

S&P 500 数据的建模: 基于 ARIMA 模型与 GARCH 模型*

统计 91 董晟渤, 2193510853

西安交通大学数学与统计学院

日期: 2022 年 6 月 8 日

摘 要

标准普尔 500 指数, 简称 S&P 500, 是美国股市的一个具有代表性的指数. 本篇报告对 2019 年 1 月 2 日至 2021 年 12 月 30 日的 S&P 500 的收盘价的对数值, 对数据建立了 ARIMA(4, 1, 4) 模型, 并对残差建立了 GARCH(1, 1) 模型.

对于 ARIMA 模型的部分, 首先, 基于 ACF、PACF 与 EACF, 筛选了合适的模型; 其次, 基于 AIC 比较了不同的模型, 并为最小化 AIC, 对对数收益率选择模型 ARMA(4, 4), 对对数收盘价选择模型 ARIMA(4, 1, 4); 为了对该模型进行参数估计, 报告中比较了 ML 估计与 CSS 估计的结果, 并选用 CSS-ML 估计的结果, 从而建立了模型; 在对对数收益率建立了 ARMA(4, 4) 模型之后, 报告中对残差的正态性与相关性进行了检验, 得到了残差不具正态性、具有平稳性, 但是残差的平方不具有平稳性的结论, 说明了模型有待改进; 最后, 基于该模型, 给出了近期对数收益率和对数收盘价的预测结果.

对于 GARCH 模型的部分, 首先, 对残差进行 ARCH 效应的检验, 说明残差需要进一步建立 GARCH 模型; 其次, 基于 AIC 比较了不同的模型, 并为最小化 AIC, 选择模型 GARCH(1, 1); 为了对该模型进行参数估计, 报告中使用了 ML 估计方法, 从而建立了模型; 在对对数收益率进一步建立了 ARMA(4, 4)-GARCH(1, 1) 模型之后, 报告中得到了残差及其平方都具有平稳性的结论, 说明了进一步建模的有效性; 最后, 基于该模型, 给出了风险值 (VaR) 的估计.

作为模型的推广, 笔者在建模过程中, 尝试基于动态神经网络, 建立了非线性自回归 (NAR) 模型, 并对预测值与残差进行了简单分析.

本篇报告实现了对真实数据进行 ARIMA 建模, 同时对残差进行 GARCH 建模. 报告的最后, 总结了本篇报告完成的工作, 并与前半学期所完成的工作进行对比.

关键词: S&P 500, 时间序列, ARMA 模型, ARIMA 模型, GARCH 模型.

*2021-2022 学年第二学期, 课程: 时间序列与金融统计, 指导老师: 惠永昌.

目录

1	概述	1
1.1	问题提出: S&P 500 数据的建模与预测	1
1.2	线性模型: MA 模型、AR 模型、ARMA 模型、ARIMA 模型	1
1.3	非线性模型: ARCH 模型与 GARCH 模型	2
1.4	方法概述: 对数据建立时间序列模型	3
2	准备工作: 数据收集与数据处理	4
2.1	数据收集: S&P 500 的历史数据	4
2.2	符号说明: 符号的说明与定义	4
2.3	数据处理: 绘制图像、取对数与作差	5
3	数据建模: ARMA 模型与 ARIMA 模型	7
3.1	模型初筛: ACF 函数、PACF 函数与 EACF 函数	7
3.2	模型选择: 最小化 AIC 准则	10
3.3	参数估计: ML、CSS 与 CSS-ML	11
3.4	模型应用: 对数收益率与对数收盘价的预测	12
3.5	残差分析: 正态性检验与相关性检验	12
3.6	理论分析: 关于残差的独立性与相关性的论证	15
4	残差建模: GARCH 模型	17
4.1	数据分析: ARCH 效应的 LM 检验	17
4.2	模型选择: 最小化 AIC 准则和阶数	17
4.3	参数估计: 对 GARCH 模型应用 ML	18
4.4	残差分析: 正态性检验与相关性检验	19
4.5	模型应用: 风险值 (VaR) 的估计	20
5	模型推广: 基于动态神经网络的 NAR 模型	21
5.1	模型概述: 动态神经网络的基本设置	21
5.2	结果分析: 预测值与残差的分析	22
6	总结	23
	参考文献	23

附录		i
A	所用软件	i
B	代码 1: 数据的 ARMA 建模	i
C	代码 2: 残差的 GARCH 建模	ii
D	代码 3: 基于动态神经网络的 NAR 建模	iv

1 概述

1.1 问题提出: S&P 500 数据的建模与预测

标准普尔 500 指数 (Standard & Poor's 500), 简称 S&P 500, 是一个由 1957 年起记录美国股市的平均记录, 观察范围达 500 只普通股, 总市值约 80%. 标准普尔 500 指数由标普道琼斯指数公司 (S&P Dow Jones Indices LLC, 标准普尔全球控股公司控制的合资公司) 开发并继续维持. S&P 500 的代表性够强, 甚至足以显示美国经济的兴衰.¹

本篇报告中, 将对 S&P 500 从 2019 年 1 月 2 日至 2021 年 12 月 30 日的历史数据, 建立时间序列模型. 具体地说, 本篇报告中处理的数据是对数收盘价和对数收益率 (分别用 p_t 和 r_t 表示), 首先对对数收盘价和对数收益率建立线性时间序列模型 (分别使用了 ARIMA 模型和 ARMA 模型), 再对上面所建立的模型的残差建立 GARCH 模型.

1.2 线性模型: MA 模型、AR 模型、ARMA 模型、ARIMA 模型

常用的线性时间序列模型有 MA 模型、AR 模型、ARMA 模型和 ARIMA 模型.²

在介绍时间序列模型时, 首先引入的是 MA(q) 模型, 其中 $q \geq 0$ 是给定的整数. 其中, “MA” 的含义是滑动平均 (Moving Average).

定义 1.1 (MA 模型). 设 $\varepsilon_t \sim \text{WN}(0, \sigma^2)$, 整数 $q \geq 0$, 若对常数 $\mu, a_1, a_2, \dots, a_q$, 有

$$X_t = \mu + \varepsilon_t + a_1\varepsilon_{t-1} + \dots + a_q\varepsilon_{t-q},$$

则记 $X_t \sim \text{MA}(q)$, 称 q 为阶数.

另外一类重要的模型是 AR(p) 模型, 其中 $p \geq 0$ 是给定的整数. 其中, AR 的含义是自回归 (Auto Regressive).

定义 1.2 (AR 模型). 设 $\varepsilon_t \sim \text{WN}(0, \sigma^2)$, 整数 $p \geq 0$, 若对常数 c, b_1, b_2, \dots, b_p , 有

$$X_t = c + b_1X_{t-1} + \dots + b_pX_{t-p} + \varepsilon_t,$$

则记 $X_t \sim \text{AR}(p)$, 称 p 为阶数.

基于 MA(q) 模型和 AR(p) 模型, 我们提出 ARMA(p, q) 模型, 其中 $p, q \geq 0$ 是给定的整数. 该模型同时包含了滑动平均和自回归的性质.

¹关于 S&P 500 的介绍, 详见 [1].

²关于时间序列模型的介绍, 详见 [2].

定义 1.3 (ARMA 模型). 设 $\varepsilon_t \sim \text{WN}(0, \sigma^2)$, 整数 $p, q \geq 0$, 若对常数 c, a_1, a_2, \dots, a_q 及 b_1, b_2, \dots, b_p , 有

$$X_t = c + b_1 X_{t-1} + \dots + b_p X_{t-p} + \varepsilon_t + a_1 \varepsilon_{t-1} + \dots + a_q \varepsilon_{t-q},$$

则记 $X_t \sim \text{ARMA}(p, q)$, 称 p, q 为阶数.

在建立时间序列模型时, 可能需要对数据进行差分, 以保证平稳性. 设进行的差分次数为正整数 $d \geq 1$, 定义差分算子 ∇ , 满足

$$\nabla X_t = X_t - X_{t-1}, \quad \nabla^d X_t = \nabla(\nabla^{d-1} X_t), \quad \forall d \geq 2.$$

定义 1.4 (ARIMA 模型). 设整数 $p, q \geq 0, d \geq 1$, 若 $\nabla^d X_t \sim \text{ARMA}(p, q)$, 则记 $X_t \sim \text{ARIMA}(p, d, q)$, 称 p, q 为阶数, d 为差分次数.

以上所介绍的模型, 将在建模中应用. 关于模型选择、参数估计、模型诊断和预测的方法, 均在 [2] 中介绍, 本篇报告将忽略一些繁琐的细节, 将关注点放在建模方法的应用上.

1.3 非线性模型: ARCH 模型与 GARCH 模型

在建立线性模型时, 常假设残差的方差是相等的. 在实际中, 残差的方差常常变化较大. 为此, 需要对残差做进一步的建模. 在这里引入非线性时间序列模型, 包括 ARCH 模型和 GARCH 模型.³

在这里, 用 r_t 表示收益率, 则有

$$r_t = \mathbb{E}(r_t | \mathcal{F}_{t-1}) + X_t,$$

其中 $\mathbb{E}(r_t | \mathcal{F}_{t-1})$ 为 r_t 的预测值, X_t 为残差. 我们需要对 X_t 进行建模, 首先引入 ARCH(p) 模型.

定义 1.5 (ARCH 模型). 设 $\varepsilon_t \sim \text{IID}(0, 1)$, ε_t 与 σ_t 独立, 整数 $p \geq 0$, 若对常数 a_0, a_1, \dots, a_p , 有

$$\begin{cases} X_t = \sigma_t \varepsilon_t, \\ \sigma_t^2 = a_0 + a_1 X_t^2 + \dots + a_p X_{t-p}^2, \end{cases}$$

则记 $X_t \sim \text{ARCH}(p)$, 称 p 为阶数.

在 ARCH 模型中, σ_t 仅与 X_1, \dots, X_{t-p} 有关. 若进一步假设 σ_t 与 $\sigma_{t-1}, \dots, \sigma_{t-q}$ 有关, 则可以建立 GARCH(p, q) 模型.

³关于 ARCH 模型和 GARCH 模型的介绍, 详见 [2].

定义 1.6 (ARCH 模型). 设 $\varepsilon_t \sim \text{IID}(0, 1)$, ε_t 与 σ_t 独立, 整数 $p, q \geq 0$, 若对常数 a_0, a_1, \dots, a_p 及 b_1, b_2, \dots, b_q , 有

$$\begin{cases} X_t = \sigma_t \varepsilon_t, \\ \sigma_t^2 = a_0 + a_1 X_t^2 + \dots + a_p X_{t-p}^2 + b_1 \sigma_{t-1}^2 + \dots + b_q \sigma_{t-q}^2, \end{cases}$$

则记 $X_t \sim \text{GARCH}(p, q)$, 称 p, q 为阶数.

通常对于数据, 建立 $\text{GARCH}(1, 1)$ 就足够了. 关于 ARCH 模型和 GARCH 模型的详细理论, 均在 [2] 中介绍, 本篇报告将直接应用结果.

1.4 方法概述: 对数据建立时间序列模型

总而言之, 本篇报告进行建模的流程如图 1 所示. 报告中, 主要考虑的模型是 ARIMA 模型 (或 ARMA 模型) 和 GARCH 模型, 主要使用的编程语言是 R 语言.

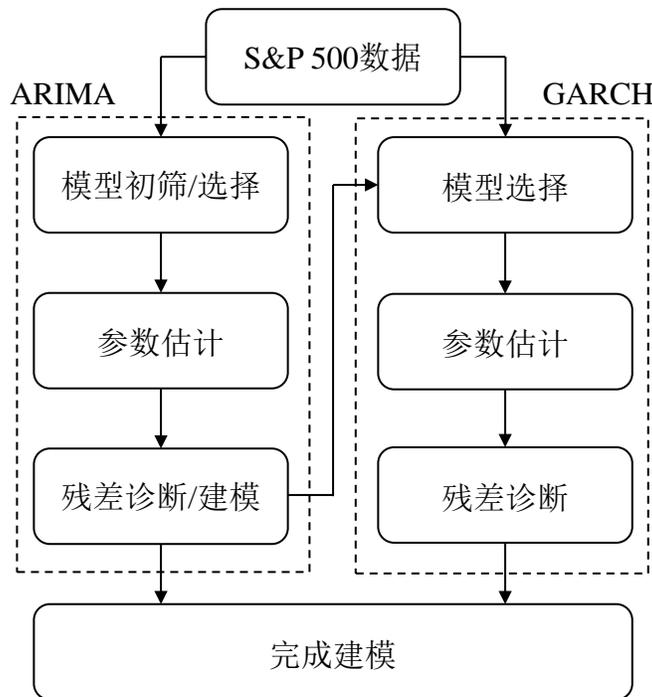


图 1: 本篇报告对数据进行时间序列建模的流程

2 准备工作: 数据收集与数据处理

2.1 数据收集: S&P 500 的历史数据

本篇报告所用的数据为 S&P 500 从 2019 年 1 月 2 日至 2021 年 12 月 30 日的历史数据, 数据来自英为财经⁴, 可供下载的数据有每月、每周和每日的历史数据. 为了建模的准确, 本篇报告在建立模型时, 选取的是按日的数据.

表 1: S&P 500 从 2019 年 1 月 2 日至 2021 年 12 月 30 日的历史数据 (按日)

日期	收盘价	开盘价	最高价	最低价
2019 年 1 月 2 日	2,510.03	2,476.96	2,519.49	2,467.47
2019 年 1 月 3 日	2,447.89	2,491.92	2,493.14	2,443.96
...
2021 年 12 月 29 日	4,793.06	4,788.64	4,804.06	4,778.08
2021 年 12 月 30 日	4,778.73	4,794.23	4,808.93	4,775.33

表 1 列出了 S&P 500 的部分历史数据. 其中:

- 收盘价是当日该股最后一笔交易前一分钟所有交易的成交量加权平均价;
- 开盘价是当日该股第一笔交易的每股成交价;
- 最高价和最低价是当日该股最高的每股成交价和最低的每股成交价.

收盘价是市场参与者们所共同认可的价格, 因此本报告建模的对象是收盘价.

2.2 符号说明: 符号的说明与定义

本报告中使用的符号、符号说明及定义如表 2 所示.

表 2: 符号说明

符号	符号说明	来源或定义
t	日期, 取值从 1 到 T	表 1 中的数据
P_t	t 时刻的收盘价	表 1 中的数据
p_t	t 时刻的对数收盘价	$p_t := \ln P_t$
r_t	t 时刻的对数收益率	$r_t := \nabla p_t$
ε_t 或 X_t	t 时刻模型的残差	ARMA 模型或 GARCH 模型

⁴历史数据见<https://cn.investing.com/indices/us-spx-500>.

其中, 在建立 ARMA 模型和 ARIMA 模型时, 由于不对残差进行建模, 从而用 ε_t 表示残差; 而在建立 GARCH 模型时, 为了体现对残差进行建模, 用 X_t 表示残差.

2.3 数据处理: 绘制图像、取对数与作差

为了处理数据, 记 2019 年 1 月 2 日对应 $t = 1$, 2021 年 12 月 30 日对应 $t = T$.

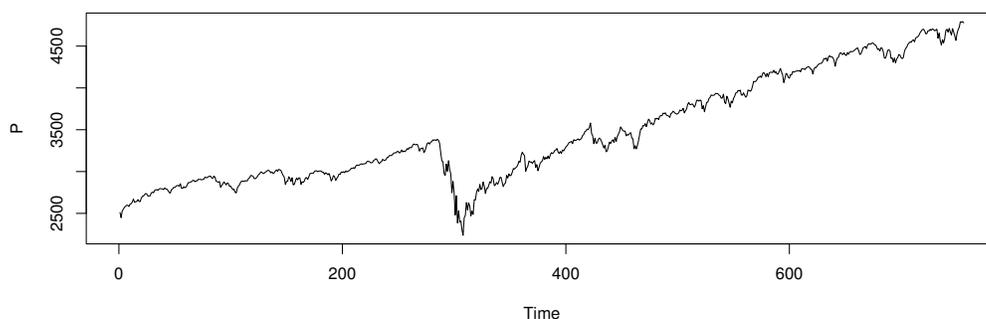


图 2: 收盘价 P_t 随时间 t 变化的图像

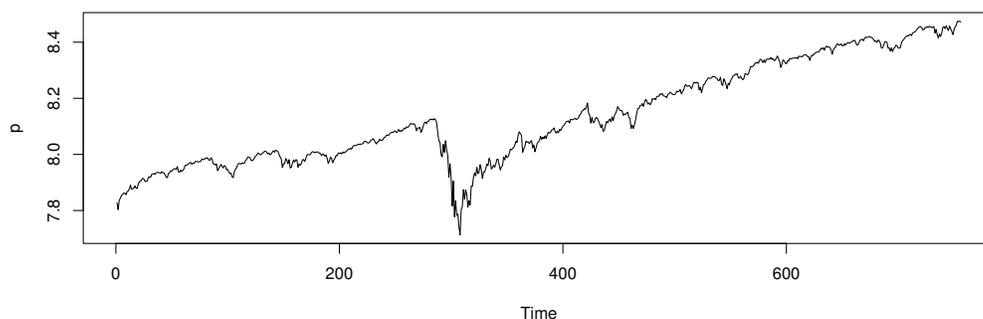


图 3: 对数收盘价 p_t 随时间 t 变化的图像

首先, 绘制出 $t - P_t$ 图像如图 2 所示. 对于金融数据, 为了看出其规律, 常常会进行取对数的操作. 记对数收盘价

$$p_t := \ln P_t, \quad t = 1, 2, \dots, T,$$

与图 2 相对比, 绘制出 $t - p_t$ 图像如图 3 所示.

接下来, 希望能对 P_t 或 p_t , 建立时间序列模型. 在此之前, 还需要考虑 P_t 和 p_t 的平稳性或相关性. 用于检验平稳性的方法有查看自相关函数 (ACF) 的图像, 以及 ADF 检验. 一方面, P_t 和 p_t 对应的自相关函数见图 4. 不管是对于 P_t 还是 p_t , 自相关函数都接近 1, 且缓

慢递减, 说明该过程具有长记忆性, 从而不具有平稳性; 另外一方面, 对 P_t 和 p_t 进行 ADF 检验, 得到的 p 值分别为 0.4746 和 0.2155, 故认为 P_t 和 p_t 存在单位根, 从而是不平稳的.

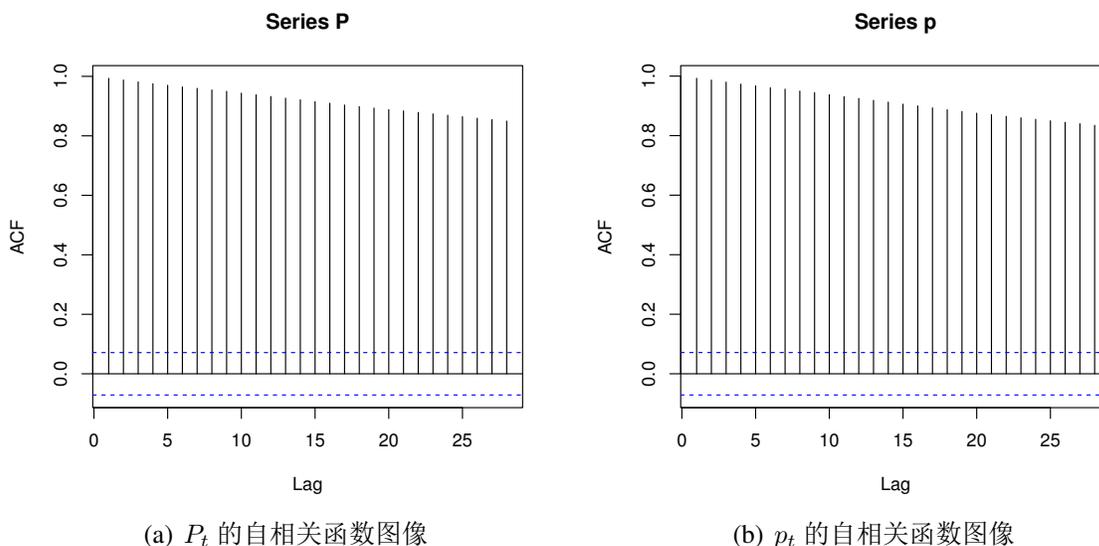


图 4: P_t 和 p_t 对应的自相关函数图像

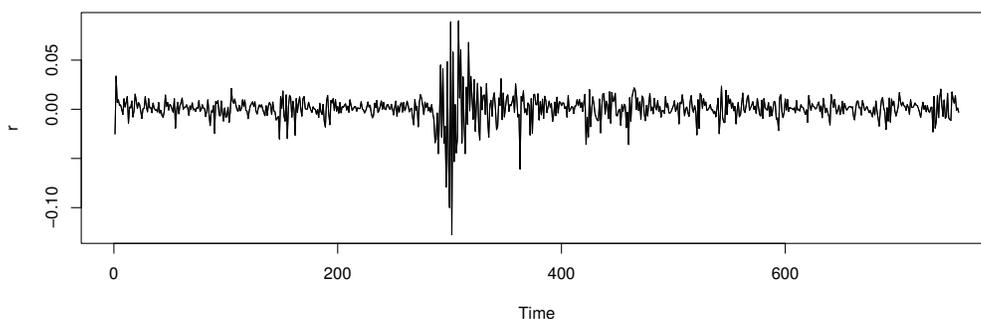


图 5: 对数收益率 r_t 随时间 t 变化的图像

为了处理不平稳的数据, 可以采用差分的方法. 考虑对对数收盘价 p_t 进行差分, 记

$$r_t := \nabla p_t = p_t - p_{t-1} = \ln \frac{P_t}{P_{t-1}}, \quad t = 2, 3, \dots, T,$$

这通常被称为对数收益率. 绘制出 $t-r_t$ 图像如图 5 所示. 对 r_t 进行 ADF 检验, 所得的 p 值为 0.01, 说明一次差分即可得到平稳的数据. 在建立模型时, 也可以考虑对 r_t 建立 ARMA(p, q) 模型, 这相当于对 p_t 建立了 ARIMA($p, 1, q$) 模型.

3 数据建模: ARMA 模型与 ARIMA 模型

3.1 模型初筛: ACF 函数、PACF 函数与 EACF 函数

在建立 MA 模型时,我们可以使用自相关函数 (ACF) 的图像来估计阶数;相应地,在建立 AR 模型时,我们可以使用偏自相关函数 (PACF) 的图像来估计阶数;对于 ARMA 模型而言,使用 ACF 和 PACF 都难以解决问题,但可以用推广的自相关系数 (EACF) 来估计阶数.

对于 P_t , 绘制 ACF 图和 PACF 图如图 6 所示. 发现此时 ACF 函数接近 1 且缓慢下降, 说明 P_t 并不是不相关的, 甚至具有较强的记忆性. 而 PACF 函数仅在 1 处有取值, 其他地方的函数值都落在置信区间内, 从而对 P_t 建立 AR(1) 模型 (相当于 MA(∞) 模型) 是合理的.

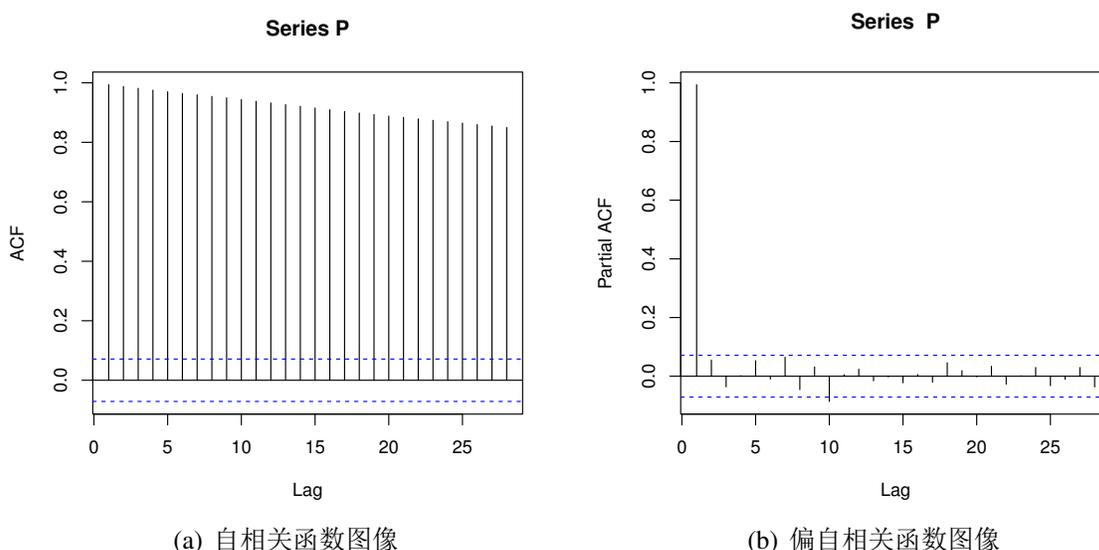


图 6: P_t 的自相关函数图像和偏自相关函数图像

另外, 对于 P_t , 计算得到的 EACF 为:

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1	x	x	o	x	o	x	x	x	x	o	o	x	x	
2	x	x	o	x	o	o	x	o	x	o	o	o	o	
3	x	x	o	x	o	o	x	o	x	o	o	o	o	
4	x	x	o	x	o	o	o	x	o	o	o	o	o	
5	x	x	x	x	o	o	o	o	x	o	o	o	o	
6	x	x	x	x	o	o	o	o	x	o	o	o	o	
7	x	x	x	x	x	x	o	o	x	o	o	o	o	

P_t 的 EACF 函数的三角形顶点大致落在 (2, 4) 处, 从而可以考虑对 P_t 建立 ARMA(2, 4) 模型. 然而, 并不能保证对任何的 $p \geq 2$ 及 $q \geq 4$, 都有 $EACF(p, q) = 0$, 例如当 $q = 8$ 时, $EACF(p, q) \neq 0$, 这说明了这一判断准则的不合理性.

对于 p_t , 绘制 ACF 图和 PACF 图如图 7 所示. 图 7 和图 6 极为相像, 考虑到如果 P_t 的相关性较强, 直觉上 $p_t = \ln P_t$ 的相关性应该也较强. 略有不同的是, 根据 PACF 图的结果, 可以考虑对 p_t 建立 AR(2) 模型, 模型的阶数略有提高.

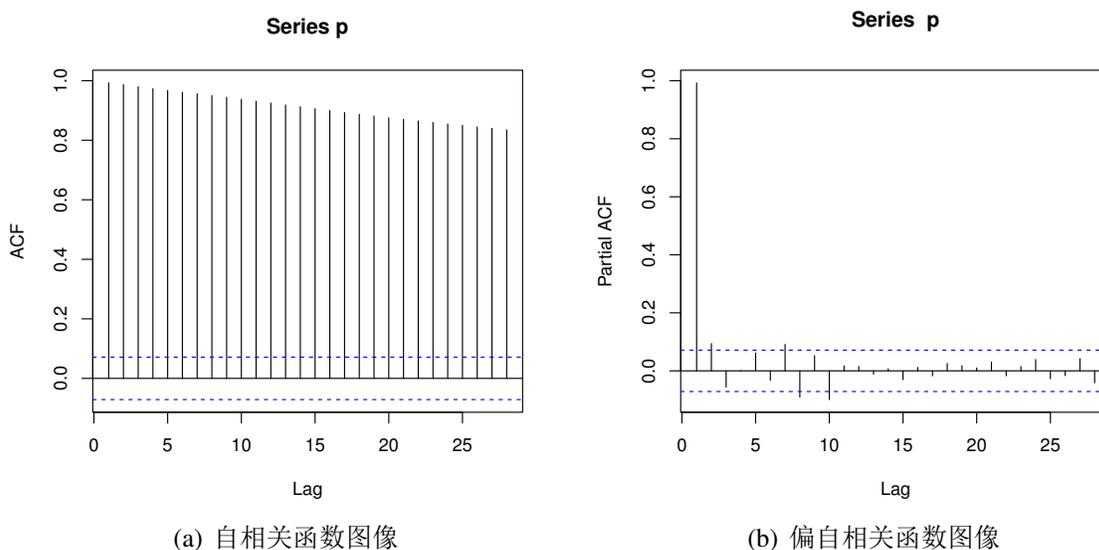


图 7: p_t 的自相关函数图像和偏自相关函数图像

另外, 对于 p_t , 计算得到的 EACF 为:

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1	x	x	o	x	x	x	x	x	x	o	x	x	x	
2	x	x	o	o	o	o	x	o	x	x	o	o	o	o
3	x	x	o	x	o	o	x	o	x	o	o	o	o	x
4	x	x	o	x	o	o	x	o	x	o	o	o	o	o
5	x	x	x	x	o	o	o	o	x	o	o	o	o	o
6	x	x	x	x	x	x	o	o	x	o	o	o	o	o
7	x	x	x	x	x	x	o	o	x	o	o	o	o	o

p_t 的 EACF 函数的三角形顶点大致落在 (2, 3) 处, 从而可以对 p_t 建立 ARMA(2, 3) 模型. 和 P_t 的 EACF 函数类似, 当 $q = 8$ 时函数值均不为 0.

对于 r_t , 绘制 ACF 图和 PACF 图如图 8 所示.

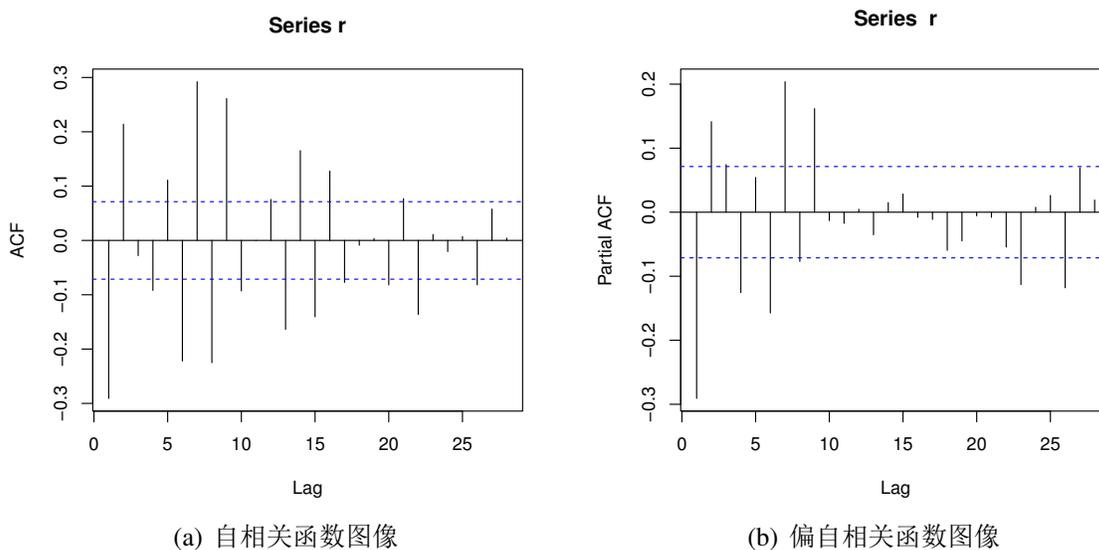


图 8: r_t 的自相关函数图像和偏自相关函数图像

另外, 对于 r_t , 计算得到的 EACF 为:

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	o	x	x	x	x	x	x	x	o	x	x	x
1	x	x	o	o	o	o	x	o	x	x	o	o	o	o
2	x	x	o	x	o	o	x	o	x	o	o	o	o	o
3	x	x	o	x	o	o	x	o	x	o	o	x	o	o
4	x	x	x	x	o	o	o	o	x	o	o	o	o	o
5	x	x	x	x	o	x	o	o	x	o	o	o	o	o
6	x	x	x	x	x	x	o	o	x	o	o	o	o	o
7	x	x	o	x	o	x	x	o	o	o	o	o	o	o

根据上述结果, 可以对 r_t 建立模型 MA(2)、AR(2) 和 ARMA(1, 2).

对于 P_t , p_t 和 r_t , 根据 ACF、PACF 和 EACF 所得到的模型如表 3 所示. 需要注意, r_t 的 ARMA(p, q) 模型实质上就是 p_t 的 ARIMA($p, 1, q$) 模型, 在表 3 中不再另外列出.

表 3: 由 ACF、PACF 和 EACF 得到的 MA 模型、AR 模型和 ARMA 模型

变量	MA 模型	AR 模型	ARMA 模型
P_t	-	AR(1)	ARMA(2, 4)
p_t	-	AR(2)	ARMA(2, 3)
r_t	MA(2)	AR(4)	ARMA(1, 2)

3.2 模型选择: 最小化 AIC 准则

另外一种进行模型选择的方式是在假定 $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ 的条件下, 计算模型的 AIC 和 BIC, 并选择使得 AIC 或者 BIC 最小的模型. 在本篇报告中, 主要考虑的是最小化 AIC, 并且为了方便, 仅考虑对 p_t 和 r_t 建立模型.

对于 p_t , 各个模型的 AIC 如表 4 所示. 表中, 空出来的部分 (例如 ARMA(3, 3)) 代表建立的模型不是平稳的, 或者进行参数估计时算法不收敛, 导致无法计算 AIC.

表 4: p_t 的 ARMA 模型的 AIC

AR/MA	0	1	2	3	4	5
0	-446.2215	-1426.307	-2104.07	-2625.097	-2931.41	-3250.085
1	-4269.619	-4314.26	-4351.963	-4352.314	-4351.536	-4349.143
2	-4331.163	-4338.882	-4338.294	-4372.177	-4379.545	-4378.7
3	-4346.388	-4346.903	-4350.897	-	-4411.69	-
4	-4349.886	-4398.016	-4412.484	-	-4409.81	-4407.952
5	-4358.985	-4401.016	-4411.971	-4410.062	-	-4418.987

对于 r_t 模型, 建立的 ARMA(p, q) 模型相当于是 p_t 的 ARIMA($p, 1, q$) 模型. 各个模型的 AIC 如表 5 所示.

表 5: r_t 的 ARMA 模型 (也即 p_t 的 ARIMA 模型, $d = 1$) 的 AIC

AR/MA	0	1	2	3	4	5
0	-4274.138	-4321.189	-4357.754	-4358.596	-4357.39	-4355.799
1	-4339.095	-4345.62	-4357.895	-4357.69	-4386.073	-4385.047
2	-4352.764	-4352.362	-4356.563	-4355.847	-4418.071	-4382.426
3	-4355.065	-4403.846	-4362.345	-4417.018	-4415.055	-4418.239
4	-4365.146	-4407.489	-4417.687	-4415.929	-4426.883	-4425.76
5	-4365.36	-4406.836	-4403.653	-4410.474	-4425.699	-4423.76

综合以上结果, 为最小化 AIC, 对 r_t 建立的模型为 ARMA(4, 4), 此时对 p_t 建立的模型为 ARIMA(4, 1, 4), AIC 的值为 -4426.883.

3.3 参数估计: ML、CSS 与 CSS-ML

根据以上过程, 我们对 r_t 建立了 ARMA(4, 4) 模型, 也即

$$r_t = c + \sum_{i=1}^4 b_i r_{t-i} + \varepsilon_t + \sum_{j=1}^4 a_j \varepsilon_{t-j}.$$

然而, 模型里面涉及的参数 c , $\mathbf{b} = (b_1, b_2, b_3, b_4)$ 和 $\mathbf{a} = (a_1, a_2, a_3, a_4)$ 还没有确定下来. 为了估计 $(c, \mathbf{b}, \mathbf{a})$, 我们可以考虑的方法有极大似然估计 (ML) 和最小二乘估计 (CSS). 其中, ML 方法需要假定 $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. 除此之外, R 还提供了使用最小二乘估计初值、使用极大似然估计参数的 CSS-ML 方法. 分别使用这些估计方法, 所得的参数的值见表 6.

表 6: ARMA(4, 4) 模型的参数估计

方法	\hat{c}	\hat{b}_1	\hat{b}_2	\hat{b}_3	\hat{b}_4	\hat{a}_1	\hat{a}_2	\hat{a}_3	\hat{a}_4
ML	9e-04	-0.5746	0.2124	-0.5326	-0.7884	0.4126	-0.1346	0.6688	0.5890
CSS	9e-04	-0.6555	0.1600	-0.4306	-0.6854	0.4978	-0.0751	0.5961	0.5190
CSS-ML	9e-04	-0.5756	0.2117	-0.5322	-0.7883	0.4138	-0.1341	0.6686	0.5894

根据表 6, 使用三种方法估计出来的参数都较为接近. 考虑到在应用 AIC 选择模型时, 假定了 $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, 因此我们更倾向于使用 CSS-ML 方法估计出的参数. 从而, 我们最终建立的模型为

$$r_t = 9 \times 10^{-4} - 0.5756 \times r_{t-1} + 0.2117 \times r_{t-2} - 0.5322 \times r_{t-3} - 0.7883 \times r_{t-4} + \varepsilon_t + 0.4138 \times \varepsilon_{t-1} - 0.1341 \times \varepsilon_{t-2} + 0.6686 \times \varepsilon_{t-3} + 0.5894 \times \varepsilon_{t-4}, \quad (1)$$

其中对数收益率

$$r_t = \nabla p_t = p_t - p_{t-1} = \ln P_t - \ln P_{t-1} = \ln \frac{P_t}{P_{t-1}}.$$

根据模型(1), 估计得到 $\hat{\sigma}^2 = 0.0001638$. 此时, 对数似然为 2223.44, AIC 为-4426.88. $(\hat{c}, \hat{\mathbf{b}}, \hat{\mathbf{a}})$ 的估计值和误差如表 7 所示.

表 7: ARMA(4, 4) 模型的 ML 估计

类别	\hat{c}	\hat{b}_1	\hat{b}_2	\hat{b}_3	\hat{b}_4	\hat{a}_1	\hat{a}_2	\hat{a}_3	\hat{a}_4
估计值	9e-04	-0.5756	0.2117	-0.5322	-0.7883	0.4138	-0.1341	0.6686	0.5894
标准差	4e-04	0.0642	0.0693	0.0774	0.0577	0.0745	0.0601	0.0640	0.0638

3.4 模型应用：对数收益率与对数收盘价的预测

在前面几节中，我们完成了模型选择、模型参数的估计等工作，成功地建立了模型(1)。基于我们所建立的模型(1)，也即对 r_t 的 ARMA(4, 4) 模型，可以通过取条件期望的方式，对未来数据进行简单的预测。首先，对对数收益率 r_t 进行预测，所得的结果如图 9 所示。

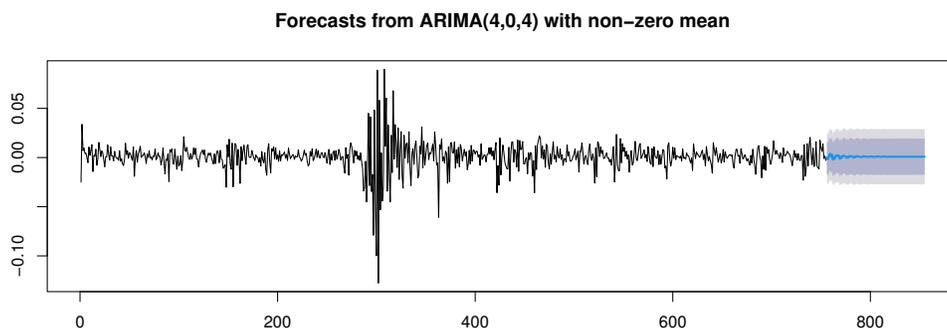


图 9: 利用 ARMA(4, 4) 模型对 r_t 进行预测

模型(1)对应着 p_t 的 ARIMA(4, 1, 4) 模型。在图 9 的预测的基础上，我们可以得到对数收盘价的预测，结果如图 10 所示。

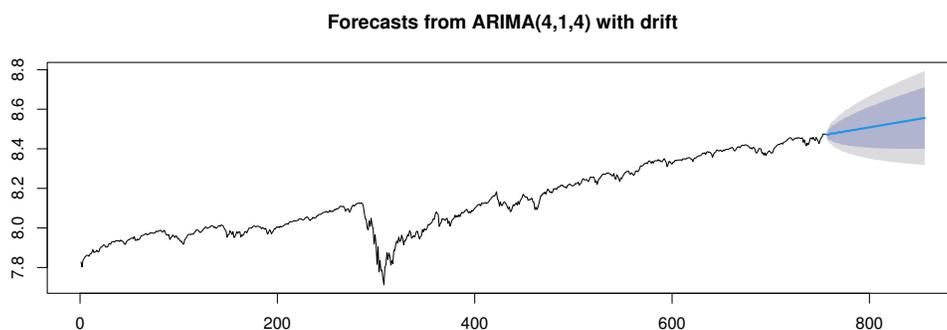


图 10: 利用 ARIMA(4, 1, 4) 模型对 p_t 进行预测

图 9 表明，对 r 的预测的方差较为稳定，而不像已知数据一样出现“波动率聚集”的现象，这是较不合理的。另外，将图 10 的预测结果与真实数据进行比较，会发现与真实数据有一定的偏差。

3.5 残差分析：正态性检验与相关性检验

本节中，我们将对模型(1)的残差 ε_t 进行诊断。根据该模型，作出 $t - \varepsilon_t$ 图如图 11 所示。

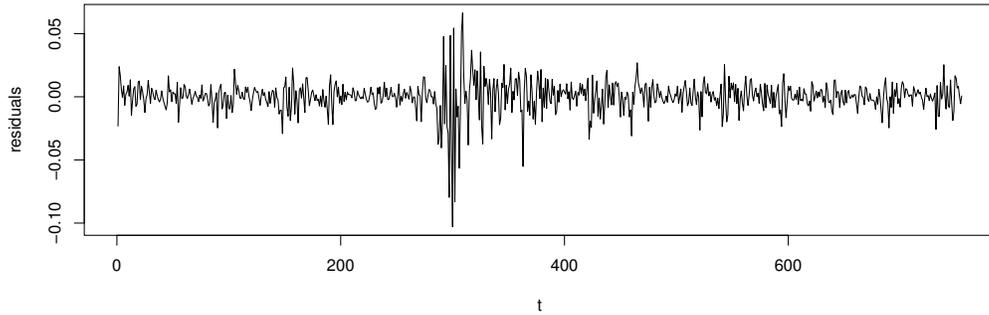


图 11: 残差 ε_t 随时间 t 变化的图像

在建立模型时, 我们假定 $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$, 得到了模型(1). 然而, 真实数据的残差 ε_t 不一定是来自正态分布的. 为了对残差进行正态性检验, 作出残差的 Q-Q 图如图 12 所示. 注意到在图中, 大部分点围绕着中间的直线, 而到了两边, 有部分点离直线较远, 说明了 ε_t 在一定程度上可以近似为正态分布, 但是尾部较厚.

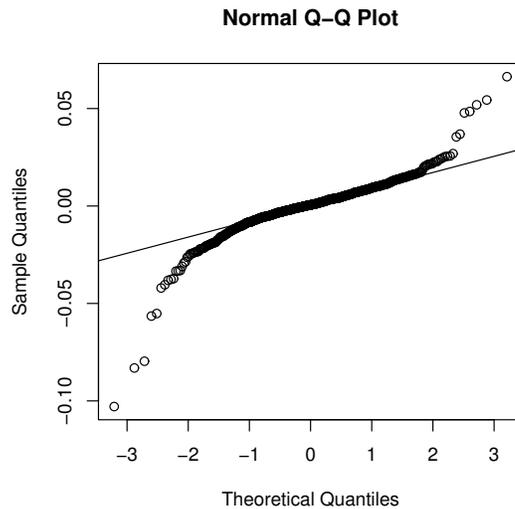
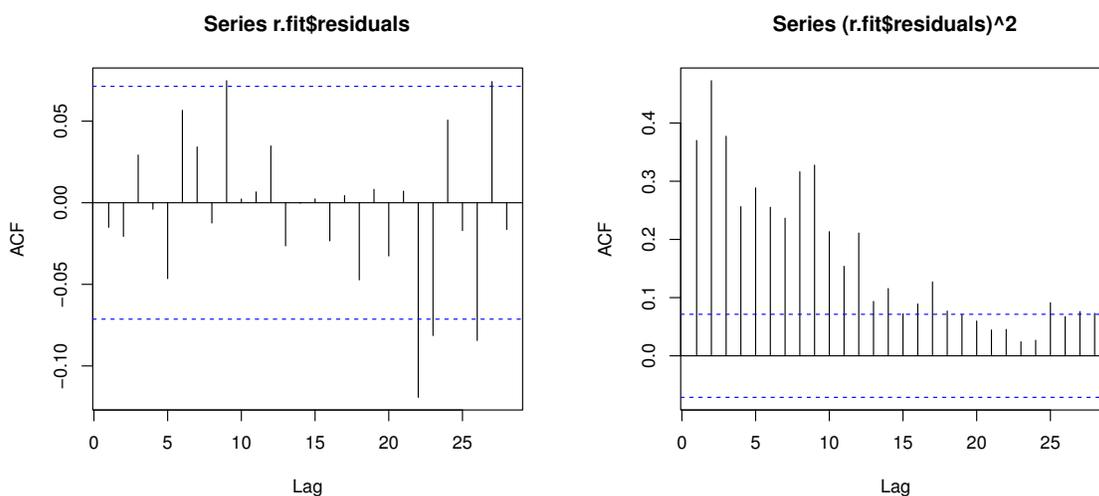


图 12: 残差 ε_t 的 Q-Q 图

使用 Q-Q 图还不能做出残差来自正态分布的判断. 在这里, 进一步对残差进行 JB 检验, 计算得到 $\chi^2 = 5067.1$, 检验的 p 值小于 $2.2e-16$, 从而拒绝原假设, 也即认为残差并不是来自正态分布的.

在建立模型时, 我们还假定了残差至少是白噪声过程. 从而, 它们一定是不相关的. 首先, 作出自相关函数 (ACF) 的图像如图 13 所示. 根据图像可以看出, ε_t 的 ACF 的取值几乎都落在置信区间内, 从而可以认为 ε_t 之间是不相关的. 然而, 考虑对 ε_t^2 绘制 ACF 图像时, 发现

相关性仍然较强.⁵



(a) ε_t 的自相关函数图像

(b) ε_t^2 的自相关函数图像

图 13: 残差 ε_t 的自相关函数图像

为了进一步检验残差之间的相关性, 在这里进行 LB 检验, 得到的 p 值如图 14 所示. 根据图像得知, 可以认为 ε_t 之间的相关性较弱.

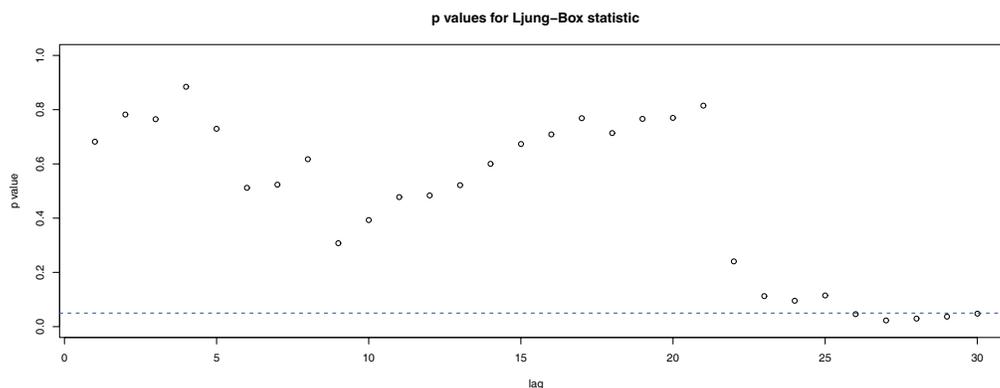


图 14: 残差 ε_t 进行 LB 检验的 p 值

本节所得的结果表明, 残差的正态性和相关性都不是非常理想. 我们认为, 残差可能与过去的残差有一定的相关性, 并且方差并不是恒定的. 在接下来, 我们希望能够对残差进行进一步的建模, 使模型更加精确. 借助 ARCH 模型与 GARCH 模型, 这一点是可以做到的.

⁵关于检验 ε_t^2 的相关性的合理性, 详见下一节.

3.6 理论分析: 关于残差的独立性与相关性的论证

在建模的时候, 我们假定白噪声 ε_t 是一个白噪声过程. 具体地说, 白噪声过程 $\{\varepsilon_t, t \geq 1\}$ 需要满足: 对任意的 $t \geq 1$, 都有

- ε_t 的均值为 0, 也即 $\mathbb{E}(\varepsilon_t) = 0$;
- ε_{t+1} 与 ε_t 不相关, 也即 $\mathbb{E}(\varepsilon_{t+1}\varepsilon_t) = \mathbb{E}(\varepsilon_{t+1}) \cdot \mathbb{E}(\varepsilon_t)$.

在建立模型时, 我们不但检验了 ε_t 之间的相关性, 还检验了 ε_t^2 之间的相关性, 并且在建立 ARMA 模型时, 发现: 即使 ε_t 之间的相关性较弱, ε_t^2 之间仍可能有一定的相关性. 这是否是偶然现象? 能否在理论上解释?

首先, 考虑随机变量 X, Y 之间的相关性. 我们举出一个反例, 来说明 X 与 Y 不相关时, X^2 与 Y^2 仍可能不是不相关的. 甚至, 如果进一步加上 X 与 Y 同分布的条件, 也不能推出 X^2 与 Y^2 不相关.

例 3.1. 设 $\theta \sim \mathcal{U}(0, 2\pi)$, $X = \cos \theta$, $Y = \sin \theta$.

- (1) X 与 Y 不相关, 且是正交的;
- (2) X 与 Y 同分布;
- (3) X^2 与 Y^2 不是不相关的.

证明. (1) 计算得

$$\mathbb{E}X = \frac{1}{2\pi} \int_0^{2\pi} \cos \theta d\theta = 0, \quad \mathbb{E}Y = \frac{1}{2\pi} \int_0^{2\pi} \sin \theta d\theta = 0,$$

以及

$$\mathbb{E}(XY) = \frac{1}{2\pi} \int_0^{2\pi} \cos \theta \sin \theta d\theta = \frac{1}{4\pi} \int_0^{2\pi} \sin 2\theta d\theta = 0,$$

因此 X 与 Y 不相关, 且是正交的.

(2) 注意到 $Y = \sin \theta = \cos\left(\theta + \frac{\pi}{2}\right)$, 因此 X 与 Y 同分布.

(3) 计算得

$$\mathbb{E}X^2 = \frac{1}{2\pi} \int_0^{2\pi} \cos^2 \theta d\theta = \frac{1}{2}, \quad \mathbb{E}Y^2 = \frac{1}{2\pi} \int_0^{2\pi} \sin^2 \theta d\theta = \frac{1}{2},$$

以及

$$\mathbb{E}(X^2Y^2) = \frac{1}{2\pi} \int_0^{2\pi} \cos^2 \theta \sin^2 \theta d\theta = \frac{1}{8},$$

因此 $\mathbb{E}(X^2Y^2) \neq \mathbb{E}(X^2) \cdot \mathbb{E}(Y^2)$, 也即 X^2 和 Y^2 不是不相关的. □

在上例的基础上, 我们来构造白噪声序列, 使得相邻两项的平方之间是不相关的.

例 3.2. 续上例, 设 $\theta \sim \mathcal{U}(0, 2\pi)$, 对 $k \geq 1$, 令

$$\varepsilon_{2k-1} = \cos \theta, \quad \varepsilon_{2k} = \sin \theta,$$

则 $\{\varepsilon_t, t \geq 1\}$ 是白噪声过程, 但是对任意的 $t \geq 1$, ε_{t+1}^2 与 ε_t^2 都不是不相关的.

接下来, 我们探讨需要加入什么样的条件, 来保证 ε_{t+1}^2 与 ε_t^2 之间的不相关性. 为此, 我们首先考虑随机变量 X 与 Y 的情形.

定理 3.1. 设 X, Y 是 $(\Omega, \mathcal{F}, \mathbb{P})$ 上的随机变量, $\mathcal{G} \subset \mathcal{F}$ 是子 σ -域, X 是关于 \mathcal{G} 可测的, Y 与 \mathcal{G} 独立.

- (1) X 与 Y 独立;
- (2) X 与 Y 不相关;
- (3) 对任意的 Borel 可测函数 f, g , $f(X)$ 与 $g(Y)$ 不相关.

证明. (1) 对任意的 $A, B \in \mathcal{B}_{\mathbb{R}}$, 记 $G = (X \in A)$, 由 X 是关于 \mathcal{G} 可测的, 知 $G \in \mathcal{G}$, 因此

$$\mathbb{P}(X \in A, Y \in B) = \mathbb{P}(G, Y \in B) = \mathbb{P}(G) \cdot \mathbb{P}(Y \in B) = \mathbb{P}(X \in A) \cdot \mathbb{P}(Y \in B),$$

这便说明了 X 与 Y 独立.

(2) 计算得

$$\mathbb{E}(XY|\mathcal{G}) = X \cdot \mathbb{E}(Y|\mathcal{G}) = X \cdot \mathbb{E}(Y),$$

因此

$$\mathbb{E}(XY) = \mathbb{E}[\mathbb{E}(XY|\mathcal{F})] = \mathbb{E}(X) \cdot \mathbb{E}(Y),$$

这便说明了 X 与 Y 不相关.

(3) 由 X 是关于 \mathcal{G} 可测的, 知 $X^{-1}(\mathcal{B}_{\mathbb{R}}) \subset \mathcal{F}$, 因此

$$[f(X)]^{-1}(\mathcal{B}_{\mathbb{R}}) = X^{-1} \circ f^{-1}(\mathcal{B}_{\mathbb{R}}) \subset X^{-1}(\mathcal{B}_{\mathbb{R}}) \subset \mathcal{F},$$

因此 $f(X)$ 是关于 \mathcal{G} 可测的; 且由 Y 与 \mathcal{G} 独立, 知对任意的 $G \in \mathcal{G}$, Y 与 G 独立, 因此对任意的 $A \in \mathcal{B}_{\mathbb{R}}$, 都有

$$\mathbb{P}(g(Y) \in A, G) = \mathbb{P}(Y \in g^{-1}(A), G) = \mathbb{P}(Y \in g^{-1}(A)) \cdot \mathbb{P}(G),$$

因此 $g(Y)$ 与 \mathcal{G} 独立. 对 $f(X)$ 与 $g(Y)$ 直接应用 (1) 中结论即可. □

在上述结论的基础上, 考虑随机过程 $\{\varepsilon_t, t \geq 1\}$, 则有如下的结论.

推论. 设 $\{\varepsilon_t, t \geq 1\}$ 是 $\{\mathcal{F}_t, t \geq 1\}$ 适应的随机过程, 且对任意的 $t \geq 1$, 都有 ε_{t+1} 与 \mathcal{F}_t 独立, 则 ε_{t+1}^2 与 ε_t^2 不相关.

推论. 设 $\{\varepsilon_t, t \geq 1\}$ 是 $\{\mathcal{F}_t, t \geq 1\}$ 适应的随机过程, 如果对某个 $t \geq 1$, ε_{t+1}^2 与 ε_t^2 不是不相关的, 则 ε_{t+1} 与 \mathcal{F}_t 不独立.

据此, 我们可以得到: 若残差的平方 ε_{t+1}^2 与 ε_t^2 是相关的, 则可以认为 ε_{t+1} 与 \mathcal{F}_t 不独立, 对于离散的随机过程而言, 通常取

$$\mathcal{F}_t = \sigma(\varepsilon_1, \dots, \varepsilon_t),$$

也就是说 ε_{t+1} 仍然受到 $\varepsilon_1, \dots, \varepsilon_t$ 的影响.

4 残差建模: GARCH 模型

4.1 数据分析: ARCH 效应的 LM 检验

根据图 11, 我们猜测残差一定的异方差结构. 在建立 GARCH 模型之前, 我们首先希望对对数收益率 r_t , 进行 ARCH 效应的 LM 检验.

表 8: 对 r_t 进行 ARCH 效应的 LM 检验的检验统计量与 p 值

滞后项数	检验统计量	p 值	滞后项数	检验统计量	p 值
1	103.8261	0	6	544.8988	0
2	273.5184	0	7	587.614	0
3	381.7222	0	8	664.093	0
4	431.724	0	9	746.3749	0
5	495.1865	0	10	781.2893	0

根据表 8 中的检验结果, 得到的结论是 r_t 存在 ARCH 效应, 从而我们接下来要做的, 是在对 r_t 建立了 ARMA 模型的基础上, 进一步对残差 ε_t 建立 GARCH 模型, 从而对 r_t 建立了 ARMA-GARCH 模型. 最后, 我们希望得到的模型是

$$\begin{cases} r_t = \mu + \beta_1 r_{t-1} + \cdots + \beta_{p_1} r_{t-p_1} + X_t + \alpha_1 X_{t-1} + \cdots + \alpha_{q_1} X_{t-q_1}, \\ X_t = \sigma_t \varepsilon_t, \\ \sigma_t^2 = a_0 + a_1 X_{t-1}^2 + \cdots + a_{p_2} X_{t-p_2}^2 + b_1 \sigma_{t-1}^2 + \cdots + b_{q_2} X_{t-q_2}, \end{cases}$$

为了方便, 我们将上面的模型记作 ARMA(p_1, q_1)-GARCH(p_2, q_2). 另外, 表 8 的一个结果是, 随着滞后项数增大, 检验统计量的值也会增大.

4.2 模型选择: 最小化 AIC 准则和阶数

根据建立线性模型时的结果, 对 r_t 我们首先建立了 ARMA(4, 4) 模型. 从而我们选取 $p_1 = 4, q_1 = 4$. 接下来, 考虑残差 X_t , 我们希望对其建立 GARCH(p, q) 模型. 为了选出最合适的模型, 首先计算各个模型的 AIC.

表 9: X_t 的 GARCH 模型的 AIC

p/q	0	1	2	3
0	-	-4698.208	-4788	-4803.258
1	-4440.277	-4870.881	-4865.492	-4858.991

2	-4434.909	-4857.031	-4859.226	-4856.918
3	-4427.832	-4844.895	-4851.392	-4843.464

根据表 9 中结果, 为最小化 AIC, 所选择的模型为 X_t 的 GARCH(1,1) 模型. 考虑到 GARCH 模型阶数较高时, 模型会太过复杂, 从而选取 GARCH(1,1) 模型是合理的.

4.3 参数估计: 对 GARCH 模型应用 ML

根据上面的结果, 我们选取 $p_2 = 1, q_2 = 1$, 并且根据 $p_1 = 4, q_1 = 4$, 最终选择的模型是 r_t 的 ARMA(4,4)-GARCH(1,1) 模型. 使用极大似然估计, 估计得到的参数如表 10 所示.

表 10: GARCH(1,1) 模型的 ML 估计

类别	\hat{a}_0	\hat{a}_1	\hat{b}_1
估计值	6.468e-06	2.470e-01	7.129e-01
标准差	1.196e-06	3.453e-02	3.312e-02

根据估计的结果, 结合表 7 中的结果, 可以建立模型(2): 假定 $\varepsilon_t \sim \mathcal{N}(0, 1)$, 则有

$$\begin{cases} X_t = \sigma_t \varepsilon_t, \\ \sigma_t^2 = 6.468 \times 10^{-6} + 0.247 \times X_{t-1}^2 + 0.7129 \times \sigma_{t-1}^2. \end{cases} \quad (2)$$

根据模型(2), 绘制波动率的置信区间如图 15 所示.

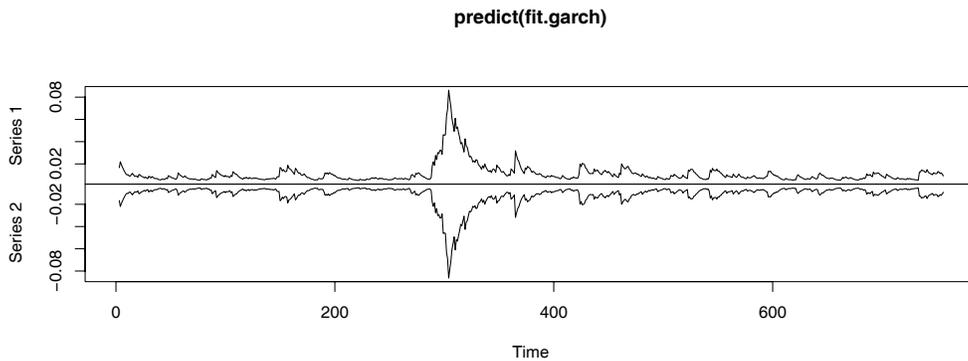


图 15: 波动率的置信区间

最终, 我们所建立的模型是(1)和(2)的结合, 参数的估计值和标准差见表 7 和表 10.

4.4 残差分析: 正态性检验与相关性检验

本节中, 我们将仿照对模型(1)的残差分析手段, 对模型(2)的残差 $\varepsilon_t = \frac{X_t}{\sigma_t}$ 进行诊断. 根据模型的结果, 作出 $t - \varepsilon_t$ 图如图 16 所示. 根据图像可以看出, 相比图 11, 残差的分布更加均匀, 没有出现 $t = 300$ 附近的剧烈变化.

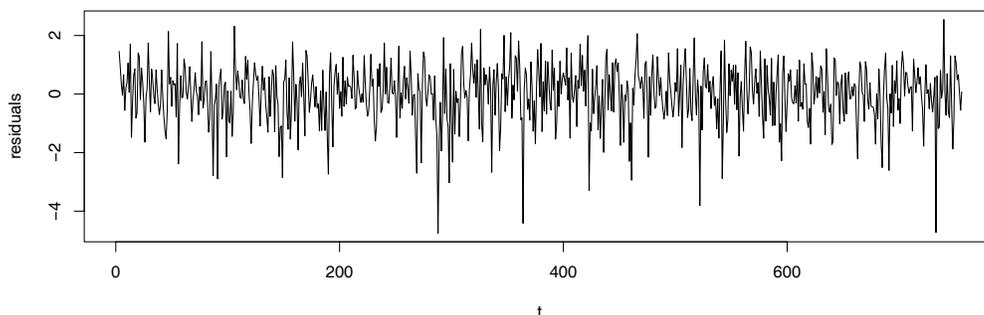


图 16: 残差 ε_t 随时间 t 变化的图像

在建立模型时, 我们假定 $\varepsilon_t \sim \text{IID}(0, 1)$. 最常用的分布是正态分布, 在这里首先对残差进行正态性检验. 作出频率分布直方图与 Q-Q 图如图 17 所示. 根据图 17, 发现 Q-Q 图中的点集中在直线附近, 但是左下方有些厚尾的情况. 另外, 对 ε_t 进行 JB 检验, 计算得到 $\chi^2 = 189.74$, 检验的 p 值小于 $2.2e-16$, 从而拒绝原假设, 也即认为残差并不服从正态分布.

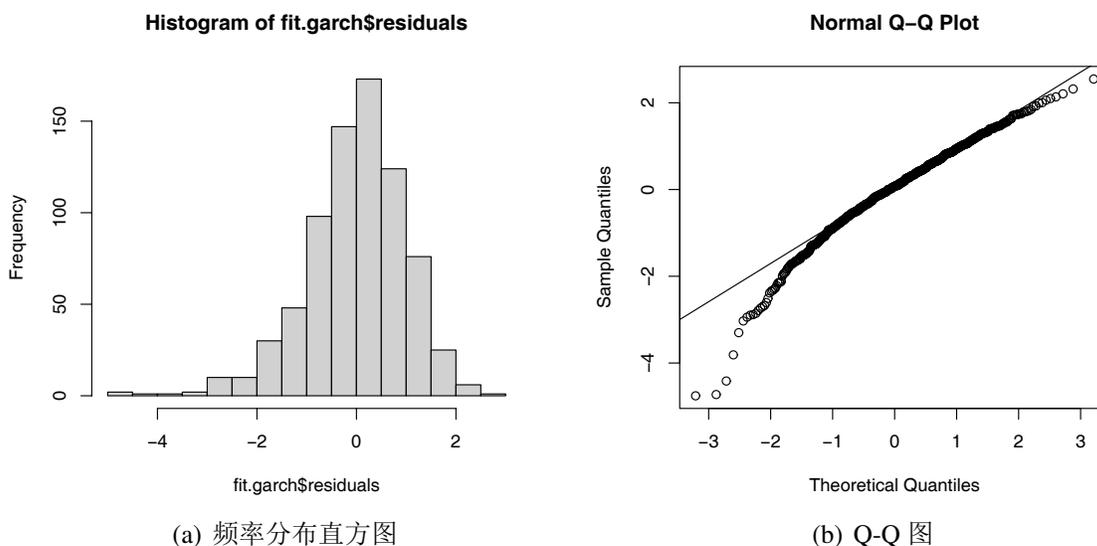


图 17: 残差 ε_t 的频率分布直方图与 Q-Q 图

更为重要的是, 我们需要对残差进行相关性检验. 绘制残差的自相关函数图像如图 18 所

示. 根据图像可以看出, 此时 ε_t 和 ε_t^2 之间的相关性较好, 从而残差的相关性较为理想. 另外, 进行 LB 检验, 计算得到 $\chi^2 = 0.32312$, 检验的 p 值为 0.5697, 从而接受原假设, 也即认为此时残差不具有相关性.

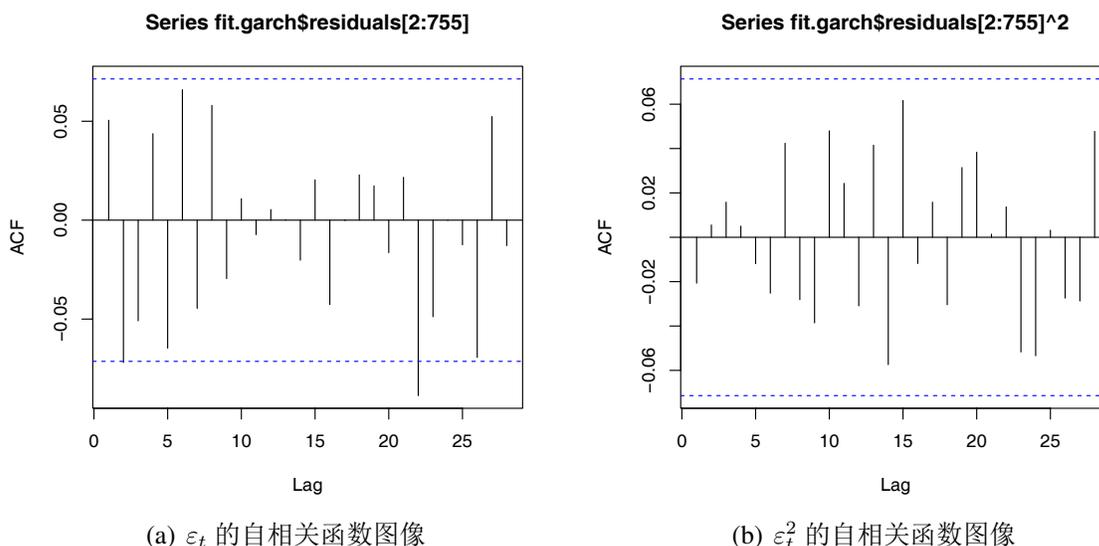


图 18: 残差 ε_t 的自相关函数图像

残差分析表明, 在建立 GARCH(1, 1) 模型时, 我们对残差的假定是合理的, 建立的模型也很好地去除了残差的 ARCH 效应. 考虑到金融数据具有后尾的特征, 通常不适合用正态分布来近似, 而选用 t 分布来描述.

4.5 模型应用: 风险值 (VaR) 的估计

风险值 (VaR) 可以描述在某个时间点的极端损失. 在建立了时间序列模型之后, 为了给出投资建议, 本篇报告中给出 VaR 的估计值, 见图 19. 图中, 蓝色的线表示 $\alpha = 0.05$ 时的 VaR, 红色的线表示 $\alpha = 0.025$ 时的 VaR. 为了给出更合理的估计, 这里使用的估计方法是使用样本分位数.

根据图 19 可以看出, 当 $\alpha = 0.025$ 时, 估计得到的 VaR 总是比 $\alpha = 0.05$ 时的 VaR 的值更小, 这是合理的. 同时, 对于大部分时刻来说, 对数收益率总是在估计的 VaR 值之上的, 这是因为 VaR 刻画的是极端损失. 当然, 图像上 $t = 300$ 附近的点尤其接近 VaR, 该时刻附近的对数收益率波动也较严重.

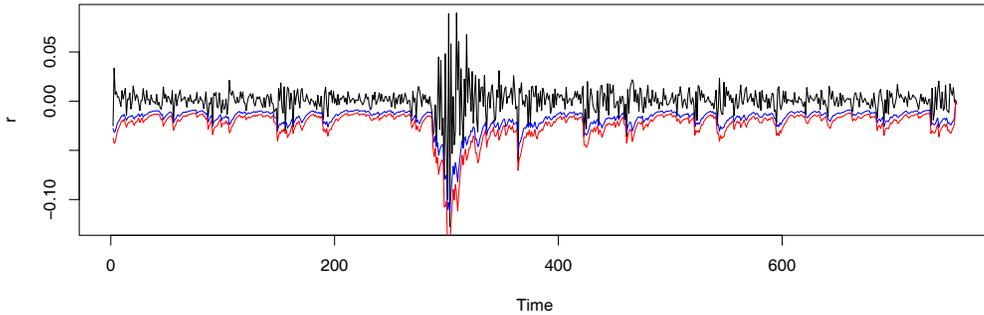


图 19: 对数收益率的 VaR 的估计值

5 模型推广: 基于动态神经网络的 NAR 模型

5.1 模型概述: 动态神经网络的基本设置

神经网络可以挖掘数据的潜在规律, 建立更为合理的模型并用于预测. 作为神经网络的应用, 笔者在本节中尝试基于动态神经网络, 对时间序列建立了非线性自回归 (NAR) 模型. 在本节中, 我们希望解决的问题是: 借助对数收益率 r_t 滞后 d 项的数据, 给出 r_t 的预测值.

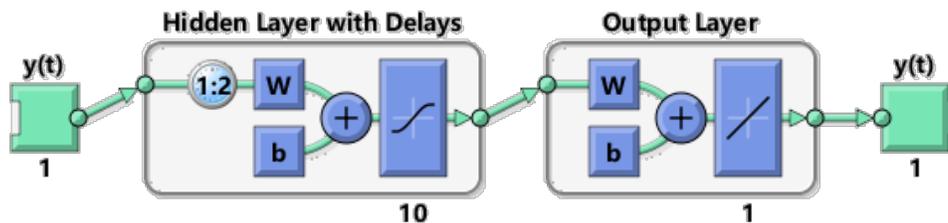


图 20: 动态神经网络示意图

本节中建立神经网络模型时, 使用了 MATLAB 的 Neural Time Series 应用. 为了合理地应用数据, 并防止过拟合现象, 本节中选取:

- 每次训练时, 将 70% 的数据作为训练集, 15% 的数据作为验证集, 15% 的数据作为测试集, 通过测试集的结果计算 MSE;
- 神经网络的隐藏层数目 $n = 10$, 滞后项 $d = 2$;
- 训练的算法为 Bayes 正则化, 该算法通常需要更多的时间, 但可以产生良好的泛化效果. 当实现自适应权重最小化 (或称为正则化) 时, 算法停止.

通过训练, 得到了最终的动态神经网络 NAR 模型.

5.2 结果分析: 预测值与残差的分析

首先, 绘制出基于 NAR 模型的预测值与真实值的图像如图 21 所示. 根据图像发现, 尽管预测仍然有一定的误差, 但是对对数收益率的预测结果与真实值较为接近, 特别地在 $t = 300$ 附近有剧烈的波动, 符合真实数据的特点. 通过测试得到, 模型的 MSE 为 $1.10556e-4$, 说明模型较为精确.

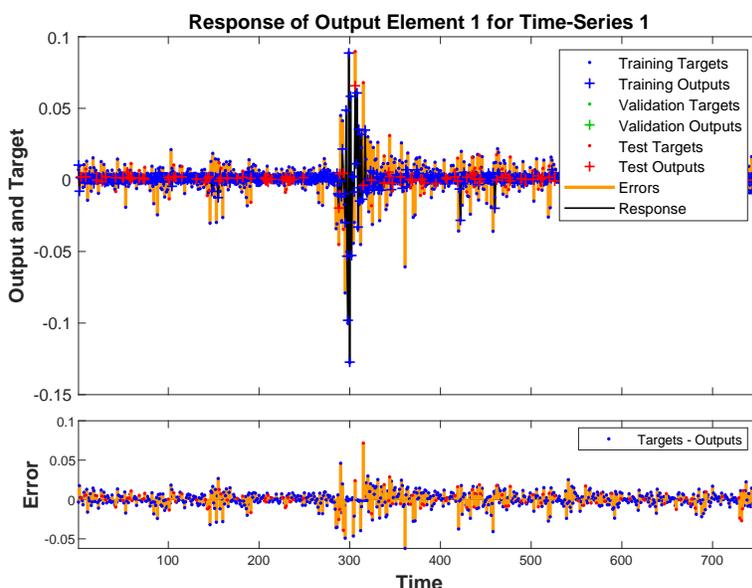


图 21: 基于 NAR 模型的预测值与真实值

同样地, 在这里需要对残差进行分析. 对残差绘制出 ACF 函数图像, 如图 22 所示. 通过图 22 可以看出, 残差之间的相关性较弱, 从而残差的白噪声假定是合理的.

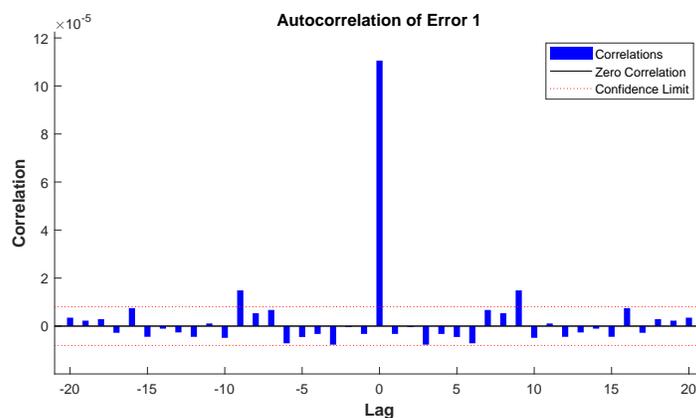


图 22: NAR 模型的残差的 ACF 函数图像

6 总结

本篇报告实现了对真实数据的 ARIMA 建模 (所得到的模型为(1), 其中 r_t 表示对数收益率, ε_t 表示残差), 以及对残差的 GARCH 建模 (所得到的模型为(2), 其中 r_t 表示对数收益率, X_t 表示残差, 主要是对 X_t 进行 GARCH 建模). 总而言之, 建立线性时间序列模型, 需要进行的步骤包括:

- 模型初筛, 选择合适的模型, 如查看 ACF 图、PACF 图或 EACF, 见图 6、图 7 和图 8; 或最小化 AIC, 见表 4 和表 5;
- 参数估计, 对选择的模型, 选取合适的方法, 如 ML、CSS 或 CSS-ML, 估计模型的参数, 并给出均方误差, 见表 7;
- 残差诊断, 对模型的 ε_t 进行正态性检验, 如画出 Q-Q 图, 见图 12, 或进行 JB 检验; 和进行相关性检验, 如画出 ACF 函数图, 见图 13, 或进行 LB 检验.

在建模完成后, 可以基于所建立的模型, 进行未来数据的预测 (见图 10 和图 9). 对于本篇报告的 S&P 500 数据, 残差的平方之间的相关性极强, 并且经过检验存在显著的 ARCH 效应 (见表 8), 从而我们需要对其进行进一步的 GARCH 建模, 需要进行的步骤包括:

- 模型选择, 最小化 AIC, 同时控制阶数不要太高, 选择合适的模型, 见表 9;
- 参数估计, 对选择的模型, 使用 ML 方法估计模型的参数, 并给出均方误差, 见表 10;
- 残差诊断, 对模型的 ε_t 进行正态性检验, 如画出 Q-Q 图, 见图 17; 和进行相关性检验, 如画出 ACF 函数图, 见图 18.

在建立了 GARCH 模型之后, 可以对波动率绘制出置信区间, 或者给出 VaR 值的估计.

作为神经网络的应用, 笔者在完成了 ARIMA 建模与 GARCH 建模之后, 尝试对数据建立了 NAR 模型 (见图 20), 并且对模型的预测值 (见图 21) 与残差 (见图 22) 进行分析, 发现结果较好. 未来需要对 S&P 500 数据进行分析时, 可以考虑结合现在热门的深度学习的方法.

本篇报告在期中报告 (对 S&P 500 数据进行线性时间序列建模) 的基础上, 修改了部分表述错误 (如通过 Q-Q 图判断残差具有正态性), 同时增加了部分内容 (如对残差的平方 ε_t^2 的相关性的检验), 最重要的是对残差进行更细致的建模, 解决了 ε_t^2 之间不相关的问题.

参考文献

- [1] 维基百科. 标准普尔 500 指数. https://zh.wikipedia.org/wiki/S%26P_500.
- [2] Jianqing Fan & Qiwei Yao. *The Elements of Financial Econometrics*. Science Press.

附录

A 所用软件

在编写本篇报告的过程中, 主要使用的软件为:

- Microsoft Excel, 用于整理收集所得的数据;
- 基于 RStudio 的 R, 用于数据处理、建立模型与绘制图像;
- Mathworks MATLAB, 用于建立动态神经网络模型;
- 基于 Visual Studio Code 的 \LaTeX , 用于排版报告.

B 代码 1: 数据的 ARMA 建模

以下代码使用 R 编写.

```
# 调用程序包

gc()
library(tseries)
library(TSA)
library(forecast)

# 原始数据

data = read.csv("Data_1.csv", header = TRUE)
P = ts(data[, 2])
plot(P) # 图像
acf(P) # ACF 图像
adf.test(P) # ADF 检验

# 对数收盘价

p = log(P)
plot(p) # 图像
acf(p) # ACF 图像
adf.test(p) # ADF 检验

# 对数收益率

r = diff(p) # 差分
```

```

plot(r) # 图像
lines(r)
acf(r) # ACF图像
pacf(r) # PACF图像
eacf(r) # EACF表

# 建立模型

r.fit = auto.arima(r)
p.fit = auto.arima(p)

# 残差诊断

plot(r.fit $ residuals, xlab = "t", ylab = "residuals") # 残差图像
tsdiag(r.fit, gof.lag = 30) # 残差的检验
qqnorm(r.fit $ residuals) # 残差的QQ图
qqline(r.fit $ residuals)
acf(r.fit $ residuals) # 残差的ACF图
acf(r.fit $ residuals ^ 2) # 残差平方的ACF图
jarque.bera.test(r.fit $ residuals) # 残差的JB检验
Box.test(r.fit $ residuals, type = "Ljung-Box") # 残差的LB检验

# 预测

plot(forecast(r.fit, 100))
plot(forecast(p.fit, 100))

```

C 代码 2: 残差的 GARCH 建模

```

# 调用程序包

gc()
library(tseries)
library(TSA)
library(forecast)
library(MTS)

# 原始数据

```

```

data = read.csv("Data_1.csv", header = TRUE)
P = ts(data[, 2]) # 收盘价
p = log(P) # 对数收盘价
r = diff(p) # 对数收益率
plot(r) # 图像

# ARMA 模型

fit.arima = auto.arima(r)
summary(fit.arima)

# ARCH 效应与 GARCH 模型

x = fit.arima $ residuals
for (i in 1 : 10)
{
  archTest(x, lag = i)
}
fit.garch = garch(x = x, order = c(1, 1))
summary(fit.garch)

# 残差分析

plot(fit.garch $ residuals, xlab = "t", ylab = "residuals") # 残差图像
hist(fit.garch $ residuals)
qqnorm(fit.garch $ residuals) # 残差的 QQ 图
qqline(fit.garch $ residuals)
acf(fit.garch $ residuals[2 : 755]) # 残差的 ACF 图
acf(fit.garch $ residuals[2 : 755] ^ 2) # 残差平方的 ACF 图

# VaR

q0.05 = ts(0, 1, 755)
q0.025 = ts(0, 1, 755)
for (i in 1 : 754)
{
  q0.05[i] = x[i] / fit.garch $ residuals[i] * quantile(fit.garch $
    residuals[2 : 755], 0.05)
  q0.025[i] = x[i] / fit.garch $ residuals[i] * quantile(fit.garch $
    residuals[2 : 755], 0.025)
}

```

```

}
plot(r)
lines(q0.05, col = "blue")
lines(q0.025, col = "red")

```

D 代码 3: 基于动态神经网络的 NAR 建模

```

function [Y,Xf,Af] = myNeuralNetworkFunction(X,Xi,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Auto-generated by MATLAB, 06-Jun-2022 18:09:33.
%
% [Y,Xf,Af] = myNeuralNetworkFunction(X,Xi,~) takes these arguments:
%
% X = 1xTS cell, 1 inputs over TS timesteps
% Each X{1,ts} = 1xQ matrix, input #1 at timestep ts.
%
% Xi = 1x2 cell 1, initial 2 input delay states.
% Each Xi{1,ts} = 1xQ matrix, initial states for input #1.
%
% Ai = 2x0 cell 2, initial 2 layer delay states.
% Each Ai{1,ts} = 10xQ matrix, initial states for layer #1.
% Each Ai{2,ts} = 1xQ matrix, initial states for layer #2.
%
% and returns:
% Y = 1xTS cell of 1 outputs over TS timesteps.
% Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
% Xf = 1x2 cell 1, final 2 input delay states.
% Each Xf{1,ts} = 1xQ matrix, final states for input #1.
%
% Af = 2x0 cell 2, final 0 layer delay states.
% Each Af{1ts} = 10xQ matrix, final states for layer #1.
% Each Af{2ts} = 1xQ matrix, final states for layer #2.
%
% where Q is number of samples (or series) and TS is the number of
% timesteps.

%#ok<*RPMT0>

```

```

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = -0.127652141156473;
x1_step1.gain = 9.20237079499445;
x1_step1.ymin = -1;

% Layer 1
b1 = [0.99950956415416325473; 4.4915887671472667364;
      1.3426849282860693346; -1.0923693014810560431;
      1.5926905947756087034; -1.9754907441639013577;
      0.046399666253847726693; 2.1680186995707821929;
      3.7159978700428633047; -0.00017784893392018341152];
IW1_1 = [-0.53229121072176388552
          8.159519471985827721;7.4761916245547981319
          -4.9540687050941834002; -1.6453664105859537869
          -1.3096414290764353261;0.06789592337161703961
          -10.658778001504030541;0.81583540570306578577
          -2.1209151702152051477;2.5759331324592800172
          2.9237415116246681102;0.53093261011454961906
          2.0291434309624265531;5.584642717712022808
          4.4425436125755899397;3.691095085246918206
          -4.555457241656318601; -0.039415391398881186036
          0.19046697338490695128];

% Layer 2
b2 = 0.13059824119900684258;
LW2_1 = [-2.3681564780612549548 3.5055460567720713172
          3.1174925238222650492 -1.948551740378621 2.8482937772703809109
          1.3988816144876319303 1.5224060805061605528 -1.9184856252572772828
          -5.3998404327372968581 0.16826752615048895523];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 9.20237079499445;
y1_step1.xoffset = -0.127652141156473;

% ===== SIMULATION =====

```

```

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX
    X = {X};
end
if (nargin < 2), error('Initial input states Xi argument needed.');
```

```

end

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
elseif ~isempty(Xi)
    Q = size(Xi{1},2);
else
    Q = 0;
end

% Input 1 Delay States
Xd1 = cell(1,3);
for ts=1:2
    Xd1{ts} = mapminmax_apply(Xi{1,ts},x1_step1);
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Rotating delay state position
    xdts = mod(ts+1,3)+1;

    % Input 1
    Xd1{xdts} = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    tapdelay1 = cat(1,Xd1{mod(xdts-[1 2]-1,3)+1});
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*tapdelay1);

```

```

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
end

% Final Delay States
finalxts = TS+(1: 2);
xits = finalxts(finalxts<=2);
xts = finalxts(finalxts>2)-2;
Xf = [Xi(:,xits) X(:,xts)];
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end

```