相比于 LS 与 ARMA 估计更小。

2、对传统的 ARMA 模型, TARMA 的 LS 估计和 Robust 估计分别计算 MSE, 可以发现 ARMA 模型的 MSE 最大, TARMA 模型的 LS 估计次之, 而 TARMA 模型的 Robust 估计保有最小的 MSE, 这也符合本文的预期, 说明 Robust 估计在应对实际数据的黑天鹅极端值现象上有着更好的表现。同时 TARMA 模型仅使用了 TARMA (1, 1), 又维持了较简洁的形式而避免了繁琐的参数。

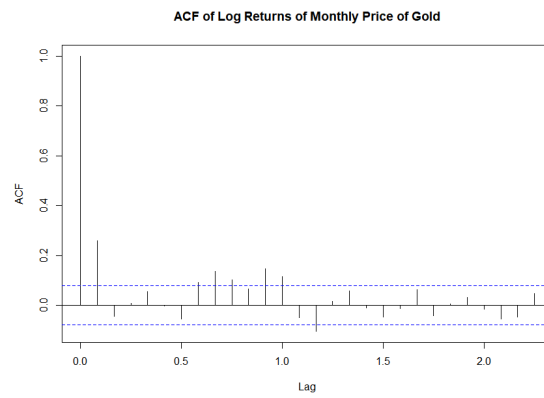
综上所述, 可认为 TARMA 模型很好地完成了任务, 有一定的应用价值。

## 7. Conclusion

本文主要回答了引言中给出的三个问题: 为什么是 TARMA 不是 ARMA(SupLM Test), 为什么是 TARMA 不是 TAR(Parsimony), 怎么对 TARMA 模型的参数进行鲁棒估计。本文的关注点异常值与门限亦或者是阈值在非线性时间序列分析以及实证研究中扮演着重要角色 [5][4], 黑天鹅般的异常值与一些“底线”般的门限在日常生活中处处存在, 比如在黄金价格中, 超出门限的区域包含了二战以来的几乎接近所有的黑天鹅与灰犀牛, 包括但不限于: 金



**Figure 5.** log returns of monthly prices of gold from 1960

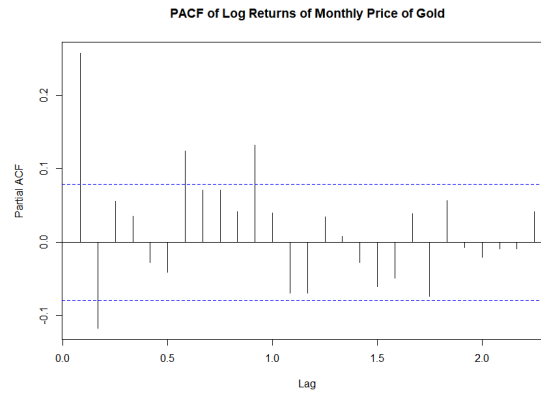


**Figure 6.** ACF of log returns of monthly prices of gold from 1972

融危机，美元加息，俄乌冲突，疫情。又或者是这一年来隔几个月就要打响一次的“上证指数三千点保卫战”，再或者是几次汇改后自由交易与央行监管双重作用下人民币汇率的波动。Threshold 就像是一只看不见的手笼罩在现实中，越界之后无论是政府的强硬限制或者是人们自发的调整行为都可能导致非理性现象的出现，留待之后研究。

## References

- [1] H. Tong, *Non-linear time series: a dynamical system approach*. Oxford university press, 1990.
- [2] K.-s. Chan, *Exploration of a Nonlinear World: An Appreciation of Howell Tong's Contributions to Statistics*. World Scientific, 2009.
- [3] P. Giordani, R. Kohn, and D. van Dijk, “A unified approach to nonlinearity, structural change, and outliers,” *Journal of Econometrics*, vol. 137, no. 1, pp. 112–133, 2007.
- [4] H. Tong, “Threshold models in time series analysis—30 years on,” *Statistics and its Interface*, vol. 4, no. 2, pp. 107–118, 2011.



**Figure 7.** PACF of log returns of monthly prices of gold from 1972

**Table 6.** result of sLM Test

threshold parameter corresponding to the sLM statistic	sLM statistic
8.009347	12.911288

- [5] B. E. Hansen, “Threshold autoregression in economics,” *Statistics and its Interface*, vol. 4, no. 2, pp. 123–127, 2011.
- [6] K.-S. Chan, B. E. Hansen, and A. Timmermann, “Guest editors’ introduction: Regime switching and threshold models,” pp. 159–161, 2017.
- [7] H. Tong, “On a threshold model,” 1978.
- [8] —, “Threshold models in time series analysis—some reflections,” *Journal of Econometrics*, vol. 189, no. 2, pp. 485–491, 2015.
- [9] G. Goracci, “Threshold autoregressive moving-average models: probabilistic structure, statistical aspects and applications,” 2019.
- [10] D. Li, W. K. Li, and S. Ling, “On the least squares estimation of threshold autoregressive and moving-average models,” *Statistics and its Interface*, vol. 4, no. 2, pp. 183–196, 2011.
- [11] G. Goracci, S. Giannerini, K.-S. Chan, and H. Tong, “Testing for threshold effects in the tarma framework,” *arXiv preprint arXiv:2103.13977*, 2021.
- [12] W.-S. Chan and S.-H. Cheung, “On robust estimation of threshold autoregressions,” *Journal of Forecasting*, vol. 13, no. 1, pp. 37–49, 1994.
- [13] L. Kilian and R. J. Vigfusson, “Are the responses of the us economy asymmetric in energy price increases and decreases?” *Quantitative Economics*, vol. 2, no. 3, pp. 419–453, 2011.
- [14] —, “Do oil prices help forecast us real gdp? the role of nonlinearities and asymmetries,” *Journal of Business & Economic Statistics*, vol. 31, no. 1, pp. 78–93, 2013.

**Table 7.** LS estimate of TARMA model

$\hat{\phi}_{1,0}$	$\hat{\phi}_{1,1}$	$\hat{\theta}_{1,1}$	$\hat{\phi}_{2,0}$	$\hat{\phi}_{2,1}$	$\hat{\theta}_{2,1}$	$\hat{\sigma}^2$	$\hat{r}$
0.2929	-0.3427	0.4149	0.4682	-0.3220	0.5377	3.0582	-1.6205

**Table 8.** robust estimate of TARMA model

$\hat{\phi}_{1,0}$	$\hat{\phi}_{1,1}$	$\hat{\theta}_{1,1}$	$\hat{\phi}_{2,0}$	$\hat{\phi}_{2,1}$	$\hat{\theta}_{2,1}$	$\hat{\sigma}^2$	$\hat{r}$
-0.1538	-0.2909	0.3457	0.7731	-0.0904	0.2742	1.3956	0.7007

- [15] H. Tong and K. S. Lim, “Threshold autoregression, limit cycles and cyclical data,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 42, no. 3, pp. 245–268, 1980.
- [16] G. Goracci, D. Ferrari, S. Giannerini *et al.*, “Robust estimation for threshold autoregressive moving-average models,” *arXiv preprint arXiv:2211.08205*, 2022.
- [17] K.-S. Chan, S. Giannerini, G. Goracci, and H. Tong, “Testing for threshold regulation in presence of measurement error with an application to the ppp hypothesis,” *arXiv e-prints*, pp. arXiv–2002, 2020.
- [18] G. Goracci, “Revisiting the canadian lynx time series analysis through tarma models,” *Statistica*, vol. 80, no. 4, pp. 357–394, 2020.

## A. 附录

### A.1. sLM 统计量渐进零分布

定义:

$\{\alpha_j\}$  满足差分方程  $\alpha_0 = 1, \alpha_j - \sum_{s=1}^q \theta_s \alpha_{j-s}, \alpha_{-1} = \dots = \alpha_{-q+1} = 0$

$$D(r) = \left( \sum_{j=0}^{\infty} \alpha_j, \sum_{j=0}^{\infty} \alpha_j X_{t-1-j}, \dots, \sum_{j=0}^{\infty} \alpha_j X_{t-p-j}, \right. \\ \left. \sum_{j=0}^{\infty} \alpha_j I_r(X_{t-d-j}), \sum_{j=0}^{\infty} \alpha_j I_r(X_{t-d-j}) X_{t-1-j}, \dots, \sum_{j=0}^{\infty} \alpha_j I_r(X_{t-d-j}) X_{t-p-j} \right)^T$$

再定义:

$$\Lambda(r) = \begin{pmatrix} \Lambda_{11} & \Lambda_{12}(r) \\ \Lambda_{12}^T(r) & \Lambda_{22}(r) \end{pmatrix} = \frac{1}{\sigma^2} E[D(r)D(r)^T]$$

其中  $\Lambda_{ij}(r)$  与  $I_{n,ij}(r)$  具有相同的维数。

Goracci 等人 (2021) [11] 发现, 若 (i) 在零假设下的 ARMA 模型的自回归多项式  $\phi(z) = 1 - \phi_{11}z - \dots - \phi_{1p}z^p$  和滑动平均多项式  $\theta(z) = 1 - \theta_{11}z - \dots - \theta_{1p}z^q$  在单位圆内无根且无公

**Table 9.** ARMA estimate

$\hat{\phi}_{1,0}$	$\hat{\phi}_{1,1}$	$\hat{\theta}_{1,1}$	$\hat{\sigma}^2$
0.5268	-0.1076	0.4057	17.99

**Table 10.** MAE of models

$\text{mae}_{robust}$	$\text{mae}_{ls}$	$\text{mae}_{arma}$
3.173626	3.498996	3.901274

共根 (ii)  $\{\epsilon_t\}$  有有限严格正密度且有有限四阶矩, 则在零假设下, sLM 统计量  $T_n$  弱收敛于

$$\sup_{r \in [r_L, r_U]} \xi(r)^T (\Lambda_{22}(r) - \Lambda_{21}(r) \Lambda_{11}^{-1} \Lambda_{12}(r))^{-1} \xi(r)$$

其中,  $\{\xi(r), r \in \mathbb{R}\}$  为中心化的高斯序列, 其协方差函数为

$$\Sigma(r_1, r_2) = \Lambda_{22}(r_1 \wedge r_2) - \Lambda_{21}(r_1) \Lambda_{11}^{-1} \Lambda_{12}(r_2)$$

## A.2. 代码

### A.2.1. supLM Test

```
library(forecast)
library(polynom)
library(graphics)
library('TSA')
library('astsa')
library('readr')
library('zoo')
library('xts')
library(fGarch, quietly = TRUE)
library('tseries')
library('MTS')
library("rugarch")
tarma.sim <- function(
  object, ntransient = 500, n = 500, phi1, phi2,
  theta1, theta2, thd, d, p, q, sigma1 = 1, sigma2 = 1,
  xstart = rep(0, max(p,d,q)), e
) {
  if(!missing(object)){#这里要和 object 对接上
    p1 = object$p1
    p2 = object$p2
    p = max(p1,p2)
    q1 = object$q1
    q2 = object$q2
    q = max(q1,q2)
```



```

Phi1 = rep(0,p+1)
Phi2 = rep(0,p+1)
theta1 = rep(0,q)
theta2 = rep(0,q)
if(object$is.constant1) phi1[1:(p1+1)]=object$qr1$coefficients
else phi1[2:(p1+1)]=object$qr1$coefficients
if(object$is.constant2) phi2[1:(p2+1)]=object$qr2$coefficients
else phi2[2:(p2+1)]=object$qr2$coefficients
d = object$d
thd = object$thd
sigma1 = object$rms1^0.5
sigma2 = object$rms2^0.5
}
if(missing(xstart)) xstart = rep(0,max(p,d))
large <- 10^7
mm = length(xstart)
x <- rep(0, (mm + ntransient + n))
boot.start <- length(xstart) + 1
x[seq(xstart)] <- xstart
if (missing(e))
  e <- rnorm(mm + ntransient + n)
for (i in (boot.start:(mm + ntransient + n))) {
  if (x[i - d] <= thd)
    x[i] <- sum(phi1 * c(1, x[(i - 1):(i - p)])) +
      sigma1 * (e[i] + sum(theta1 * e[(i - 1):(i - q)]))
  else x[i] <- sum(phi2 * c(1, x[(i - 1):(i - p)])) +
    sigma2 * (e[i] + sum(theta2 * e[(i - 1):(i - q)]))
}
invisible(list(y = x[(ntransient + mm + 1):(ntransient + mm + n)], e = e[(ntransient +
}

eps_phi_10 <- function(t, theta) {
  temp <- 0
  for (j in 0:(t - 1)) {
    temp <- temp + theta^j
  }
  return(temp)
}

eps_phi_11 <- function(t, theta, x) {
  temp <- 0
  for (j in 0:(t - 1)) {
    if ((t - 1-j) == 0) {
      next
    }
    temp <- temp + theta^j * x[t - 1 - j]
  }
  return(temp)
}

```

```

eps_theta_11 <- function(t, theta, epsilon) {
  temp <- 0
  for (j in 0:(t - 1)) {
    if ((t - 1-j) == 0) {
      next
    }
    temp <- temp + theta^j * epsilon[t - 1 - j]
  }
  return(temp)
}

eps_psi_10 <- function(t, theta, x, r) {
  temp <- 0
  for (j in 0:(t - 1)) {
    if ((t - 1-j) == 0) {
      next
    }
    if (x[t - 1 - j] <= r) {
      temp <- temp + theta^j
    }
  }
  return(temp)
}

eps_psi_11 <- function(t, theta, x, r) {
  temp <- 0
  for (j in 0:(t - 1)) {
    if ((t - 1-j) == 0) {
      next
    }
    if (x[t - 1 - j] <= r) {
      temp <- temp + theta^j * x[t - 1 - j]
    }
  }
  return(temp)
}

l_psi <- function(n, sigma_2, theta, x, epsilon, r) {
  temp1 <- 0
  temp2 <- 0
  for (t in 1:n) {
    temp1 <- temp1 + epsilon[t] / sigma_2 * eps_psi_10(t, theta, x, r)
    temp2 <- temp2 + epsilon[t] / sigma_2 * eps_psi_11(t, theta, x, r)
  }
  return(c(temp1, temp2))
}

I_11 <- function(n, sigma_2, theta, x, epsilon) {
  A <- matrix(rep(0, 9), nrow = 3)
  for (t in 1:n) {
    A[1, 1] <- A[1, 1] + 1 / sigma_2 * eps_phi_10(t, theta)^2
    A[1, 2] <- A[1, 2] + 1 / sigma_2 * eps_phi_10(t, theta) * eps_phi_11(t, theta, x)
  }
}

```

```

    A[1, 3] <- A[1, 3] + 1 / sigma_2 * eps_phi_10(t, theta) * eps_theta_11(t, theta, epsilon)
    A[2, 2] <- A[2, 2] + 1 / sigma_2 * eps_phi_11(t, theta, x)^2
    A[2, 3] <- A[2, 3] + 1 / sigma_2 * eps_phi_11(t, theta, x) * eps_theta_11(t, theta, epsilon)
    A[3, 3] <- A[3, 3] + 1 / sigma_2 * eps_theta_11(t, theta, epsilon)^2
  }
  A[2, 1] <- A[1, 2]
  A[3, 1] <- A[1, 3]
  A[3, 2] <- A[2, 3]
  return(A)
}

I_12 <- function(n, sigma_2, theta, x, epsilon, r) {
  B <- matrix(rep(0, 6), nrow = 3)
  for (t in 1:n) {
    B[1, 1] <- B[1, 1] + 1 / sigma_2 * eps_phi_10(t, theta) * eps_psi_10(t, theta, x, r)
    B[1, 2] <- B[1, 2] + 1 / sigma_2 * eps_phi_10(t, theta) * eps_psi_11(t, theta, x, r)
    B[2, 1] <- B[2, 1] + 1 / sigma_2 * eps_phi_11(t, theta, x) * eps_psi_10(t, theta, x, r)
    B[2, 2] <- B[2, 2] + 1 / sigma_2 * eps_phi_11(t, theta, x) * eps_psi_11(t, theta, x, r)
    B[3, 1] <- B[3, 1] + 1 / sigma_2 * eps_theta_11(t, theta, epsilon) * eps_psi_10(t, theta, x, r)
    B[3, 2] <- B[3, 2] + 1 / sigma_2 * eps_theta_11(t, theta, epsilon) * eps_psi_11(t, theta, x, r)
  }
  return(B)
}

I_22 <- function(n, sigma_2, theta, x, r) {
  C <- matrix(rep(0, 4), nrow = 2)
  for (t in 1:n) {
    C[1, 1] <- C[1, 1] + 1 / sigma_2 * eps_psi_10(t, theta, x, r)^2
    C[1, 2] <- C[1, 2] + 1 / sigma_2 * eps_psi_10(t, theta, x, r) * eps_psi_11(t, theta, x, r)
    C[2, 2] <- C[2, 2] + 1 / sigma_2 * eps_psi_11(t, theta, x, r)^2
  }
  C[2, 1] <- C[1, 2]
  return(C)
}

supLM_r <- function(epsilon, sigma_2, phi_10, phi_11, theta, d, r, x, n) { # 符号与 supLM 一致
  temp1 <- solve(I_11(n, sigma_2, theta, x, epsilon))
  temp2 <- I_22(n, sigma_2, theta, x, r) - t(I_12(n, sigma_2, theta, x, epsilon, r)) %*% temp1
  temp3 <- solve(temp2)
  temp4 <- t(l_psi(n, sigma_2, theta, x, epsilon, r)) %*% temp3 %*% l_psi(n, sigma_2, theta, x, epsilon)
  return(temp4)
}

trimming <- function(data) { # data is a vector
  l <- length(data)
  index_ext <- ceiling(length(data) / 4)
  extreme_index <- order(data)[-((index_ext + 1):(l - index_ext))]
  return(data[-extreme_index])
}

supLM <- function(data) {

```

```

trimmed_data <- trimming(data)
min_tdata <- range(trimmed_data)[1]
max_tdata <- range(trimmed_data)[2]
range_tdata <- seq(min_tdata, max_tdata, length.out = 100)
sLM_r <- cbind(range_tdata, 0)
mod <- arima(data, order = c(1,0,1), include.mean = TRUE)
# non-zero mean
epsilon <- mod$residuals
coef <- mod$coef # coef : ar1:phi_11, ma1:theta_11, mean:phi_10
sigma2 <- mod$sigma2
for (i in 1:100) {
  sLM_r[i, 2] <- supLM_r(epsilon = epsilon, sigma_2 = sigma2, phi_10 = coef[3], phi_11 =
}
return(sLM_r[which.max(sLM_r[, 2]), ])
}
data0 <- read.csv(
  'C:\\Users\\cokez\\Desktop\\paper\\CMO-Historical-Data-Monthly.csv', header = T, na.strin
data_gold <- ts(100*diff(log(data0$Gold))[400:740], start = 1)

```

### A.2.2. est for TARMA

```

TarmaRho1<-function(r,d,x,alpha,t,snew = c(1,0.9,-1,0.5,0.8,-1,1)){##给定r, d求使得损失函数
w<-NULL
s<-rep(1,7)##记录TARMA (1, 1) 的参数
l=0
e<-NULL
enew<-NULL
if(r>=0) {
  enew[1]<-x[1]-snew[1]
}else enew[1]<-x[1]-snew[4]
for(i in (1+d):t){
  enew[i]<-epsilon(snew,r,d,i,x)
}
while(sum((snew-s)^2)>=10^(-6)){##IRLS的方法求出在给定r, d下的最优
s<-snew
e<-enew
w<-qieta(t,alpha,abs(e)^(1/2),s[7])
for(i in 1:length(w)){
  w[i]<-ifelse(w[i]>=10,10,w[i])
  w[i]<-ifelse(w[i]==0,1,w[i])
}
snew<-wlm(x,e,t,r,d,w)$s##用加权最小二乘的方法计算出最优的参数
enew<-wlm(x,e,t,r,d,w)$re
fit1<-wlm(x,e,t,r,d,w)$fit1
fit2<-wlm(x,e,t,r,d,w)$fit2
l<-l+1
print(l)

```

```

    print(sum((snew-s)^2))
  }
  print(s)
  return(list(s=s, fit1=fit1, fit2=fit2, re=enew))
}
sigma1<-function(e,t){
  sigma<-0
  for(i in 1:t){
    sigma<-sigma+e[i]^2
  }
  sigma<-sigma/t
  return(sigma)
}
epsilon<-function(s,r,d,t,x){##新息函数
  if(t<=d) {
    if(r>=0) {
      return(x[t]-s[1])
    }else return(x[t]-s[4])
  }
  if(x[t-d]<=r){
    e<-x[t]-s[1]-s[2]*x[t-1]-s[3]*epsilon(s,r,d,t-1,x)
  }else{
    e<-x[t]-s[4]-s[5]*x[t-1]-s[6]*epsilon(s,r,d,t-1,x)
  }
  return(e)
}
qieta<-function(t,alpha,e,sigma){##导数
  if(alpha>0){
    q<-exp(-e^2*alpha/(2*sigma^2))/((2*pi*sigma^2)^(alpha/2))*(e/sigma^2)
  }else if(alpha==0)
    q<-rep(1,length(e))
  ##q<-e/sigma^2
  return(q)
}
TarmaRubust<-function(x,alpha){##求出参数
  y<-trimming(x)
  r<-quantile(y)[2]
  d<-1
  e<-NULL
  w<-NULL
  T1<-TarmaRho1(r,d,y,0,length(y))
  s<-T1$s
  if(alpha == 0){
    yita<-c(s,r,d)
    fit1<-T1$fit1
    fit2<-T1$fit2
    re <- T1$re
    return(list(yita=yita, fit1=fit1, fit2=fit2, re = re))
  }
}

```

```

}
T2<- TarmaRho1(r,d,x,alpha,length(x),snew = s)
s <- T2$s
if(r>=0) {
  e[1]<-x[1]-s[1]
}else e[1]<-x[1]-s[4]
for(i in (1+d):length(x)){
  e[i]<-epsilon(s,r,d,i,x)
}
w<-qieta(length(x),alpha,abs(e)^(1/2),s[7])
for(i in 1:length(x)){
  w[i]<-ifelse(w[i]>=10,10,w[i])
  w[i]<-ifelse(w[i]==0,1,w[i])
}
q=1
seq1<-seq(from=quantile(x)[2],to=quantile(x)[4],by=0.1)
pho<-matrix(0,nrow=length(seq1),ncol=1)
for(i in seq1){
  e<-wlm(x,e,length(x),i,d,w)$re
  pho[q,d]<-TarmaRho2(e,length(x),alpha,s[7])
  q=q+1
}
print(pho)
m<-which.min(pho)
r<-seq1[m]
list1<-TarmaRho1(r,1,x,alpha,length(x),snew = s)
s<-list1$s
yita<-c(s,r,d)
fit1<-list1$fit1
fit2<-list1$fit2
re <- list1$re
return(list(yita=yita,fit1=fit1,fit2=fit2,re = re))
}

TarmaRho2<-function(e,n,alpha,sigma){
  if(alpha>0){
    pho=0
    for(i in 1:n){
      pho<-pho-1/alpha*(exp(-e[i]^2*alpha/2/sigma^2)/(2*pi*sigma^2)^(alpha/2)-1)
    }
  }else if(alpha==0){
    pho<-n*log(2*pi*sigma^2)/2
    for(i in 1:n){
      pho<-pho+e[i]^2/(2*sigma^2)
    }
  }
  return(pho)
}

```

```

wlm<-function(x,e,t,r,d,w){
  s<-NULL
  y1<-NULL
  y2<-NULL
  x1<-NULL
  x2<-NULL
  w1<-NULL
  w2<-NULL
  re<-NULL
  index1<-NULL
  index2<-NULL
  for(i in (d+1):t){
    if(x[i-d]<=r){
      y1<-c(y1,x[i])
      n<-c(x[i-1],e[i-1])
      x1<-rbind(x1,n)
      w1<-c(w1,w[i])
      index1<-c(index1,i)
    }else {
      y2<-c(y2,x[i])
      n<-c(x[i-1],e[i-1])
      x2<-rbind(x2,n)
      w2<-c(w2,w[i])
      index2<-c(index2,i)
    }
  }
  if(length(y1)==0) {
    s[1]<-0
    s[2]<-0
    s[3]<-0
    fit1<-lm(1~1)
  }else{
    fit1<-lm(y1~x1,weights=w1)
    s[1]<-fit1$coefficients[1]
    s[2]<-fit1$coefficients[2]
    s[3]<-fit1$coefficients[3]
  }
  if(length(y2)==0) {
    s[4]<-0
    s[5]<-0
    s[6]<-0
    fit2<-lm(1~1)
  }else{
    fit2<-lm(y2~x2,weights=w2)
    s[4]<-fit2$coefficients[1]
    s[5]<-fit2$coefficients[2]
    s[6]<-fit2$coefficients[3]
  }
}

```



```

s[7]<-sqrt((length(y1)*stats::sigma(fit1)^2+length(y2)*stats::sigma(fit2)^2)/(length(y1)
s[is.na(s)]<-0
if(r>=0) {
  re[1]<-x[1]-s[1]
}else re[1]<-x[1]-s[4]
re[index1]<-fit1$residuals
re[index2]<-fit2$residuals
return(list(s=s,re=re,fit1=fit1,fit2=fit2))
}

wlm1<-function(x,e,t,r,d,w,s){
  l<-seq(from=(s[4]-1),to=(s[4]+1),by=0.1)
  for(i in l){
    fit1<-lm(ynew~xnew,w=w1)

  }
}

tarma.sim <- function(
  object, ntransient = 500, n = 500, phi1, phi2,
  theta1, theta2, thd, d, p, q, sigma1 = 1, sigma2 = 1,
  xstart = rep(0, max(p,d,q)), e
) {
  if(!missing(object)){#这里要和 object 对接上
    p1 = object$p1
    p2 = object$p2
    p = max(p1,p2)
    q1 = object$p1
    q2 = object$p2
    q = max(q1,q2)
    Phi1 = rep(0,p+1)
    Phi2 = rep(0,p+1)
    theta1 = rep(0,q)
    theta2 = rep(0,q)
    if(object$is.constant1) phi1[1:(p1+1)]=object$q1$coefficients
    else phi1[2:(p1+1)]=object$q1$coefficients
    if(object$is.constant2) phi2[1:(p2+1)]=object$q2$coefficients
    else phi2[2:(p2+1)]=object$q2$coefficients
    d = object$d
    thd = object$thd
    sigma1 = object$rms1^0.5
    sigma2 = object$rms2^0.5
  }
  if(missing(xstart)) xstart = rep(0,max(p,d))
  large <- 10^7
  mm = length(xstart)
  x <- rep(0, (mm + ntransient + n))

```

```

boot.start <- length(xstart) + 1
x[seq(xstart)] <- xstart
if (missing(e))
  e <- rnorm(mm + ntransient + n)
for (i in (boot.start:(mm + ntransient + n))) {
  if (x[i - d] <= thd)
    x[i] <- sum(phi1 * c(1, x[(i - 1):(i - p)])) +
      sigma1 * (e[i] + sum(theta1 * e[(i - 1):(i - q)]))
  else x[i] <- sum(phi2 * c(1, x[(i - 1):(i - p)])) +
    sigma2 * (e[i] + sum(theta2 * e[(i - 1):(i - q)]))
}
invisible(list(y = x[(ntransient + mm + 1):(ntransient + mm + n)], e = e[(ntransient +
}

phi<-function(x,w,e,t,r,d){##x分为x1, x2
y1<-NULL
y2<-NULL
x1<-NULL
x2<-NULL
w1<-NULL
w2<-NULL
e1<-NULL
e2<-NULL
phi<-NULL
for(i in (d+1):t){
  if(x[i-d]>=r){
    y1<-c(y1,x[i])
    n<-c(1,x[i-1],e[i-1])
    x1<-rbind(x1,n)
    w1<-c(w1,w[i])
    e1<-c(e1,e[i])
  }else {
    y2<-c(y2,x[i])
    n<-c(1,x[i-1],e[i-1])
    x2<-rbind(x2,n)
    w2<-c(w2,w[i])
    e2<-c(e2,e[i])
  }
}
}
w1<-diag(w1)
w2<-diag(w2)
phi[1:3]<-t(solve(t(x1)%*%w1%*%x1)%*%t(x1)%*%w1%*%e1)
phi[4:6]<-t(solve(t(x2)%*%w2%*%x2)%*%t(x2)%*%w2%*%e2)

return(phi)
}

```

```
trimming <- function(data) { # data is a vector
  l <- length(data)
  index_ext <- ceiling(length(data) / 20)
  extreme_index <- order(data)[-((index_ext + 1):(l - index_ext))]
  return(data[-extreme_index])
}
```

### A.2.3. some fundamental function of TARMA

```
predict.tarma <- function (x, n.ahead = 1, n.sim = 1000, ..., p, q, d, phi1, phi2, theta1,
  res = NULL
  xstart = rev(rev(x)[1:max(d, p, q)])
  for (i in 1:n.sim) {
    res = rbind(res, tarma.sim(p=p, q=q, d=d, phi1=phi1, phi2=phi2, theta1=theta1, theta2=
  }
  colnames(res) = paste(1:n.ahead, "step", sep='-')
  quan = apply(res, 2, quantile, c(0.5,0.025,0.975))
  if(plot==T){
    plot(x, type="l", xlim=c(0,(length(x)+n.ahead)),...)
    lines(x=(length(x)+1):(length(x)+n.ahead), y = quan[1,], type="l", col=2)
    lines(x=(length(x)+1):(length(x)+n.ahead), lty=3, y = quan[2,], type="l", col=4)
    lines(x=(length(x)+1):(length(x)+n.ahead), lty=3, y = quan[3,], type="l", col=4)
    legend("topleft", col=c(2,4), lty=c(1,3), legend = c("prediction", "95%interval"))
  }
  invisible(list(pred.matrix = res, fit=quan[1,], pred.interval=quan[-1, ]))
}

xp<-rep(0,100)
s<-NULL
for(i in 1900:1999){
  list1<-TarmaRubust(x[1:i],0.8)
  s<-list1$yita
  model <- predict.tarma(x[1:i], n.ahead=1, phi1 = s[1:2], phi2 = s[4:5], theta1 = s[3], t
  xp[i-1899] <-model$fit[1]
  print(i-1899)
}
err(x[1901:2000],xp,length(model$fit))

Tarmaforecast<-function(x,t,alpha){##x为数据, t为用于测试的数据量,alpha为调整参数
  xp<-rep(0,t)
  s<-NULL
  for(i in (length(x)-t):(length(x)-1)){
    list1<-TarmaRubust(x[1:i],alpha)
    s<-list1$yita
    model<-predict.tarma(x[1:i],n.ahead=1,phi1=s[1:2],phi2=s[4:5],theta1=s[3],theta2=s[6], t
    xp[i-length(x)+t+1] <-model$fit[1]
```

```
    print(i-length(x)+t+1)
  }
  err(x[(length(x)-t+1):length(x)],xp,t)
}

err<-function(x,xp,t){## 计算t个
  m<-mean(abs(x-xp))
  print(m)
}

ARMAforecast<-function(x,t){
  xp<-rep(0,t)
  s<-NULL
  for(i in (length(x)-t):(length(x)-1)){
    fit1<-arima(x[1:i],c(1,0,1))
    xp[i-length(x)+t+1]<-predict(fit1,n.ahead=1,se.fit = FALSE)
  }
  err(x[(length(x)-t+1):length(x)],xp,t)
}
```