

SSE-DP-2026-5

Rによる時系列モデリング入門

－ 中国語版 －

北川源四郎(原著)

趙宇・国友直人(監修)

東京理科大学・統計数理研究所

2026年5月

SSE-DP(ディスカッションペーパー・シリーズ)は以下のサイトから無料で入手可能です。

<https://stat-expert.ism.ac.jp/training/discussionpaper/>

このディスカッション・ペーパーは、関係者の討論に資するための未定稿の段階にある草稿である。著者の承諾なしに引用・複写することは差し控えられたい。

SSE-DP-2026-5

Introduction to Time Series Modeling with
Applications in R : Chinese Edition

Genshiro Kitagawa

edited by Yu Zhao and Naoto Kunitomo

Tokyo Science University · Institute of Statistical Mathematics

May 2026

(Summary)

This is the Chinese Edition of “Introduction to Time Series Modeling with Applications in R (2nd edition)” by Genshiro Kitagawa (Japanese Edition: Iwanami-Shoten, 2020, Tokyo, Japan; English Edition: CRC Press 2021). The translation was done as a project of the statistical expert training program promoted by the ISM (Institute of Statistical Mathematics),

中国語版への前書き

統計数理研究所が推進している統計エキスパート人材育成プログラムでは統計学を専門としてない各分野の若手研究者と統計家であるメンターにより統計エキスパート演習を実施している。2025 年度に実施したグループ演習では時系列解析の分野で定評のある北川源四郎「Rによる時系列モデリング入門」(岩波書店, 2020; “Introduction to Time Series Modeling with Applications in R (2nd edition)” Genshiro Kitagawa, CRC Press, 2021). の中国語版を作成した。翻訳は趙宇(東京理科大学), 何協(東京理科大学), 張文婷(新潟県立大学), 薛玉傑(統計数理研究所)が分担したが、各章の翻訳の担当は以下の通りである。

まえがき 趙宇

R と時系列解析パッケージ TSSS 趙

第 1 章 張文婷

第 2 章 何協

第 3 章 張文婷

第 4 章 何協

第 5 章 趙宇

第 6 章 張文婷

第 7 章 薛玉傑

第 8 章 張文婷

第 9 章 薛玉傑

第 10 章 何協

第 11 章 趙宇

第 12 章 趙宇

第 13 章 何協

第 14 章 薛玉傑

第 15 章 薛玉傑

第 16 章 趙宇

付録 A, B, C, D/参考文献/章末問題解答/索引 趙宇

なお原著の原稿ファイルを提供された北川源四郎氏、翻訳原案について有益なコメントを寄せてくれた姜興起氏、原著(第 2 版)中国語版の公開を了解された岩波書店、大連海事大学出版社(第 1 版の中国語版を出版)に感謝する。

趙宇・国友直人

2026年 5 月 19 日

时间序列分析入门

北川源四郎

2020年10月

译者前言

本书是以日文撰写的北川源四郎 (2020) 《(基于 R 的) 时间序列建模入门》，岩波书店 (英文版 “Introduction to Time Series Modeling with Applications in R (2nd edition)” Genshiro Kitagawa, CRC Press, 2021) 的中文版。

感谢提供原著原稿的北川源四郎老师，感谢对翻译初稿提出宝贵意见的姜兴起老师，并感谢同意第 2 版 · 中文版公开的岩波书店 (日语第 2 版) 和大连海事大学出版社 (出版了中文版 · 第 1 版)。

译者

新版序言

自《时间序列分析入门》出版以来已经过去了15年。在此期间，数据环境、计算环境和数据利用技术取得了飞跃式的发展，数据驱动型社会正在成为现实。因此，将数理、数据科学和AI视为现代社会的基本素养，面向所有学生而不分专业领域的教育也即将启动。在这种数据科学发展的背景下，计算环境发生了显著变化。如今，网络上存在大量免费的函数，尝试进行数据分析的人通过寻找R或Python等适当的函数，甚至可以进行大规模数据的分析。考虑到这样的时代背景，本书在修订时采用了引入R分析实例的形式。

本书自初版以来，一直以使读者能够针对个别问题自行设计模型并加以实现为目标。为了时间序列分析的未来发展，培养具有这种能力的人才是不可或缺的，此外，使人们能够利用现有软件快速地应用于实际数据也很重要。幸运的是，在作者此前所属的统计数理研究所，以本书前身《FORTRAN77 时间序列分析编程》中刊载的FORTRAN程序为基础，开发了统计计算语言R的程序包TSSS (Time Series Analysis with State Space Model)。通过使用TSSS，本书中介绍的大部分建模方法和计算方法都能够简单地实现。因此，本书中几乎所有例题都使用TSSS程序包的函数进行计算，同时也对其使用方法进行了说明。期望通过活用此次全面采用的R函数，在学习方法的同时进行实际的数据分析，从而进一步加深对时间序列分析的理解。

在修订中，除此之外还进行了若干修正，包括添加了自动搜索ARMA模型阶数的方法，以及在随机数生成方法中加入Mersenne Twister等。此外，将第15章的标题从蒙特卡罗滤波改为粒子滤波。作者一直提倡蒙特卡罗滤波，但自该方法开发至今已过去四分之一世纪，粒子滤波这一名称已在全球范围内确立。

本第2版的修订正是由于TSSS程序包的开发才得以实现，在此衷心感谢负责开发的嵯峨雅美女士和中野纯司教授。此外，本修订版的出版是由岩波书店编辑局自然科学书编辑部的濱门麻美子女士和高野照子女士提议的，感谢二位 in 出版过

程中给予的帮助。实际上,本次修订工作是英译版先行进行的。旧版的英译版于2008年由CRC Press, Chapman & Hall出版,借第2版出版之机,将R的例题纳入其中。感谢提议出版第2版的编辑David Grubbs先生,以及在英语版出版过程中给予帮助的岩波书店编辑局版权管理部的森川裕美女士。

北川源四郎

东京, 2020年10月

序言

21 世纪的现代社会在地球环境、经济、安全性等各方面面临着严峻的问题。这些问题大多涉及复杂、不确定且动态的现象，需要用于解析这些复杂系统、进行预测和信息提取的方法。另一方面，在统计科学领域，以信息量准则 AIC 的引入为契机，统计建模方法取得了飞速发展，加之大量数据的积累和计算能力的飞跃式提升，众多新模型被开发并投入实用。

本书旨在介绍在这些快速发展的统计建模方法中具有代表性的时间序列建模方法。时间序列分析试图从与过去变动的关联中捕捉某一现象的变动。由于统计学所研究的许多现象都是时间序列，而且这种捕捉现象的方式是统计建模的本质——条件分布构建的典型范例，因此学习时间序列分析方法是十分重要的。本书介绍了时间序列分析中使用的主要模型，其中一个特色是特别说明了基于状态空间模型的统一方法。此外，本书展示了时间序列建模所必需的最小二乘法、最大似然法以及通过递推滤波实现高效估计的方法，同时介绍了基于信息量准则 AIC 的一贯模型评价与选择方法。

需要注明的是，本书是以 1993 年作为“岩波计算机科学”系列之一出版的《FORTRAN77 时间序列分析编程》为基础，经过修订的入门书。自前著出版至今已过去十余年，统计科学取得了切实的进步，与此同时，作为信息处理基础设施的计算机也经历了巨大的发展和普及。个人计算机已渗透到各种环境中，而许多领域中大量数据已变得可以利用，因此比以往更加需要活用这些资源、对数据进行适当建模并提取适当信息的方法。

另一方面，在实际分析中，相比以往在大型计算机上运行 FORTRAN 程序的情况，在各种平台上使用多种多样的编程语言进行分析的情况日益增多。因此，本书删除了 FORTRAN 源代码及其说明部分使之更加紧凑，同时通过新设关于蒙特卡罗滤波的章节，力求尽可能地适应追求更自由建模的新时代需求。不过，新版中删除的计算程序及其说明已安排在网站 (<http://www.ism.ac.jp/~kitagawa/>) 上供下载。此外，产业技术综合研究所的松本则夫博士开发了可以在 Web 上执行这些程序的分析系统，并在以下网站公开 (<http://www.aist.go.jp/RIODB/gxwell/GSJ/analysis/index.html>)。通过利用这些资源，读者可以用自己的数据尝试本书中介

绍的几乎所有模型和分析方法。^{*1}此外,本书在各章末新增了若干习题。希望这些习题能有助于读者理解正文中介绍的代表性模型。

最后,在本书问世之前的时间序列分析研究中,作者得到了许多人的帮助。无需赘言,以统计数理研究所前所长赤池弘次为首的众多前辈和同事,在日常的讨论和交谈中给予了作者极大的影响。此外,本书中介绍的许多非平稳模型,是与夏威夷大学的 Gersch 教授、东京海洋大学的大津皓平先生、北海道大学的高波铁夫先生、产业技术综合研究所的松本则夫先生等众多人士合作研究的成果。这些合作者还慷慨提供了本书例题中使用的宝贵数据。在本书原稿的撰写和计算过程中,福岛靖子女士和小野节子女士给予了帮助。在本书的出版过程中,岩波书店编辑部的吉田宇一先生和松永研先生给予了莫大的帮助。在此向所有这些人士表示衷心的感谢。

2005 年 1 月

北川源四郎

^{*1} 由于计算环境发生了重大变化,上述两个网站已无法使用,对计算机软件感兴趣的读者请使用本书所采用的 R 函数。

R 与时间序列分析程序包 TSSS

R 是一种开源的统计计算和可视化编程语言及开发执行环境，可在 Linux、Mac 或 Windows 上运行。本书例题中使用的数据和分析程序大多包含在 R 的程序包 TSSS (Time Series Analysis with State Space model) 中。

R 的获取

R 可以在下述 CRAN (Comprehensive R Archive Network) 的网页上选择 Linux、(Mac)OS X 或 Windows 中的任一操作系统进行获取。

<https://cran.r-project.org/>

<http://cran.ism.ac.jp> (日本镜像站点)

R 的时间序列分析程序包 TSSS

TSSS 是由统计数理研究所的嵯峨雅美女士和中野纯司教授，以本书前身《FORTRAN77 时间序列分析编程》(岩波书店) 中刊载的 FORTRAN 程序为基础开发的。

TSSS 程序包也可以从 CRAN 下载，统计数理研究所的以下网站包含 R 在内的详细安装说明。

<http://jasp.ism.ac.jp/ism/TSSS/>

TSSS 中提供了以下函数。

<code>arfit</code>	一元 AR 模型的拟合
<code>armachar</code>	一元 ARMA 模型的特征提取 (脉冲响应函数、自协方差函数、PARCOR、谱、特征根的计算)
<code>armafit</code>	一元 ARMA 模型的拟合
<code>armafit2</code>	多个一元 ARMA 模型的自动估计
<code>boxcox</code>	Box-Cox 变换
<code>crscor</code>	互协方差函数和互相关函数的计算
<code>fftper</code>	用 FFT 计算周期图
<code>klinfo</code>	K-L 信息量的计算
<code>lsar</code>	用局部平稳 AR 模型将时间序列划分为局部平稳区间
<code>lsar.chgpt</code>	用局部平稳 AR 模型估计时间序列的变化点
<code>lsqr</code>	用 Householder 变换的最小二乘法进行三角函数回归模型的估计
<code>marfit</code>	用 Yule-Walker 法拟合多元 AR 模型
<code>marlsq</code>	用最小二乘法拟合 AR 模型
<code>marspc</code>	互谱、相干性和功率贡献率的计算
<code>ngsim</code>	非高斯型状态空间模型的仿真
<code>ngsmth</code>	非高斯型平滑
<code>pdfunc</code>	概率密度函数的计算
<code>period</code>	周期图的计算
<code>pfilter</code>	粒子滤波和粒子平滑
<code>pfilterNL</code>	非线性模型的粒子滤波和粒子平滑
<code>polreg</code>	多项式回归模型的估计
<code>season</code>	季节调整
<code>simssm</code>	高斯型状态空间模型的仿真
<code>trend</code>	趋势估计
<code>tsmooth</code>	利用状态空间模型的时间序列预测和插值
<code>tvar</code>	时变系数 AR 模型的估计
<code>tvspc</code>	用时变系数 AR 模型计算时变谱
<code>tvvar</code>	时变方差模型的估计
<code>unicor</code>	自协方差函数和自相关函数的计算

TSSS 程序包中还包含本书使用的以下数据。

BLSALLFOOD	美国食品产业从业人数
Haibara	地下水位数据
HAKUSAN	船舶航行中的多元数据
MYE1F	地震波数据
Nikkei225	日经 225 股价数据
NLmodel	非线性状态空间模型的测试数据
PfilterSample	均值发生偏移的时间序列人工数据
Rainfall	东京每月各日两年间的降雨次数
Sunspot	太阳黑子数数据
Temperature	东京日最高气温数据
WHARD	硬件批发额数据

目次

第 1 章	时间序列数据的分析及其准备工作	15
1.1	时间序列数据	15
1.2	时间序列的分类	20
1.3	时间序列分析的目的	21
1.4	时间序列数据的预处理	22
	章末习题	28
第 2 章	协方差函数	29
2.1	时间序列的分布与平稳性	29
2.2	平稳序列的自协方差函数	32
2.3	多元时间序列与散点图	35
2.4	互协方差函数及互相关函数	37
	章末习题	41
第 3 章	谱与周期图	42
3.1	谱	42
3.2	周期图	46
3.3	周期图的平均与平滑化	49
3.4	周期图的计算方法	53
3.5	基于 FFT 的周期图计算	54
	章末习题	57
第 4 章	建模	58
4.1	概率分布与统计模型	58
4.2	K-L 信息量与熵最大化原理	61

4.3	K-L 信息量的估计与对数似然	64
4.4	基于最大似然法的参数估计	66
4.5	AIC (赤池信息量准则)	68
4.6	数据变换	72
	章末习题	75
第 5 章	最小二乘法	77
5.1	回归模型与最小二乘法	77
5.2	基于 Householder 法的最小二乘法求解方法	78
5.3	通过 AIC 进行阶数选择	80
5.4	数据的追加与分割处理	83
5.5	通过 AIC 进行变量选择	84
	章末习题	85
第 6 章	基于 ARMA 模型的时间预测分析	86
6.1	ARMA 模型	86
6.2	冲激响应函数	87
6.3	自协方差函数	88
6.4	AR 系数与 PARCOR 的关系	89
6.5	功率谱	90
6.6	特性方程式	93
6.7	多元 AR 模型	96
	章末习题	101
第 7 章	AR 模型的估计	103
7.1	AR 模型的拟合	103
7.2	Yule-Walker 方法与 Levinson 算法	104
7.3	用最小二乘法估计 AR 模型	106
7.4	用 PARCOR 法估计 AR 模型	107
7.5	AR 系数估计量的误差分布	109
7.6	数值例	111
7.7	用 Yule-Walker 方法估计多变量 AR 模型	112

7.8	用最小二乘法估计多变量 AR 模型	115
	章末习题	119
第 8 章	局部平稳 AR 模型	120
8.1	局部平稳 AR 模型	120
8.2	自动划分为任意个区间	121
8.3	变化时点的精密估计	126
8.4	变化时点的事后概率	128
	章末习题	130
第 9 章	利用状态空间模型的时间序列分析	131
9.1	状态空间模型	131
9.2	利用卡尔曼滤波进行状态估计	133
9.3	平滑算法	134
9.4	状态的长期预测	135
9.5	时间序列的预测	136
9.6	时间序列模型的似然计算与参数估计	139
9.7	缺测值的插补	142
	章末习题	145
第 10 章	ARMA 模型的估计	146
10.1	ARMA 模型的状态空间表示	146
10.2	AR 模型的初始状态分布	147
10.3	ARMA 模型的初始状态	149
10.4	ARMA 模型的最大似然估计	150
10.5	关于参数的初始值	152
	章末习题	154
第 11 章	趋势的估计	155
11.1	多项式回归模型	155
11.2	趋势分量模型 – 结构随机变化的模型	159
11.3	趋势模型	161

章末习题	165
第 12 章 季节调整模型	167
12.1 季节分量模型	167
12.2 标准季节调整模型	169
12.3 包含平稳 AR 分量的分解	172
12.4 包含星期效应项的分解	175
章末习题	179
第 13 章 时变系数 AR 模型	181
13.1 时变方差模型	181
13.2 时变系数 AR 模型	183
13.3 时变谱的估计	188
13.4 时变系数 AR 模型的系统噪声假设	190
13.5 关于系数的剧烈变化	190
章末习题	192
第 14 章 非高斯型模型	193
14.1 非高斯型模型的必要性	193
14.2 非高斯型状态空间模型与状态估计	194
14.3 状态估计公式的数值实现	195
14.4 非高斯型趋势模型	198
14.5 非对称分布——时变方差模型	201
14.6 非高斯型状态空间模型的应用	206
章末习题	209
第 15 章 粒子滤波与平滑化	210
15.1 非线性·非高斯型状态空间模型与分布的近似	210
15.2 粒子滤波	212
15.3 粒子平滑化	216
章末习题	222
第 16 章 仿真	223

16.1	均匀随机数的生成	223
16.2	白噪声的生成	225
16.3	ARMA 模型的仿真	229
16.4	利用状态空间模型的仿真	230
16.5	利用非高斯型状态空间模型的仿真	233
	章末习题	237
附录 A	非线性最优化的算法	238
附录 B	Levinson 算法的推导	240
附录 C	卡尔曼滤波和平滑算法的推导	242
C.1	卡尔曼滤波	242
C.2	平滑	243
附录 D	粒子滤波的算法	245
	参考文献	248
	章末习题解答	251
	索引	262

第 1 章

时间序列数据的分析及其准备工作

时间序列数据具有多种不同特征，因此作为时间序列分析的第一步，将时间序列作图并充分把握其特征是非常重要的。通过图形展示，往往能够得到关于下一步应采用的分析方法或者应使用的模型的启示。本章前半部分将把第 2 章以后数值例中使用的的时间序列数据加以图示，阐明其特征，并一般性地说明时间序列的分类以及时间序列建模的目的。后半部分将介绍变量变换、差分等时间序列预处理方法。本书中几乎所有数值例的计算都使用统计计算环境 R。对于这些例子，也将以 TSSS 包为中心说明 R 函数的使用方法。

1.1 时间序列数据

随时间推移而不规则变化的现象记录称为**时间序列** (time series)。气压、气温、降雨量等气象信息，地震波记录，股价与汇率等经济现象记录，脑电波与心电图等医学数据，汽车、船舶、飞机的操纵记录等，即便只从我们日常生活周边举例，也能想到大量的时间序列。

作为时间序列分析的第一步，首先把数据可视化是很重要的。这不仅可以把握时间序列的大致特征，也能据此制定必要的预处理策略，以及后续应进行何种分析的方针。

本书通篇将以下数据集作为例题使用。这些数据包含在 R 包 TSSS 中，可按下面所示用函数 `data` 读入。

```

> data( HAKUSAN ) # 船舶的多变量数据
> data( Sunspot ) # 太阳黑子数数据
> data( Temperature )# 东京日最高气温数据
> data( BLSALLFOOD ) # 食品产业从业人数数据
> data( WHARD ) # 某硬件的批发额数据
> data( MYE1F ) # 地震波数据
> data( Nikkei225 ) # 日经 225 平均股价数据
> data( Rainfall ) # 两年内下雨天数数据
> data( Haibara ) # 地下水位数据

```

此外, 如果想分析这些之外的数据集, 可以用函数 `read.csv` 读取 csv 形式的数据。这里, 函数 `as.ts` 用于将读入的数据转换为时间序列对象。并且, 可以用参数 `start` 指定时间序列的起始时点, 用 `frequency` 登记单位时间内的观测点数。默认值为 `start=1, frequency=1`。对于月度数据, 令 `frequency=12`, 并如 `start=c(1998,5)` 那样即可登记数据开始的年份与月份。

```

> sunspot <- as.ts( read.csv( "sunspot.csv" ) )
>
> blsfood <- as.ts( read.csv( "blsfood_new.csv" ), start = c(1967,1), frequency =
12 )

```

一旦数据读入 R, 即可用函数 `plot` 绘制图形。在该函数中, 可用参数 `ylim` 以及 `main` 指定纵轴上下限以及图的标题。

```

> par( mar=c(2,2,3,1)+0.1 )
> plot( HAKUSAN[,1],main="(a) 船舶的偏航角速度")
> plot( Sunspot,main="(b) 太阳黑子数")
> plot( Temperature,main="(c) 东京最高气温")
> plot( BLSALLFOOD,main="(d) 食品产业从业人数")
> plot( WHARD,main="(e) 批发额")
> plot( MYE1F,main="(f) 地震波东西方向")
> plot( Nikkei225,main="(g) 日经 225 平均股价")
> plot( Haibara,main="(h) 地下水位 (上段) 与气压 (下段) ")
> plot( HAKUSAN[,c(2,3,4)],main="(i) 船舶横摇 (上段)、纵摇 (中段)、舵角 (下段) " )

```

图 1.1 给出了之后数值例中使用的典型时间序列。虽然会稍长一些, 下面先对各数据的特征作说明。

(a) 这是 4 变量 HAKUSAN 数据的第 1 列, 为航行于外洋中的船舶在保针控制下保持航向时, 以每秒一次记录的**偏航角速度** (yaw rate) 时间序列。由于进行了保持航向的保针控制, 偏航角速度在 0 附近波动。(东京海洋大学 大津皓

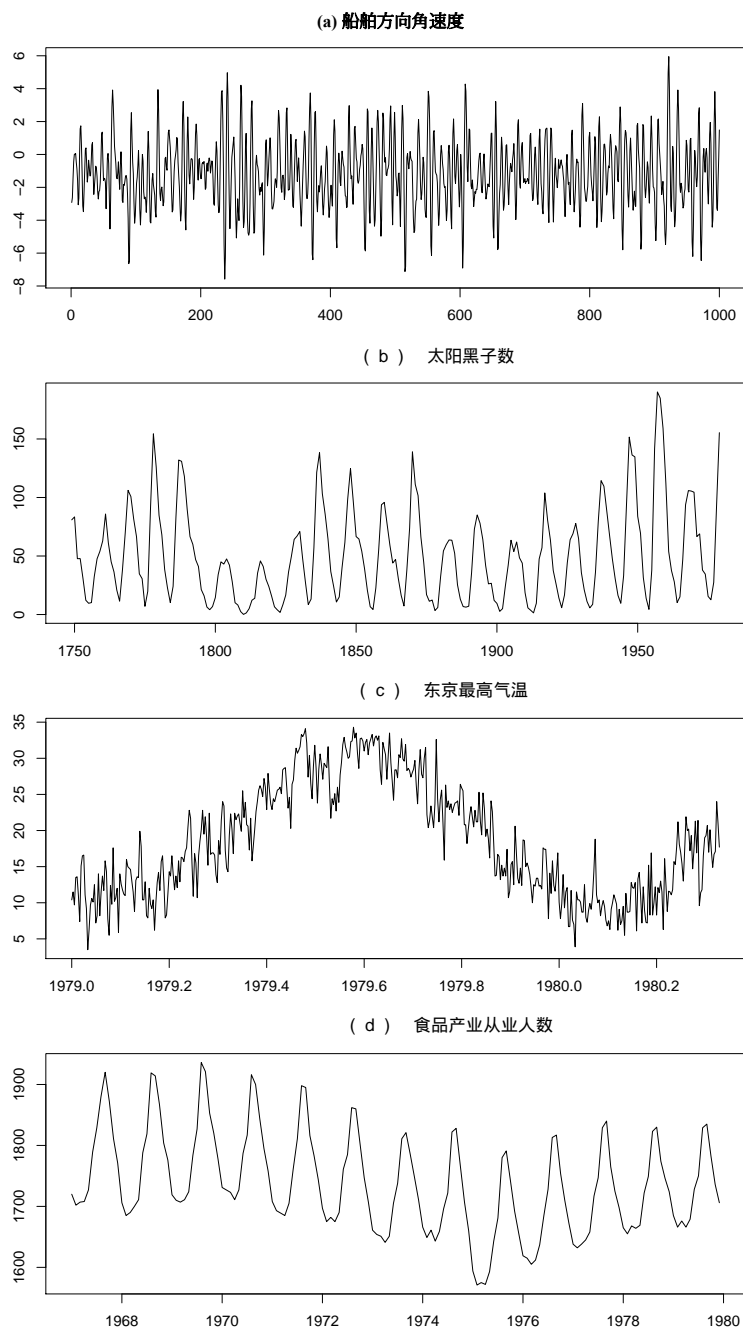


图 1.1 各种时间序列的例子

平氏提供)

(b) 记录每年的**太阳黑子数** (Wolfer sunspot number) 的时间序列。呈现近似 11 年周期的类似模式，反复增减。

(c) 记录东京每日**最高气温**、持续 16 个月的数据。可以看到占主导的年周期 (趋势)，并且在年周期周围存在不规则波动。(东京管区气象台公布)

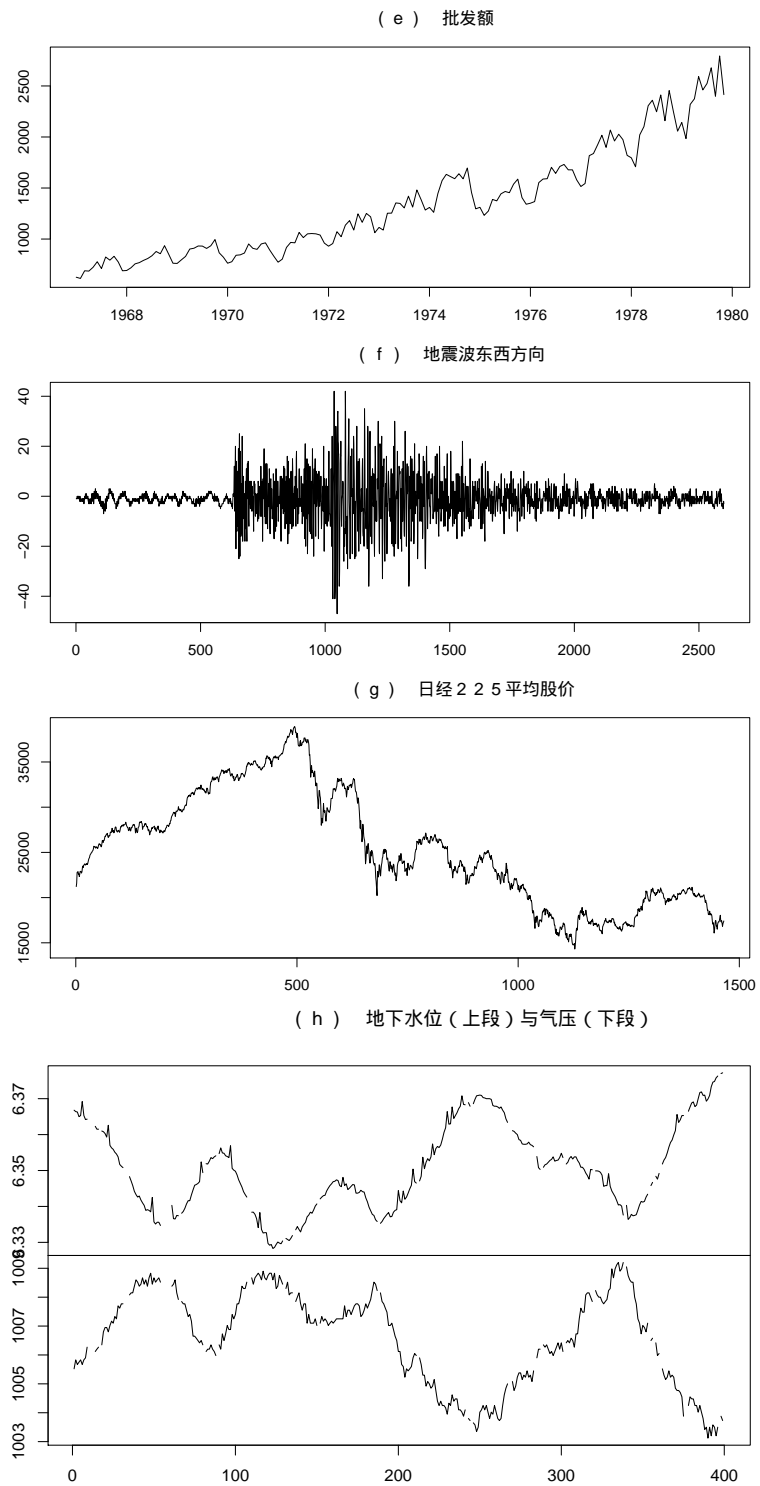


图 1.1 各种时间序列的例子 (续)

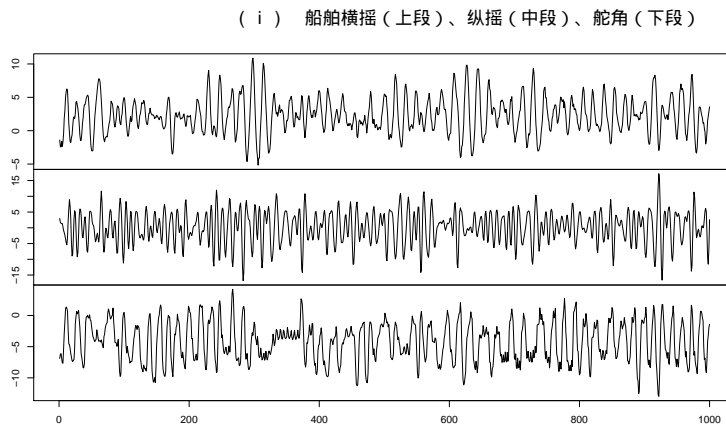


图 1.1 各种时间序列的例子 (续)

- (d) 美国食品产业从业人数的月度时间序列, 称为 BLSALLFOOD 数据。包含经济时间序列特有的缓慢变化的趋势成分, 以及每年重复相似波动的季节成分。(美国劳工统计局 Bureau of Labor Statistics (BLS) 公布)
- (e) 记录某硬件每月**批发额**的数据, 称为 WHARD(Wholesale hardware) 数据。可以看到经济数据特有的特征: 每年以几乎固定比例增加, 并随之趋势周围的波动也逐渐变大。(美国 Bureau of Labor Statistics (BLS) 公布)
- (f) 以约 0.02 秒间隔记录的**地震波**东西向分量时间序列。随着 P 波 (纵波) 与 S 波 (横波) 的到达, 波动幅度显著变化。若进一步细看, 会发现不仅幅度, 波动方式本身也发生变化。(北海道大学大学院理学系研究科 高波鐵夫氏提供)
- (g) 日经 225 平均**股价数据**, 为 1988 年 1 月 4 日至 1993 年 12 月 30 日的日收盘价。到 1989 年末为止几乎单调上升, 而泡沫经济崩溃后在反复大幅波动中逐渐下降。股价数据的分析中, 常常在取对数变换后再取差分来进行。
- (h) 在东海地区某观测点对**地下水**位与**气压**以 10 分钟间隔观测得到的二变量时间序列。折线中空白部分表示由于某种原因未进行观测之处。另外, 局部可见向上突出的值, 推测为观测设备问题导致的异常值。为了有效利用通过长期连续观测得到的宝贵数据, 有必要开发即便面对这种数据也能进行分析的方法。(独立行政法人产业技术综合研究所 高桥诚氏、松本则夫氏提供)
- (i) 航行于太平洋中的船舶以每秒一次记录的多变量时间序列, 包括横摇 (rolling)、纵摇 (pitching) 以及舵角 (rudder angle)。可以看出横摇与舵角都具有十几秒左右的周期性波动; 而纵摇则以 10 秒以下的较短周期波动。(东京

海洋大学 天津皓平氏提供)

1.2 时间序列的分类

如图 1.1 所示, 时间序列有多种类型。这些时间序列可以从以下诸多角度进行分类。

连续时间序列与离散时间序列

用模拟记录仪等连续记录的数据称为**连续时间** (continuous time) 时间序列。与之相对, 如每隔一小时测量一次的气压那样, 在某一时间间隔下观测得到的称为**离散时间** (discrete time) 时间序列。离散时间序列又可分为**等间隔** (equally-spaced) 观测与**不等间隔** (unequally-spaced) 观测。图 1.1 中的时间序列虽然用实线连续连接, 但实际上全部都是离散时间序列。在使用计算机进行分析时, 多数情况下处理的是离散观测值, 因此今后本书将把等间隔的离散时间序列简称为时间序列。

单变量时间序列与多变量时间序列

像图 1.1 中 (a)~(g) 那样, 在每个观测时点只得到一种信息的称为**单变量** (univariate) 时间序列。相反, 像 (h) 与 (i) 那样同时记录两种以上信息的称为**多变量** (multi-variate) 时间序列。不过, 与某一现象同时发生的现象可以有很多, 因此时间序列是一变量还是多变量, 以及在多变量情况下同时考虑哪些变量, 并非由分析对象唯一决定, 而是受测量限制、目的以及此前分析结果等因素影响而确定。从建模角度看, 选择用于分析的变量本身就是极其重要的统计问题。

平稳时间序列与非平稳时间序列

时间序列会随时间不规则变化, 但在时间序列分析中, 用概率模型来表示这种不规则波动。此时, 即便表面上看似不规则的现象, 也可能被视为来自一个随时间不变的固定概率模型的实现值。这类时间序列称为**平稳** (stationary) 时间序列。在图 1.1 中, (a) 可视为典型的平稳时间序列。相反, 像 (c)、(d)、(e)、(g) 那样均值随时间变化, 或像 (f) 那样均值周围的波动方式随时间变化的, 称为**非平稳** (nonstationary) 时间序列。

高斯型时间序列与非高斯型时间序列

分布服从正态分布 (高斯分布) 的时间序列称为**高斯型** (Gaussian) 时间序列, 否则为**非高斯型** (non-Gaussian) 时间序列。本书所处理的许多模型都假定服从高斯

分布的高斯型时间序列模型。即便像 (b) 那样波动模式上下不对称、周边分布看起来难以认为是正态的时间序列，通过对数据进行适当变换，有时也能近似看作高斯型时间序列。

线性时间序列与非线性时间序列

能够表示为线性模型输出的时间序列称为**线性** (linear) 时间序列；需要非线性模型的则称为**非线性** (nonlinear) 时间序列。

缺失值与异常值

与以上分类方法不同，但在实际时间序列分析中需要注意的还有缺失值与异常值。如 (h) 所示，如果由于某种原因未记录观测值，该部分称为**缺失值** (missing value)。此外，若由于被观测对象的异常行为、观测装置异常、填写或数据传输错误等原因出现明显异常的数据，这些称为**异常值** (outlier, 离群值)。(h) 中的水位数据中有几处向上突出的观测值可视为异常值。

1.3 时间序列分析的目的

本书将说明用于时间序列统计分析的方法。本书所考虑的问题可大致分为以下 4 类，统称为**时间序列分析** (time series analysis)。

- **描述与可视化** (description, visualization)

通过作图，或利用样本自协方差函数、样本自相关函数、周期图等基本描述统计量，对时间序列特征进行简洁表达。由于时间序列分析会输出大量数值，用图形表达更为有效。

- **建模** (modeling)

对给定时间序列，估计刻画其波动方式的时间序列模型，并分析其随机结构。时间序列具有多样特征，因此需要根据分析对象与目的选择合适的模型，并估计模型中的参数。

- **预测与仿真** (prediction, simulation)

利用时间序列之间存在相关性这一事实，根据截至当前获得的信息预测未来变化。尤其本书将讨论如何利用估计得到的模型进行预测与仿真。

- **信号提取** (signal extraction)

从时间序列中按目的提取所需信号与信息。进行符合对象特征与目的的恰

当建模非常重要。

1.4 时间序列数据的预处理

当时间序列数据具有非平稳性或非对称性时，在进入第 2 章以后的分析之前，有时会对数据进行预处理（一次处理）。本节介绍若干用于平稳化的方法。但在第 11 章以后，将讨论不依赖此类预处理、而直接分析非平稳时间序列的方法。

1.4.1 变量变换

如图 1.1 的 (e) 和 (g) 那样，金额或人数等取正值的数据中，很多都具有“随数值增大波动幅度也增大”的特征。此时，以原序列 y_n 的对数变换 $z_n = \log y_n$ 代替 y_n ，可能会使方差更均匀，或使误差分布更接近正态分布。

图 1.2 给出了 WHARD 数据及其对数序列。对数变换使得趋势周围的波动更均匀，同时趋势也更接近直线。

```
> data( WHARD )
> log_WHARD <- log10( WHARD )
> par( mfrow = c(2,1), mar = c(2,4,1,1)+0.1 )
> plot( WHARD, type = "l" )
> plot( log_WHARD, type = "l" )
```

作为对数变换的特殊情形所包含的更一般 Box-Cox 变换、以及其自动确定等细节，将在 4.8 节讲解。另外，对于取值在 $(0,1)$ 上的概率或比例等时间序列 y_n ，可以用 logit 变换

$$z_n = \log \left(\frac{y_n}{1-y_n} \right) \quad (1.1)$$

得到取值在 $(-\infty, \infty)$ 上的时间序列 z_n 。变换后的 z_n 分布偏斜较小，建模往往更容易。

1.4.2 差分（阶差）

当时间序列 y_n 像图 1.1 的 (c)、(e)、(g) 那样包含趋势时，有时会先求差分序列

$$z_n = \Delta y_n = y_n - y_{n-1} \quad (1.2)$$

并对 z_n 进行分析。这是因为当 y_n 为直线 $y_n = a + bn$ 时，

$$z_n = \Delta y_n = b \quad (1.3)$$

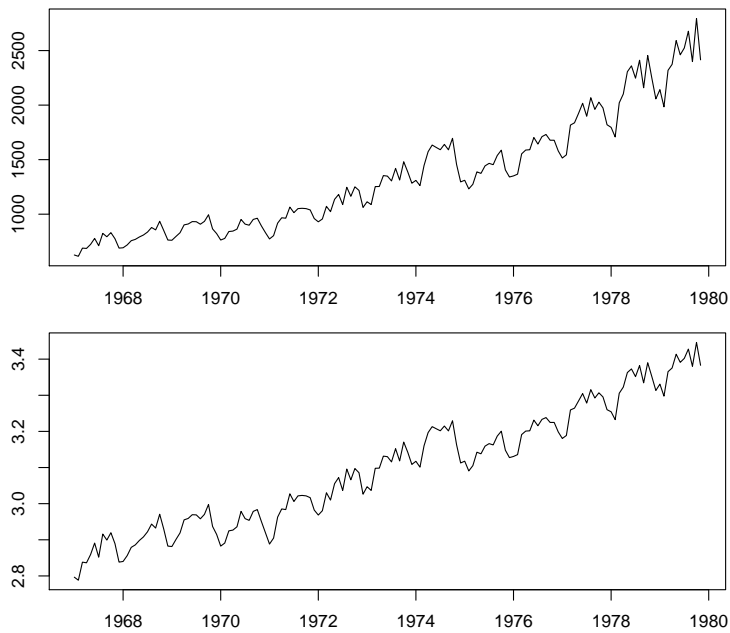


图 1.2 WHARD 数据 (上段) 与对数变换数据 (下段)

从而可以去除直线斜率。此外, 当 y_n 为二次式 $y_n = a + bn + cn^2$ 时, 再对 z_n 求差分可得

$$\begin{aligned}
 \Delta z_n &= z_n - z_{n-1} \\
 &= \Delta y_n - \Delta y_{n-1} \\
 &= (b + 2cn) - (b + 2c(n-1)) \\
 &= 2c
 \end{aligned} \tag{1.4}$$

从而能够去除二次成分与一次成分。由于 z_n 的差分为 $\Delta z_n = \Delta^2 y_n$, 也称为 y_n 的二阶差分。

```
> plot( diff( log( Nikkei225 ) ) )
```

R 的函数 `diff` 用于计算数据差分。图 1.3 给出了图 1.1(g) 日经 225 数据的对数值差分序列 $r_n = \log y_n - \log y_{n-1}$ 。这种对数差分在金融领域常用作收益率的替代量。可以看出, 以 $n = 500$ 附近为界, 波动方差发生了明显变化。这将在 13.1 节与 14.5 节中讨论。

当如图 1.1(e) 那样存在季节性波动时, 有时会使用与一个周期之前值的差分:

$$\Delta_p y_n = y_n - y_{n-p} \tag{1.5}$$

Δ_p 称为季节差分 (季节阶差)。季节差分能去除每年重复的波动, 从而使景气波动等更加清晰。

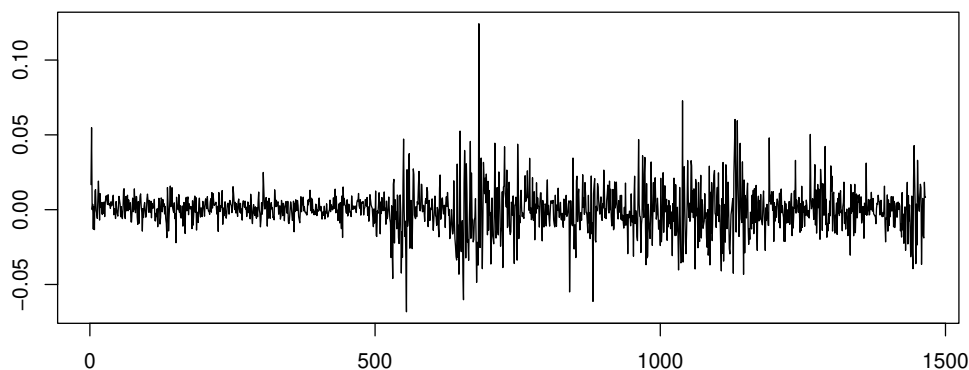


图 1.3 日经 225 数据的对数差分序列

1.4.3 环比、同比

特别是像图 1.1(e) 那样的经济时间序列中, 有时会分析原序列 y_n 的环比或同比:

$$z_n = \frac{y_n}{y_{n-1}}, \quad x_n = \frac{y_n}{y_{n-p}} \quad (1.6)$$

若时间序列 y_n 可表示为趋势与噪声的乘积

$$y_n = T_n w_n \quad (1.7)$$

并且 T_n 按 $T_n = (1 + \alpha)T_{n-1}$ 以增长率 α 指数增长, 则

$$z_n = \frac{y_n}{y_{n-1}} = \frac{T_n w_n}{T_{n-1} w_{n-1}} = (1 + \alpha) \frac{w_n}{w_{n-1}} \quad (1.8)$$

从而在忽略噪声时即可检测到增长率 α 。

另外, 若 y_n 可表示为周期为 p 的周期函数 s_n 与噪声的乘积:

$$y_n = s_n \cdot w_n, \quad s_n = s_{n-p} \quad (1.9)$$

则

$$x_n = \frac{y_n}{y_{n-p}} = \frac{s_n w_n}{s_{n-p} w_{n-p}} = \frac{w_n}{w_{n-p}} \quad (1.10)$$

从而可以去除周期成分。但由于会出现噪声 w_n 之比, 当趋势或季节成分不明显时反而可能变得更复杂, 需注意。

同比与环比可用 R 的函数 `lag` 轻松计算。

```
> plot( WHARD/lag(WHARD) ) # change from previous month
> plot( WHARD/lag(WHARD,k=-12) ) # year-over-year change
```

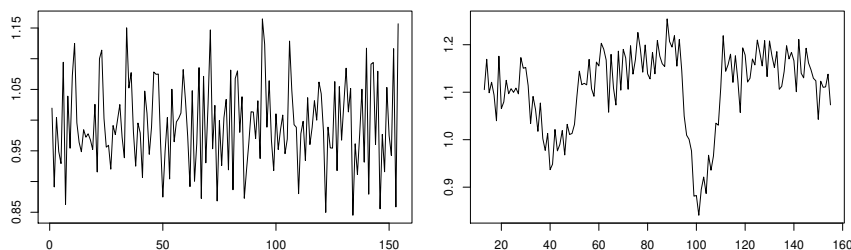


图 1.4 WHARD 数据的环比（左）与同比数据（右）

图 1.4 给出了图 1.1(e) 所示 WHARD 数据的环比与同比。可见环比去除了趋势成分；而同比去除了年周期成分，使 $n = 40$ 与 100 附近的下滑更为显著。

1.4.4 移动平均

将波动剧烈的时间序列平滑化的一种简便方法是移动平均。给定时间序列 y_n , $2k+1$ 项移动平均定义为在 y_n 前后各取 k 个，共 $2k+1$ 个值的平均：

$$T_n = \frac{1}{2k+1} \sum_{j=-k}^k y_{n+j} \quad (1.11)$$

若原时间序列可表示为直线与噪声之和：

$$y_n = t_n + w_n, \quad t_n = a + bn \quad (1.12)$$

其中 w_n 为均值 0、方差 σ^2 的独立噪声，则移动平均为

$$T_n = t_n + \frac{1}{2k+1} \sum_{j=-k}^k w_{n+j}. \quad (1.13)$$

对独立噪声和有

$$\begin{aligned} \mathbb{E} \left[\sum_{j=-k}^k w_{n+j} \right] &= \sum_{j=-k}^k \mathbb{E}[w_{n+j}] = 0, \\ \mathbb{E} \left[\left(\sum_{j=-k}^k w_{n+j} \right)^2 \right] &= \sum_{j=-k}^k \mathbb{E}[(w_{n+j})^2] = (2k+1)\sigma^2, \end{aligned} \quad (1.14)$$

因此移动平均的均值为 t_n ，方差为

$$\text{Var} \left(\frac{1}{2k+1} \sum_{j=-k}^k w_{n+j} \right) = \frac{\sigma^2}{2k+1}. \quad (1.15)$$

这表明移动平均的期望值 T_n 与 t_n 相同，而方差降为原时间序列 y_n 的 $1/(2k+1)$ 。

时间序列移动平均可用下面的 R 代码计算。同样的结果也可用 TTR 包的函数 SMA 得到。

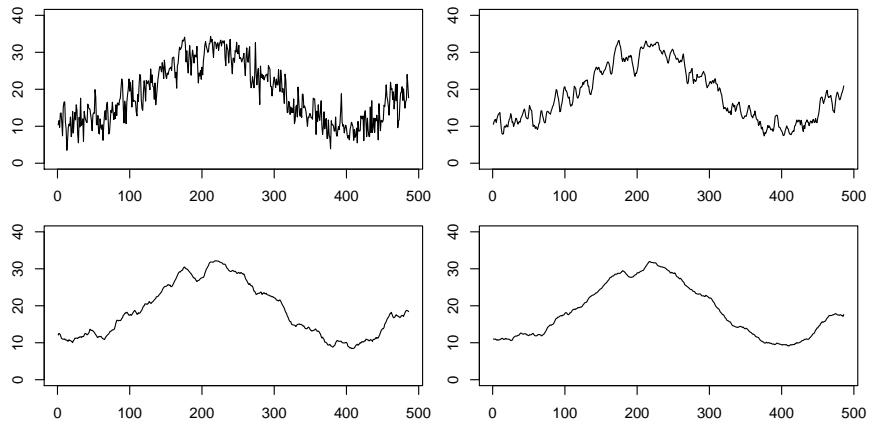


图 1.5 最高气温数据及其移动平均。左上：原数据，右上： $k=2$ 的情况，左下： $k=8$ 的情况，右下： $k=14$ 的情况。

```

> data( Temperature )
> plot( Temperature,ylim=c(0,40) ) # 数据展示
> y <- Temperature
> #
> # (2*kfilter+1) 项移动平均滤波器
> #
> kfilter <- 2 # kfilter = 2,8,14
> ndata <- length( Temperature )
> y[1:ndata] <- NA
> for(i in 1:ndata) {
> i0 <- max(i-kfilter,1)
> i1 <- min(i+kfilter,ndata)
> y[i] <- mean( Temperature[i0:i1] )
> }
> plot( y, ylim = c(0,40), type = "l")

```

图 1.5 给出了 $k=2, 8, 14$ 三种情况下，对图 1.1(c) 最高气温数据进行移动平均的结果。随着 k 增大，可得到更平滑的曲线。

移动平均的推广形式是加权移动平均：

$$T_n = \sum_{j=-k}^k w_j y_{n-j} \quad (1.16)$$

其中 w_j 为满足 $\sum_{j=-k}^k w_j = 1$, $w_j \geq 0$ 的权重系数。

将移动平均定义中的平均换成中位数，可得到 $(2k+1)$ 项移动中位数：

$$T_n = \text{median} \{y_{n-k}, \dots, y_n, \dots, y_{n+k}\} \quad (1.17)$$

移动中位数在噪声水平较低时比移动平均更能快速检测趋势变化，且不易受异常值影响，这是它的特点。

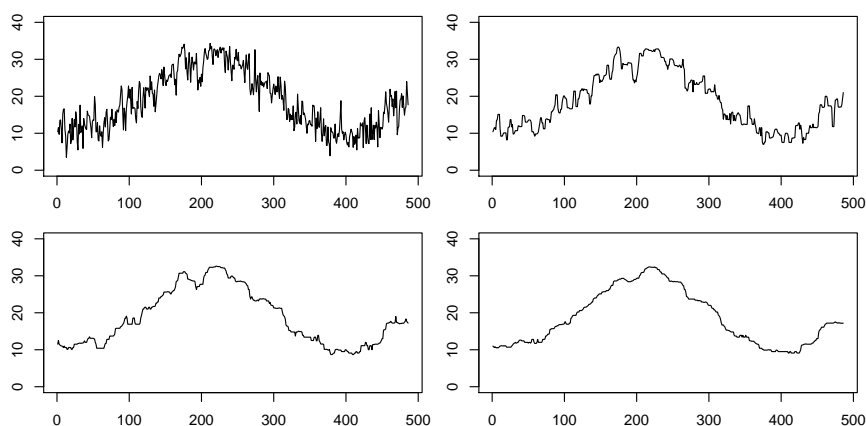


图 1.6 最高气温数据及其移动中位数。左上：原数据，右上：移动中位数 $k = 2$ ，左下： $k = 8$ ，右下： $k = 14$ 。

计算移动中位数时，只需在移动平均的 R 代码中将 mean（平均）替换为 median（中位数）即可。

```
> # (2*kfilter+1)-项移动中位数滤波器
> #
> for(i in 1:ndata) {
> i0 <- max(i-kfilter,1)
> i1 <- min(i+kfilter,ndata)
> y[i] <- median( Temperature[i0:i1] )
> }
```

图 1.6 给出了图 1.1(c) 所示最高气温数据在 $k = 2, 8, 14$ 时得到的移动中位数。可以看出移动中位数不易受原数据中较大偏差的影响。

● 章末习题

1. 说明将连续时间序列离散化时需要注意的事项。
2. 给出非高斯型时间序列的例子及其特征。
3. (1) 求 logit 变换 (1.1) 的逆变换。
(2) 求从 (a, b) 到 $(-\infty, \infty)$ 的变换及其逆变换。
4. 说明用差分实现平稳化的方法存在的问题。
5. 说明用同比 (与上一年同月相比) 去除周期成分的方法存在的问题。
6. (1) 当真实趋势为直线时, 验证三项移动平均不会改变均值, 且误差方差变为 $1/3$ 。
(2) 说明移动平均与移动中位数在特征上的差异。

第 2 章

协方差函数

本章将介绍协方差函数，它是表现和可视化平稳序列特征最基本的方法。自协方差函数表达了时间序列与自身历史变动之间的关联强度，而互协方差函数则表达了与其它序列历史变动之间的关联强度。通过这些协方差函数，可以从相关的角度捕捉时间序列特征，同时协方差函数在谱估计和时间序列模型估计中也发挥着重要作用。

2.1 时间序列的分布与平稳性

捕捉随机现象特征的手段通常采用平均 (mean) 和方差 (variance)。此外，有时也会绘制直方图 (histogram) 来粗略掌握各种数值出现的比例。在这种情况下，我们认为通过取得平均、方差或直方图，可以在一定程度上把握数据的特征。

```
> data( HAKUSAN )
> par( mfrow=c(3,2), mar=c(2,2.5,3,1)+0.1 )
> hist( HAKUSAN[,1], main="(a) Yaw rate" )
> hist( HAKUSAN[,2], main="(b) Rolling",breaks=seq(-6,12,1.2) )
> x <- as.ts( HAKUSAN[,1] )
> y <- as.ts( HAKUSAN[,2] )
> plot( x, lag(x,k=2), pch=20, main="(c) lag=2" )
> plot( y, lag(y,k=2), pch=20, main="(d) lag=2" )
> plot( x, lag(x,k=4), pch=20, main="(e) lag=4" )
> plot( y, lag(y,k=4), pch=20, main="(f) lag=4" )
```

首先，让我们探讨这种方法在时间序列中仍是否有效。上方显示了用于绘制船舶数据直方图以及当前值与历史值散点图的 R 代码。其中，参数 `par` 用于设置单张画布显示的图形数量 (`mfrow`) 以及图形的边距大小 (`mar`)。

图 1.1 中 (a) 所示的船舶偏航角速度和 (b) 的横摇这两个时间序列的直方图如图 2.1 的 (a) 和 (b) 所示。然而，可以观察到，即使在图 1.1 中特征明显不同的两个时

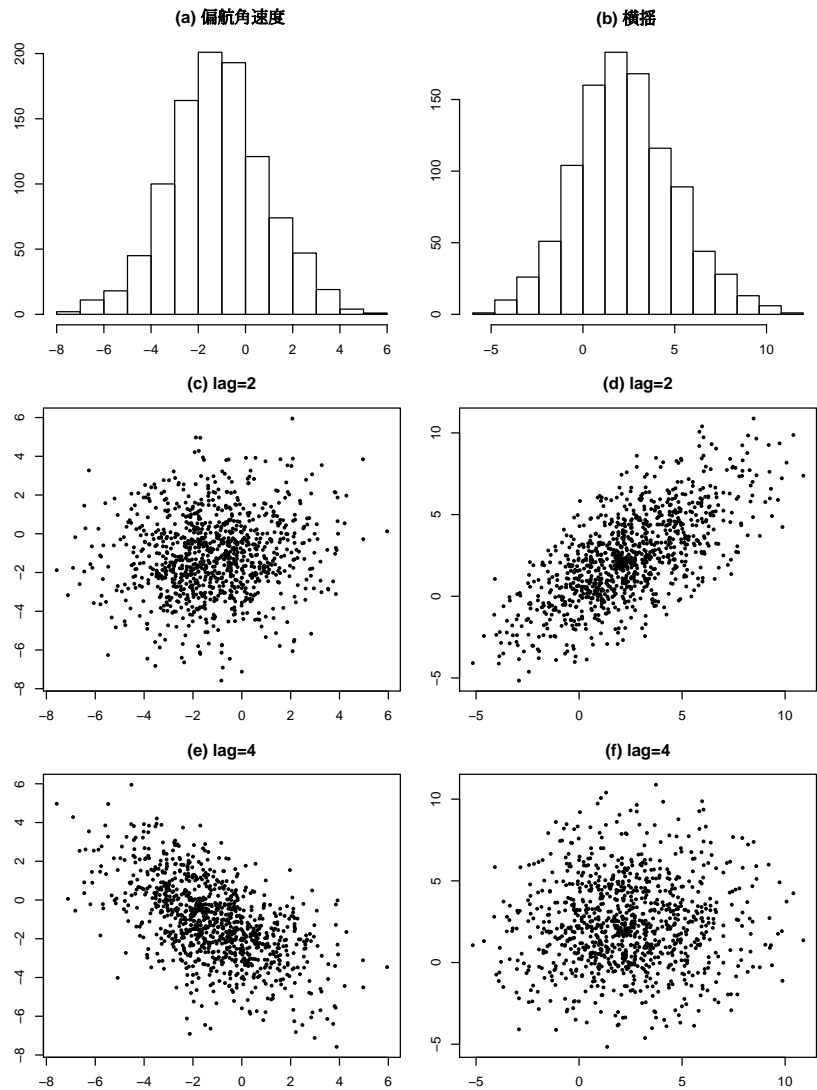


图 2.1 偏航角速度与横摇数据的直方图与散点图

间序列，仅看直方图几乎无法区分。这表明直方图无法捕捉到视觉上明显的特征差异。

另一方面，图 2.1 的 (c) 和 (d) 显示了以 y_n 为纵轴，2 期前的值 y_{n-2} 为横轴将数据绘制在二维平面上的散点图 (scatter plot)。(e) 和 (f) 则是以 4 期前的值 y_{n-4} 为横轴的结果。可以看到，(c) 的散点图在原点附近呈近似圆形分布，这表明在偏航角速度的情况下， y_n 与 y_{n-2} 几乎没有相关性。而 (d) 的散点图集中在斜率为正的直线周围，表明该时间序列与 2 期前的值有很强的正相关。但在 (e) 的散点图中可以看到斜率为负的直线关系，表明 y_n 与 y_{n-4} 之间存在负相关，而在 (f) 的情况下则几乎看不到这种相关性。

上述例子表明, 在时间序列分析中, 如果忽略这种时间上的相关性, 而仅仅考察 y_n 的分布 (边缘分布), 是无法把握时间序列的特征的。因此, 为了刻画时间序列波动的特性, 不仅需要考察 y_n 的分布, 还需要考察 y_n 与 y_{n-1} 、 y_n 与 y_{n-2} 等, 更一般地说, 需要通过考察 y_n 与 y_{n-k} 的**联合分布** (joint distribution), 来分析它们之间存在何种关联。如图 2.1(c)~(f) 所示的相关性差异, 通常可以通过 y_n 与 y_{n-k} 的协方差或相关系数加以简洁地描述。

设时间序列为 $\{y_1, \dots, y_N\}$, y_n 的期望值

$$\mu_n = E[y_n] \quad (2.1)$$

被称为时间序列 y_n 的**平均值函数** (mean value function)。此外, 时间序列 y_n 与时间平移了 k 的 y_{n-k} 之间的协方差

$$\text{Cov}(y_n, y_{n-k}) = E[(y_n - \mu_n)(y_{n-k} - \mu_{n-k})] \quad (2.2)$$

被称为时间序列 y_n 的**自协方差** (autocovariance)。特别地, 当 $k=0$ 时, 可以得到时间序列的方差函数 $\text{Var}(y_n)$ 。

不过, 由于平均值和协方差随时间变化的非平稳序列分析需要更高级的方法, 因此这类情况将留在第 8 章以后讨论, 目前我们考虑平均、方差、协方差不随时间平移而变化的情况。也就是说, 假设当 ℓ 表示任意时间平移量时, 下式成立:

$$\begin{aligned} E[y_n] &= E[y_{n-\ell}] \\ \text{Var}(y_n) &= \text{Var}(y_{n-\ell}) \\ \text{Cov}(y_n, y_m) &= \text{Cov}(y_{n-\ell}, y_{m-\ell}) \end{aligned} \quad (2.3)$$

这种时间序列被称为**弱平稳** (二阶平稳, weakly stationary, covariance stationary)。虽然在数据分布遵循正态分布 (高斯分布) 的情况下, 仅靠平均、方差和协方差就能捕捉分布特征, 但一般情况下这种假设并不成立。因此, 即使具有相同平均和方差的数据, 绘制直方图时也可能呈现完全不同的分布。

同理, 对于时间序列, 平均、方差和自协方差函数并不总能完全捕捉其特征。一般需要研究 y_1, \dots, y_N 的**联合概率密度函数** (joint probability density function) $f(y_1, \dots, y_N)$ 。为此, 只要能确定对于任意整数 k 以及任意时刻 $i_1 < i_2 < \dots < i_k$, y_{i_1}, \dots, y_{i_k} 的联合概率密度函数 $f(y_{i_1}, \dots, y_{i_k})$ 即可。特别是当此分布由 k 元正态分布表示时, 称为**高斯型时间序列**。由于正态分布完全由平均和方差协方差矩阵决定, 因此高斯型时间序列的特征可以通过求解平均向量和方差协方差矩阵来完全捕捉。

当时间序列的分布对时间平移具有不变性, 即其概率分布不随时间轴方向移动而改变时, 该时间序列被称为**强平稳** (strongly stationary)。也就是说, 时间序列强平稳是指对于任意的时间平移量 ℓ 以及任意的 i_1, \dots, i_k , 下式成立:

$$f(y_{i_1}, \dots, y_{i_k}) = f(y_{i_1-\ell}, \dots, y_{i_k-\ell}) \quad (2.4)$$

对于高斯型时间序列, 弱平稳性与强平稳性是等价的。

2.2 平稳序列的自协方差函数

假设时间序列具有平稳性, 平均值函数 μ_n 成为与时刻 n 无关的常数值, 因此记作

$$\mu = E[y_n] \quad (2.5)$$

并称之为时间序列 y_n 的**平均**。此外, y_n 与 y_{n-k} 的协方差 $\text{Cov}(y_n, y_{n-k})$ 成为仅依赖于时间差 k 的量, 因此记作

$$C_k = \text{Cov}(y_n, y_{n-k}) = E[(y_n - \mu)(y_{n-k} - \mu)] \quad (2.6)$$

并称之为平稳序列的**自协方差函数** (autocovariance function)。 k 也被称为**滞后** (lag)。特别是当 $k=0$ 时, 自协方差函数等于 y_n 的方差。自协方差函数是偶函数 ($C_\ell = C_{-\ell}$), 且满足 $|C_k| \leq C_0$ 。此外, 将 y_n 与 y_{n-k} 的相关系数

$$R_k = \frac{\text{Cov}(y_n, y_{n-k})}{\sqrt{\text{Var}(y_n)\text{Var}(y_{n-k})}} \quad (2.7)$$

视为滞后 k 的函数, 称为**自相关函数** (autocorrelation function)。在平稳序列的情况下, 由于

$$\text{Var}(y_n) = \text{Var}(y_{n-k}) = C_0 \quad (2.8)$$

成立, 自相关函数可以用自协方差函数简单地表示为:

$$R_k = \frac{C_k}{C_0} \quad (2.9)$$

例: 白噪声

当时间序列 y_n 是不相关随机变量的实现值, 且其自协方差函数满足

$$C_k = \begin{cases} \sigma^2 & k = 0 \\ 0 & k \neq 0 \end{cases} \quad (2.10)$$

时, 被称为方差为 σ^2 的**白噪声** (white noise)。

自协方差函数与自相关函数的估计

给定平稳序列 $\{y_1, \dots, y_N\}$ 时, 平均 μ 、自协方差函数 C_k 以及自相关函数 R_k 的估计值可以通过以下公式求得:

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^N y_n \quad (2.11)$$

$$\hat{C}_k = \frac{1}{N} \sum_{n=k+1}^N (y_n - \hat{\mu})(y_{n-k} - \hat{\mu}) \quad (2.12)$$

$$\hat{R}_k = \frac{\hat{C}_k}{\hat{C}_0} \quad (2.13)$$

$\hat{\mu}$ 被称为**样本平均** (sample mean), \hat{C}_k 被称为**样本自协方差函数** (sample autocovariance function), \hat{R}_k 被称为**样本自相关函数** (sample autocorrelation function)。

对于高斯过程, 样本自相关函数 \hat{R}_k 的方差近似给出如下:

$$\text{var}(\hat{R}_k) \simeq \frac{1}{N} \sum_{j=-\infty}^{\infty} (R_j^2 + R_{j-k}R_{j+k} - 4R_kR_jR_{j-k} + 2R_j^2R_k^2) \quad (2.14)$$

因此, 关于自相关, 当对于所有大于某个 m 的 $j > m$ 都有 $R_j = 0$ 成立时, \hat{R}_k ($k > m$) 的方差近似如下:

$$\text{var}(\hat{R}_k) \simeq \frac{1}{N} \sum_{j=-\infty}^{\infty} R_j^2 = \frac{1}{N} \left(1 + 2 \sum_{j=1}^{\infty} R_j^2 \right) \quad (2.15)$$

特别地, 当时间序列为白噪声, 即对于所有 $k > 0$ 都有 $R_k = 0$ 成立时, 该近似式可以简化如下:

$$\text{var}(\hat{R}_k) \simeq \frac{1}{N} \quad (2.16)$$

该近似式可用于判断时间序列是否为白噪声。例如, 当 $N = 100, 1,000, 10,000$ 时, \hat{R}_k 的标准误差分别为 0.1, 0.032, 0.01。

例: 自相关函数的计算

时间序列的自相关函数 $R_k, (k = 0, \dots, \text{lag})$ 可以通过 TSSS 包的函数 `unicor` 进行计算和绘图。通过参数 `lag` 可以指定自相关函数的最大滞后数, 若未明确指定, 则使用默认值 $2\sqrt{n}$, 其中 n 为数据量。自相关函数也可以使用 R 的标准函数 `acf` 进行计算。

```
> unicor( HAKUSAN[,1],lag=50 )
> unicor( log(Sunspot),lag=50 )
> unicor( Temperature,lag=50 )
> unicor( BLSALLFOOD,lag=50 )
> unicor( log(WHARD),lag=50 )
> unicor( MYE1F,lag=50 )
```

图 2.2 展示了对图 1.1(a)-(f) 所示的各种时间序列计算样本自相关函数的结果。(a) 的平稳序列情况下, 样本自相关函数随着滞后增加呈振荡状态并迅速衰减至 0。

在 (b) 的情况下, 对应于数据中观察到的约 11 年的重复性, 样本自相关函数也呈现出约 11 年周期的波动, 但其振幅随着滞后增加逐渐衰减。由于数据表现出显著的上下非对称性, 因此图中显示的是经过对数转换后的数据自相关函数。

在 (c) 的情况下, 由于 1 年周期的平滑趋势占据主导地位, 样本自相关函数的衰减极其缓慢。这种现象在平均值随时间变化的非平稳序列中非常典型。

在 (d) 的经济时间序列中, 由于存在年周期和趋势, 样本自相关函数也呈现出年周期, 但衰减比 (b) 还要缓慢。

在 (e) 的经济时间序列中, 数值随年份增加, 波动幅度也随之变大, 因此首先进行了对数转换, 然后求得该数据的样本自相关函数。

在 (f) 的地震数据中, 最初急剧下降后, 以约滞后 10 的间隔持续振荡, 其衰减相当缓慢。

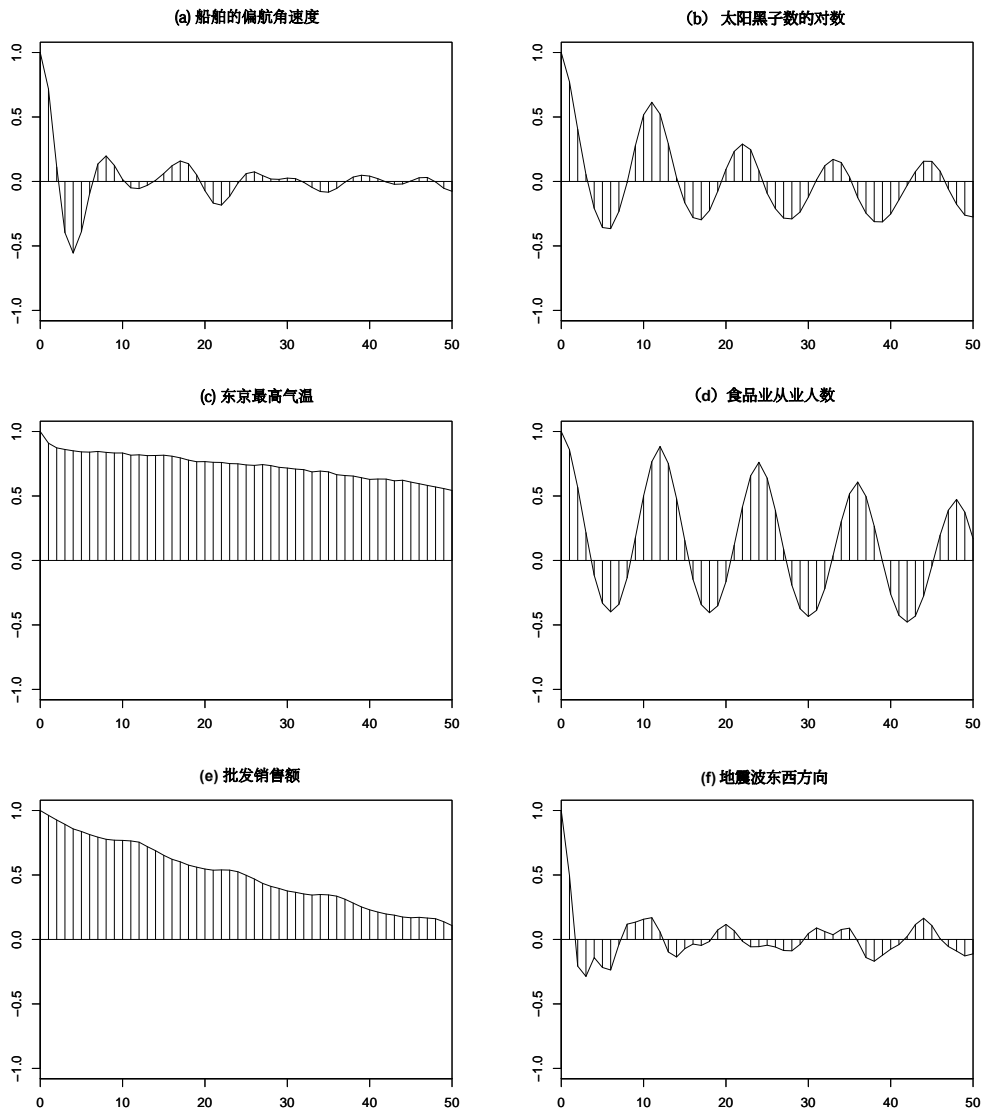


图 2.2 样本自相关函数

2.3 多元时间序列与散点图

同时观测并记录多个随机现象所得到的是多元时间序列。设同时记录的序列数量为 ℓ ，各时间序列表示为 $y_n(j)$ ($j = 1, \dots, \ell$) 时，多元时间序列成为 ℓ 维向量，表示为 $y_n = (y_n(1), \dots, y_n(\ell))^T$ 。此后， v^T 表示 v 的转置。

虽然描述一元时间序列特征使用了自协方差函数和自相关函数，但为了描述多元时间序列的特征，还需要观察不同变量之间的关系。

在前一章中，我们强调了在时间序列分析中首先将时间序列可视化是非常重要的，而对于多元时间序列 $y_n = (y_n(1), \dots, y_n(\ell))^T$ ，通过查看散点图可以在一定程度

上掌握变量间的相关性。时间序列 $y_n(i)$ 与 $y_n(j)$ 的散点图是以 $y_n(i)$ 为 x 轴, $y_n(j)$ 为 y 轴, 在二维平面上绘制其离散情况的图形。

例: 多元时间序列的直方图与散点图

使用 R 的标准函数 `pairs`, 可以绘制多元数据的直方图和散点图。

```
> pairs( HAKUSAN[,c(2,3,4)] )
```

图 2.3 显示了对图 1.1(i) 所示的三变量船舶数据 (横摇、纵摇和舵角) 中, $y_n(i)$ 与 $y_n(j)$ 的散点图。需要说明的是, 对角线上显示的是时间序列 $y_n(i)$ 的直方图。从散点图可以看到, 横摇与纵摇之间、以及纵摇与舵角之间均呈现出负相关关系; 而横摇与舵角的散点图则近似呈圆形分布, 表明这两个变量在至少同一时刻下几乎不存在相关关系。

```
> pairs( Haibara )
```

图 2.4 给出了针对图 1.1 (h) 所示的地下水位与气压数据所绘制的相应散点图。可以看到, 所有数据点几乎都集中在具有负斜率的直线附近, 由此可知地下水位的升降与气压的增减之间具有极为紧密的对应关系。

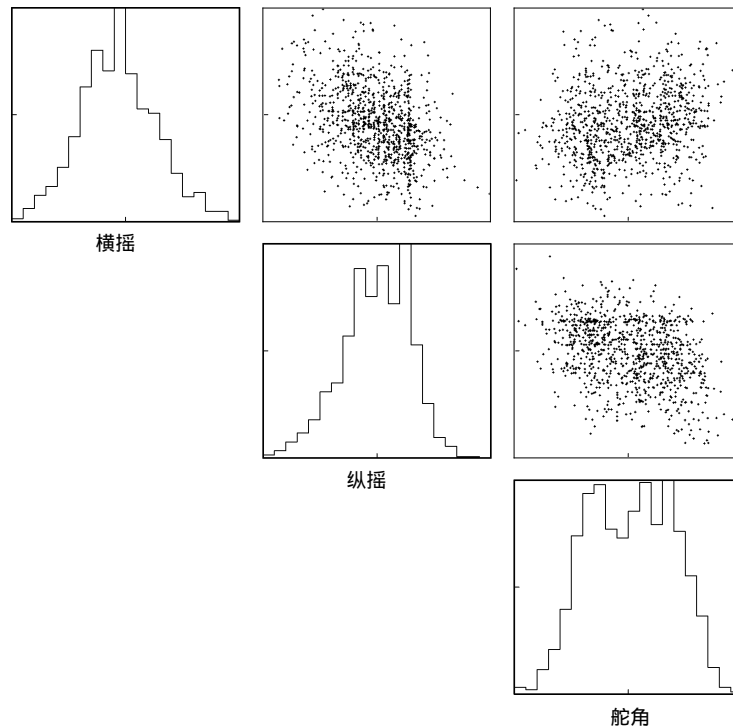


图 2.3 船舶数据的直方图及散点图

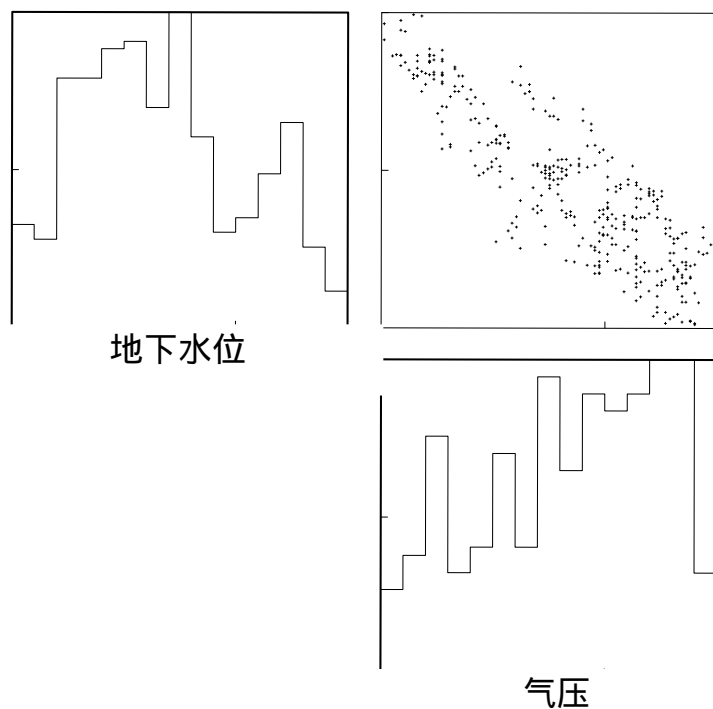


图 2.4 地下水水位与气压的直方图及散点图

由此可见, 通过观察散点图, 可以在一定程度上把握两个变量之间的关系。然而, 正如在刻画一元时间序列特征时, 需要考察 y_n 与其滞后变量 y_{n-k} 之间的关系一样, 来自其他变量的影响不仅体现在同一时刻, 还应当考虑伴随时间滞后的影响。

因此, 若要详细考察多变量之间的关系, 原则上需要对所有 i, j, k 的组合, 绘制 $y_n(i)$ 与 $y_{n-k}(j)$ 的散点图。但为了高效地表示这种包含时间滞后的变量间关系, 第 2.4 节中介绍的互协方差函数和互相关函数是一种十分便利的工具。

2.4 互协方差函数及互相关函数

在一元时间序列的情况下, 平均、自协方差函数及自相关函数是描述时间序列特征的基本统计量, 而在多元时间序列 $y_n = (y_n(1), \dots, y_n(\ell))^T$ 的情况下, 平均向量、互协方差函数及互相关函数则对应于此。首先, 第 i 个时间序列 $y_n(i)$ 的平均定义为:

$$\mu(i) = E[y_n(i)] \quad (2.17)$$

此时 $\boldsymbol{\mu} = (\mu(1), \dots, \mu(\ell))^T$ 被称为多元时间序列 y_n 的**平均向量** (mean vector)。此外, 时间序列 $y_n(i)$ 与时间序列 $y_{n-k}(j)$ 的协方差通过下式求得:

$$\begin{aligned} C_k(i, j) &= \text{Cov}(y_n(i), y_{n-k}(j)) \\ &= \text{E}[(y_n(i) - \mu(i))(y_{n-k}(j) - \mu(j))^T] \end{aligned} \quad (2.18)$$

此时, $\ell \times \ell$ 矩阵

$$C_k = \begin{bmatrix} C_k(1, 1) & \cdots & C_k(1, \ell) \\ \vdots & \ddots & \vdots \\ C_k(\ell, 1) & \cdots & C_k(\ell, \ell) \end{bmatrix} \quad (2.19)$$

被称为滞后 k 的互协方差。此外, 当将 C_k ($k = 0, 1, 2, \dots$) 视为滞后 k 的函数时, 称为**互协方差函数** (cross-covariance function)。互协方差函数的对角元 $C_k(i, i)$ 即为第 i 个时间序列 $y_n(i)$ 的自协方差函数。此外, 将时间序列 $y_n(i)$ 与 $y_{n-k}(j)$ 的相关系数表示为

$$\begin{aligned} R_k(i, j) &= \text{Cor}(y_n(i), y_{n-k}(j)) \\ &= \frac{C_k(i, j)}{\sqrt{C_0(i, i)C_0(j, j)}} \end{aligned} \quad (2.20)$$

则

$$R_k = \begin{bmatrix} R_k(1, 1) & \cdots & R_k(1, \ell) \\ \vdots & \ddots & \vdots \\ R_k(\ell, 1) & \cdots & R_k(\ell, \ell) \end{bmatrix} \quad (2.21)$$

被称为**互相关函数** (cross-correlation function)。自协方差函数及自相关函数是偶函数, 满足 $C_{-k} = C_k$, $R_{-k} = R_k$, 但对于互协方差函数及互相关函数, 此性质不成立。然而, 由于存在

$$C_{-k} = C_k^T, \quad R_{-k} = R_k^T \quad (2.22)$$

的关系, 因此在多元时间序列的情况下, 也只需计算 $k \geq 0$ 的 C_k 和 R_k 即可。观测到长度为 N 的多元时间序列 $\{y_1(j), \dots, y_N(j)\}$ ($j = 1, \dots, \ell$) 时, 平均 $\mu(i)$ 、互协方差函数 $C_k(i, j)$ 以及互相关函数 $R_k(i, j)$ 的估计值可以通过以下公式求得:

$$\hat{\boldsymbol{\mu}}(i) = \frac{1}{N} \sum_{n=1}^N y_n(i) \quad (2.23)$$

$$\hat{C}_k(i, j) = \frac{1}{N} \sum_{n=k+1}^N (y_n(i) - \hat{\boldsymbol{\mu}}(i))(y_{n-k}(j) - \hat{\boldsymbol{\mu}}(j)) \quad (2.24)$$

$$\hat{R}_k(i, j) = \frac{\hat{C}_k(i, j)}{\sqrt{\hat{C}_0(i, i)\hat{C}_0(j, j)}} \quad (2.25)$$

ℓ 维向量 $\hat{\boldsymbol{\mu}}$ 被称为**样本平均向量** (sample mean vector), $\ell \times \ell$ 矩阵 \hat{C} 和 \hat{R} 分别被称为**样本互协方差函数** (sample cross-covariance function) 和**样本互相关函数** (sample cross-correlation function)。

例：船舶数据

多元时间序列的互相关函数可以如下所示，通过 TSSS 包的函数 `crscor` 进行计算。该函数若未明确指定滞后 `lag`，则使用默认值 $2\sqrt{n}$ 进行计算，其中 n 表示数据量。

```
> data( HAKUSAN )
> par( mar=c(2,3,2,1)+0.1 )
> crscor( HAKUSAN[,2:4], lag = 50 ) # Rolling, Pitching, Rudder
```

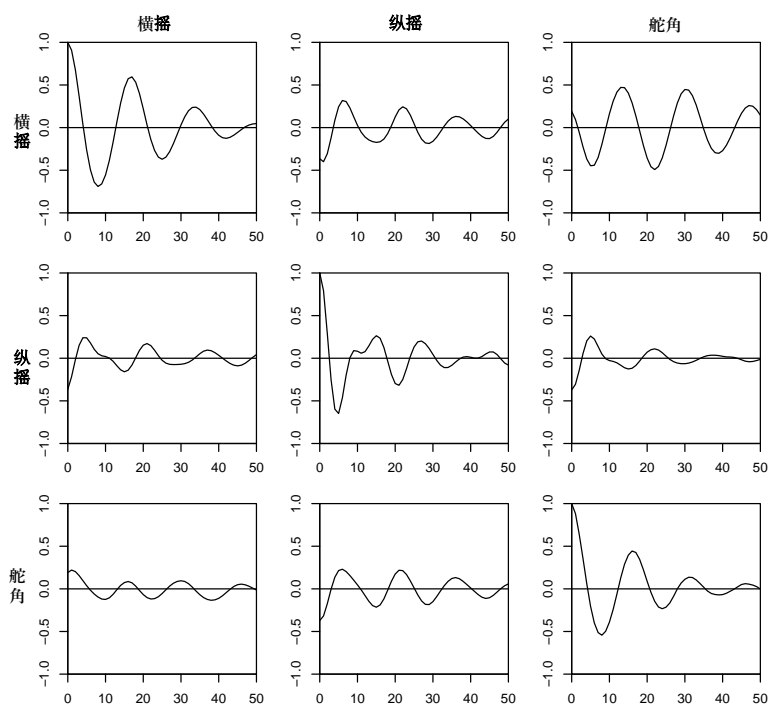


图 2.5 船舶数据的自相关函数与互相关函数

图 2.5 显示了对图 1.1 (i) 的船舶数据（横摇、纵摇、舵角）求得的样本互相关函数。纵向为 i ，横向为 j ，在各个位置显示了相关函数 $R_k(i, j)$ ($k = 0, \dots, 50$)。因此，对角线上的三个图代表自相关函数 $R_k(i, i)$ ，其余 6 个代表互相关函数 $R_k(i, j)$ 。从这些图中可以看出，横摇和舵角表现出相当强的周期性波动。另一方面，纵摇呈现出稍复杂的波动。此外，横摇和舵角的自相关函数极其相似，且由于从舵角到横摇的互相关 $R_k(1, 3)$ 极高，可知横摇与舵角之间存在密切关系。

例：地下水位数据

```
> data( Haibara )  
> crscor( Haibara,lag=50 )
```

图 2.6 显示了图 1.1 (h) 的地下水位与气压的互相关函数。由于两个变量都包含缺失值, 因此采用了与 2.3 节相同的方法进行处理。两个变量之间表现出极其强的相关性。

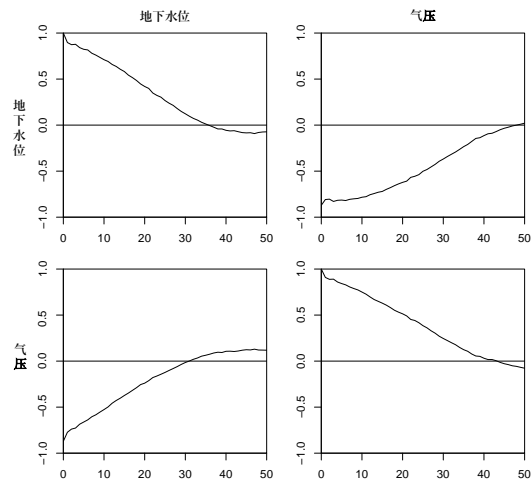


图 2.6 地下水位与气压的自相关函数与互相关函数

● 章末习题

1. 思考弱平稳的高斯型时间序列为何是强平稳的。
2. 强平稳序列是否可以说成是弱平稳的?
3. 证明当时间序列平稳时, 自协方差函数是偶函数。
4. 设 v_n 为平均为 0、方差为 1 的白噪声, 求时间序列 $y_n = v_n - cv_{n-1}$ ($|c| < 1$) 的自协方差函数。
5. 设 C_0, C_1, \dots , 为平稳序列的自协方差函数, 证明矩阵

$$C = \begin{bmatrix} C_0 & C_1 & \cdots & C_{k-1} \\ C_1 & C_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & C_1 \\ C_{k-1} & \cdots & C_1 & C_0 \end{bmatrix}$$

是非负定的。

6. (1) 求样本自协方差函数 \hat{C}_k 的期望值。
(2) 思考在样本自协方差函数的定义式中, 为何除以 N 而不是 $N-k$ 。
7. (1) 当时间序列为平均为 0 的白噪声时, 求 \hat{C}_k 、 \hat{R}_k 的分布。
(2) 利用 (1) 的结果, 思考一种判断时间序列是否为白噪声的简单方法。

第 3 章

谱与周期图

时间序列的一种分析方法，是把时间序列的波动分解为具有各种周期的三角函数，并用各个周期成分的强弱来表征该时间序列特征的谱分析方法。本章将定义时间序列的功率谱以及周期图，并说明其计算方法、改进方式、平滑化方法等。另外，还将说明如何利用快速傅里叶变换 FFT 高效地计算周期图。

3.1 谱

当滞后 k 变大时自协方差函数 C_k 快速衰减，并且满足

$$\sum_{k=-\infty}^{\infty} |C_k| < \infty$$

时，就可以定义 C_k 的傅里叶变换 (Fourier transform)。此时，在频率 $-\frac{1}{2} \leq f \leq \frac{1}{2}$ 上定义的函数

$$p(f) = \sum_{k=-\infty}^{\infty} C_k e^{-2\pi i k f} \quad (3.1)$$

称为功率谱密度函数 (power spectral density function)，或简称为谱 (spectrum)*¹。

由于自协方差函数为偶函数且满足 $C_k = C_{-k}$ ，因此谱也可表示为

$$p(f) = \sum_{k=-\infty}^{\infty} C_k \cos 2\pi k f = C_0 + 2 \sum_{k=1}^{\infty} C_k \cos 2\pi k f \quad (3.2)$$

当用各种周期的三角函数来表示时间序列的波动时，可以把谱理解为：这些不同周期成分以怎样的比例包含在时间序列之中。反过来，若给定谱，则自协方差函数可由傅里叶逆变换得到：

$$C_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} p(f) e^{2\pi i k f} df = \int_{-\frac{1}{2}}^{\frac{1}{2}} p(f) \cos 2\pi k f df \quad (3.3)$$

*¹ (译注) 原文为“スペクトル” (spectrum)。在中文文献中，该术语的翻译因学科领域而异：物理学中多译为“谱”，信号处理领域常译为“频谱”，而在时间序列分析和统计学领域中一般简称为“谱”。本书统一采用“谱”这一译法。

从而得到 C_k 。

例 白噪声的谱

白噪声的自协方差函数为 $C_0 = \sigma^2$, $C_k = 0$ ($k \neq 0$)。因此此时

$$p(f) = \sum_{k=-\infty}^{\infty} C_k \cos 2\pi k f = C_0 = \sigma^2 \quad (3.4)$$

可知白噪声的谱与频率 f 无关, 为常数。这表明白噪声以相同的比例包含各种频率的波动成分。

图 3.1 (a)~(c) 给出了方差 $\sigma^2 = 1$ 的白噪声的自相关函数、功率谱以及通过仿真生成的实现值。关于生成时间序列的仿真方法将在第 16 章说明。

例 自回归模型的谱

令 w_n 为方差 σ^2 的白噪声。若时间序列按 $y_n = ay_{n-1} + w_n$ (一阶自回归模型) 生成, 则自协方差函数为 $C_k = \sigma^2(1-a^2)^{-1}a^{|k|}$ 。因此该时间序列的谱为

$$p(f) = \frac{\sigma^2}{|1 - ae^{-2\pi if}|^2} = \frac{\sigma^2}{1 - 2a \cos 2\pi f + a^2} \quad (3.5)$$

图 3.2 的 (a)~(c) 给出了取 $a = 0.9$ 时的自相关函数、谱以及仿真实实现值示例。同样地, 图 3.3 给出了取 $a = -0.9$ 时的情况。

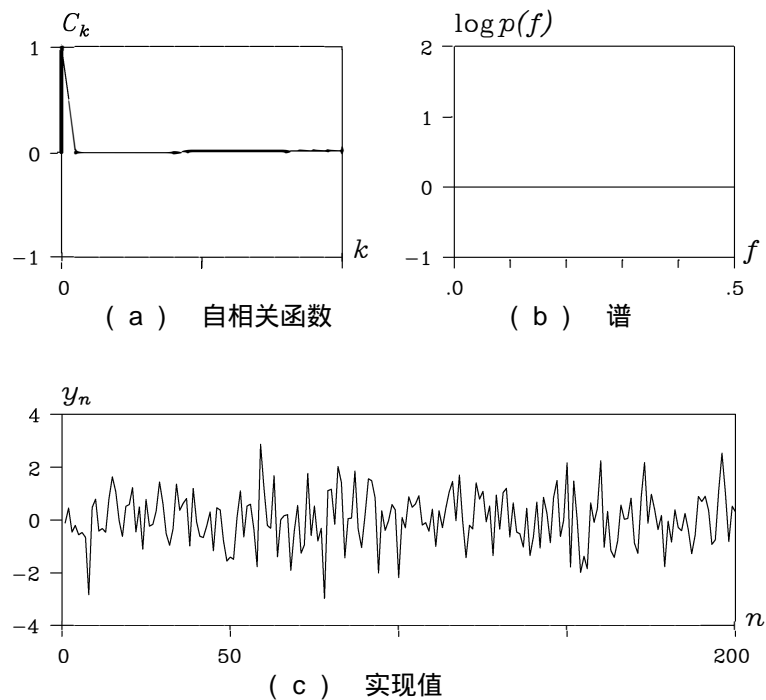


图 3.1 白噪声的自相关函数、谱以及实现值

此外，若时间序列服从 $y_n = a_1 y_{n-1} + a_2 y_{n-2} + w_n$ (二阶自回归模型)，则自相关函数满足

$$R_1 = \frac{a_1}{1 - a_2}, \quad R_k = a_1 R_{k-1} + a_2 R_{k-2} \quad (3.6)$$

相应的谱为

$$\begin{aligned} p(f) &= \frac{\sigma^2}{|1 - a_1 e^{-2\pi i f} - a_2 e^{-4\pi i f}|^2} \\ &= \frac{\sigma^2}{1 - 2a_1(1 - a_2) \cos 2\pi f - 2a_2 \cos 4\pi f + a_1^2 + a_2^2} \end{aligned} \quad (3.7)$$

图 3.4 给出了取 $a_1 = 0.9\sqrt{3}$, $a_2 = -0.81$ 时的 (a) 自相关函数、(b) 谱以及 (c) 时间序列实现值。

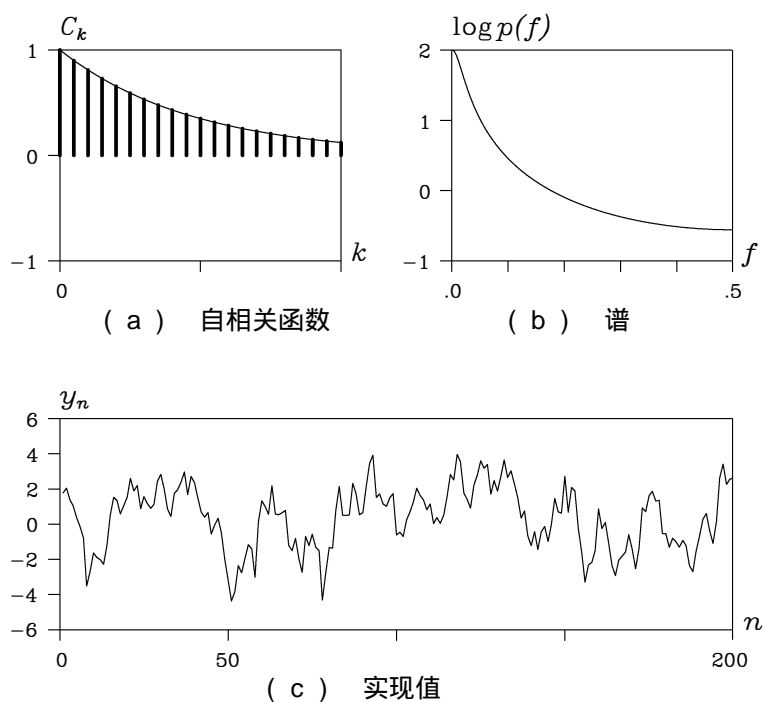


图 3.2 一阶自回归模型 ($a = 0.9$) 的自相关函数、谱以及实现值

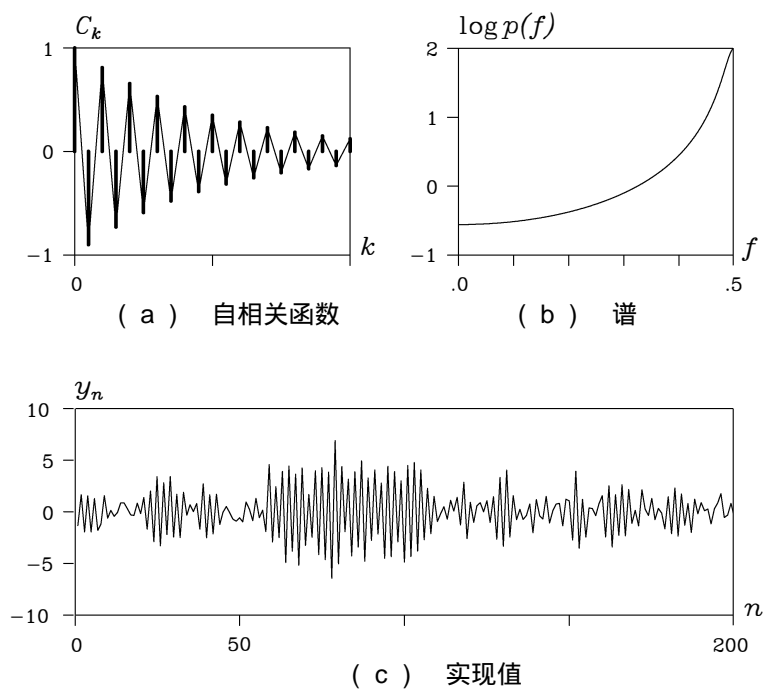


图 3.3 一阶自回归模型 ($a = -0.9$) 的自相关函数、谱以及实现值

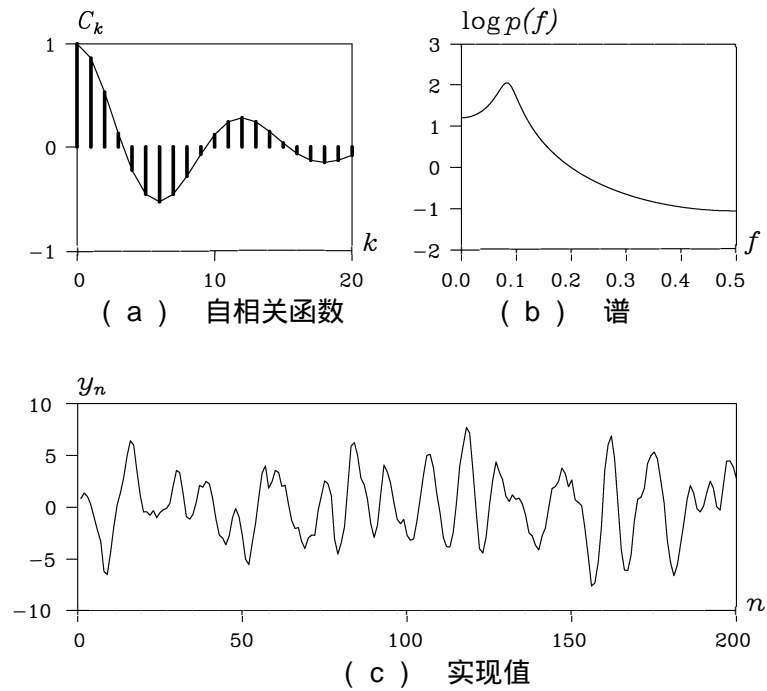


图 3.4 二阶自回归模型的自相关函数、谱以及实现值

3.2 周期图

给定时间序列 y_1, \dots, y_N 时, 用样本自协方差函数替代 (3.1) 或 (3.2) 中的 C_k 可得

$$p_j = \sum_{k=-N+1}^{N-1} \hat{C}_k e^{-2\pi i k f_j} = \hat{C}_0 + 2 \sum_{k=1}^{N-1} \hat{C}_k \cos 2\pi k f_j \quad (3.8)$$

这称为**周期图** (periodogram)。其中频率只考虑自然频率 $f_j = j/N$, $j = 0, \dots, [N/2]$ 。

$[N/2]$ 表示不超过 $N/2$ 的最大整数。此外, 将周期图的定义域扩展到连续区间

$$\hat{p}(f) = \sum_{k=-N+1}^{N-1} \hat{C}_k e^{-2\pi i k f}, \quad -0.5 \leq f \leq 0.5 \quad (3.9)$$

称为**样本谱** (sample spectrum)。周期图也可以看作是仅在自然频率处计算得到的

样本谱。对应 (3.3), 样本谱与样本自协方差函数之间满足:

$$\hat{C}_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{p}(f) e^{2\pi i k f} df, \quad k = 0, \dots, N-1 \quad (3.10)$$

例 在 R 包 TSSS 中, 函数 period 把参数设为 window=0 即可计算周期图。

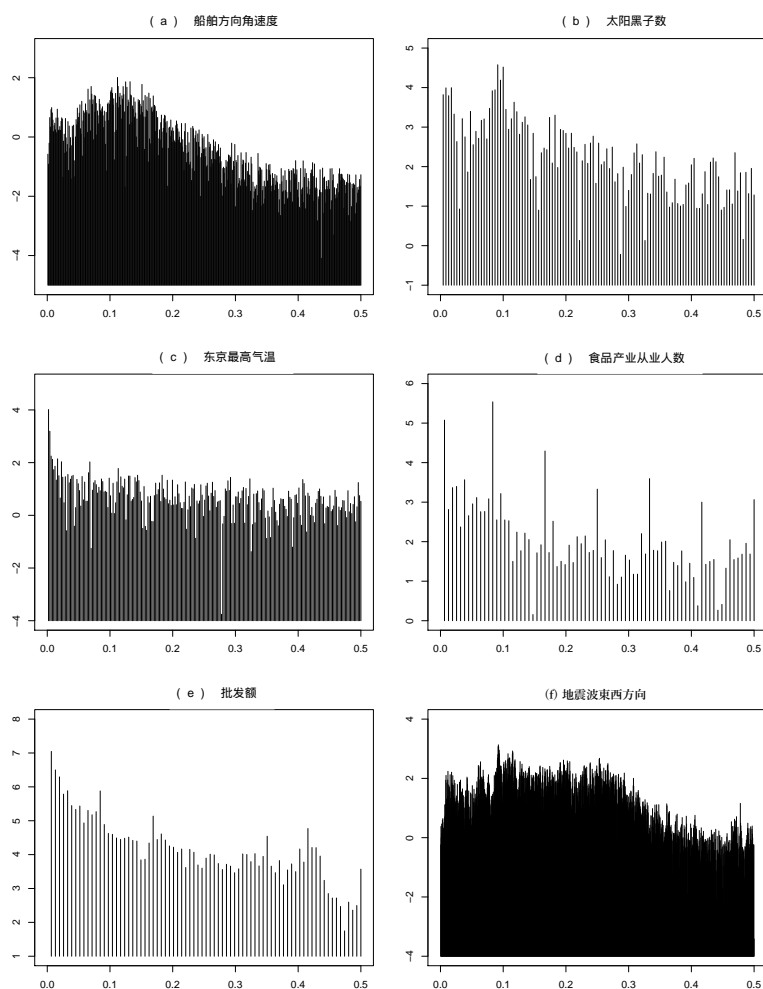


图 3.5 图 1.1 数据的周期图 (纵轴以对数刻度表示)

```

> period( HAKUSAN[,1],window=0 )
> period( Sunspot, window=0 )
> period( Temperature,window=0 )
> period( BLSALLFOOD, window=0)
> period( WHARD,window=0 )
> period( MYE1F, window=0 )

```

图 3.5 给出了图 1.1(a)~(f) 所示单变量时间序列的周期图。其中纵轴以对数刻度表示。

(a) 为船舶偏航角速度数据, 可见约 10 秒 ($f = 0.1$) 的周期波动成分相对较强。

(b) 为太阳黑子数数据, 可见约 11 年 ($f = 0.09$) 的强周期成分。在 $f = 0.18$ 附近也有峰, 但由于波动较大, 因此并不十分清晰。

(c) 最高气温数据在 $f = 0$ 处有很强的趋势成分, 除 $f = 0.064$ 附近的弱峰外, 几乎看不到明显周期性。

(d) 食品产业从业人数数据对应季节波动, 在 $f = \frac{1}{12}$ 处出现显著峰值。右侧 5 个峰的位置是 f 的整数倍, 可视为由波形非线性导致的高次谐波。

(e) 批发额数据也能看到与 (d) 类似的峰, 但由于趋势较强, 因此不如 BLSALL-FOOD 那样显著。

(f) 地震数据在 $f = 0.1$ 及 $f = 0.25$ 附近有隆起, 尤其在 $f = 0.07$ 与 0.1 处有尖锐峰值。这可能对应 P 波与 S 波的频率特性。

下面稍微考察一下按上述方式得到的周期图的性质。首先, 周期图与样本谱在渐近意义下是不偏的, 即

$$\lim_{N \rightarrow \infty} E[\hat{p}(f)] = p(f) = \sum_{k=-\infty}^{\infty} C_k \cos 2\pi k f \quad (3.11)$$

成立。这意味着在各频率 f 上, 样本谱的期望会随着样本量增加而逼近真实谱 $p(f)$ 。但是, 这并不意味着由数据计算得到的 $\hat{p}(f)$ 会在每个频率上随数据增多而收敛到 $p(f)$ (一致性)。实际上,

$$\frac{2\hat{p}(f_1)}{p(f_1)}, \dots, \frac{2\hat{p}(f_{[\frac{N}{2}-1]})}{p(f_{[\frac{N}{2}-1]})} \quad (3.12)$$

彼此独立并服从自由度 2 的 χ^2 分布; 两端的 $\hat{p}(0)/p(0)$ 与 $\hat{p}(0.5)/p(0.5)$ 服从自由度 1 的 χ^2 分布。因此, 周期图的方差与样本数 N 无关而保持常数, 所以它不可能成为一致估计量。

例: 样本自相关函数与周期图

在 R 中可用包 TSSS 的函数 `unicor` 与 `period` 计算样本自相关函数与周期图。`rnorm` 用于生成正态随机数序列 (长度由参数指定)。下面用 `rnorm` 生成长度为 3,200 的白噪声, 并从中截取长度为 200、800、3,200 的部分作为测试数据。

```

> r <- as.ts( rnorm(3200) )
> unicolor( r, lag=50 )
> x <- period( r, window=0 )
> plot( log10(x$period), type="l", ylim=c(-4,1), xaxt="n" )
> axis( side=1, at=c(0,320,640,960,1280,1600), labels=c(0.0,0.1,0.2,0.3,0.4,0.5) )
>
> r1 <- r[1:800]
> unicolor(r1, lag=50)
> x1 <- period(r1, window=0)
> plot(log10(x1$period), type="l", ylim=c(-4,1), xaxt="n")
> axis( side=1, at=c(0,80,160,240,320,400), labels=c(0.0,0.1,0.2,0.3,0.4,0.5) )
>
> r2 <- r[1:200]
> unicolor(r2, lag=50)
> x2 <- period(r2, window=0)
> plot(log10(x2$period), type="l", ylim=c(-4,1), xaxt="n")
> axis( side=1, at=c(0,20,40,60,80,100), labels=c(0.0,0.1,0.2,0.3,0.4,0.5) )

```

对样本数 $N = 200$ 的白噪声实现值（生成方式与图 3.1(c) 相同），图 3.6(a) 给出样本自相关函数，(b) 给出周期图。样本自相关函数接近 0，大多落在误差范围 $[-1/\sqrt{200}, 1/\sqrt{200}] \simeq [-0.07, 0.07]$ 之内。白噪声谱的理论值为常数，此处 $\log p(f) \equiv 0$ 但周期图上下剧烈波动，并不是对真实谱的良好估计。

图 3.6 的 (c)~(f) 给出了将样本数 N 分别增至 800、3200（每次扩大 4 倍）后得到的样本自相关函数与周期图。样本自相关函数随着样本数增加而收敛到真实值 $C_k = 0$ 。而周期图即便样本数增加，波动幅度也不减小，只是上下波动次数随 N 成比例增加，完全不会收敛到真实谱。这表明周期图不是谱的一致估计量。

3.3 周期图的平均与平滑化

接下来考虑如何构造能够收敛到真实谱的估计量。用 (3.8) 的替代形式，对 $f_j = \frac{j}{2L}$ ($j = 0, \dots, L$) 计算

$$p_j = \hat{C}_0 + 2 \sum_{k=1}^{L-1} \hat{C}_k \cos 2\pi k f_j \quad (3.13)$$

其中 L 取为小于 N 的适当整数。这样得到的 p_j 称为原始谱 (raw spectrum)。图 3.7(a) 与 (b) 分别对应图 3.6(d) 与 (e) 的数据，固定 $L = 200$ 计算得到 p_j 。此时随着样本数增加，上下波动幅度大致逐次减半，看起来像在收敛到真实谱。

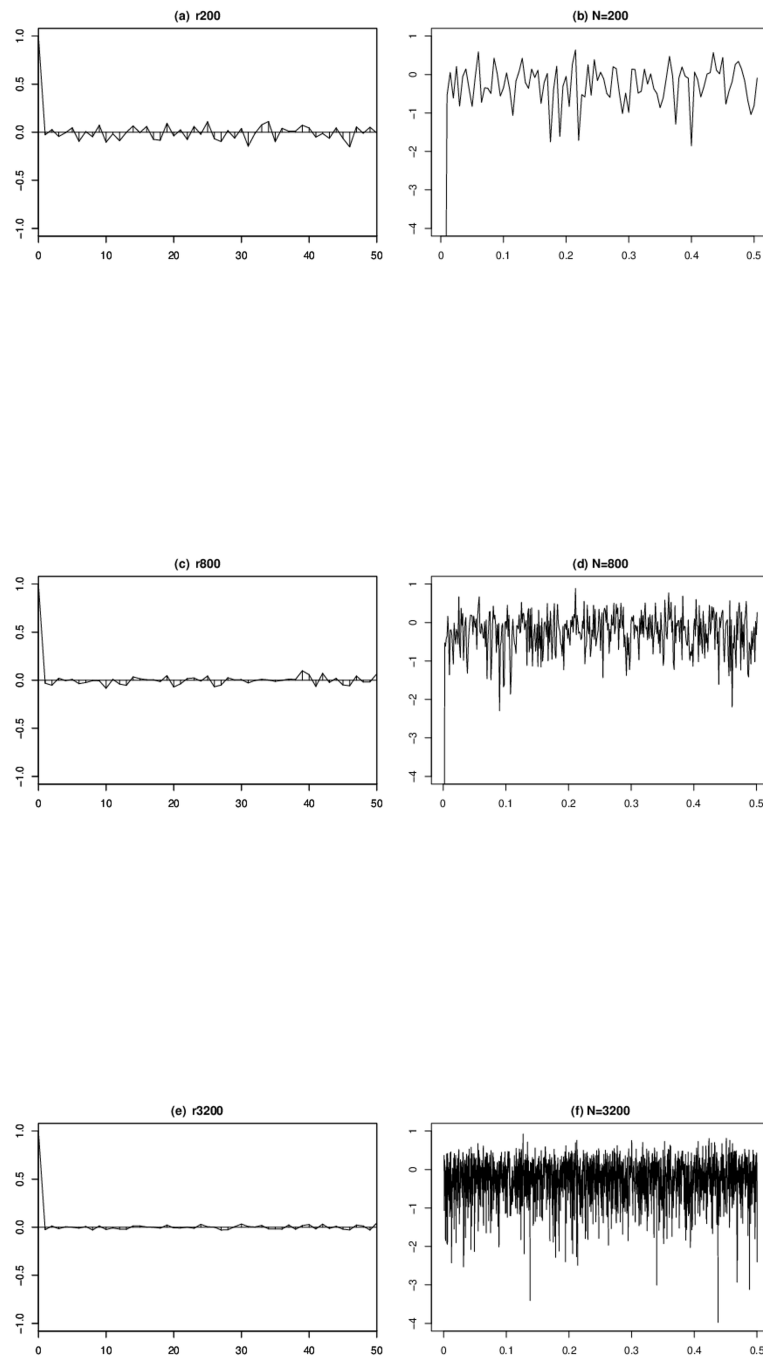


图 3.6 正态白噪声的样本自相关函数与周期图。自上而下数据量分别为 $n = 200$ 、800、1,600。

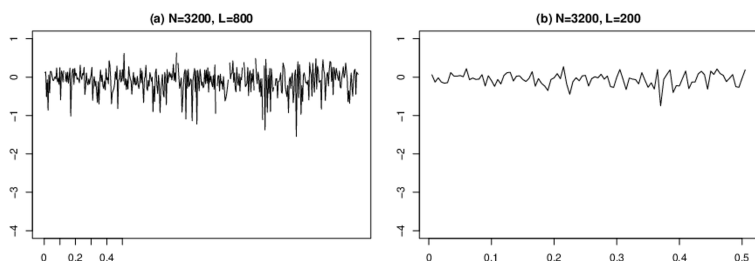


图 3.7 取 $L = 200$ 计算得到的白噪声原始谱。左: $N = 800$, 右: $N = 3,200$ 。

```

> x <- period( r,window=0, lag=800 )
> plot(log10( x$period), type="l", ylim=c(-4,1), xaxt="n" )
> axis( side=1, at=c(0,80,160,240,320,400), labels=c(0.0,0.1,0.2,0.3,0.4,0.5) )
>
> x <- period( r,window=0, lag=200 )
> plot( log10(x$period), type="l", ylim=c(-4,1), xaxt="n" )
> axis( side=1, at=c(0,80,160,240,320,400), labels=c(0.0,0.1,0.2,0.3,0.4,0.5) )

```

当样本数 N 增加时, 周期图定义 (3.8) 要求样本自协方差函数的滞后一直接到 $N-1$ 。若不使用全部滞后, 而是固定到某个常数 $L-1$, 原始谱就会收敛到真实谱。当样本数 N 按 $N = \ell L$ ($\ell = 1, 2, \dots$) 增长时, 用固定滞后 $L-1$ 计算原始谱, 等价于如下处理:

先将时间序列 y_1, \dots, y_n 按长度 L 分割为 N/L 段, 对每段计算周期图 $p_j^{(i)}$ ($j = 0, \dots, [L/2]; i = 1, \dots, N/L$), 再对每个 $j = 0, \dots, [L/2]$ 对这 N/L 个估计取平均:

$$p_j = \frac{L}{N} \sum_{i=1}^{N/L} p_j^{(i)} \quad (3.14)$$

从而得到平均后的周期图。在这种情况下, 即使样本数 N 变为 ℓL 并随 ℓ 倍增加, 各 $p_j^{(i)}$ 的方差不变, 但由于取了 ℓ 个平均, p_j 的方差变为 $\frac{1}{\ell}$, 并随 ℓ 增大而收敛到 0。因此, 简单周期图无论样本数如何增加都不收敛到真实谱; 但只要把用于傅里叶变换的自协方差个数固定, 就能构造出收敛到真实值的估计量。

不过, (3.13) 定义的原始谱与 (3.14) 得到的平均周期图在严格意义下并不一致, 并且可能出现 $p_j < 0$ 。为避免这一点, 可按 (3.14) 直接计算; 而在实际进行谱估

表 3.1 由 (3.8) 与 (3.14) 得到的周期图与对数周期图的方差。

数据量 N	200	800	3200
p_j by (3.8)	1.006	0.998	1.010
p_j by (3.14)	1.006	0.250	0.061
$\log p_j$ by (3.8)	0.318	0.309	0.315
$\log p_j$ by (3.14)	0.318	0.053	0.012

窗函数	m	W_0	W_1
Hanning	1	0.5	0.25
Hamming	1	0.54	0.23

计时, 通常会进一步用谱窗 (spectral window) 对原始谱做平滑化: 若给定谱窗 W_i ($i = 0, \pm 1, \dots, \pm m$), 则

$$\hat{p}_j = \sum_{i=-m}^m W_i p_{j-i}, \quad (j = 0, 1, \dots, [\frac{L}{2}]) \quad (3.15)$$

得到谱估计值。其中规定 $p_{-j} = p_j$, $p_{[\frac{L}{2}]+i} = p_{[\frac{L}{2}]-i}$ 。这不仅能避免负估计值, 也能减小估计方差。谱窗的常见形式如下 (均为左右对称, 满足 $W_{-i} = W_i$)。另外, 为保证 $\hat{p}_j > 0$, 必须有 $W_i > 0$ 。

例: 平滑周期图

平滑周期图也可用函数 `period` 计算。该函数默认设置为 `lag=2√N` 且 `window=1`。也可以通过参数 `lag` 与 `window` 指定其他滞后 L 或窗类型。

`window`: 窗类型

0: boxcar (矩形窗)

1: Hanning

2: Hamming

`lag`: 自协方差函数的最大滞后。若为 Null (默认) 则

`window=0`: `lag=n-1`

`window>0`: `lag=2√n`

其中 n 为数据量

```
> period( HAKUSAN[,1] )
> period( Sunspot )
> period( Temperature )
> period( BLSALLFOOD )
> period( WHARD )
> period( MYE1F )
```

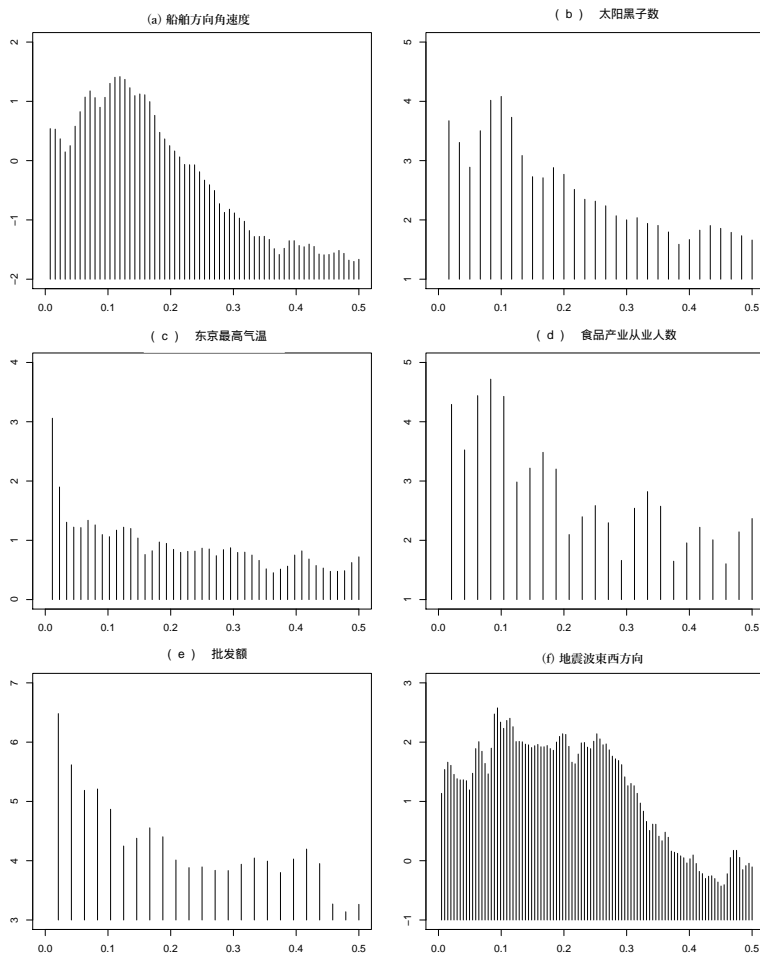


图 3.8 图 1.1 所示数据的平滑周期图。横轴: 周波数 f , 纵轴: 对数周期图 $\log p(f)$ 。

图 3.8 给出了取 $L = 2\sqrt{N}$ 并用 Hanning 窗平滑后的周期图 (数据与图 3.5 对应)。当 HAKUSAN[1] 或 MYE1F 这类数据量较大时, 波动得到抑制, 能得到较好的估计值。但对 BLSALLFOOD 或 WHARD 这类具有显著周期性的情况, 图 3.5 中可见的线状谱会因平滑而变得不够清晰。为解决这一问题, 可利用第 6 章的时间序列模型进行谱估计, 这是有效的方法。

3.4 周期图的计算方法

为了高效计算傅里叶变换, 有 Görtzel 法。一般而言, 按定义计算数列 x_0, \dots, x_{L-1} 的傅里叶余弦变换与傅里叶正弦变换

$$X_c(f) = \sum_{n=0}^{L-1} x_n \cos 2\pi n f, \quad X_s(f) = \sum_{n=0}^{L-1} x_n \sin 2\pi n f$$

则各自需要 L 次乘加运算与 L 次三角函数计算。利用三角函数加法公式的 Görtzel 法, 只需要计算两次三角函数: $\cos 2\pi f$ 与 $\sin 2\pi f$ 。其思想是令 $a_0 = 0, a_1 = 1$, 并对 $n = 2, \dots, L-1$ 用递推 $a_n = 2a_{n-1} \cos 2\pi f - a_{n-2}$ 求得 a_n , 则有

$$\sin 2\pi n f = a_n \sin 2\pi f, \quad \cos 2\pi n f = a_n \cos 2\pi f - a_{n-1}$$

从而得到 $\sin 2\pi n f$ 与 $\cos 2\pi n f$ 。

3.5 基于 FFT 的周期图计算

上一节通过对样本自协方差函数做离散傅里叶变换来求周期图, 但一般而言, 长度 N 数列的离散傅里叶变换需要 N^2 次乘加运算, 当 N 很大时会耗费很长计算时间。相比之下, 使用 **FFT** (fast Fourier transform) 时, 当数据量满足 $N = p^\ell$ 形式, 可用大约 $Np\ell$ 次运算完成。因此运算量可由 N^2 降为 $Np\ell$, 比例为 $Np\ell/N^2 = p\ell/N$ 。例如, $N = 1,024 = 2^{10}$ 时运算量约为 $\frac{1}{50}$, $N = 32,768 = 2^{15}$ 时约为 $\frac{1}{1000}$ 。

不过, 按 (3.8) 通过傅里叶变换样本自协方差函数计算周期图时, 还需要先计算样本自协方差函数 \hat{C}_k ($k = 0, \dots, N-1$), 这也需要 $\frac{N^2}{2}$ 次运算。因此用 FFT 计算周期图时, 通常采用直接对时间序列做傅里叶变换的方法。该方法先计算

$$\begin{aligned} X_j &= \sum_{n=1}^N y_n e^{-2\pi i(n-1)j/N} \\ &= \sum_{n=1}^N y_n \cos \frac{2\pi(n-1)j}{N} - i \sum_{n=1}^N y_n \sin \frac{2\pi(n-1)j}{N} \\ &\equiv FC_j - iFS_j \end{aligned} \quad (3.16)$$

即时间序列 y_1, \dots, y_N 的傅里叶变换 $X_j, j = 0, \dots, \left[\frac{N}{2}\right]$, 然后用

$$p_j = \frac{|X_j|^2}{N} = \frac{1}{N} \left| \sum_{n=1}^N y_n e^{-2\pi i(n-1)j/N} \right|^2 = \frac{FC_j^2 + FS_j^2}{N} \quad (3.17)$$

得到周期图。可以容易验证 (3.17) 与 (3.8) 中定义的周期图是一致的。对长度为 $N = 2^\ell$ 的时间序列, FFT 尤其容易实现。

对于不能写成 $N = 2^\ell$ 形式的一般长度序列, 可利用素因数分解 $N = p_1^{\ell_1} \times \dots \times p_m^{\ell_m}$; 也可在数据末尾补 0 使长度变为 $NN = 2^\ell$, 再进行 FFT。该方法对频率 $f_j = \frac{j}{NN}$ ($j = 0, \dots, \frac{NN}{2}$) 而言, 会得到与 (3.13) 计算的样本谱相同的数值。但需注意: (3.13) 可以计算任意频率 f 处的傅里叶变换, 而 FFT 在数据量为 N 时, 只能在 $f_k = \frac{k}{N}$ 这些频率点上计算。因此当 $N \neq 2^\ell$ 时, FFT 与 3.2 节直接计算周期图所使用的频率点

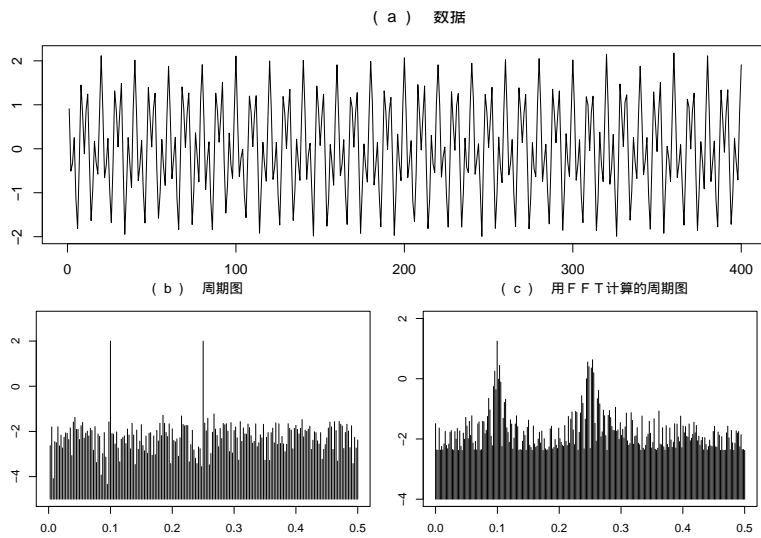


图 3.9 具有线状谱的数据、按 (3.8) 计算的周期图，以及补上 112 个 0 后用 FFT 得到的周期图。

不同。当真实谱连续时，这种差异几乎不是问题；但当真实谱不连续、具有线状谱时，可能得到差异很大的形状，需要注意。

```

> r <- as.ts(rnorm(3200))
> # simulation of 2 cosine function + noise
> t <- 1:400
> r <- rnorm(400)
> y <- rep(0,400)
> for (i in t) {
> y[i] <- cos(2*pi*i/10) + cos(2*pi*i/4) + r[i]*0.1
> }
> y <- as.ts(y)
> plot(y,main="(a) Generated data")
> period(y>window=0)
> fftper(y>window=0)

```

图 3.9(a) 给出了具有两条线状谱的模型

$$y_n = \cos \frac{2\pi n}{10} + \cos \frac{2\pi n}{4} + w_n \quad (3.18)$$

的实现值，其中 $N = 400$ ， $w_n \sim N(0, 0.01)$ 。由数据 y_1, \dots, y_{400} 得到的周期图如图 3.9 下方左图所示。对应两项余弦函数，在周频率 $f = 0.05$ 与 $f = 0.125$ 处可见两条线状谱；其余频率处则围绕白噪声对应的常数值上下波动。右图则是在 y_1, \dots, y_{400} 后补 112 个 0 构造 $NN = 512 = 2^9$ 的数据后，用 FFT 得到的周期图。由于计算频率点与线状谱位置不一致，因此与左图相比差异很大。

```

> data( HAKUSAN )
> period( HAKUSAN[,1], window=0 )
> fftper( HAKUSAN[,1], window=0 )

```

图 3.10 (a) 为船舶偏航角速度时间序列，(b) 为按定义直接计算的周期图，(c) 为

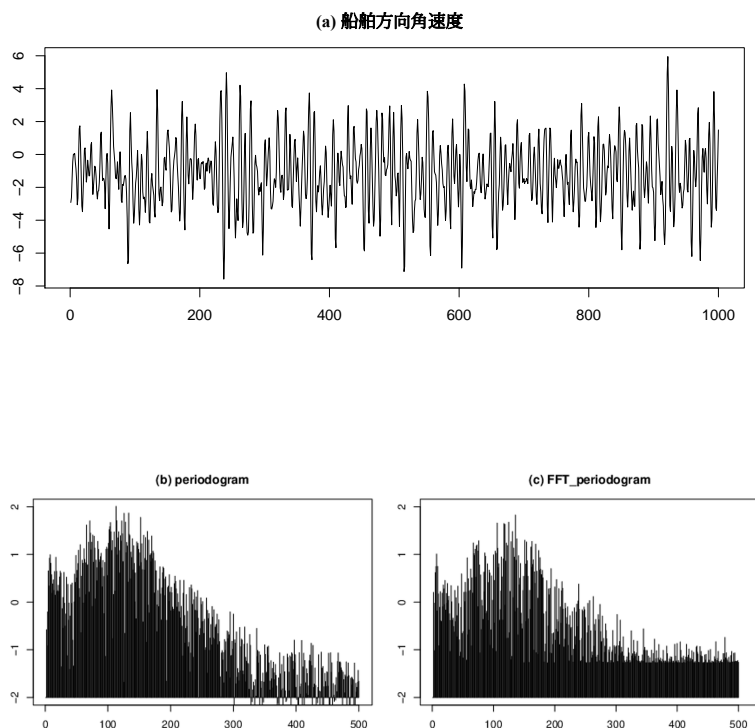


图 3.10 船舶偏航角速度数据、按定义计算的周期图，以及用 FFT 计算的周期图。

在 (a) 数据末尾补 24 个 0 后用 FFT 计算得到的周期图。像本例这样真实谱连续且不具有线状谱时，即便补 0 后用 FFT 计算周期图，与按定义直接计算得到的结果在主要部分也几乎没有差别，能得到相近的估计值。

● 章末习题

1. 验证功率谱能够像 (3.2) 式那样用余弦表示。
2. 利用第 2 章习题 4 的结果, 令 v_n 为方差为 1 的白噪声, 求时间序列 $y_n = v_n - cv_{n-1}$ 的谱。
3. 当自协方差函数为 $C_k = \sigma^2(1-a^2)^{-1}a^{|k|}$ 时, 验证谱可由 (3.5) 式得到。
4. 证明由 (3.17) 可得到 (3.8) 的周期图。
5. 关于样本谱 $\hat{p}(f)$, 证明渐近不偏性 $\lim_{n \rightarrow \infty} E[\hat{p}(f)] = p(f)$ 成立。

第 4 章

建模

在时间序列分析等统计分析中, 包含随机性的现象可以看作是服从某一概率分布的随机变量的实现值。统计模型和时间序列模型, 都是基于数据对该概率分布进行估计而得到的。本章将引入用于评价真实概率分布与模型概率分布之间接近程度的标准, 并在该标准基础上, 说明参数估计、模型选择等统计模型构建的统一方法。

4.1 概率分布与统计模型

当 Y 为**随机变量** (random variable) 时, 对于所有实数 $y \in R$, 都可以定义事件 $Y \leq y$ 发生的概率 $\text{Prob}(Y \leq y)$ 。将其视为 y 的函数, 记为

$$G(y) = \text{Prob}(Y \leq y) \quad (4.1)$$

此时, $G(y)$ 被称为随机变量 Y 的**概率分布函数** (probability distribution function)。

在时间序列分析中使用的随机变量通常是所谓的**连续型**, 其分布函数可以表示为满足 $g(t) \geq 0$ ($-\infty < t < \infty$) 的函数的积分形式:

$$G(y) = \int_{-\infty}^y g(t) dt \quad (4.2)$$

此时, $g(x)$ 被称为**概率密度函数** (density, density function)。

反过来, 若给定分布函数或密度函数, 则对于任意 $a < b$, 事件 $a < Y \leq b$ 发生的概率可以由

$$G(b) - G(a) = \int_a^b g(x) dx \quad (4.3)$$

计算得到。

在统计分析中, 为了刻画数据的表现形式, 会使用各种概率分布模型。下面给出一些具有代表性的概率密度函数。

(a) **正态分布** (高斯分布, normal distribution)

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad -\infty < x < \infty \quad (4.4)$$

该分布的均值为 μ , 方差为 σ^2 , 记为 $N(\mu, \sigma^2)$ 。特别地, $N(0, 1)$ 被称为标准正态分布。

(b) **柯西分布** (Cauchy distribution)

$$g(x) = \frac{\tau}{\pi\{(x-\mu)^2 + \tau^2\}}, \quad -\infty < x < \infty \quad (4.5)$$

μ 和 τ^2 分别称为位置参数和尺度参数, τ^2 的平方根称为尺度参数。

(c) **皮尔逊分布族** (Type VII Pearson family of distributions)

$$g(x) = \frac{C}{\{(x-\mu)^2 + \tau^2\}^b}, \quad -\infty < x < \infty \quad (4.6)$$

其中 $C = \tau^{2b-1}\Gamma(b)/(\Gamma(b-\frac{1}{2})\Gamma(\frac{1}{2}))$ 。 b 被称为形状参数, 当 $b=1$ 时, 该分布与柯西分布一致。另外, 令 k 为正整数, 当 $b=(k+1)/2$ 时, 该分布称为自由度为 k 的 t 分布 (t -distribution)。

(d) **指数分布** (exponential distribution)

$$g(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \text{ 时} \\ 0, & x < 0 \text{ 时} \end{cases} \quad (4.7)$$

指数分布的均值为 λ^{-1} , 方差为 λ^{-2} 。

(e) **χ^2 分布** (chi-square distribution)

$$g(x) = \begin{cases} \frac{1}{2^{k/2}\Gamma(\frac{k}{2})} e^{-\frac{x}{2}} x^{\frac{k}{2}-1}, & x \geq 0 \text{ 时} \\ 0, & x < 0 \text{ 时} \end{cases} \quad (4.8)$$

k 被称为自由度。特别地, 当 $k=2$ 时, 该分布退化为指数分布。 χ^2 分布的均值与方差分别为 k 和 $2k$ 。 k 个正态随机变量平方和服从自由度为 k 的 χ^2 分布。

(f) **双指数分布** (double exponential distribution)

$$g(x) = e^{x-e^x} \quad (4.9)$$

双指数分布的均值与方差分别为 $-\gamma$ 和 $\pi^2/6$, 其中 γ 为欧拉常数 (0.577224)。指数分布取对数后服从双指数分布。

(g) **均匀分布** (uniform distribution)

$$g(x) = \begin{cases} (b-a)^{-1}, & a < x \leq b \text{ 时} \\ 0, & \text{其他情况} \end{cases} \quad (4.10)$$

均匀分布的均值为 $(a+b)/2$, 方差为 $(b-a)^2/12$ 。

例 密度函数的表示

上述概率密度函数可以使用 TSSS 包中的函数 `pdfunc` 进行绘制。其中, 参数 `xmin` 和 `xmax` 分别用于指定密度函数定义域的下限和上限。

```
> par(mar=c(2,2,3,1)+0.1)
> # 正态分布
> pdfunc(model = "norm", xmin = -4, xmax = 4)
> # 柯西分布
> pdfunc(model = "Cauchy", xmin = -4, xmax = 4)
> # 皮尔逊分布
> pdfunc(model = "Pearson", shape = 2, xmin = -4, xmax = 4)
> # 指数分布
> pdfunc(model = "exp", xmin = 0, xmax = 8)
> #  $\chi^2$  分布
> pdfunc(model = "Chi2", df = 3, xmin = 0, xmax = 8)
> # 双指数分布
> pdfunc(model = "dexp", xmin = -4, xmax = 4)
```

图 4.1 给出了 (a)~(f) 所示的概率密度函数。

利用第 16 章中介绍的模拟方法, 可以按照密度函数高度的比例生成取不同数值的数据 y_1, \dots, y_N 。这类数据称为随机变量的**实现值** (realization)。

图 4.2 给出了 (a)~(c) 以及 (f) 各分布在 $N = 20$ 时的实现值示例。

如上所述, 给定随机变量即可生成服从该分布的数据; 而在统计分析中, 情况正好相反, 即当观测到数据 y_1, \dots, y_N 时, 将其视为某一随机变量 Y 的实现值。也就是说, 在我们实际观测到的数据背后, 假定存在一个随机变量, 而数据正是该随机变量的实现结果。

此时, 用于刻画该随机变量的概率密度函数 $g(y)$ 称为**真模型** (true model)。

通常, 该真模型是未知的, 因此需要根据给定数据来推断生成这些数据的概率分布。也就是说, 当给定如图 4.2 所示的一组数据时, 需要反向估计如图 4.1 所示的密度函数。此时, 由数据估计得到的密度函数称为**统计模型** (statistical model), 记为 $f(y)$ 。

在一般的统计分析中, 只需为每一个观测值指定相应的概率分布即可; 但对于时间序列数据, 如第 2 章所示, 必须考虑其联合分布 $f(y_1, \dots, y_N)$ 。

在第 2 章中, 我们使用样本均值 $\hat{\mu}$ 以及样本自协方差函数 \hat{C}_k 来刻画时间序列 y_1, \dots, y_N 。这等价于假定 N 维向量 $y = (y_1, \dots, y_N)^T$ 服从均值向量 $\hat{\mu} = (\hat{\mu}_1, \dots, \hat{\mu}_N)^T$ 以

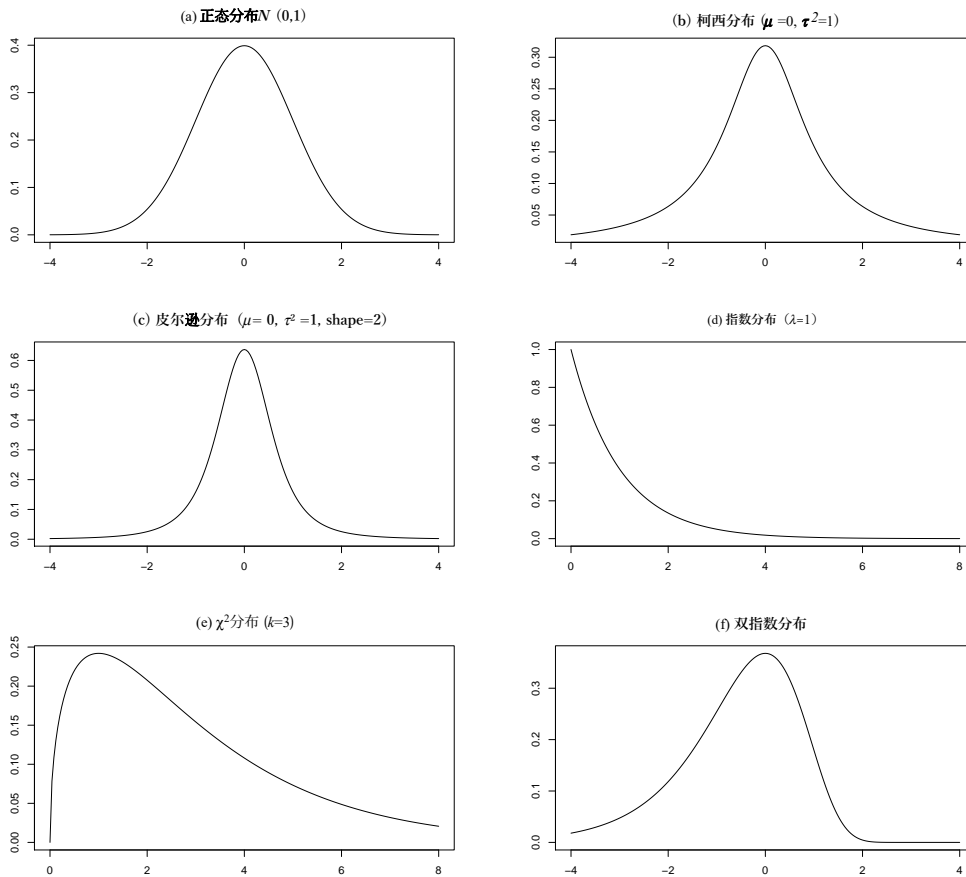


图 4.1 各种分布的概率密度函数

及协方差矩阵

$$\hat{C} = \begin{bmatrix} \hat{C}_0 & \hat{C}_1 & \cdots & \hat{C}_{N-1} \\ \hat{C}_1 & \hat{C}_0 & \cdots & \hat{C}_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{N-1} & \hat{C}_{N-2} & \cdots & \hat{C}_0 \end{bmatrix} \quad (4.11)$$

的多元正态分布。

这种模型能够灵活地刻画服从正态分布的平稳时间序列，但相对于数据量 N ，需要估计 $N+1$ 个未知参数 $\hat{\mu}, \hat{C}_0, \dots, \hat{C}_{N-1}$ ，并不能有效地压缩数据所包含的信息。

相比之下，下一章及之后介绍的平稳时间序列模型，可以使用少量参数来表示式 (4.11) 中的协方差矩阵结构。

4.2 K-L 信息量与熵最大化原理

设生成现实数据的真实模型为 $g(y)$ ，用来近似该真实模型的统计模型为 $f(y)$ 。在统计建模中，寻找一个尽可能接近 $g(y)$ 的 $f(y)$ 是最主要的目标，而为此需要一个

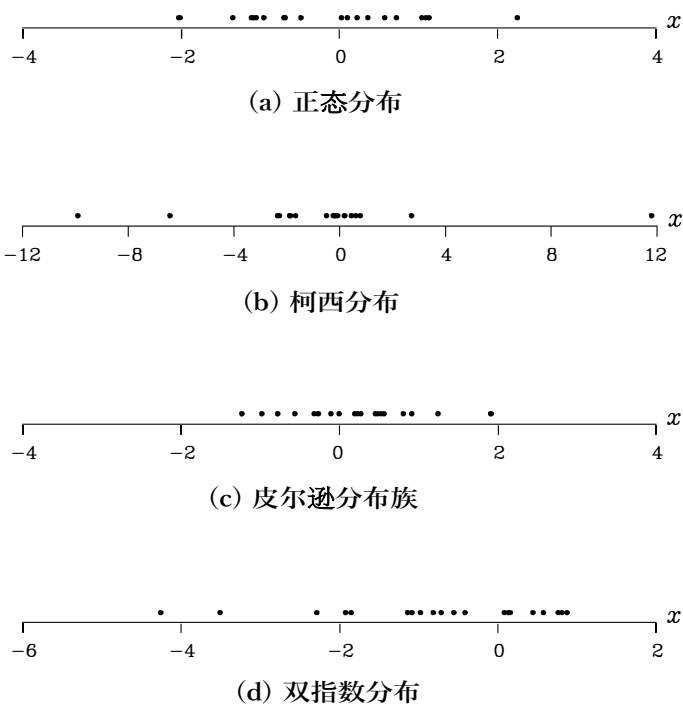


图 4.2 各种分布的实现值

能够客观评价模型 $f(y)$ 优劣的准则。这里采用的评价准则是 **Kullback - Leibler 信息量** (Kullback - Leibler information, 以下称为 K-L 信息量)

$$I(g; f) = \mathbb{E}_Y \left[\log \left\{ \frac{g(Y)}{f(Y)} \right\} \right] = \int_{-\infty}^{\infty} \log \left\{ \frac{g(y)}{f(y)} \right\} g(y) dy \quad (4.12)$$

其中, 第二个等式对应于模型为连续型概率分布时的表示形式。该 K-L 信息量具有如下性质:

$$\begin{aligned} \text{(i)} \quad & I(g; f) \geq 0 \\ \text{(ii)} \quad & I(g; f) = 0 \iff g(y) = f(y) \end{aligned} \quad (4.13)$$

此外, 将 K-L 信息量取相反数得到的量 $B(g; f) = -I(g; f)$ 也被称为广义的**熵** (entropy)。当从假定的分布 $f(y)$ 中取出 n 个实现值时, 其相对频率分布与真实分布 $g(y)$ 相一致的概率可以近似表示为 $\frac{1}{n}$ 。因此, K-L 信息量越小, 概率分布 f 就越接近 g 。

统计模型是基于数据 y_1, \dots, y_N 对真实分布 $g(y)$ 进行近似得到的, 其优劣可以通过 K-L 信息量 $I(g; f)$ 进行评价。在统计建模中, 通过最大化 $B(g; f) = -I(g; f)$ 来构建模型的思想, 称为**熵最大化原理** (entropy maximization principle)。

例: 正态分布模型的 K-L 信息量 设真实模型 $g(y)$ 及其近似模型 $f(y)$ 均为正

态分布:

$$\begin{aligned} g(y|\mu, \sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\} \\ f(y|\xi, \tau^2) &= \frac{1}{\sqrt{2\pi\tau^2}} \exp\left\{-\frac{(y-\xi)^2}{2\tau^2}\right\} \end{aligned} \quad (4.14)$$

此时,

$$\log\left\{\frac{g(y)}{f(y)}\right\} = \frac{1}{2}\left\{\log\frac{\tau^2}{\sigma^2} - \frac{(y-\mu)^2}{\sigma^2} + \frac{(y-\xi)^2}{\tau^2}\right\} \quad (4.15)$$

因此, K-L 信息量为

$$\begin{aligned} I(g;f) &= E_Y\left[\log\frac{g(Y)}{f(Y)}\right] = \frac{1}{2}\left\{\log\frac{\tau^2}{\sigma^2} - \frac{E_Y[(Y-\mu)^2]}{\sigma^2} + \frac{E_Y[(Y-\xi)^2]}{\tau^2}\right\} \\ &= \frac{1}{2}\left\{\log\frac{\tau^2}{\sigma^2} - 1 + \frac{\sigma^2 + (\mu-\xi)^2}{\tau^2}\right\} \end{aligned} \quad (4.16)$$

例如, 当 g 为标准正态分布 $N(0,1)$, f 为 $N(0.1,1.5)$ 时, 有 $I(g;f) = (\log 1.5 - 1 + 1.01/1.5)/2 = 0.03940$ 。

如上所述, 当 g 与 f 均服从正态分布时, K-L 信息量可以较为容易地计算得到; 然而, 对于一般分布 g 与 f 的组合, $I(g;f)$ 并不总是能够以解析形式求得。因此, 在一般情况下, K-L 信息量需要通过数值计算的方法来求取。

TSSS 软件包中的函数 `klinfo` 用于计算相对于真实分布 $g(x)$ 的模型分布 $f(x)$ 的 K-L 信息量。其中, 当 `distg=1` 时, 真实分布 $g(x)$ 为正态分布, 当 `distg=2` 时为柯西分布; 相应地, 当 `distf=1` 时, 模型分布 $f(x)$ 为正态分布, 当 `distf=2` 时为柯西分布。各分布的参数分别通过 `paramf` 与 `paramg` 进行指定。在数值积分中, 若给定积分区间上限 `xmax`, 则积分区间的下限将自动设置为 `-xmax`。

```
> # g: 正态分布, f: 正态分布
> klinfo(distg = 1, paramg = c(0, 1), distf = 1, paramf = c(0.1, 1.5), xmax = 4)
> klinfo(distg = 1, paramg = c(0, 1), distf = 1, paramf = c(0.1, 1.5), xmax = 6)
> klinfo(distg = 1, paramg = c(0, 1), distf = 1, paramf = c(0.1, 1.5), xmax = 8)
```

表 4.1 为了检验数值计算的精度, 针对两个密度函数 $g(y)$ 与 $f(y)$, 给出了在区间 $[x_0, x_k]$ 上利用梯形公式

$$\hat{I}(g;f) = \frac{\Delta x}{2} \sum_{i=1}^k \{h(x_i) + h(x_{i-1})\} \quad (4.17)$$

计算得到的 K-L 信息量结果。其中, $x_0 = -x_k$, 且

$$x_i = x_0 + (x_k - x_0) \frac{i}{k} \quad (4.18)$$

$$h(x) = g(x) \log \frac{g(x)}{f(x)} \quad (4.19)$$

表 4.1 K-L 信息量 (g : 正态分布, f : 正态分布)

x_k	k	Δx	$\hat{I}(g;f)$	$\hat{G}(x_k)$
4.0	8	1.000	0.03974041	0.99986319
4.0	16	0.500	0.03962097	0.99991550
4.0	32	0.250	0.03958692	0.99993116
4.0	64	0.125	0.03957812	0.99993527
6.0	12	1.000	0.03939929	1.00000000
6.0	24	0.500	0.03939924	1.00000000
6.0	48	0.250	0.03939924	1.00000000
6.0	96	0.125	0.03939923	1.00000000
8.0	16	1.000	0.03939926	1.00000000
8.0	32	0.500	0.03939922	1.00000000
8.0	64	0.250	0.03939922	1.00000000
8.0	128	0.125	0.03939922	1.00000000

成立。

表 4.1 中给出了 x_0 取 4、6、8 三种情况, 以及 k 取 8、16、32、64 四种情况时的 $\hat{I}(g;f)$ 与 $\hat{G}(x_k)$ 的数值结果。

由该表可以看出, 只要取较大的 x_0 , 即使在 k 取值较小、 $\Delta x = 0.5$ 这样较为粗糙的积分条件下, 也能够得到令人惊讶的良好近似结果。这是因为在此假定 g 为正态分布, 其密度函数在远离原点时会以极快的速度收敛到 0。表中最右一列给出了将密度函数 $g(y)$ 在区间 $-x_k$ 到 x_k 上进行数值积分所得的结果。当采用收敛速度不如正态分布快的密度函数作为 $g(y)$ 时, 可以通过检查该积分值是否接近 1, 来在一定程度上判断数值计算的精度。

```
> # g: 正态分布, f: 柯西分布
> klinfo(distg = 1, paramg = c(0, 1), distf = 2, paramf = c(0, 1), xmax = 8)
```

另一方面, 表 4.2 给出了当 g 取标准正态分布, 而 f 取标准柯西分布 ($\mu = 0$, $\tau^2 = 1$) 时的计算结果。

4.3 K-L 信息量的估计与对数似然

在上一节中, 我们将 K-L 信息量作为统计模型的评价标准加以介绍, 但除去蒙特卡洛实验等少数特殊情况外, K-L 信息量几乎不会被直接用于实际的统计模型评价。这是因为在进行统计分析时, 真实分布通常是未知的, 从而无法直接计算

表 4.2 K-L 信息量 (g : 正态分布, f : 柯西分布)

x_k	k	Δx	$\hat{I}(g, f)$	$\hat{G}(x_k)$
8.0	16	1.000	0.25620181	1.00000001
8.0	32	0.500	0.25924202	1.00000000
8.0	64	0.250	0.25924453	1.00000000
8.0	128	0.125	0.25924453	1.00000000

K-L 信息量。在现实情况下, 我们能够得到的只是数据 y_1, \dots, y_N , 而非真实分布 $g(y)$ 。下面假设数据 y_1, \dots, y_N 是从 $g(y)$ 中独立观测得到的, 在此基础上, 我们来考虑如何估计模型 $f(y)$ 的 K-L 信息量。

根据熵最大化原理, 为了得到最优模型, 只需寻找使 $B(g; f)$ 达到最大, 即使 $I(g; f)$ 最小的模型即可。首先, K-L 信息量可以分解为

$$I(g; f) = \mathbb{E}_Y[\log g(Y)] - \mathbb{E}_Y[\log f(Y)] \quad (4.20)$$

两个部分。其中, 右侧第一项只有在真实分布 $g(y)$ 已知时才能计算, 但该项与模型 $f(y)$ 无关, 因此可以视为常数并忽略不计。于是, 只需最大化右侧第二项, 即可得到更优的模型。

右侧第二项称为**平均对数似然** (expected log-likelihood)。对于具有密度函数的连续型模型, 该量可以表示为以下形式:

$$\mathbb{E}_Y[\log f(Y)] = \int \log f(y) g(y) dy \quad (4.21)$$

因此, 当真实模型 $g(y)$ 未知时, 该平均对数似然同样无法直接计算。然而, 由于数据 y_n 是按照密度函数 $g(y)$ 生成的, 根据**大数定律**, 当数据个数 $N \rightarrow \infty$ 时,

$$\frac{1}{N} \sum_{n=1}^N \log f(y_n) \rightarrow \mathbb{E}_Y[\log f(Y)] \quad (4.22)$$

成立。

因此, 用对数似然 $\sum_{n=1}^N \log f(y_n)$ 来代替原本的评价标准 K-L 信息量 $I(g; f)$, 并选择使其最大的模型, 即可在近似意义下实现熵最大化。

式 (4.22) 左边乘以 N 后, 在观测值相互独立的情况下, 得到的量称为**对数似然** (log-likelihood), 其表达式为

$$\ell = \sum_{n=1}^N \log f(y_n) \quad (4.23)$$

此外, 对其取指数可得

$$L = \prod_{n=1}^N f(y_n) \quad (4.24)$$

该量称为**似然性** (likelihood)。

在时间序列分析模型中, 通常不满足观测值相互独立的假设。在这种一般情况下, 似然函数应当使用 y_1, \dots, y_N 的联合分布来定义, 即

$$L = f(y_1, \dots, y_N) \quad (4.25)$$

当在 (4.25) 中假定独立性时, 便可得到 (4.24), 因此 (4.25) 是 (4.24) 的自然推广形式。在这种情形下, 对数似然可以写为

$$\ell = \log L = \log f(y_1, \dots, y_N) \quad (4.26)$$

4.4 基于最大似然法的参数估计

当模型是以参数 θ 为参数的参数化模型, 并且具有 $f(y) = f(y|\theta)$ 这种形式时, 对数似然 ℓ 可以视为参数 θ 的函数。因此, 将 θ 显式写出,

$$\ell(\theta) = \begin{cases} \sum_{n=1}^N \log f(y_n|\theta), & \text{独立情况下} \\ \log f(y_1, \dots, y_N|\theta), & \text{一般情况下} \end{cases} \quad (4.27)$$

称为关于 θ 的**对数似然函数**。

由于对数似然函数 $\ell(\theta)$ 是用于评价由 θ 所确定模型优劣的量, 因此, 通过选择使 $\ell(\theta)$ 达到最大的 θ , 即可确定参数化模型 $f(y|\theta)$ 的最优参数值。这种通过最大化对数似然或似然来估计参数的方法, 称为**最大似然法** (maximum likelihood method)。此外, 由最大似然法得到的参数估计值记为 $\hat{\theta}$, 称为**最大似然估计值** (maximum likelihood estimate)。

例 考虑均值为 μ , 方差为 1 的正态分布模型

$$f(y|\mu) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(y-\mu)^2}{2}\right\} \quad (4.28)$$

下面用最大似然法来估计参数 μ 。在这种情况下, 对数似然函数为

$$\ell(\mu) = -\frac{N}{2} \log 2\pi - \frac{1}{2} \sum_{n=1}^N (y_n - \mu)^2 \quad (4.29)$$

因此, 为了使 $\ell(\mu)$ 达到最大, 只需寻找使

$$S(\mu) = \sum_{n=1}^N (y_n - \mu)^2 \quad (4.30)$$

达到最小的 μ 。于是, 令 $S(\mu)$ 的一阶导数为 0, 可得

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^N y_n \quad (4.31)$$

式(4.31)这种通过最小化平方和来估计参数的方法,称为**最小二乘法**(least squares method)。关于最小二乘法的一般求解方法,将在第5章中进行说明。在假定正态分布的模型中,参数的最大似然估计值往往与最小二乘估计值一致,并且在很多情况下可以解析求解。

然而,时间序列模型的似然函数或对数似然函数一般具有较为复杂的形式,能够以解析方式求得最大似然估计或其近似解的情形,仅限于AR模型(第7章)以及多项式趋势模型(第11章)等少数例外。一般而言,为了求取时间序列模型参数 θ 的最大似然估计,通常需要采用基于拟牛顿法的**数值最优化**(numerical optimization)算法(见附录A)。在该方法中,当给定参数 θ 的初始值 θ_0 ,以及在某一 θ 取值下的对数似然值 $\ell(\theta)$ 和一阶导数 $\frac{\partial \ell}{\partial \theta}$ 时,通过反复迭代

$$\theta_k = \theta_{k-1} + \lambda_k H_{k-1} \frac{\partial \ell}{\partial \theta} \quad (4.32)$$

即可自动求得 $\ell(\theta)$ 的极大点。其中,步长 λ_k 以及海森矩阵的逆矩阵 H_{k-1} 由算法自动确定。

例 10个数据

-1.10	-0.40	-0.20	-0.02	0.02
0.71	1.35	1.46	1.74	3.89

假设给定了上述数据。此时,柯西分布模型

$$f(y|\mu, \tau^2) = \frac{1}{\pi} \frac{\tau}{(y - \mu)^2 + \tau^2} \quad (4.33)$$

的对数似然可以表示为

$$\ell(\mu, \tau^2) = 5 \log \tau^2 - 10 \log \pi - \sum_{n=1}^{10} \log \{(y_n - \mu)^2 + \tau^2\} \quad (4.34)$$

此外,关于参数 $\theta = (\mu, \tau^2)^T$ 的一阶导数如下式所示。

$$\frac{\partial \ell}{\partial \mu} = 2 \sum_{n=1}^{10} \frac{y_n - \mu}{(y_n - \mu)^2 + \tau^2} \quad (4.35)$$

$$\frac{\partial \ell}{\partial \tau^2} = \frac{5}{\tau^2} - \sum_{n=1}^{10} \frac{1}{(y_n - \mu)^2 + \tau^2} \quad (4.36)$$

表4.3总结了以 $\theta_0 = (0, 1)^T$ 作为初始值,通过数值方法求取柯西分布参数最大似然估计值的计算过程。可以看到, $\frac{\partial \ell}{\partial \mu}$ 和 $\frac{\partial \ell}{\partial \tau^2}$ 的绝对值迅速减小,在5次迭代后便得到了最大似然估计值。

表 4.3 基于拟牛顿法的柯西分布参数估计

k	μ	τ^2	对数似然	$\partial\ell/\partial\mu$	$\partial\ell/\partial\tau^2$
0	0.00000	1.00000	-19.1901	2.10968	-0.92404
1	0.38588	0.83098	-18.7140	-0.21335	-0.48655
2	0.34795	0.62966	-18.6536	-0.35810	0.06627
3	0.26819	0.60826	-18.6396	0.00320	-0.01210
4	0.26752	0.60521	-18.6395	0.00000	-0.00002
5	0.26752	0.60520	-18.6395	0.00000	0.00000

表 4.4 利用数值微分的拟牛顿法进行的柯西分布参数估计

k	μ	τ^2	对数似然	$\partial\ell/\partial\mu$	$\partial\ell/\partial\tau^2$
0	0.00000	1.00000	-19.1901	2.10967	-0.92404
1	0.38588	0.83098	-18.7140	-0.21335	-0.48655
2	0.34795	0.62966	-18.6536	-0.35810	0.06627
3	0.26819	0.60826	-18.6396	0.00320	-0.01210
4	0.26752	0.60521	-18.6395	0.00000	-0.00000

如本例所示, 当对数似然及其一阶导数可以解析地求得时, 只需给出这些值, 即可求得最大似然估计值。然而, 许多时间序列模型的对数似然函数形式极其复杂, 或者需要通过卡尔曼滤波等递推公式来计算, 因此往往难以构造用于求取对数似然一阶导数的算法。在这种情况下, 可以通过数值微分来近似计算一阶导数, 从而仅利用对数似然的数值来求取最大似然估计值。

表 4.4 给出了采用这种方法、仅利用对数似然进行优化所得到的结果。即使在这种情形下, 所得结果也与表 4.2 几乎完全一致, 并且迭代过程反而提前一次终止。

4.5 AIC (赤池信息量准则)

由于对数似然是平均对数似然的自然估计量, 因此在参数化模型参数估计中, 可以采用最大化对数似然的最大似然法。同样地, 当存在多个参数化模型时, 可以先分别利用最大似然法估计各模型的参数, 再根据对应的对数似然值, 即**最大对数似然** (maximum log-likelihood) $\ell(\hat{\theta})$ 来对模型进行比较。

然而, 实际上最大对数似然并不能直接用于不同模型之间的比较。其原因在于, 对于由最大似然估计值 $\hat{\theta}$ 所确定的模型, 量 $N^{-1}\ell(\hat{\theta})$ 作为 $E_Y[\log f(Y|\hat{\theta})]$ 的估计量具有正偏差 (见图 4.3)。这种偏差源于在参数估计与模型评价过程中, 对同一组数

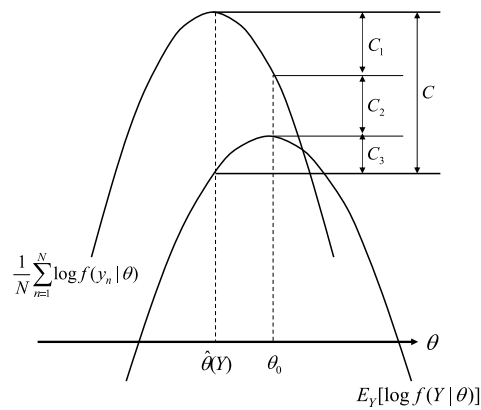


图 4.3 平均对数似然与对数似然的差异

据进行了两次使用。

当用 $N^{-1}\ell(\hat{\theta}) \equiv N^{-1}\sum_{n=1}^N \log f(y_n|\hat{\theta})$ 来估计 $E_Y[\log f(Y|\hat{\theta})]$ 时, 由此产生的平均偏差可定义为:

$$C \equiv E_X \left[E_Y[\log f(Y|\hat{\theta})] - N^{-1} \sum_{n=1}^N \log f(y_n|\hat{\theta}) \right] \quad (4.37)$$

此时, 若对 $N^{-1}\ell(\hat{\theta})$ 进行 C 的修正, 即考虑 $N^{-1}\ell(\hat{\theta}) + C$, 即可得到 $E_Y[\log f(Y|\hat{\theta})]$ 的无偏估计量。这里可以近似地得到 $C = -N^{-1}k$, 由此便可导出**赤池信息量准则** (AIC: Akaike Information Criterion):

$$\begin{aligned} \text{AIC} &= -2\ell(\hat{\theta}) + 2k \\ &= -2(\text{最大对数似然}) + 2(\text{参数个数}) \end{aligned} \quad (4.38)$$

在本书中, 将采用 AIC 作为模型选择的评价标准。下面将说明 AIC 的推导过程; 对于对模型选择准则本身不感兴趣的读者, 可以跳过以下内容。

设真实分布为 $g(y)$, 模型分布为 $f(y|\theta)$, 并将基于数据 $X = (x_1, \dots, x_N)$ 得到的参数 θ 的最大似然估计值记为 $\hat{\theta} \equiv \hat{\theta}(X)$ 。令 θ_0 为使平均对数似然 $E_Y[\log f(Y|\theta)]$ 达到最大的参数, 并称其为真实参数。此时有

$$\frac{\partial}{\partial \theta} E_Y[\log f(Y|\theta_0)] = 0$$

成立。另一方面, 由于 $\hat{\theta}$ 是使对数似然函数 $\ell(\theta) = \sum_{n=1}^N \log f(x_n|\theta)$ 达到最大的参数,

因此有下式成立:

$$\frac{\partial}{\partial \theta} \sum_{n=1}^N \log f(x_n | \hat{\theta}) = 0$$

这里将式 4.37 分解为如下三个部分 (见图 4.3):

$$\begin{aligned} C &= E_X \{ E_Y [\log f(Y | \hat{\theta})] - E_Y [\log f(Y | \theta_0)] \} \\ &+ E_X \left\{ E_Y [\log f(Y | \theta_0)] - N^{-1} \sum_{n=1}^N \log f(x_n | \theta_0) \right\} \\ &+ E_X \left\{ N^{-1} \sum_{n=1}^N \log f(x_n | \theta_0) - N^{-1} \sum_{n=1}^N \log f(x_n | \hat{\theta}) \right\} \\ &\equiv C_1 + C_2 + C_3 \end{aligned} \quad (4.39)$$

对 C_1 的评估

将由最大似然估计值确定的模型的平均对数似然 $E_Y [\log f(Y | \hat{\theta})]$ 在 θ_0 附近作二阶泰勒展开, 并交换求导与取期望的顺序, 得到

$$\begin{aligned} E_Y [\log f(Y | \hat{\theta})] &\approx E_Y [\log f(Y | \theta_0)] + \left\{ \frac{\partial}{\partial \theta} E_Y [\log f(Y | \theta_0)] \right\} (\hat{\theta} - \theta_0) \\ &+ \frac{1}{2} (\hat{\theta} - \theta_0)^T \left\{ \frac{\partial^2}{\partial \theta \partial \theta'} E_Y [\log f(Y | \theta_0)] \right\} (\hat{\theta} - \theta_0) \\ &= E_Y [\log f(Y | \theta_0)] - \frac{1}{2} (\hat{\theta} - \theta_0)^T J (\hat{\theta} - \theta_0) \end{aligned}$$

其中

$$J \equiv -E_Y \left\{ \frac{\partial^2}{\partial \theta \partial \theta'} \log f(Y | \theta_0) \right\}$$

成立。此外, $\hat{\theta}$ 是 θ_0 的最大似然估计量, 根据中心极限定理, 令 I 为费舍尔信息矩阵 (Fisher information matrix)

$$I \equiv E_Y \left[\left\{ \frac{\partial}{\partial \theta} \log f(Y | \theta_0) \right\} \left\{ \frac{\partial}{\partial \theta} \log f(Y | \theta_0) \right\}^T \right]$$

则 $\sqrt{N}(\hat{\theta} - \theta_0)$ 渐近服从均值为 0、协方差矩阵为 $J^{-1}IJ^{-1}$ 的正态分布。因此, 对 X 取期望可得

$$E_X [(\hat{\theta} - \theta_0)^T J (\hat{\theta} - \theta_0)] = \frac{1}{N} \text{trace}\{IJ^{-1}\} \approx \frac{k}{N} \quad (4.40)$$

其中 k 为矩阵 I 的维数。特别地, 当 $g(y) = f(y | \theta_0)$ 成立时, 有 $J = I$, 上述近似关系完全成立。

由此得到以下结果:

$$C_1 \equiv E_X \{ E_Y [\log f(Y | \hat{\theta})] - E_Y [\log f(Y | \theta_0)] \} \approx -\frac{k}{2N} \quad (4.41)$$

对 C_3 的评估

将 $N^{-1} \sum_{n=1}^N \log f(x_n | \theta_0)$ 在 $\hat{\theta}$ 附近作泰勒展开, 得到

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \log f(x_n | \theta_0) &\approx \frac{1}{N} \sum_{n=1}^N \log f(x_n | \hat{\theta}) + \frac{1}{N} \sum_{n=1}^N \frac{\partial}{\partial \theta} \log f(x_n | \hat{\theta}) (\theta_0 - \hat{\theta}) \\ &\quad + \frac{1}{2} (\theta_0 - \hat{\theta})^T \left\{ \frac{1}{N} \sum_{n=1}^N \frac{\partial^2}{\partial \theta \partial \theta'} \log f(x_n | \hat{\theta}) \right\} (\theta_0 - \hat{\theta}) \end{aligned}$$

这里, 由于 $\hat{\theta}$ 是最大似然估计值, 理解右边第二项为 0。此外, 根据大数定律, 当 $N \rightarrow \infty$ 时,

$$\frac{1}{N} \sum_{n=1}^N \frac{\partial^2}{\partial \theta \partial \theta'} \log f(x_n | \hat{\theta}) \rightarrow E_Y \left\{ \frac{\partial^2}{\partial \theta \partial \theta'} \log f(Y | \theta_0) \right\} = -J \quad (4.42)$$

成立, 因此有

$$\frac{1}{N} \sum_{n=1}^N \log f(x_n | \theta_0) \approx \frac{1}{N} \sum_{n=1}^N \log f(x_n | \hat{\theta}) - \frac{1}{2} (\theta_0 - \hat{\theta})^T J (\theta_0 - \hat{\theta})$$

于是, 对 X 取期望, 与式 (4.41) 类似, 可得

$$C_3 = E_X \left\{ \frac{1}{N} \sum_{n=1}^N \log f(x_n | \theta_0) - \frac{1}{N} \sum_{n=1}^N \log f(x_n | \hat{\theta}) \right\} \approx -\frac{k}{2N} \quad (4.43)$$

对 C_2 的评估

对于固定的 θ_0 , $\log f(x_n | \theta_0)$ 的期望值即为平均对数似然, 因此下式成立。

$$C_2 = E_X \left\{ E_Y [\log f(Y | \theta_0)] - \frac{1}{N} \sum_{n=1}^N \log f(x_n | \theta_0) \right\} = 0 \quad (4.44)$$

对 C 的评估与 AIC

将以上 (4.41), (4.43), (4.44) 三式相加, 得到

$$C = E_X \left\{ E_Y [\log f(Y | \hat{\theta})] - \frac{1}{N} \sum_{n=1}^N \log f(x_n | \hat{\theta}) \right\} \approx -\frac{k}{N} \quad (4.45)$$

由此可知, $N^{-1} \ell(\hat{\theta})$ 在平均意义下比 $E_Y [\log f(Y | \hat{\theta})]$ 大 $\frac{k}{N}$ 。

因此,

$$N^{-1} \ell(\hat{\theta}) + C \approx N^{-1} (\ell(\hat{\theta}) - k) \quad (4.46)$$

可以近似看作是最大似然模型平均对数似然 $E_Y [\log f(Y | \hat{\theta})]$ 的一个无偏估计量。

赤池信息量准则 (AIC) 将上述结果乘以 $-2N$, 并定义为:

$$\begin{aligned} \text{AIC} &= -2 \ell(\hat{\theta}) + 2k \\ &= -2 (\text{最大对数似然}) + 2 (\text{参数个数}) \end{aligned} \quad (4.47)$$

由于最小化 AIC 在近似意义下等价于最小化 K-L 信息量, 因此, 通过选择使 AIC 取最小值的模型, 可以得到近似最优的模型。由此, AIC 所提供的模型选择方法能够实现合理且自动化的模型选择。

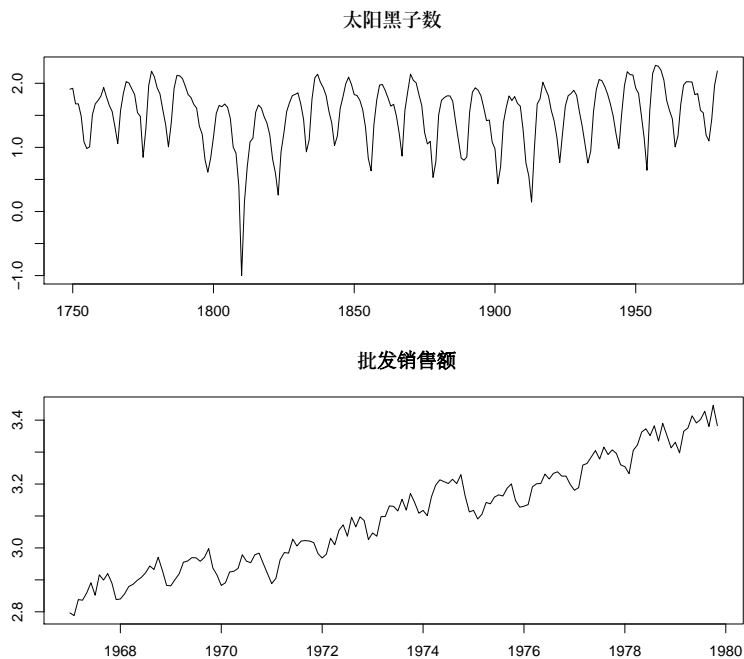


图 4.4 对数变换后的数据

4.6 数据变换

当对某一事件发生的次数、人数或金额等始终取正值的变量进行观测并将所得数据作图时，常会发现随着均值的增加方差也随之增大，或者分布呈现出显著的偏态。由于这类数据在不同位置具有明显不同的性质，且其分布与正态分布存在显著差异，因此若不采用第 11~13 章中的非平稳模型或第 14 章中的非高斯模型，通常很难用简单模型对其进行分析。

```
> par(mar=c(2,2,2,1)+0.1)
> data(WHARD)
> plot(log10(WHARD))
> data(Sunspot)
> plot(log10(Sunspot))
```

即便在上述情形下，对数据进行对数变换 $z_n = \log y_n$ 后，数据的波动往往会变得近似均匀，其边际分布也可能更接近正态分布。图 4.4 给出了图 2.1 中 (b) 与 (e) 所示数据的对数变换结果。在图 2.1(b) 的太阳黑子数数据中可以观察到明显的上下非对称性，而在进行对数变换后的图 4.4 中，这种非对称性已明显减弱，趋于对称。此外，在图 1.1(e) 的批发额数据中，随时间推移波动不断加剧，而通过对数变

换后, 得到的序列在整体上呈现出较为均匀的波动。

作为包含对数变换在内的一般数据变换, **Box - Cox 变换**

$$z_n = \begin{cases} \lambda^{-1}(y_n^\lambda - 1), & \lambda \neq 0 \text{ 时} \\ \log y_n, & \lambda = 0 \text{ 时} \end{cases} \quad (4.48)$$

已被广泛使用。忽略常数项时, 当 $\lambda = 0$ 时该变换等价于对数变换, 当 $\lambda = -1$ 时为取倒数的变换, 当 $\lambda = 0.5$ 时为平方根变换, 而当 $\lambda = 1.0$ 时则与原始数据一致。

利用 AIC 可以选择适合数据变换的参数 λ 。若经 Box - Cox 变换后的数据 $z_n = h(y_n)$ 服从密度函数 $f(z)$, 则原始数据 y_n 的密度函数为

$$g(y) = \left| \frac{dh}{dy} \right| f(h(y)) \quad (4.49)$$

其中, $|dh/dy|$ 称为变换的**雅可比行列式**(Jacobian)。式 (4.49) 表明, 对变换后数据所建立的模型, 同样为变换前的数据隐含地确定了一个模型。

例如, 设对原始数据 y_n 与变换后数据 z_n 分别拟合正态分布, 对应的 AIC 值为 AIC_y 与 AIC_z 。此时, 定义

$$AIC'_z = AIC_z - 2 \sum_{i=1}^N \log \left| \frac{dh}{dy} \right|_{y=y_i} \quad (4.50)$$

并将其与 AIC_y 进行比较, 即可判断原始数据与变换后数据中哪一个更接近正态分布。具体而言, 若 $AIC_y < AIC'_z$, 则原始数据更为合适; 若 $AIC_y > AIC'_z$, 则进行变换更为合适。此外, 还可以通过使 AIC'_z 达到最小, 来选择 Box - Cox 变换中最优的 λ 值。

需要注意的是, 在实际的时间序列数据分析中, 通常会在进行 Box - Cox 变换后, 再拟合多种时间序列模型。在这种情况下, 有必要利用 Box - Cox 变换的雅可比项对时间序列模型的 AIC 进行相应修正。

例: 太阳黑子数数据的 Box-Cox 变换

```
> par( mar=c(2,2,2,1)+0.1 )
> data( Sunspot )
> boxcox( Sunspot )
```

TSSS 包的函数 `boxcox` 用于计算 Box-Cox 变换, 并在对变换后的数据拟合正态分布模型时计算 AIC 的值。表 4.5 总结了将不同取值的 λ 的 Box-Cox 变换应用于图 2.1(b) 所示太阳黑子数数据时得到的结果。在本例中, 当 $\lambda = 0.4$ 时的变换, 即 $z_n = 2.5(y_n^{0.4} - 1)$, 成为最优的 Box-Cox 变换, 此时 AIC 的值为 2267.00。

表 4.5 太阳黑子数数据的各种 Box-Cox 变换及其对应的 AIC 值

λ	AIC'	log-L	AIC	log-L	平均	方差
1.0	2360.37	-1178.19	2360.37	-1178.19	49.109	1576.35
0.8	2313.88	-1154.94	1992.09	-994.04	25.917	320.08
0.6	2281.75	-1138.87	1638.16	-817.08	14.382	69.16
0.4	2267.00	-1131.50	1301.62	-648.81	8.442	16.11
0.2	2274.40	-1135.20	987.23	-491.61	5.261	4.13
0.0	2313.40	-1154.70	704.44	-350.22	3.483	1.21
-0.2	2405.33	-1200.67	474.58	-235.29	2.441	0.45
-0.4	2587.43	-1291.71	334.88	-165.44	1.800	0.25
-0.6	2881.56	-1438.78	307.22	-151.61	1.386	0.22
-0.8	3260.47	-1628.23	364.33	-180.17	1.103	0.28
-1.0	3685.11	-1840.56	467.18	-231.59	8.996	0.43

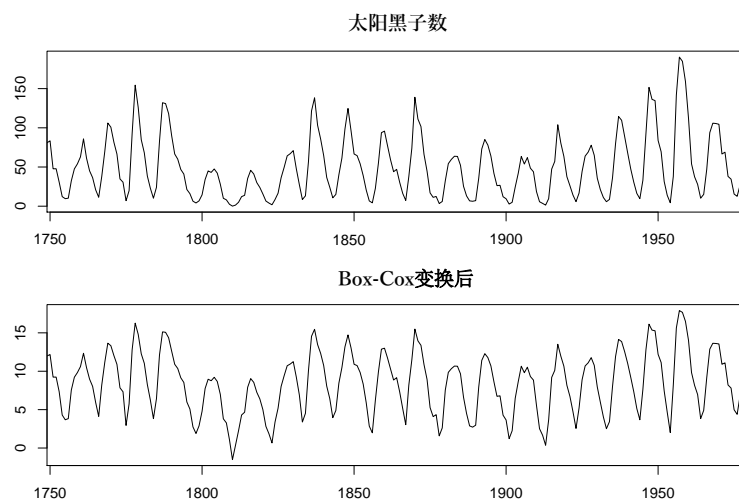


图 4.5 太阳黑子数数据与最优的 Box-Cox 变换

图 4.5 展示了太阳黑子数数据以及进行最优 Box-Cox 变换后的数据。可以看出，通过该变换，原本上下不对称的时间序列被转换为近似对称的数据。

同样地，表 4.6 总结了批发销售额数据的 Box-Cox 变换结果。在该例中，当 $\lambda = -0.3$ 时 AIC 取到最小值 2355.33。图 4.6 展示了批发销售额数据以及使 AIC 最小的 $\hat{\lambda} = -0.3$ 时对应的 Box-Cox 变换。通过该 Box-Cox 变换可以看出，围绕趋势的波动幅度变得近似恒定。

表 4.6 批发销售额数据的 Box-Cox 变换及其对应的 AIC 值

λ	AIC'	log-L	AIC	log-L	平均	方差
1.0	2387.94	-1191.97	2387.94	-1191.97	1348.090	2.80e+05
0.8	2378.55	-1187.27	1936.27	-966.14	393.043	1.52e+04
0.6	2370.74	-1183.37	1486.19	-741.09	121.978	8.33e+02
0.4	2364.52	-1180.26	1037.70	-516.85	41.379	4.61e+01
0.2	2359.90	-1177.95	590.81	-293.40	15.886	2.58e+00
0.0	2356.89	-1176.44	145.51	-70.76	7.133	1.46e-01
-0.2	2355.45	-1175.73	-298.19	151.10	3.796	8.33e-03
-0.3	2355.33	-1175.66	-519.46	261.73	2.939	2.00e-03
-0.4	2355.59	-1175.79	-740.33	372.17	2.354	4.81e-04
-0.6	2357.27	-1176.63	-1180.93	592.46	1.644	2.80e-05
-0.8	2360.46	-1178.23	-1620.01	812.01	1.246	1.65e-06
-1.0	2365.12	-1180.56	-2057.62	1030.81	0.999	9.80e-08

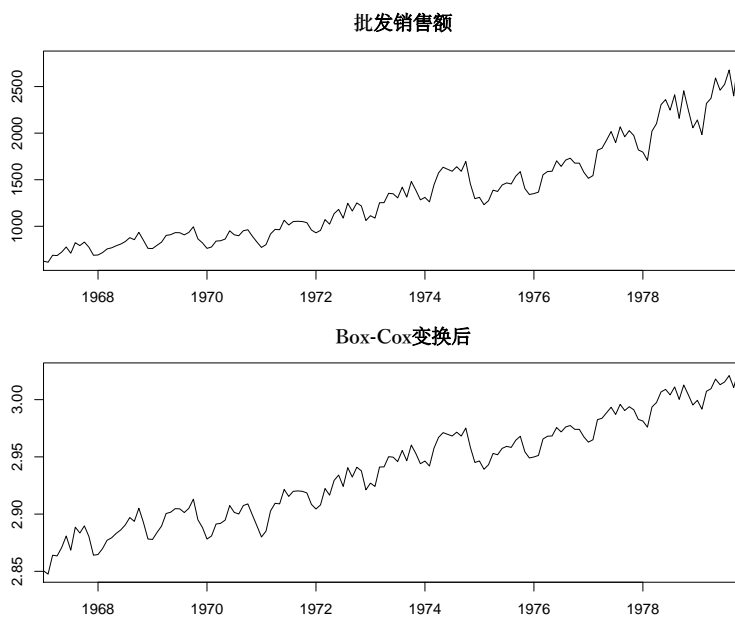


图 4.6 批发销售额数据与使 AIC 最小的 Box-Cox 变换

● 章末习题

- 当给定 n 个整数值数据 $\{m_1, \dots, m_n\}$ 时, 对于泊松分布模型 $f(m|\lambda) = e^{-\lambda} \lambda^m / m!$, 求 λ 的最大似然估计值.
- 对于服从正态分布的两组数据 $\{x_1, \dots, x_n\}$, $\{y_1, \dots, y_m\}$,
 - 在方差相同的假设下, 说明如何利用 AIC 判断均值是否相等.
 - 在均值相同的假设下, 说明如何利用 AIC 判断方差是否相等.

3. 对于数据 $\{y_1, \dots, y_n\}$, 考虑其模型为正态分布或柯西分布两者之一。请思考如何利用 AIC 判断哪一种分布更为合适.
4. 进行 n 次抛硬币试验, 其中正面出现了 m 次。
 - (1) 求正面出现概率的最大似然估计值.
 - (2) 思考如何利用 AIC 判断该硬币是否可以认为正反面等概率.
5. 设真实分布为 $g(y)$, 模型为 $f(y|\theta)$ 。特别地, 当存在某个 θ_0 使得 $f(y|\theta_0) = g(y)$ 成立时, 证明 $J = I$.
6. 当数据 y 的对数变换 $z = \log y$ 服从正态分布 $N(\mu, \sigma^2)$ 时, 求 y 的分布.

第 5 章

最小二乘法

在假设残差服从正态分布的回归模型和时间序列模型中，最小二乘估计值往往与最大似然估计值一致，或能给出其良好的近似值。本章将介绍求解回归模型最小二乘解的基本方法——Householder 法。如后续各章所示，利用该方法不仅能高精度地估计模型系数，还能通过 AIC 高效地进行阶数选择和解释变量选择等时间序列建模操作。

5.1 回归模型与最小二乘法

设 y_n 为目标变量， x_{n1}, \dots, x_{nm} 为解释变量，将 y_n 的变动用解释变量的线性组合表示的模型

$$y_n = \sum_{i=1}^m a_i x_{ni} + \varepsilon_n \quad (5.1)$$

称为**回归模型** (regression model)。 a_i 是与解释变量 x_{ni} 对应的**回归系数** (regression coefficient)， m 是解释变量的个数，称为**阶数** (order)。此外， y_n 的变动中无法由解释变量的变动解释的部分 ε_n 称为**残差** (residual)，假设其为服从均值为 0、方差为 σ^2 的正态分布的独立随机变量。这里，定义 N 维向量 y 和 $N \times m$ 矩阵 Z 为

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad Z = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ x_{21} & \cdots & x_{2m} \\ \vdots & & \vdots \\ x_{N1} & \cdots & x_{Nm} \end{bmatrix} \quad (5.2)$$

则回归模型可以

$$y = Za + \varepsilon \quad (5.3)$$

这一矩阵-向量表达式简洁地描述。 y 称为目标变量向量， Z 称为解释变量矩阵 (设计矩阵)。此外， $a = (a_1, \dots, a_m)^T$ 为回归系数向量， $\varepsilon = (\varepsilon_1, \dots, \varepsilon_N)^T$ 为残差向量。

(5.1) 的回归模型以回归系数 a_1, \dots, a_m 和方差 σ^2 为参数，将它们合并记为 $\theta = (a_1, \dots, a_m, \sigma^2)^T$ 。给定 N 组观测值 $\{y_n, x_{n1}, \dots, x_{nm}\}$ ($n = 1, \dots, N$) 时，回归模型

的似然函数和对数似然函数为 θ 的函数, 分别为

$$L(\theta) = \prod_{n=1}^N p(y_n | \theta, x_{n1}, \dots, x_{nm}) \quad (5.4)$$

$$\ell(\theta) = \sum_{n=1}^N \log p(y_n | \theta, x_{n1}, \dots, x_{nm}) \quad (5.5)$$

其中, 由 (5.1) 可知右边各项为

$$p(y_n | \theta, x_{n1}, \dots, x_{nm}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} \left(y_n - \sum_{i=1}^m a_i x_{ni} \right)^2 \right\} \quad (5.6)$$

$$\log p(y_n | \theta, x_{n1}, \dots, x_{nm}) = -\frac{1}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \left(y_n - \sum_{i=1}^m a_i x_{ni} \right)^2 \quad (5.7)$$

因此对数似然函数为

$$\ell(\theta) = -\frac{N}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{n=1}^N \left(y_n - \sum_{i=1}^m a_i x_{ni} \right)^2 \quad (5.8)$$

通过求使该 $\ell(\theta)$ 最大的 θ , 可以求得参数 θ 的最大似然估计值 $\hat{\theta} = (\hat{a}_1, \dots, \hat{a}_m, \hat{\sigma}^2)^T$ 。

对于任意给定的回归系数 a_1, \dots, a_m , 为求使对数似然 (5.8) 最大的方差 σ^2 , 令

$$\frac{\partial \ell(\theta)}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{n=1}^N \left(y_n - \sum_{i=1}^m a_i x_{ni} \right)^2 = 0 \quad (5.9)$$

求解可得

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N \left(y_n - \sum_{i=1}^m a_i x_{ni} \right)^2 \quad (5.10)$$

因此, $\hat{\sigma}^2$ 可以独立于 a_i 简单地求得。接下来, 将其代入 (5.8), 对数似然函数变为仅关于回归系数 a_1, \dots, a_m 的函数

$$\ell(a_1, \dots, a_m) = -\frac{N}{2} \log 2\pi\hat{\sigma}^2 - \frac{N}{2} \quad (5.11)$$

考虑到对数函数是单调递增函数, 要使对数似然 (5.11) 最大的回归系数 a_1, \dots, a_m , 只需使 (5.10) 的方差 $\hat{\sigma}^2$ 最小即可。由此可知, 回归模型参数的最大似然估计值可通过**最小二乘法** (least squares method) 求得。

5.2 基于 Householder 法的最小二乘法求解方法

如前节所示, 回归系数的最大似然估计值可以通过使 (5.10) 最小的最小二乘法求得。这里, 利用 (5.2) 和 (5.3) 的矩阵-向量表达, 得到

$$\sum_{n=1}^N \left(y_n - \sum_{i=1}^m a_i x_{ni} \right)^2 = \|y - Za\|_N^2 = \|\varepsilon\|_N^2 \quad (5.12)$$

这一简洁的表达式。其中, $\|y\|_N$ 表示 N 维向量 y 的欧几里得范数。

最小二乘法一种广为人知的求解方法是, 对 (5.12) 中的 $\|y - Za\|_N^2$ 关于 a 求偏导并令其为 0, 导出正规方程 $Z^T y = Z^T Z a$, 并求解 $\hat{a} = (Z^T Z)^{-1} Z^T y$ 。然而在实际计算中, 由于计算精度更高且便于进行建模中的各种操作, 基于正交变换的下述方法更为方便。

设 U 为任意的 $N \times N$ 正交矩阵, 将向量 $y - Za$ 经 U 变换后其范数不变。因此,

$$\|y - Za\|_N^2 = \|U(y - Za)\|_N^2 = \|Uy - UZa\|_N^2 \quad (5.13)$$

成立, 使 $\|Uy - UZa\|_N^2$ 最小的向量 a 与使 $\|y - Za\|_N^2$ 最小的 a 相同。这意味着, 为了求回归系数向量 a 的最小二乘解, 可以先通过适当的正交变换将 UZ 变形为便于处理的形式, 然后求使 (5.13) 最小的向量 a 即可。

基于 (5.13) 的关系, 最小二乘法的求解可以利用 **Householder 变换** (Householder transformation) 高效地实现如下。首先, 在解释变量矩阵 Z 的右侧附加目标变量向量 y , 构成 $N \times (m+1)$ 矩阵

$$X = [Z \mid y] \quad (5.14)$$

对该矩阵 X 施加适当的 Householder 变换 U , 可将其变形为如下上三角矩阵 S 。

$$UX = S = \begin{bmatrix} s_{11} & \cdots & s_{1m} & s_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & s_{mm} & s_{m,m+1} \\ & & & s_{m+1,m+1} \\ & \mathbf{O} & & \end{bmatrix} \quad (5.15)$$

此时, S 的第 1~ m 列对应于 UZ , 第 $m+1$ 列对应于 Uy , 因此

$$\begin{aligned} \|Uy - UZa\|_N^2 &= \left\| \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{m,m+1} \\ s_{m+1,m+1} \\ \mathbf{O} \\ \vdots \\ \mathbf{O} \end{bmatrix} - \begin{bmatrix} s_{11} & \cdots & s_{1m} \\ & \ddots & \vdots \\ & & s_{mm} \\ & & & \mathbf{O} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \right\|_N^2 \\ &= \left\| \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{m,m+1} \end{bmatrix} - \begin{bmatrix} s_{11} & \cdots & s_{1m} \\ & \ddots & \vdots \\ & & s_{mm} \\ & & & \mathbf{O} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \right\|_m^2 + s_{m+1,m+1}^2 \end{aligned} \quad (5.16)$$

成立。其中最右端的第 2 项 $s_{m+1,m+1}^2$ 不依赖于 a 的值而取固定值。因此, 使第 1 项最小即为 0 的向量 $a = (a_1, \dots, a_m)^T$ 就是最小二乘解。这意味着 a 的最小二乘估计值可以作为一次方程

$$\begin{bmatrix} s_{11} & \cdots & s_{1m} \\ & \ddots & \vdots \\ & & s_{mm} \\ \mathbf{O} & & \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{m,m+1} \end{bmatrix} \quad (5.17)$$

的解来求得。由于 (5.17) 的一次方程左边的矩阵为上三角形, 因此可以通过后向代入简单地求解。即

$$\begin{aligned}\hat{a}_m &= \frac{s_{m,m+1}}{s_{mm}} \\ \hat{a}_i &= \frac{(s_{i,m+1} - s_{i,i+1}\hat{a}_{i+1} - \cdots - s_{i,m}\hat{a}_m)}{s_{ii}}, \quad i = m-1, \cdots, 1\end{aligned}\quad (5.18)$$

即可。此外, 由于 $s_{m+1,m+1}^2$ 给出了残差向量长度的平方, 因此 m 阶回归模型的残差方差 σ^2 的估计值为

$$\hat{\sigma}_m^2 = \frac{s_{m+1,m+1}^2}{N} \quad (5.19)$$

5.3 通过 AIC 进行阶数选择

将由 (5.19) 求得的残差方差估计值代入 (5.11), 最大对数似然为

$$\ell(\hat{\theta}) = -\frac{N}{2} \log 2\pi \hat{\sigma}_m^2 - \frac{N}{2} \quad (5.20)$$

此外, m 阶回归模型包含的参数为 a_1, \cdots, a_m 和 σ^2 共 $m+1$ 个。因此, m 阶回归模型的 AIC 为

$$\begin{aligned}\text{AIC}_m &= -2\ell(\hat{\theta}) + 2(\text{参数个数}) \\ &= N(\log 2\pi \hat{\sigma}_m^2 + 1) + 2(m+1)\end{aligned}\quad (5.21)$$

利用 Householder 法, 一旦求得 (5.15) 的上三角矩阵 S , 就不仅可以求 m 阶回归模型, 还可以求所有低于该阶数的回归模型。即当 $j \leq m$ 时, 以 x_{n1}, \cdots, x_{nj} 为解释变量的 j 阶回归模型

$$y_n = \sum_{i=1}^j a_i x_{ni} + \varepsilon_n \quad (5.22)$$

的残差方差估计值和 AIC 为

$$\begin{aligned}\hat{\sigma}_j^2 &= \frac{1}{N} \sum_{i=j+1}^{m+1} s_{i,m+1}^2 \\ \text{AIC}_j &= N(\log 2\pi \hat{\sigma}_j^2 + 1) + 2(j+1)\end{aligned}\quad (5.23)$$

可由此计算。为了估计回归系数, 只需通过后向代入求解一次方程

$$\begin{bmatrix} s_{11} & \cdots & s_{1j} \\ & \ddots & \vdots \\ & & s_{jj} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} s_{1,m+1} \\ \vdots \\ s_{j,m+1} \end{bmatrix} \quad (5.24)$$

即可。

利用 AIC 进行阶数选择, 只需通过 (5.23) 计算 $\text{AIC}_0, \cdots, \text{AIC}_m$, 找出取最小值的阶数即可。这里值得注意的是, 一旦求得上三角矩阵 S , 无需进行回归系数的估计,

就可以通过 (5.23) 立即计算出所有阶数的回归模型的 AIC。因此, 回归系数只需对 AIC 最小的阶数的模型通过 (5.24) 来求解即可。

例: 三角函数回归模型

程序包 TSSS 中的函数 `lsqr` 通过 Householder 法的最小二乘法拟合三角函数回归模型。所考虑模型的最高阶数 k 由参数 `lag` 指定, 为 $2 \times \text{lag} + 1$ 。当未指定 `lag` 时, 默认值为数据个数 N 的平方根 \sqrt{N} 。

```
> data( Temperature )
> lsqr( Temperature, lag=10 )
> lsqr( Temperature )
```

表 5.1 是对图 1.1(c) 所示的最高气温数据拟合各种阶数的三角函数模型

$$y_n = a + \sum_{j=1}^m b_j \sin j\omega n + \sum_{j=1}^{\ell} c_j \cos j\omega n + \varepsilon_n \quad (5.25)$$

时的残差方差和 AIC 值的汇总。其中, $L = \text{lag}$, $0 \leq m \leq L$, ℓ 为 m 或 $m - 1$ 。最右列的数字表示与最小 AIC 的差值。解释变量按 $\{1, \sin \omega n, \cos \omega n, \dots, \sin L\omega n, \cos L\omega n\}$ 的顺序采用。因此, 最高阶数的模型的参数向量为 $\theta = (a, b_1, c_1, \dots, b_L, c_L)^T$ 。由于未指定最高阶数, 默认设置下最高阶模型具有 45 个解释变量。由于该数据存在明显的年周期, 取 $\omega = \frac{2\pi}{365}$ 。

TSSS 程序包的函数 `lsqr` 可计算 AIC、 σ^2 、各阶数的回归系数、使 AIC 最小的阶数以及由 MAICE 模型得到的回归曲线, 并将结果分别返回到成分 `aic`, `sigma2`, `regress`, `maice.order`, `tripoly` 中。因此, 例如需要 AIC 的图时, 可以按如下方式操作。

```
> z <- lsqr( Temperature )
> x <- seq( 0,45,length=46 )
> par( mfrow = c(1,2) )
> plot( x,z$aic,type="b",pch=20 )
> plot( x,z$aic,type="b",pch=20,ylim=c(2430,2490) )
```

图 5.1 展示了解释变量个数从 0 到 45 时 AIC 的变化。由表 5.1 和图 5.1 可知, 阶数为 29 的模型, 即包含常数项、14 个正弦分量和 14 个余弦分量的模型, AIC 最小。

图 5.2 展示了最高气温数据和由该 AIC 最小模型得到的回归曲线。可以看出该曲线在一定程度上较好地捕捉了气温的变化。

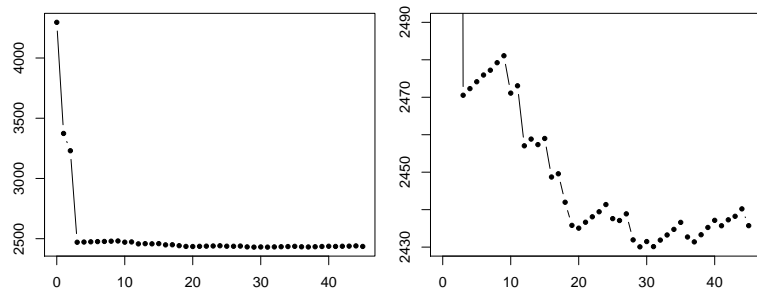


图 5.1 最高气温数据的三角函数回归模型的 AIC。右图为纵轴放大显示。

表 5.1 各种阶数的三角函数回归模型的残差方差和 AIC

阶数	残差方差	AIC	AIC 之差	阶数	残差方差	AIC	AIC 之差
0	402.618	4298.23	1866.16	23	8.027	2439.44	9.37
1	60.086	3375.76	943.68	24	8.025	2441.36	11.29
2	44.538	3232.23	800.16	25	7.931	2437.61	7.53
3	9.291	2475.54	40.47	26	7.890	2437.11	7.03
4	9.287	2474.31	42.24	27	7.887	2438.88	8.80
5	9.284	2476.14	44.08	28	7.743	2431.93	1.86
6	9.280	2477.94	45.86	29	7.682	2430.07	0.00
7	9.266	2479.23	47.15	30	7.672	2431.49	1.41
8	9.266	2481.22	49.15	31	7.620	2430.13	0.06
9	9.263	2483.08	51.01	32	7.615	2431.88	1.80
10	9.038	2473.11	41.04	33	7.605	2433.24	3.17
11	9.037	2475.06	42.99	34	7.598	2434.74	4.66
12	8.708	2459.04	26.97	35	7.596	2436.60	6.53
13	8.705	2460.84	28.77	36	7.504	2432.70	2.63
14	8.643	2459.36	27.29	37	7.453	2431.40	1.32
15	8.636	2461.00	28.93	38	7.452	2433.30	3.23
16	8.421	2450.71	18.64	39	7.451	2435.25	5.17
17	8.401	2451.57	19.50	40	7.449	2437.15	7.08
18	8.237	2443.98	11.90	41	7.397	2435.70	5.63
19	8.099	2437.80	5.73	42	7.390	2437.29	7.21
20	8.054	2438.06	4.99	43	7.374	2438.23	8.15
21	8.047	2438.65	6.57	44	7.374	2440.21	10.14
22	8.038	2438.09	8.02	45	7.276	2435.72	5.65

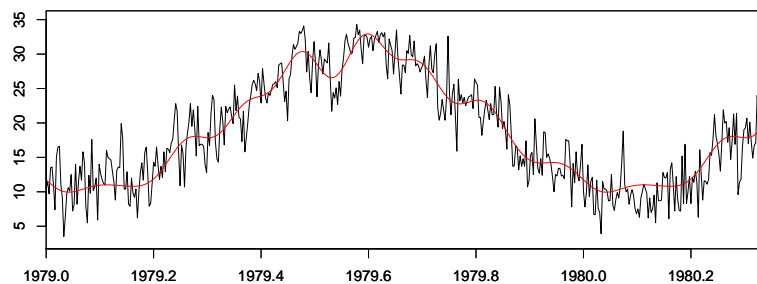


图 5.2 最高气温数据的原始数据与 29 阶三角函数回归曲线

5.4 数据的追加与分割处理

利用正交变换的性质, 可以方便地通过追加数据来更新模型。当对大量数据拟合回归模型时, 如果试图构造 (5.14) 的矩阵, 可能会因内存不足而无法执行。即使在这种情况下, 只要确保 $L \times (m+1)$ 矩阵 (其中 $L > m+1$) 的存储空间, 通过反复应用本节介绍的方法, 就可以将数据分成每 $L-m-1$ 个一组进行处理。

假设已从 N 个数据 $\{y_n, x_{n1}, \dots, x_{nm}\}$ ($n = 1, \dots, N$) 得到 (5.15) 的上三角矩阵 S 。如 5.2 节所示, 利用该 S 可以方便地求得回归模型。现在假设获得了 M 组新数据 $\{y_n, x_{n1}, \dots, x_{nm}\}$ ($n = N+1, \dots, N+M$)。此时, 要对全部 $N+M$ 个数据拟合回归模型, 需要与 (5.14) 类似地构造 $(N+M) \times (m+1)$ 矩阵

$$X_1 = \begin{bmatrix} x_{11} & \cdots & x_{1m} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{N+M,1} & \cdots & x_{N+M,m} & y_{N+M} \end{bmatrix} \quad (5.26)$$

并通过 Householder 变换 $S' = U'X_1$ 将其变换为上三角矩阵。但是, 这种方法不仅浪费了此前的计算, 而且需要为 X_1 准备非常大的存储空间。

然而, 由于 Householder 变换是正交变换, 首先在 (5.15) 的上三角矩阵下方附加新的矩阵, 构成 $(M+m+1) \times (m+1)$ 矩阵

$$X_2 = \begin{bmatrix} s_{11} & \cdots & s_{1m} & s_{1,m+1} \\ 0 & \ddots & \vdots & \vdots \\ \vdots & \ddots & s_{mm} & s_{m,m+1} \\ 0 & \cdots & 0 & s_{m+1,m+1} \\ x_{N+1,1} & \cdots & x_{N+1,m} & y_{N+1} \\ \vdots & \ddots & \vdots & \vdots \\ x_{N+M,1} & \cdots & x_{N+M,m} & y_{N+M} \end{bmatrix} \quad (5.27)$$

将其通过 Householder 变换缩约为上三角矩阵, 即可得到与 S' 相同的矩阵。因此, 要对长于 L 的数据进行 Householder 变换, 只需先令 $N=L$ 求得 S , 然后每次追加 $M=L-m-1$ 个数据即可。此外, 当新数据 $\{y_n, x_{n1}, \dots, x_{nm}\}$ ($n = N+1, \dots, N+M$) 已经得到上三角矩阵 S_2 时, 可通过

$$X_3 = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \quad (5.28)$$

构造 $2(m+1) \times (m+1)$ 矩阵并将其缩约为上三角矩阵, 同样可以得到与 S' 相同的矩阵。当 $M \gg m$ 时, 矩阵 X_3 的第 1 个下标的大小小于 X_1 或 X_2 , 因此 X_3 的 Householder 变换所需的计算量显著减小。该方法可用于第 8 章中局部平稳 AR 模型的估计。

5.5 通过 AIC 进行变量选择

在 5.3 节中说明了利用 AIC 进行阶数选择的方法, 该方法中解释变量的采用顺序是预先确定的, 仅考虑了

$$y_n = \sum_{i=1}^j a_i x_{ni} + \varepsilon_n \quad (5.29)$$

这种形式的模型。这种模型选择方式在后述的自回归模型 (6.1 节) 和多项式回归模型 (11.1 节) 等中是极为自然的。但是, 在一般的多元回归模型和多变量时间序列模型等中, 作为解释变量使用的变量的优先顺序通常并未预先确定, 因此需要以 (ℓ_1, \dots, ℓ_m) 为表示解释变量采用优先顺序的索引向量, 在

$$y_n = \sum_{i=1}^j a_{\ell_i} x_{n, \ell_i} + \varepsilon_n \quad (5.30)$$

这些模型中选择最优模型。在这种情况下, 即使确定了阶数 j , 根据索引向量的取法, 也存在 mC_j 种模型。这样的模型称为子集回归模型 (subset regression model)。

为了估计以 (ℓ_1, \dots, ℓ_m) 为解释变量优先顺序的模型, 需要利用 Householder 矩阵将上三角矩阵 S 从左依次变换为由非零元素个数为 $j_1, \dots, j_m, m+1$ 个列向量组成的矩阵 T 。其中, (j_1, \dots, j_m) 是索引向量 (ℓ_1, \dots, ℓ_m) 的逆函数, 满足 $\ell_{j_i} = i$ 。例如, 当 $m=4$, $(\ell_1, \ell_2, \ell_3, \ell_4) = (2, 4, 3, 1)$ 时, $(j_1, \dots, j_m) = (4, 1, 3, 2)$,

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} & t_{15} \\ t_{21} & 0 & t_{23} & t_{24} & t_{25} \\ t_{31} & 0 & t_{33} & 0 & t_{35} \\ t_{41} & 0 & 0 & 0 & t_{45} \\ 0 & 0 & 0 & 0 & t_{55} \end{bmatrix} \quad (5.31)$$

即可。

此时, 按照优先顺序仅使用前 j 个解释变量的模型的残差方差和 AIC 为

$$\hat{\sigma}^2(\ell_1, \dots, \ell_j) = \frac{1}{N} \sum_{i=j+1}^{m+1} t_{i, m+1}^2$$

$$\text{AIC}(\ell_1, \dots, \ell_j) = N \log 2\pi \hat{\sigma}^2(\ell_1, \dots, \ell_j) + N + 2(j+1) \quad (5.32)$$

可由此计算。为了估计回归系数, 需要通过后向代入求解一次方程

$$\begin{bmatrix} t_{1, \ell_1} & \cdots & t_{1, \ell_j} \\ & \ddots & \vdots \\ 0 & & t_{j, \ell_j} \end{bmatrix} \begin{bmatrix} a_{\ell_1} \\ \vdots \\ a_{\ell_j} \end{bmatrix} = \begin{bmatrix} t_{1, m+1} \\ \vdots \\ t_{j, m+1} \end{bmatrix} \quad (5.33)$$

即可。但在实际计算中, 不需要交换列将 T 还原为上三角形式,

$$\hat{a}_{\ell_j} = t_{j, \ell_j}^{-1} t_{j, m+1} \quad (5.34)$$

$$\hat{a}_{\ell_i} = t_{i, \ell_i}^{-1} (t_{i, m+1} - t_{i, \ell_{i+1}} \hat{a}_{\ell_{i+1}} - \cdots - t_{j, \ell_j} \hat{a}_{\ell_j}), \quad (i = j-1, \dots, 1)$$

通过上述后向代入, 可直接从 (5.31) 的矩阵 T 求得回归系数。

● 章末习题

1. 在 (5.1) 式中, 求方差 σ^2 已知时的 AIC。
2. 给定 N 个数据 $(x_1, y_1), \dots, (x_N, y_N)$ 时
 - (1) 求过原点的二次多项式模型 $y_n = ax_n^2 + bx_n + \varepsilon_n$ 的系数 a, b 的最小二乘解。
 - (2) 给出回归式过原点和 $(c, 0)$ 的二次多项式模型, 并思考其最小二乘解的求法。

第 6 章

基于 ARMA 模型的时间预测分析

通过使用时间序列模型，不仅可以简洁地表达时间序列的特征，而且还可以用于时间序列的预测、信号提取以及决策等。本章将从一元 ARMA 模型出发，介绍冲激响应函数、自协方差函数、偏自相关系数 (PARCOR)、功率谱以及求特性方程根的方法，并说明 AR 系数与 PARCOR 之间的关系。此外，还将说明基于多元 AR 模型的互谱以及噪声贡献率的计算方法。

6.1 ARMA 模型

将时间序列 y_n 表示为过去观测值 y_{n-i} 与白色噪声当前及过去值的线性和的模型

$$y_n = \sum_{i=1}^m a_i y_{n-i} + v_n - \sum_{i=1}^{\ell} b_i v_{n-i} \quad (6.1)$$

称为**自回归移动平均模型** (AutoRegressive Moving Average model, ARMA 模型)。其中， m 与 a_i 分别称为自回归的**阶数** (order)，以及**自回归系数** (AR coefficient)。另外， ℓ 与 b_i 分别称为移动平均的阶数以及**移动平均系数** (MA coefficient)。也常把两个阶数合起来称为 (m, ℓ) 阶的 ARMA 模型。同时， v_n 是假定与时间序列过去值 y_{n-i} 独立、服从均值 0、方差 σ^2 的正态分布的**白色噪声**。也就是说， v_n 满足以下条件：

$$\begin{aligned} E[v_n] &= 0, & E[v_n^2] &= \sigma^2, \\ E[v_n v_m] &= 0, & n &\neq m \text{ のとき} \\ E[v_n y_m] &= 0, & n &> m \text{ 时} \end{aligned} \quad (6.2)$$

服从 ARMA 模型的时间序列 y_n 称为 ARMA 过程。

在实际应用中，通常假设 $\ell = 0$ ，即用时间序列的过去值与时刻 n 的白色噪声 v_n 来表示时间序列的模型

$$y_n = \sum_{i=1}^m a_i y_{n-i} + v_n \quad (6.3)$$

最为重要，称为 m 阶**自回归模型** (AutoRegressive model, AR 模型)。需要注意的是，

定常时间序列分析中相当大的一部分仅用 AR 模型就可以实现。另外令 $m = 0$ 时, 可以得到 ℓ 阶移动平均模型 (Moving Average model, MA 模型)。

6.2 冲激响应函数

用 $By_n \equiv y_{n-1}$ 定义的时间移位算子 (lag operator) B 表示 ARMA 模型, 可写成

$$\left(1 - \sum_{i=1}^m a_i B^i\right) y_n = \left(1 - \sum_{i=1}^{\ell} b_i B^i\right) v_n \quad (6.4)$$

在这里, 将 AR 算子与 MA 算子分别定义为

$$a(B) \equiv \left(1 - \sum_{i=1}^m a_i B^i\right), \quad b(B) \equiv \left(1 - \sum_{i=1}^{\ell} b_i B^i\right)$$

则 ARMA 模型可简洁地写成

$$a(B)y_n = b(B)v_n \quad (6.5)$$

将 (6.5) 两边除以 $a(B)$, ARMA 模型可写成 $y_n = a(B)^{-1}b(B)v_n$ 。因此, 若用形式上得到的无穷级数

$$g(B) \equiv a(B)^{-1}b(B) = \sum_{i=0}^{\infty} g_i B^i \quad (6.6)$$

来定义 $g(B)$, 则服从 ARMA 模型的时间序列 y_n 可以仅用到目前为止加入的白色噪声 v_n 表示为

$$y_n = g(B)v_n = \sum_{i=0}^{\infty} g_i v_{n-i} \quad (6.7)$$

即表示为无限阶的移动平均模型。 $\{g_i; i = 0, 1, \dots\}$ 表示在某一时刻加入的噪声在 i 期后对时间序列产生影响的大小, 称为 ARMA 模型的冲激响应函数 (impulse response function)。 g_i 可以通过逐次代入很容易地计算:

$$\begin{aligned} g_0 &= 1 \\ g_i &= \sum_{j=1}^i a_j g_{i-j} - b_i, \quad i = 1, 2, \dots \end{aligned} \quad (6.8)$$

其中当 $j > m$ 时令 $a_j = 0$, 当 $j > \ell$ 时令 $b_j = 0$ 。

例 考虑下面 4 个模型。

- (a) 一阶 AR 模型: $y_n = 0.9y_{n-1} + v_n$
- (b) 二阶 AR 模型: $y_n = 0.9\sqrt{3}y_{n-1} - 0.81y_{n-2} + v_n$
- (c) 二阶 MA 模型: $y_n = v_n - 0.9\sqrt{2}v_{n-1} + 0.81v_{n-2}$
- (d) (2,2) 阶 ARMA 模型:

$$y_n = 0.9\sqrt{3}y_{n-1} - 0.81y_{n-2} + v_n - 0.9\sqrt{2}v_{n-1} + 0.81v_{n-2}$$

图 6.1 的 (a)、(b)、(c)、(d) 是对 4 个模型用 (6.8) 求得的冲激响应函数的示意图。

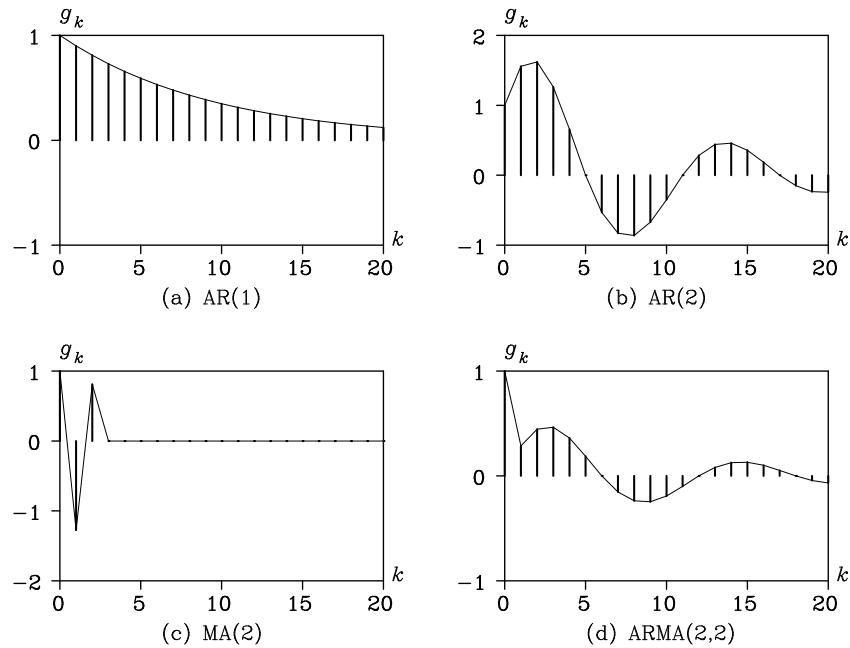


图 6.1 4 个模型的冲激响应函数

6.3 自协方差函数

在 (6.1) 两边乘以 y_{n-k} 并取期望, 可得

$$E[y_n y_{n-k}] = \sum_{i=1}^m a_i E[y_{n-i} y_{n-k}] + E[v_n y_{n-k}] - \sum_{i=1}^{\ell} b_i E[v_{n-i} y_{n-k}] \quad (6.9)$$

这里, 关于右边时间序列 y_n 与白色噪声 v_n 的协方差, 注意到用 (6.6) 的冲激响应函数表示有

$$E[v_{n-i} y_{n-k}] = \sum_{j=0}^{\infty} g_j E[v_{n-i} v_{n-k-j}] = \begin{cases} 0, & i > k \\ \sigma^2 g_{i-k}, & i \leq k \end{cases} \quad (6.10)$$

于是对于 ARMA 模型的自协方差函数 $C_k \equiv E[y_n y_{n-k}]$ 可得到如下方程:

$$\begin{aligned} C_0 &= \sum_{i=1}^m a_i C_i + \sigma^2 \left\{ 1 - \sum_{i=1}^{\ell} b_i g_i \right\} \\ C_k &= \sum_{i=1}^m a_i C_{k-i} - \sigma^2 \sum_{i=1}^{\ell} b_i g_{i-k}, \quad k = 1, 2, \dots \end{aligned} \quad (6.11)$$

因此, 当给定 ARMA 模型的阶数 m 、 ℓ 以及参数 a_i 、 b_i 、 σ^2 时, 可先用 (6.8) 求冲激响应函数 g_1, \dots, g_{ℓ} , 再通过解 (6.11) 求得自协方差函数 C_0, C_1, \dots 。特别地, 在 $\ell = 0$ 的 AR 模型情形下,

$$\begin{aligned} C_0 &= \sum_{i=1}^m a_i C_{-i} + \sigma^2 \\ C_k &= \sum_{i=1}^m a_i C_{k-i} \end{aligned} \quad (6.12)$$

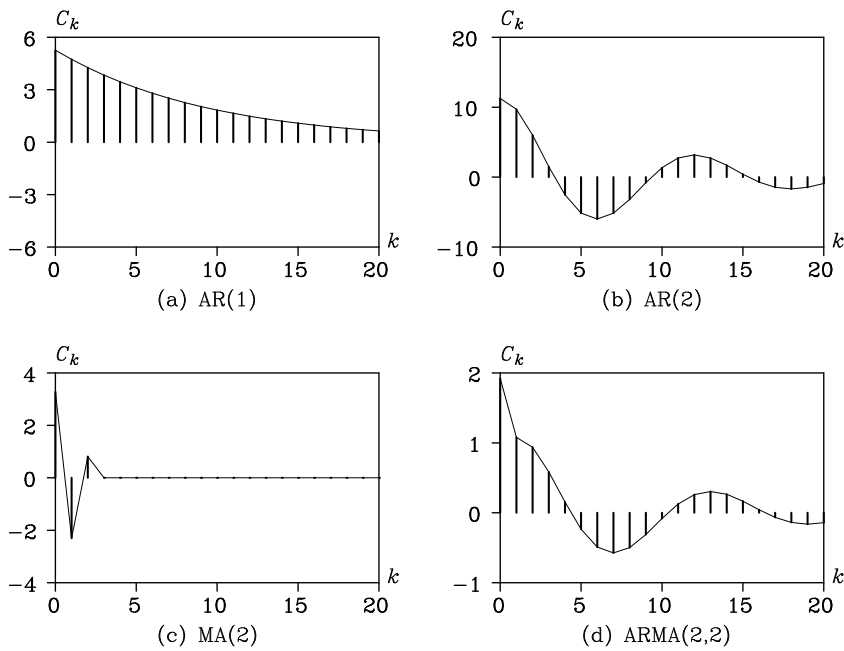


图 6.2 4 个模型的自协方差函数

称为 **Yule-Walker 方程** (Yule-Walker equation)。

例 图 6.2 给出了图 6.1 所用的 4 个模型 (a)~(d) 的自协方差函数。在 (c) 的 MA 模型情形中, 当 $k > 2$ 时有 $C_k = 0$, 而在 (b) 与 (d) 中 ($m > 0$), 自协方差函数会在振荡的同时衰减。

6.4 AR 系数与 PARCOR 的关系

如附录 B 所示, 对给定的自协方差函数 C_0, C_1, \dots , 与之对应的 $m-1$ 阶 AR 模型系数 a_i^{m-1} 与 m 阶 AR 模型系数 a_i^m 之间满足如下关系:

$$a_i^m = a_i^{m-1} - a_m^m a_{m-i}^{m-1}, \quad (i = 1, \dots, m-1) \quad (6.13)$$

其中 a_m^m 称为 m 阶 **PARCOR** (partial autocorrelation coefficient, 偏自相关系数)。若给定到 M 阶的 PARCOR, 则反复应用 (6.13) 即可求出从 2 阶到 M 阶的所有 AR 模型系数。

反过来, 若给定最高阶 AR 模型的系数 a_1^m, \dots, a_m^m , 由 (6.13) 可得

$$a_i^{m-1} = a_i^m - a_m^m (a_{m-i}^m + a_m^m a_i^{m-1}) \quad (6.14)$$

解之可得

$$a_i^{m-1} = \frac{a_i^m + a_m^m a_{m-i}^m}{1 - (a_m^m)^2} \quad (6.15)$$

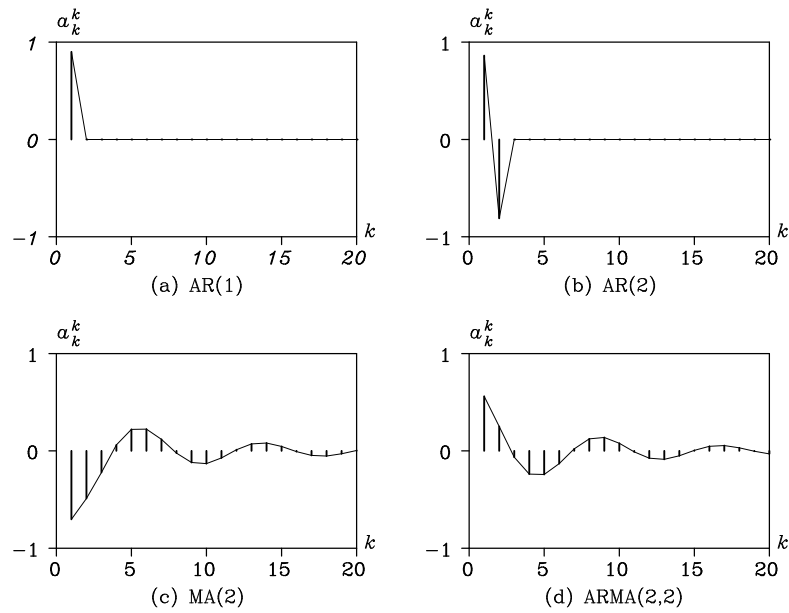


图 6.3 4 个模型的 PARCOR

从而得到 $m-1$ 阶 AR 模型系数。重复这一过程即可求得所有 PARCOR a_1^1, \dots, a_m^m 。

由此可知, 估计 m 阶 AR 模型系数 a_1^m, \dots, a_m^m 与估计到 m 阶的 PARCOR a_1^1, \dots, a_m^m 是等价的。

例 图 6.3 给出了 (a)~(d) 四个模型的 PARCOR。与自协方差函数相反, 对于 AR 模型有 $a_k^k = 0, k > m$, 但对于 MA 模型与 ARMA 模型, 即使 $k > l$ 也不会出现 $a_k^k = 0$, 而是逐渐衰减。

6.5 功率谱

正如在给定时间序列的 ARMA 模型后可以计算自协方差函数一样, 也可以求得功率谱。ARMA 过程 (6.1) 的功率谱由 (6.7) 可表示为

$$\begin{aligned}
 p(f) &= \sum_{k=-\infty}^{\infty} C_k e^{-2\pi i k f} & (6.16) \\
 &= \sum_{k=-\infty}^{\infty} E[y_n y_{n-k}] e^{-2\pi i k f} \\
 &= \sum_{k=-\infty}^{\infty} E \left[\left(\sum_{j=0}^{\infty} g_j v_{n-j} \right) \left(\sum_{p=0}^{\infty} g_p v_{n-k-p} \right) \right] e^{-2\pi i k f} \\
 &= \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} \sum_{p=0}^{\infty} g_j g_p E[v_{n-j} v_{n-k-p}] e^{-2\pi i k f}
 \end{aligned}$$

这里利用 (6.2) 以及当 $p < 0$ 时 $g_p = 0$ 可得

$$\begin{aligned}
 p(f) &= \sigma^2 \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} g_j g_{j-k} e^{-2\pi i k f} \\
 &= \sigma^2 \sum_{j=0}^{\infty} \sum_{k=-\infty}^j g_j e^{-2\pi i j f} g_{j-k} e^{-2\pi i (k-j) f} \\
 &= \sigma^2 \sum_{j=0}^{\infty} \sum_{p=0}^{\infty} g_j e^{-2\pi i j f} g_p e^{2\pi i p f} \\
 &= \sigma^2 \left| \sum_{j=0}^{\infty} g_j e^{-2\pi i j f} \right|^2
 \end{aligned} \tag{6.17}$$

其中 $\sum_{j=0}^{\infty} g_j e^{-2\pi i j f}$ 是冲激响应函数的傅里叶变换, 称为**频率响应函数** (frequency response function)。另一方面, 在 (6.6) 中令 $B = e^{-2\pi i f}$ 可得

$$\sum_{j=0}^{\infty} g_j e^{-2\pi i j f} = \left\{ 1 - \sum_{j=1}^m a_j e^{-2\pi i j f} \right\}^{-1} \left\{ 1 - \sum_{j=1}^{\ell} b_j e^{-2\pi i j f} \right\} \tag{6.18}$$

因此代入 (6.17), ARMA 模型的功率谱为

$$p(f) = \sigma^2 \frac{\left| 1 - \sum_{j=1}^{\ell} b_j e^{-2\pi i j f} \right|^2}{\left| 1 - \sum_{j=1}^m a_j e^{-2\pi i j f} \right|^2} \tag{6.19}$$

给出。

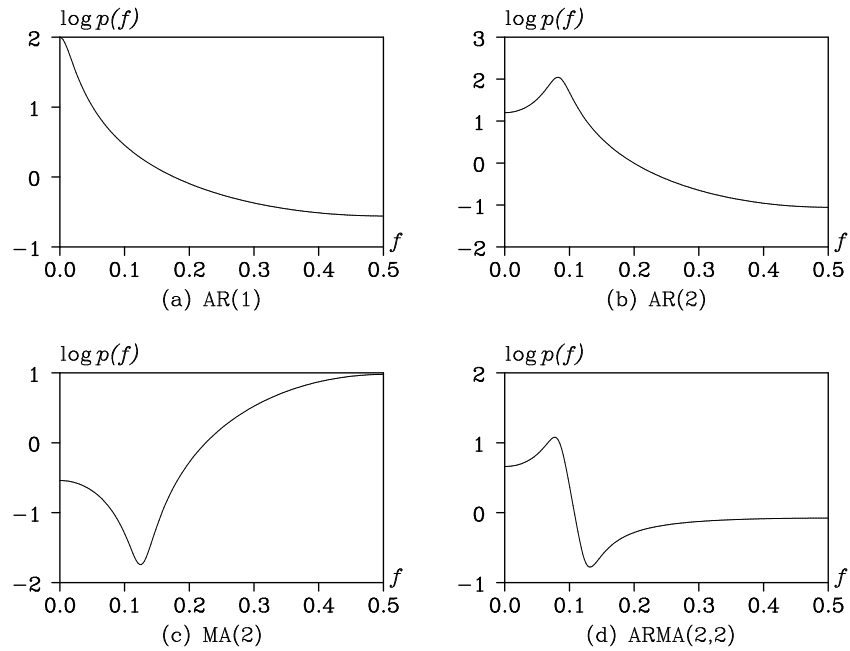


图 6.4 4 个模型的功率谱

例 图 6.4 给出了 4 个模型功率谱的对数值。(a) 中的一阶 AR 模型的谱是一条向右下倾斜的平滑曲线。而 (b) 中的二阶 AR 模型谱出现一个峰, (c) 中的二阶 MA 模型则出现一个谷。(d) 的 (2,2) 阶 ARMA 模型则同时出现一个峰和一个谷。

从以上例子可以推测, AR 阶数与 MA 阶数与谱的峰谷之间存在密切关系。图 6.4 中的 $\log p(f)$ 可表示为

$$\log p(f) = \log \sigma^2 - 2 \log \left| 1 - \sum_{j=1}^m a_j e^{-2\pi i j f} \right| + 2 \log \left| 1 - \sum_{j=1}^{\ell} b_j e^{-2\pi i j f} \right| \quad (6.20)$$

因此, 在 $|1 - \sum_{j=1}^m a_j e^{-2\pi i j f}|$ 的极小点会出现谱的峰, 在 $|1 - \sum_{j=1}^{\ell} b_j e^{-2\pi i j f}|$ 的极小点会出现谱的谷。峰谷的数量对应于下一节所述 AR 算子与 MA 算子特性方程的根的个数: 要表示 k 个峰, 需要至少 $2k$ 阶的 AR; 要表示 k 个谷, 需要至少 $2k$ 阶的 MA。此外, 谱峰谷的位置与高度与特性方程复根的辐角与模有关。

特别地, 当 AR 算子与 MA 算子的复根辐角一致时, 可以表示线状的谱。

$$\begin{aligned} m = 2, \quad a_1 = 0.99\sqrt{2}, \quad a_2 = -0.99^2 \\ \ell = 2, \quad b_1 = 0.95\sqrt{2}, \quad b_2 = -0.95^2 \end{aligned}$$

此时二者都在 $f = 0.125 (= 45^\circ)$ 处有根, $\log p(f)$ 如图 6.5 所示。

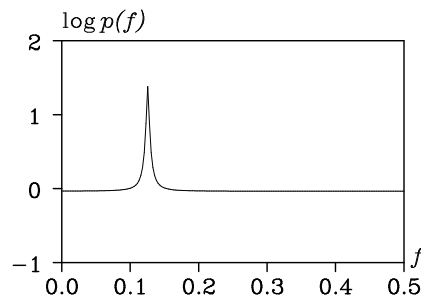


图 6.5 ARMA(2,2) 模型的线状谱

6.6 特性方程式

ARMA 模型的性质可由以下两个多项式的根来刻画:

$$a(B) = 1 - \sum_{j=1}^m a_j B^j = 0 \quad (6.21)$$

$$b(B) = 1 - \sum_{j=1}^{\ell} b_j B^j = 0 \quad (6.22)$$

(6.21) 称为 AR 算子的**特性方程** (characteristic equation), (6.22) 称为 MA 算子的特性方程。这些方程的根称为**特性根** (characteristic root)。当 AR 算子的特性方程 $a(B) = 0$ 的所有根都在单位圆之外时, 某一时刻加入的扰动影响会随时间衰减, ARMA 模型为**定常**。另一方面, 当 MA 算子的特性方程 $b(B) = 0$ 的所有根都在单位圆之外时, $b(B)^{-1} = \sum_{i=0}^{\infty} h_i B^i$ 的系数 h_i 收敛, 从而可用无限阶 AR 模型表示为

$$y_n = - \sum_{i=1}^{\infty} h_i y_{n-i} + v_n \quad (6.23)$$

因此该时间序列称为**可逆** (invertible)。

如前一节所述, 这两个特性多项式根的位置与谱的形状也密切相关。当 AR (MA)

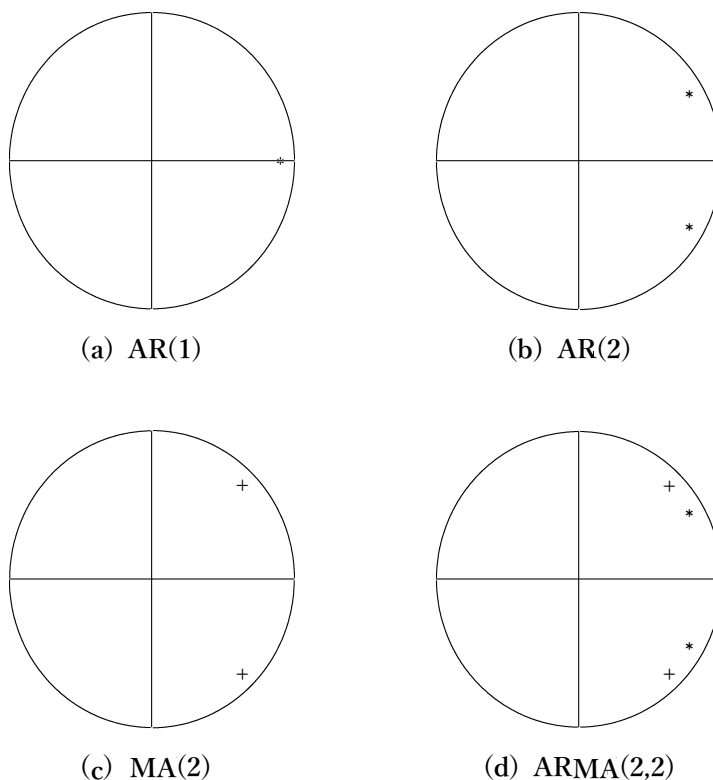


图 6.6 4 个模型的特性根。(a) 一阶 AR 模型, (b) 二阶 AR 模型, (c) 二阶 MA 模型, (d) (2,2) 阶 ARMA 模型

算子的复根可写成

$$z = \alpha + i\beta = re^{i\theta} \quad (6.24)$$

的形式时, 在 $f = \frac{\theta}{2\pi}$ 处会出现谱的峰 (谷)。并且 r 越接近 1, 谱的峰或谷越尖锐。图 6.6 示意了前述 4 个模型的特性根位置。符号 * 表示 AR 算子的特性根, 符号 + 表示 MA 算子的特性根位置。为便于作图, 这里实际显示的是 $z^{-1} = r^{-1}e^{-i\theta}$ 的位置, 而不是 z 本身。

TSSS 包的函数 `armachar` 可以计算并绘制 ARMA 模型的冲激响应函数、自协方差函数、PARCOR、功率谱以及 AR 算子与 MA 算子的特性根。该函数需要以下参数:

- `arcoef`: AR 系数向量
- `macoef`: MA 系数向量
- `v`: 白色噪声的方差
- `lag`: 自协方差的滞后, 默认值为 50
- `nf`: 用于计算谱的频率点数, 默认值为 200

从函数 `armachar` 可得到以下输出:

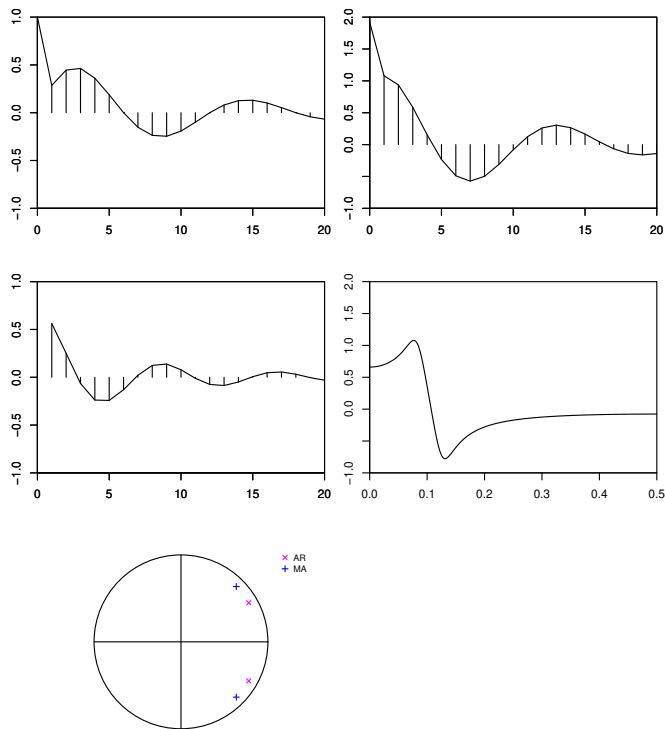


图 6.7 ARMA(2,2) 情形下函数 `armachar` 的输出。左上：冲激响应函数，右上：自协方差函数，中左：PARCOR，中右：功率谱（对数尺度），下：特性根

`impuls`: 冲激响应函数
`acov`: 自协方差函数
`parcor`: 偏自相关系数
`spec`: 功率谱
`croot.ar`: AR 算子的特性根
`croot.ma`: MA 算子的特性根

```

> # AR model : y(n) = a(1)*y(n-1) + a(2)*y(n-2) + v(n)
> a <- c(0.9 * sqrt(3), -0.81)
> armachar(arcoef = a, v = 1.0, n = 1000, lag = 20)

> # MA model : y(n) = v(n) - b(1)*v(n-1) - b(2)*v(n-2)
> b <- c(0.9 * sqrt(2), -0.81)
> armachar(macoeef = b, v = 1.0, n = 1000, lag = 20)

> # ARMA model : y(n) = a(1)*y(n-1) + a(2)*y(n-2)
> # + v(n) - b(1)*v(n-1) - b(2)*v(n-2)
> armachar(arcoef = a, macoeef = b, v = 1.0, n = 1000, lag = 20)

> # ARMA model with line spectrum
> a <- c(0.99 * sqrt(2), -0.9801)
> b <- c(0.95 * sqrt(2), -0.9025)
> armachar(arcoef = a, macoeef = b, v = 1.0, n = 1000, lag = 20)

```

图 6.7 给出了 ARMA(2,2) 模型情形下函数 `armachar` 的输出, 包括冲激响应函数、自协方差函数、PARCOR、功率谱 (对数尺度) 以及特性根。

6.7 多元 AR 模型

对于多元时间序列 $y_n = (y_n(1), \dots, y_n(\ell))^T$, 也与一元情形类似, 可以用时间序列当前值由过去值 y_{n-1}, \dots, y_{n-M} 与白色噪声表示的模型

$$y_n = \sum_{m=1}^M A_m y_{n-m} + v_n \quad (6.25)$$

称为多元自回归模型 (multivariate AR model)。其中, A_m 是一个 $\ell \times \ell$ 矩阵, 其 (i, j) 元素为 $a_m(i, j)$, 称为自回归系数矩阵。 v_n 是 ℓ 维白色噪声, 满足

$$\begin{aligned} E[v_n] &= \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}, & E[v_n v_n^T] &= \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1\ell} \\ \vdots & \ddots & \vdots \\ \sigma_{\ell 1} & \cdots & \sigma_{\ell\ell} \end{bmatrix} = W \\ E[v_n v_m^T] &= O, & n \neq m \text{ のとき} & \\ E[v_n y_m^T] &= O, & n > m \text{ のとき} & \end{aligned} \quad (6.26)$$

其中 O 表示所有元素为 0 的 $\ell \times \ell$ 矩阵, W 是对称矩阵, 并满足 $\sigma_{ij} = \sigma_{ji}$ 。

$y_n(i)$ 与 $y_n(j)$ 的互协方差函数定义为 $C_k(i, j) = E[y_n(i) y_{n-k}(j)]$ 。此时, 把 $C_k(i, j)$ 作为 (i, j) 元素的 $\ell \times \ell$ 矩阵 C_k 称为互协方差函数, 并可写为 $C_k = E[y_n y_{n-k}^T]$ 。对于服从

多元 AR 模型的时间序列, C_k 与一元情形类似, 满足 Yule - Walker 方程

$$C_0 = \sum_{j=1}^M A_j C_{-j} + W \quad (6.27)$$

$$C_k = \sum_{j=1}^M A_j C_{k-j} \quad (k = 1, 2, \dots) \quad (6.28)$$

互协方差函数 $C_k(s, j)$ 的傅里叶变换

$$\begin{aligned} p_{sj}(f) &= \sum_{k=-\infty}^{\infty} C_k(s, j) e^{-2\pi i k f} \\ &= \sum_{k=-\infty}^{\infty} C_k(s, j) \cos 2\pi k f - i \sum_{k=-\infty}^{\infty} C_k(s, j) \sin 2\pi k f \end{aligned} \quad (6.29)$$

称为互谱密度函数 (cross-spectral density function), 或简称互谱 (cross spectrum)。

由于互协方差函数不是偶函数, 互谱会像 (6.29) 第二项那样出现 sin 变换项, 从而成为复数。再定义 $\ell \times \ell$ 矩阵 $P(f)$ 为

$$P(f) = \begin{bmatrix} p_{11}(f) & \cdots & p_{1\ell}(f) \\ \vdots & \ddots & \vdots \\ p_{\ell 1}(f) & \cdots & p_{\ell\ell}(f) \end{bmatrix} \quad (6.30)$$

则与一元时间序列类似, C_k 与 $P(f)$ 之间满足

$$P(f) = \sum_{k=-\infty}^{\infty} C_k e^{-2\pi i k f} \quad (6.31)$$

$$C_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} P(f) e^{2\pi i k f} df \quad (6.32)$$

的关系。

对于服从多元 AR 模型的时间序列, 其互谱可由

$$P(f) = A(f)^{-1} W (A(f)^{-1})^* \quad (6.33)$$

求得。这里, A^* 表示矩阵 A 的共轭转置, $A(f)$ 表示一个 $\ell \times \ell$ 矩阵, 其 (j, k) 元素为

$$A_{jk}(f) = \sum_{m=0}^M a_m(j, k) e^{-2\pi i m f} \quad (6.34)$$

其中规定 $a_0(j, j) = -1$, 且当 $(j \neq k)$ 时 $a_0(j, k) = 0$ 。互谱为复数, 令

$$\begin{aligned} \alpha_{jk}(f) &= \sqrt{(\Re\{p_{jk}(f)\})^2 + (\Im\{p_{jk}(f)\})^2} \\ \phi_{jk}(f) &= \arctan \Im\{p_{jk}(f)\} / \Re\{p_{jk}(f)\} \end{aligned}$$

则

$$p_{jk}(f) = \alpha_{jk}(f) e^{i\phi_{jk}(f)} \quad (6.35)$$

其中 \Re 与 \Im 分别表示复数的实部与虚部。此时 $\alpha_{jk}(f)$ 称为**幅度谱** (amplitude spectrum), $\phi_{jk}(f)$ 称为**相位谱** (phase spectrum)。此外

$$\text{coh}_{jk}(f) = \frac{\alpha_{jk}(f)^2}{p_{jj}(f)p_{kk}(f)} \quad (6.36)$$

表示时间序列 $y_n(j)$ 与 $y_n(k)$ 在频率 f 处频率分量之间相关系数的平方, 称为**相干性** (coherency)。

下文为简化起见, 将 $A(f)^{-1}$ 记为 $B(f) = (b_{jk}(f))$ 。当白色噪声 v_n 各分量之间互不相关, 且其方差协方差矩阵为对角矩阵 $W = \text{diag}\{\sigma_1^2, \dots, \sigma_\ell^2\}$ 时, 第 i 个分量的功率谱可写为

$$p_{ii}(f) = \sum_{j=1}^{\ell} b_{ij}(f)\sigma_j^2 b_{ij}(f)^* \equiv \sum_{j=1}^{\ell} |b_{ij}(f)|^2 \sigma_j^2 \quad (6.37)$$

这表示第 i 个分量在频率 f 处的波动功率谱可以分解为 ℓ 个噪声源的影响, 其影响大小由 $|b_{ij}(f)|^2 \sigma_j^2$ 表示。因此令

$$r_{ij}(f) = \frac{|b_{ij}(f)|^2 \sigma_j^2}{p_{ii}(f)} \quad (6.38)$$

则 $r_{ij}(f)$ 表示 $y_n(i)$ 在频率 f 处的波动中由 $v_n(j)$ 引起的比例。 $r_{ij}(f)$ 称为**相对功率贡献率** (relative power contribution), 是分析反馈系统的有效手段。

TSSS 包的函数 `marsspc` 用于计算多元时间序列的互谱、相干性以及功率贡献率。需要以下参数:

`arcoef`: AR 系数矩阵
`v`: 噪声的方差协方差矩阵

该函数输出如下:

`spec`: 互谱
`amp`: 幅度谱
`phase`: 相位谱
`coh`: 相干性
`power`: 功率谱分解
`rpowor`: 相对功率贡献率

```

> data(HAKUSAN)
> yy <- as.matrix(HAKUSAN[, c(1,2,4)])
> nc <- dim(yy)[1]
> n <- seq(1, nc, by = 2)
> y <- yy[n, ]
> # Fit MAR model
> z <- marfit(y, lag = 20)
> # Draw Cross-spectra, coherency, power contribution
> marspc(z$arcoef, v = z$v)

```

例： 船舶数据的功率贡献率

图 6.7 展示了利用从图 1.1(a) 与 (h) 所示的方向角速度、横摇、舵角三条时间序列中每隔一点抽样得到的数据 ($N=500$, $\Delta t = 2$ 秒) 拟合的三元 AR 模型所计算得到的互谱。9 幅图中对角线上的 3 幅分别表示方向角速度、横摇、舵角的功率谱对数值。方向角速度的功率谱在 $f = 0.25$ (周期 8 秒) 附近出现最大峰值; 横摇与舵角则在 $f = 0.125$ (周期 16 秒) 附近出现最大峰值。另一方面, 对角线之上 3 幅图表示互谱的绝对值, 也就是幅度谱的对数值; 对角线之下 3 幅图表示相位谱。相位谱中可见不连续的跳变, 这是因为相位是在 $[-\pi, \pi]$ 范围内绘制的。

图 6.8 展示了功率谱与相干性。对角线上的 3 幅与图 6.7 相同的功率谱, 其余 3 幅表示相干性。

图 6.9 表示功率贡献率。最上方两幅分别表示方向角速度的功率谱分解 (左图) 与相对功率贡献率 (右图)。在两幅图中, 从下到上依次表示方向角速度、横摇、舵角的贡献部分。在 $f < 0.13$ 时可看到舵角影响相当明显, 但在方向角速度主要功率所在的 $f = 0.14$ 、 0.23 附近几乎看不到舵角影响。这暗示该数据是在传统自动驾驶仪进行自动操舵时获得的, 并使用了抑制 $f < 0.13$ 以下低频振动的控制律。从上往下第二组两幅图表示横摇的噪声贡献率。在功率谱较强的 $f = 0.1$ 附近, 舵角影响接近 50 另一方面, 在 $f > 0.14$ 时可以看到方向角速度的强影响, 但实际功率非常小。从上往下第三组两幅图表示舵角的功率贡献率。在主要功率所在的 $f = 0.12$ 附近, 方向角速度的影响极强。并且在 $f < 0.08$ 范围内, 频率越低方向角速度的贡献越高。

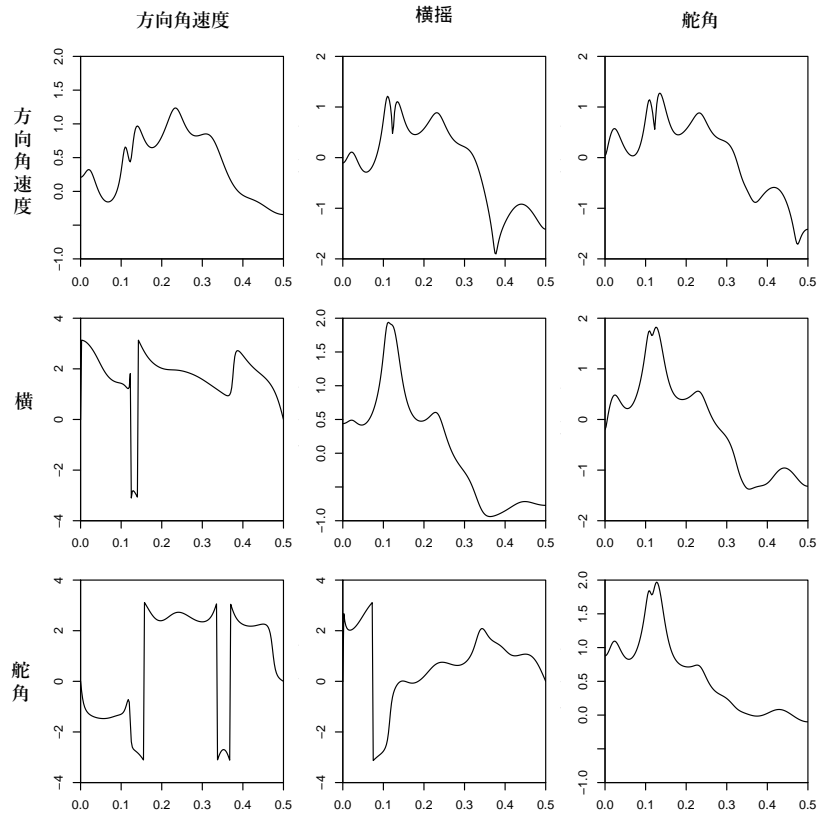


图 6.8 船舶三元数据（方向角速度、横摇、舵角）的谱（对角线），幅度谱（对角线以上），相位谱（对角线以下）

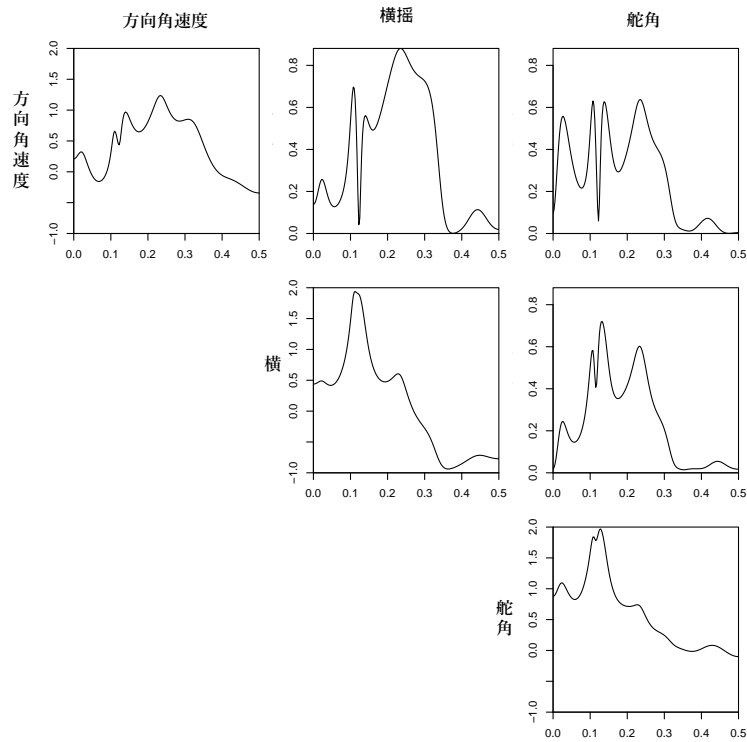


图 6.9 三元船舶数据的功率谱（对角线上的 3 幅）与相干性

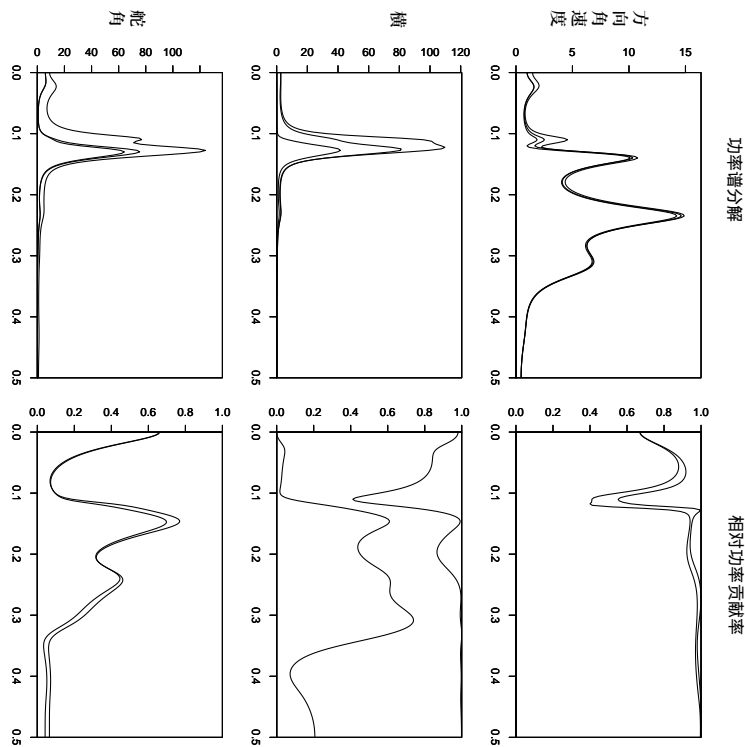


图 6.10 三元船舶数据的功率谱分解 (左) 与相对功率贡献率 (右)

● 章末习题

- (1) 给出 AR(1) 的定常条件。
(2) 给出 AR(2) 的定常条件。
- AR(1), $y_n = ay_{n-1} + v_n$, $v_n \sim N(0, \sigma^2)$ 时
 - 求 1 期预测误差方差。
 - 求 2 期预测误差方差。
 - 求 k 期预测误差方差。
- 当时间序列 y_n 服从下列模型时, 求自协方差函数 C_k 。其中 v_n 为均值 0、方差 1 的正态白色噪声。
 - 一阶 AR 模型: $y_n = -0.9y_{n-1} + v_n$.
 - 二阶 AR 模型: $y_n = 1.2y_{n-1} - 0.6y_{n-2} + v_n$
 - 一阶 MA 模型: $y_n = v_n - bv_{n-1}$
 - (1,1) 阶 ARMA 模型: $y_n = ay_{n-1} + v_n - bv_{n-1}$.
- 当时间序列服从一阶 AR 模型 $y_n = ay_{n-1} + v_n$, $v_n \sim N(0, 1)$ 时
 - 若噪声项 v_n 不是白色噪声, 而是服从一阶 AR 过程 $v_n = bv_{n-1} + w_n$, 证明

y_n 为二阶 AR 过程。

- (2) 在 y_n 上叠加独立观测噪声得到 $x_n = y_n + w_n, w_n \sim N(0, 0.1)$, 求其自协方差函数 $C_k, k = 0, 1, 2, 3$ 。
5. (1) 利用问题 3(3) 的结果与谱定义, 证明一阶 MA 模型的谱可写为 $p(f) = |1 - be^{-2\pi if}|^2$ 。并说明右边也可写为 $1 + b^2 - 2b\cos(2\pi f)$ 。
- (2) 当 $\sigma^2 = 1$ 时, 利用一阶 AR 模型 $y_n = ay_{n-1} + v_n$ 的谱 $p(f) = (1 - 2a\cos(2\pi f) + a^2)^{-1}$, 证明谱的最大值出现在 $f = 0$ 或 $f = 0.5$ 。并进一步说明在什么条件下 $p(f)$ 达到最大?
6. 关于二阶 AR 模型 $y_n = a_1y_{n-1} + a_2y_{n-2} + v_n, v_n \sim N(0, \sigma^2)$
- (1) 给出求自协方差函数 C_0, C_1, \dots 的递推式。
- (2) 对同一模型, 给出求功率谱 $p(f)$ ($0 \leq f \leq 1/2$) 的公式。
- (3) 当 $a_1 = 0.8, a_2 = -0.6, \sigma^2 = 1$ 时, 求 C_0, C_1, C_2 。
- (4) 同一情形下, 写出功率谱 $p(f)$ 的计算式, 并考察该 $p(f)$ 在哪个频率 f 处谱最大。
7. 设时间序列 y_n 服从一阶 MA 模型 $y_n = v_n - bv_{n-1}, v_n \sim N(0, 1)$ 。
- (1) 求 y_n 的自协方差函数 $C_k, k = 0, 1, 2, 3$ 。
- (2) 将 y_n 用 AR 模型表示时会是什么形式?
8. (1) 对服从一阶 MA 模型 $y_n = v_n - bv_{n-1}$ 的时间序列, 求长期预测误差 $\epsilon_{n+k|n}$ 的方差。其中 $v_n \sim N(0, 1)$ 。
- (2) 将一阶 AR 模型 $y_n = ay_{n-1} + v_n$ 表示成 (无限阶) MA 模型形式, 并利用它进行 (k 期) 长期预测时, 求误差方差的大小。
- (3) 将随机游走模型 $y_n = y_{n-1} + v_n$ 形式展开, 求其 MA 表示, 并利用它计算进行 (k 期) 长期预测时误差方差的大小。

第 7 章

AR 模型的估计

在上一章中介绍的平稳时间序列模型中, 对于 AR 模型存在极其高效的估计方法。本章将介绍利用 Yule-Walker 方法、最小二乘法以及 PARCOR 方法对一元 AR 模型参数进行估计的方法, 以及利用 AIC 确定模型阶数的方法。此外, 还将介绍基于 Yule-Walker 方法和最小二乘法的多变量 AR 模型估计方法。

7.1 AR 模型的拟合

当给定时间序列 y_1, \dots, y_N 时, 考虑对其进行如下模型拟合: **自回归模型** (autoregressive model, AR 模型)

$$y_n = \sum_{i=1}^m a_i y_{n-i} + v_n \quad (7.1)$$

其中, m 为自回归的**阶数**, a_i 为自回归系数, v_n 为均值为 0、方差为 σ^2 的正态白噪声。为了对数据拟合 AR 模型, 需要确定阶数 m , 并估计参数 a_1, \dots, a_m 和分散 σ^2 。以下将这些参数记为 $\theta = (a_1, \dots, a_m, \sigma^2)^T$ 。为了对数据拟合 AR 模型, 需要确定阶数 m , 并估计参数 a_1, \dots, a_m 和分散 σ^2 。以下将这些参数记为 $\theta = (a_1, \dots, a_m, \sigma^2)^T$ 。

首先假设阶数 m 已知, 考虑利用极大似然法估计参数 θ 。若时间序列服从 AR 模型, 则 $y = (y_1, \dots, y_N)^T$ 的联合分布为多元正态分布。

在假设模型 (7.1) 成立时, y 的均值向量为 0, 协方差矩阵由自协方差函数 C_k 表示为: 首先假设阶数 m 已知, 考虑利用极大似然法估计参数 θ 。若时间序列服从 AR 模型, 则 $y = (y_1, \dots, y_N)^T$ 的联合分布为多元正态分布。

在假设模型 (7.1) 成立时, y 的均值向量为 0, 协方差矩阵由自协方差函数 C_k 表示为:

$$\Sigma = \begin{bmatrix} C_0 & C_1 & \cdots & C_{N-1} \\ C_1 & C_0 & \cdots & C_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-1} & C_{N-2} & \cdots & C_0 \end{bmatrix} \quad (7.2)$$

因此, AR模型的似然函数为:

$$\begin{aligned} L(\boldsymbol{\theta}) &= p(y_1, \dots, y_N | \boldsymbol{\theta}) \\ &= (2\pi)^{-\frac{N}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} y^T \Sigma^{-1} y \right\} \end{aligned} \quad (7.3)$$

但该方法需要计算 $N \times N$ 矩阵的逆与行列式, 当数据量较大时计算困难。此外, 通过最大化 (7.3), 可以得到 $\boldsymbol{\theta}$ 的极大似然估计值, 但由于似然函数是参数 $\boldsymbol{\theta}$ 的复杂函数, 因此其极大化需要借助数值方法。

将时间序列模型的似然表示为如下条件分布的乘积, 可以高效地计算似然。

$$\begin{aligned} L(\boldsymbol{\theta}) &= p(y_1, \dots, y_N | \boldsymbol{\theta}) \\ &= p(y_1, \dots, y_{N-1} | \boldsymbol{\theta}) p(y_N | y_1, \dots, y_{N-1}, \boldsymbol{\theta}) \\ &= \dots \\ &= \prod_{n=1}^N p(y_n | y_1, \dots, y_{n-1}, \boldsymbol{\theta}) \end{aligned} \quad (7.4)$$

利用卡尔曼滤波, 对于 ARMA 模型以及其他时间序列模型, 可以高效且严格地计算 (7.4) 右边的各项, 并求得极大似然估计值。不过, 这种方法将在第9章中说明。本章将介绍对 (7.4) 进行近似, 从而高效求取极大似然估计近似值的方法。

通过最大化 (7.4) 得到 AR 模型的极大似然估计值 $\hat{\boldsymbol{\theta}}$ 后, 该模型的 AIC 为

$$\begin{aligned} \text{AIC} &= -2(\text{最大对数尤度}) + 2(\text{パラメータ数}) \\ &= -2\log L(\hat{\boldsymbol{\theta}}) + 2(m+1) \end{aligned} \quad (7.5)$$

由此可以计算。因此, 为了通过 AIC 最小化法确定 AR 的阶数 m , 只需计算到预先设定的最大阶数 M 为止的 AR 模型的 AIC, 即 $\text{AIC}_0, \dots, \text{AIC}_M$, 并从中选取取最小值的阶数即可。

7.2 Yule-Walker 方法与 Levinson 算法

如第6章所示, m 阶 AR 模型 (7.1) 的自协方差函数满足 Yule-Walker 方程 (Yule-Walker equation)

$$C_0 = \sum_{i=1}^m a_i C_i + \sigma^2 \quad (7.6)$$

$$C_j = \sum_{i=1}^m a_i C_{j-i} \quad (7.7)$$

反过来, 当给定时间序列时, 首先计算样本自协方差函数 \hat{C}_k , 将其代入 (7.7), 便得到以自回归系数 a_1, \dots, a_m 为未知数的线性方程组

$$\begin{bmatrix} \hat{C}_0 & \hat{C}_1 & \cdots & \hat{C}_{m-1} \\ \hat{C}_1 & \hat{C}_0 & \cdots & \hat{C}_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{m-1} & \hat{C}_{m-2} & \cdots & \hat{C}_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \hat{C}_1 \\ \hat{C}_2 \\ \vdots \\ \hat{C}_m \end{bmatrix} \quad (7.8)$$

解这个方程即可得到 AR 系数的估计值 \hat{a}_i 。此外, σ^2 的估计值可由 (7.6)

$$\hat{\sigma}^2 = \hat{C}_0 - \sum_{i=1}^m \hat{a}_i \hat{C}_i \quad (7.9)$$

求得。由这种方法得到的 $\hat{a}_1, \dots, \hat{a}_m, \hat{\sigma}^2$ 称为 **Yule-Walker 估计值** (Yule-Walker estimates)。

以 a_i 为系数的 AR 模型所对应的预测误差方差为

$$\begin{aligned} E[v_n^2] &= E\left[\left(y_n - \sum_{i=1}^m a_i y_{n-i}\right)\right]^2 \\ &= C_0 - 2 \sum_{i=1}^m a_i C_i + \sum_{i=1}^m \sum_{j=1}^m a_i a_j C_{i-j} \end{aligned} \quad (7.10)$$

因此

$$\frac{\partial E[v_n^2]}{\partial a_i} = -2C_i + 2 \sum_{j=1}^m a_j C_{i-j} = 0 \quad (7.11)$$

由此可得到 (7.7)。因此, 将 (7.11) 中的 C_i 替换为 \hat{C}_i , 并求解所得到的 (7.8), 所得到的 Yule-Walker 估计值可以解释为近似地使预测误差方差最小的估计值。

为了求得 m 阶 Yule-Walker 估计值, 需要求解 (7.8) 中的 m 元线性方程组。另一方面, 为了通过 AIC 最小化法确定 AR 模型的阶数, 需要事先得到直到预先设定的最高阶数 M 为止的所有 AR 模型的 AIC, 因此必须分别求解从 1 阶到 M 阶的线性方程组各一次, 从而得到所有阶数模型的估计值。然而, 如果使用下面给出的 **Levinson 算法** (Levinson's algorithm), 就可以逐次且高效地计算这些估计值。以下将 m 阶 AR 模型的系数记为 a_m^m , 方差记为 σ_m^2 。

1. $\hat{\sigma}_0^2 = \hat{C}_0$

$$\text{AIC}_0 = N(\log 2\pi \hat{\sigma}_0^2 + 1) + 2$$

2. $m = 1, \dots, M$ について

- (a) $\hat{a}_m^m = \left(\hat{C}_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} \hat{C}_{m-j}\right) (\hat{\sigma}_{m-1}^2)^{-1}$

- (b) $\hat{a}_i^m = \hat{a}_i^{m-1} - \hat{a}_m^m \hat{a}_{m-i}^{m-1}$

- (c) $\hat{\sigma}_m^2 = \hat{\sigma}_{m-1}^2 \{1 - (\hat{a}_m^m)^2\}$

$$(d) \text{AIC}_m = N(\log 2\pi\hat{\sigma}_m^2 + 1) + 2(m+1)$$

在 Levinson 算法中, 第 6 章引入的 PARCOR \hat{a}_m^m 起着重要作用。关于该算法, 将在附录 B 中作详细说明。

7.3 用最小二乘法估计 AR 模型

本节说明第 5 章中介绍的利用最小二乘法估计 AR 模型的方法。令 $\theta = (a_1, \dots, a_m, \sigma^2)^T$, 则由 (7.4) 可知, AR 模型的对数似然为

$$\ell(\theta) = \sum_{n=1}^N \log p(y_n | y_1, \dots, y_{n-1}) \quad (7.12)$$

这里, 对于 m 阶 AR 模型, 由于 y_n 的分布由 y_{n-1}, \dots, y_{n-m} 的值决定, 因此, 对于 (7.12) 中第 $m+1$ 项及其之后的各项, 有

$$\begin{aligned} p(y_n | y_1, \dots, y_{n-1}) &= p(y_n | y_{n-m}, \dots, y_{n-1}) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{1}{2\sigma^2} \left(y_n - \sum_{i=1}^m a_i y_{n-i} \right)^2 \right\} \\ \log p(y_n | y_1, \dots, y_{n-1}) &= -\frac{1}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \left(y_n - \sum_{i=1}^m a_i y_{n-i} \right)^2 \end{aligned} \quad (7.13)$$

因此, 如果忽略 (7.12) 的最初 M 项 ($M \geq m$), 则可由

$$\ell(\theta) = -\frac{N-M}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} \sum_{n=M+1}^N \left(y_n - \sum_{i=1}^m a_i y_{n-i} \right)^2 \quad (7.14)$$

得到 AR 模型对数似然的近似值。与回归模型的情形类似, 对于任意给定的自回归系数 a_1, \dots, a_m , 使 (7.14) 达到最大值的分散 σ^2 满足

$$\frac{\partial \ell(\theta)}{\partial \sigma^2} = -\frac{N-M}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{n=M+1}^N \left(y_n - \sum_{i=1}^m a_i y_{n-i} \right)^2 = 0 \quad (7.15)$$

由此得到

$$\hat{\sigma}^2 = \frac{1}{N-M} \sum_{n=M+1}^N \left(y_n - \sum_{i=1}^m a_i y_{n-i} \right)^2 \quad (7.16)$$

将其代入 (7.14), 则对数似然成为仅关于自回归系数 a_1, \dots, a_m 的函数

$$\ell(a_1, \dots, a_m) = -\frac{N-M}{2} \log 2\pi\hat{\sigma}^2 - \frac{N-M}{2} \quad (7.17)$$

这里注意到对数函数是单调递增函数, 因此可知, 近似对数似然 (7.17) 的最大化可以通过最小化分散 σ^2 来实现。这样, 自回归模型的极大似然估计的近似值便可以用最小二乘法求得。

为了利用第 5 章中的 Householder 方法求出直到 M 阶为止的 AR 模型的全部最小二乘解, 只需令

$$Z = \begin{bmatrix} y_M & y_{M-1} & \cdots & y_1 \\ y_{M+1} & y_M & \cdots & y_2 \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-1} & y_{N-2} & \cdots & y_{N-M} \end{bmatrix}, \quad y = \begin{bmatrix} y_{M+1} \\ y_{M+2} \\ \vdots \\ y_N \end{bmatrix} \quad (7.18)$$

即可。因此, 在实际计算中, 构造 $(N-M) \times (M+1)$ 矩阵

$$X = [Z | y] = \begin{bmatrix} y_M & \cdots & y_1 & y_{M+1} \\ y_{M+1} & \cdots & y_2 & y_{M+2} \\ \vdots & \ddots & \vdots & \vdots \\ y_{N-1} & \cdots & y_{N-M} & y_N \end{bmatrix} \quad (7.19)$$

并通过 Householder 变换将其化为上三角矩阵

$$HX = \begin{bmatrix} S \\ O \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots & s_{1M} & s_{1,M+1} \\ & \ddots & \vdots & \vdots \\ & & s_{MM} & s_{M,M+1} \\ & & & s_{M+1,M+1} \\ & & & 0 \end{bmatrix} \quad (7.20)$$

这时, j 阶 ($0 \leq j \leq M$) 自回归模型的残差分散以及 AIC 可由

$$\hat{\sigma}_j^2 = \frac{1}{N-M} \sum_{i=j+1}^{M+1} s_{i,M+1}^2$$

$$\text{AIC}_j = (N-M)(\log 2\pi \hat{\sigma}_j^2 + 1) + 2(j+1) \quad (7.21)$$

计算得到。此外, 自回归系数只需通过回代求解线性方程组

$$\begin{bmatrix} s_{11} & \cdots & s_{1j} \\ & \ddots & \vdots \\ & & s_{jj} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} s_{1,M+1} \\ \vdots \\ s_{j,M+1} \end{bmatrix} \quad (7.22)$$

即可。

7.4 用 PARCOR 法估计 AR 模型

7.2 节中的 Levinson 算法利用了这样一个事实: 当给定自协方差函数 C_0, C_1, \dots 时, 对于 $m-1$ 阶 AR 模型和 m 阶 AR 模型的系数, 有

$$a_j^m = a_j^{m-1} - a_m^m a_{m-j}^{m-1} \quad (7.23)$$

这样的关系成立。因此, 一旦估计出 PARCOR a_m^m , 其余系数便自动确定。在 Levinson 算法中, 作为 a_m^m 的估计值, 使用了附录 (B.8) 中将样本自协方差函数 $\hat{C}_0, \dots, \hat{C}_m$ 代

入后得到的

$$\begin{aligned}\hat{a}_m^m &= \left\{ \hat{C}_0 - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} \hat{C}_j \right\}^{-1} \left\{ \hat{C}_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} \hat{C}_{m-j} \right\} \\ &= \left(\hat{\sigma}_{m-1}^2 \right)^{-1} \left\{ \hat{C}_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} \hat{C}_{m-j} \right\}\end{aligned}\quad (7.24)$$

而本节将说明一种不经过样本自协方差函数, 而是直接从时间序列 y_1, \dots, y_N 估计 PARCOR a_m^m 的方法。

首先, 令 w_n^{m-1} 为 $m-1$ 阶后向 AR 模型

$$y_n = \sum_{j=1}^{m-1} a_j^{m-1} y_{n+j} + w_n^{m-1} \quad (7.25)$$

的预测误差。对于一元时间序列, 由于自协方差函数是偶函数, 可以利用前向和后向 AR 模型的系数相同这一性质。此时由 (B.2) 可得

$$\begin{aligned}C_m - \sum_{j=1}^{m-1} a_j^{m-1} C_{m-j} &= E \left[\left(y_n - \sum_{j=1}^{m-1} a_j^{m-1} y_{n-j} \right) y_{n-m} \right] \\ &= E [v_n^{m-1} y_{n-m}] \\ &= E [v_n^{m-1} w_{n-m}^{m-1}]\end{aligned}\quad (7.26)$$

由此可知, (7.26) 左边可以用

$$\frac{1}{N-m} \sum_{n=m+1}^N v_n^{m-1} w_{n-m}^{m-1} \quad (7.27)$$

来近似。另一方面, 由 (B.4) 可得

$$\begin{aligned}C_0 - \sum_{j=1}^{m-1} a_j^{m-1} C_j &= E \left[\left(y_{n-m} - \sum_{j=1}^{m-1} a_j^{m-1} y_{n-m+j} \right) y_{n-m} \right] \\ &= E [w_{n-m}^{m-1} y_{n-m}] \\ &= E [(w_{n-m}^{m-1})^2]\end{aligned}\quad (7.28)$$

但考虑到 $E[(w_{n-m}^{m-1})^2] = E[(v_n^{m-1})^2]$ 成立, 于是可由 (7.28) 导出如下几种不同的估计方法:

$$\frac{1}{N-m} \sum_{n=m+1}^N (w_{n-m}^{m-1})^2 \quad (7.29)$$

$$\frac{1}{N-m} \left\{ \sum_{n=m+1}^N (w_{n-m}^{m-1})^2 \sum_{n=m+1}^N (v_n^{m-1})^2 \right\}^{\frac{1}{2}} \quad (7.30)$$

$$\frac{1}{2(N-m)} \left\{ \sum_{n=m+1}^N (w_{n-m}^{m-1})^2 + \sum_{n=m+1}^N (v_n^{m-1})^2 \right\} \quad (7.31)$$

与之对应, 可以得到三种 PARCOR 的估计方法:

$$\hat{a}_m^m = \sum_{n=m+1}^N v_n^{m-1} w_{n-m}^{m-1} \left\{ \sum_{n=m+1}^N (w_{n-m}^{m-1})^2 \right\}^{-1} \quad (7.32)$$

$$\hat{a}_m^m = \sum_{n=m+1}^N v_n^{m-1} w_{n-m}^{m-1} \left\{ \sum_{n=m+1}^N (w_{n-m}^{m-1})^2 \sum_{n=m+1}^N (v_n^{m-1})^2 \right\}^{-\frac{1}{2}} \quad (7.33)$$

$$\hat{a}_m^m = 2 \sum_{n=m+1}^N v_n^{m-1} w_{n-m}^{m-1} \left\{ \sum_{n=m+1}^N (w_{n-m}^{m-1})^2 + \sum_{n=m+1}^N (v_n^{m-1})^2 \right\}^{-1} \quad (7.34)$$

此外, 还可以考虑在 (7.32) 中用 $(v_n^{m-1})^2$ 替换 $(w_{n-m}^{m-1})^2$ 的形式。由 (7.32) 得到的 PARCOR 估计值, 是将前向预测误差 v_n^{m-1} 对后向预测误差 w_{n-m}^{m-1} 作回归时的回归系数。另外, (7.33) 的估计值是 v_n^{m-1} 与 w_{n-m}^{m-1} 的相关系数, 与 PARCOR 的定义非常吻合。进一步地, (7.34) 的估计值相当于最小化预测误差方差与后向预测误差方差平均值, 因此可得到基于最大熵方法 (MEM) 的 Burg 算法。

为了利用这种 PARCOR 方法实际从时间序列 y_1, \dots, y_N 估计 AR 模型, 只需遵循下述步骤。这里假定 y_n 的均值为 0。

1. 令 $v_n^0 = w_n^0 = y_n$, ($n = 1, \dots, N$)。此外, 对于 0 阶模型, 令 $\hat{\sigma}_0^2 = N^{-1} \sum_{n=1}^N y_n^2$, 并令 $\text{AIC}_0 = N(\log 2\pi \hat{\sigma}_0^2 + 1) + 2$ 。
2. 对于 $m = 1, \dots, M$
 - (a) 用 (7.32)、(7.33)、(7.34) 中的任一种方法估计 PARCOR \hat{a}_m^m 。
 - (b) 由 (7.23) 求得 $\hat{a}_1^m, \dots, \hat{a}_{m-1}^m$ 。
 - (c) 对于 $n = m+1, \dots, N$, 由 $v_n^m = v_n^{m-1} - \hat{a}_m^m w_{n-m}^{m-1}$ 求得前向预测误差。
 - (d) 对于 $n = m+1, \dots, N$, 由 $w_{n-m}^m = w_{n-m}^{m-1} - \hat{a}_m^m v_n^{m-1}$ 求得后向预测误差。
 - (e) 由 $\hat{\sigma}_m^2 = \hat{\sigma}_{m-1}^2 \{1 - (\hat{a}_m^m)^2\}$ 估计 m 阶 AR 模型的分散。
 - (f) 由 $\text{AIC}_m = N(\log 2\pi \hat{\sigma}_m^2 + 1) + 2(m+1)$ 求得 AIC。

7.5 AR 系数估计量的误差分布

在假设时间序列服从阶数为 m 的 AR 模型的前提下, 当样本数 n 较大时, AR 参数估计值的分布近似为

$$\hat{a}_j \sim N(a_j, n^{-1} \sigma^2 \Sigma) \quad (7.35)$$

其中, Σ 是由自协方差函数生成的 (7.2) 中的 Toeplitz 矩阵, σ^2 是创新项的方差。另一方面, 当时间序列服从阶数为 m 的 AR 模型, 且 $j > m$ 时, PARCOR 的估计值 \hat{a}_j^j ,

表 7.1 针对太阳黑子数据, 不同阶 AR 模型的方差及 AIC

m	σ_m^2	AIC_m	m	σ_m^2	AIC_m	m	σ_m^2	AIC_m
0	0.22900	317.05	7	0.06694	46.95	14	0.05766	26.45
1	0.09204	108.49	8	0.06573	44.73	15	0.05716	26.47
2	0.07058	49.17	9	0.05984	25.02	16	0.05701	27.84
3	0.06959	47.90	10	0.05829	20.96	17	0.05701	29.84
4	0.06868	46.85	11	0.05793	21.52	18	0.05669	30.53
5	0.06815	47.08	12	0.05780	23.02	19	0.05661	32.21
6	0.06805	48.72	13	0.05766	24.47	20	0.05615	32.32

即阶数为 j ($j > m$) 的 AR 模型中的第 j 个自回归系数, 近似相互独立, 其方差为 $1/n$ 。因此,

$$\text{Var}(\hat{a}_j) \simeq \frac{1}{n} \quad \text{for } j > m \quad (7.36)$$

成立。这一性质可以用于检验所估计阶数的合理性。关于 AIC 与 PARCOR 估计值之间的关系, 将在章末习题 1 中讨论。

例: 太阳黑子数据的 AR 建模 R 包 TSSS 中的函数 `arfit` 基于 AIC 对 AR 模型进行估计。需要指定如下参数:

`lag`: AR 模型的最大阶数。默认值为 $2\sqrt{N}$, 其中 N 为数据个数。

`method`: AR 系数的估计方法:

- 1: Yule-Walker 方法
- 2: 最小二乘法 (Householder 方法)
- 3: PARCOR 方法 (偏相关系数)
- 4: PARCOR 方法 (PARCOR)
- 5: PARCOR 方法 (Burg 算法)

该函数的输出包

括:

`sigma2`: 创新项的方差
`maice.order`: 使 AIC 最小的阶数
`aic`: 各阶 AR 模型的 AIC
`arcoef`: 各阶 AR 模型的 AR 系数
`parcor`: PARCOR
`spec`: AIC 最优 AR 模型的功率谱 (对数尺度)

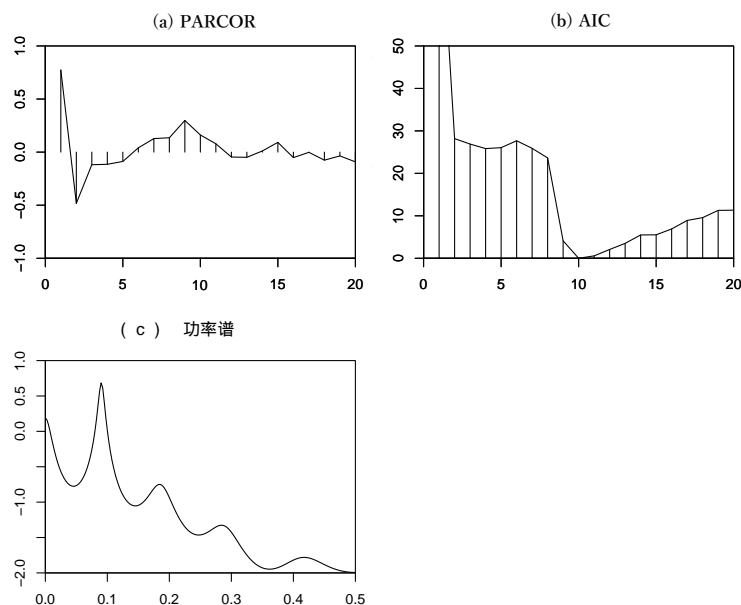


图 7.1 不同阶 AR 模型

7.6 数值例

```
> data( Sunspot )
> arfit( log10(Sunspot), lag=20, method=1 )
```

表 7.1 汇总了对图 1.1(b) 中的太阳黑子数据的对数值应用 Yule-Walker 方法拟合至 20 阶 AR 模型的结果。AIC 在 $m = 10$ 时取得最小值，此后逐渐增大。此外，图 7.1(a) 给出了估计得到的 PARCOR 值，(b) 为 AIC 值的变化图，(c) 则是采用 AIC 最优阶数 $m = 10$ 的 AR 模型所估计得到的谱。可以看到在 $f = 0.09$ 附近存在明显峰值，对应约 11 年的周期。

```
> arfit( HAKUSAN[,1] )
> arfit( log10(Sunspot) )
> arfit( Temperature )
> arfit( BLSALLFOOD )
> arfit( WHARD )
> arfit( MYE1F, lag=60 )
```

图 7.2 展示了对图 1.1 中的数据应用 Yule-Walker 方法拟合 AR 模型时，PARCOR 与 AIC 的变化，以及 AIC 最优模型所估计的功率谱。对于方向角速度数据，选择了

AR(17), 其功率谱在 $f = 0.8, 1.2, 1.7$ 附近出现峰值。对于最高气温数据, 选择了 AR(12), 在 $f = 0$ 处对应年周期的功率非常显著, 而其他频率处没有明显峰值。对于食品产业从业人数数据, 选择了 AR(15), 在 $f = 1/12$ 及其整数倍处出现非常尖锐的峰值, 对应季节性变化。对于批发额数据, 选择了 AR(13), 同样可见与季节变化对应的峰值。对于地震波数据, 选择了 AR(41), 其功率谱中存在多个峰值。

7.7 用 Yule-Walker 方法估计多变量 AR 模型

以下说明多变量 AR 模型的估计方法。本节中用 k 表示变量的个数 (维数)。 m 阶多变量 AR 模型

$$y_n = \sum_{i=1}^m A_i^m y_{n-i} + v_n, \quad v_n \sim N(0, V_m) \quad (7.37)$$

的参数是 AR 系数矩阵 A_1^m, \dots, A_m^m 以及残差 v_n 的方差协方差矩阵 V_m 。给定多变量 AR 模型后, 可以由 (6.28)、(6.29) 求得互协方差函数; 反过来也可以利用这些方程, 根据样本互协方差函数估计多变量 AR 模型的参数。

实际上, 与一元 AR 模型的情况类似, 也可以通过下面的算法高效地求得估计值。不过, 在该算法中, 除 (7.37) 之外, 还要考虑后向的多变量 AR 模型

$$y_n = \sum_{i=1}^m B_i^m y_{n+i} + u_n, \quad u_n \sim N(0, U_m) \quad (7.38)$$

不仅需要同时估计 A_i^m 和 V_m , 还需要同时估计 B_i^m 和 U_m 。这是因为, 在一元时间序列的情况下, 由于自协方差函数是偶函数, 前向 AR 模型与后向 AR 模型是一致的; 但在多变量时间序列的情况下, 这一性质不再成立。

1. $\hat{V}_0 = \hat{U}_0 = C_0$

$$\text{AIC}_0 = N(k \log 2\pi + \log |\hat{V}_0| + k) + k(k+1)$$

2. 对于 $m = 1, \dots, M$

- (a) $W_m = C_m - \sum_{i=1}^{m-1} A_i^{m-1} C_{m-i}$

- (b) $A_m^m = W_m U_{m-1}^{-1}$

$$B_m^m = W_m^T V_{m-1}^{-1}$$

- (c) $A_i^m = A_i^{m-1} - A_m^m B_{m-i}^{m-1}$

$$B_i^m = B_i^{m-1} - B_m^m A_{m-i}^{m-1}$$

- (d) $V_m = C_0 - \sum_{i=1}^m A_i^m C_i^T$

$$U_m = C_0 - \sum_{i=1}^m B_i^m C_i$$

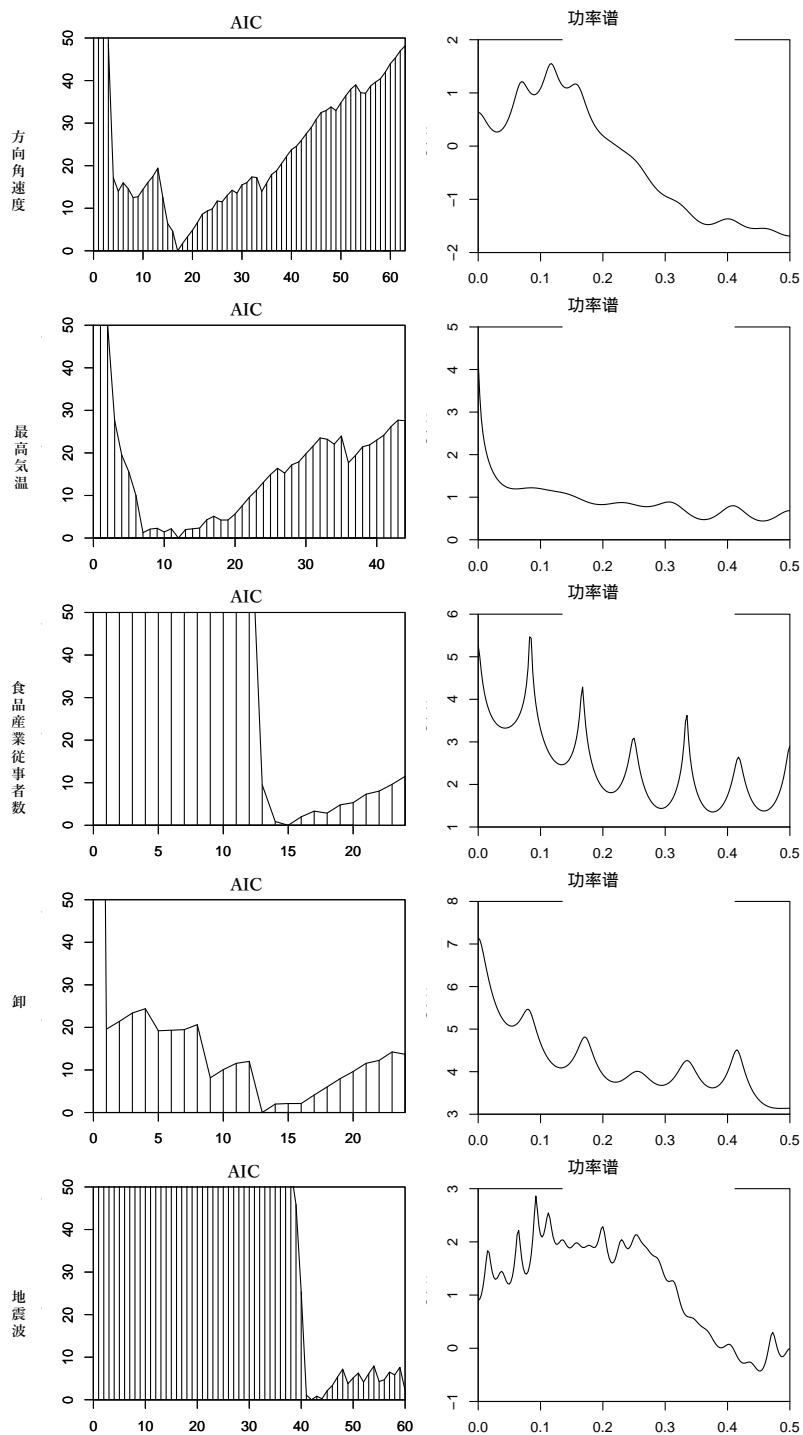


图 7.2 AR 模型的 AIC 变化及估计得到的功率谱

$$(e) AIC_m = N(k \log 2\pi + \log |\hat{V}_m| + k) + k(k+1) + 2k^2m$$

按照上述算法计算 AIC_0, \dots, AIC_M , 并将使其取最小值的 m 作为多变量 AR 模型的阶数。因此, 在这种方法中, 估计得到的是一个对所有 i, j 而言, 系数 $a_m(i, j)$ 都具有相同阶数的模型。

例 包 TSSS 中的函数 R 用 Yule-Walker 方法估计多变量 AR 模型, 并通过 AIC 最小化自动确定阶数。默认的最大 AR 阶数为 $2\sqrt{N}$, 但也可以通过参数 lag 显式指定最大阶数。该函数的输出为:

```
maice.order: 使 AIC 最小的阶数。
aic:         阶数 0, ..., lag 的 AR 模型的 AIC。
arcoef:      各阶 AR 模型的 AR 系数矩阵。
v:          AIC 最优模型的方差协方差矩阵。
```

```
> # Yaw rate, Pitching, Rudder angle
> data( HAKUSAN )
> yy <- as.matrix( HAKUSAN[, c(1,2,4)] )
> nc <- dim( yy )[1]
> n <- seq( 1, nc, by = 2 )
> y <- yy[n, ]
> marfit( y, lag=20 )
```

表 7.2 汇总了对图 1.1 中的船舶数据应用 Yule-Walker 方法拟合到 20 阶的 3 变量 AR 模型所得的结果。AIC 在 $m = 10$ 时最小, 此后逐渐增大。由 AIC 最小的 10 阶多变量 AR 模型估计得到的功率谱、互谱、相干性以及噪声贡献率等结果已在第 6 章中给出。

表 7.2 多变量 AR 模型的 AIC

m	AIC_m	m	AIC_m	m	AIC_m
0	7091.71	7	5105.83	14	5100.98
1	6238.80	8	5096.35	15	5113.05
2	5275.36	9	5087.91	16	5116.52
3	5173.02	10	5083.79	17	5129.42
4	5135.20	11	5093.79	18	5136.06
5	5136.63	12	5091.42	19	5143.56
6	5121.02	13	5097.98	20	5157.37

7.8 用最小二乘法估计多变量 AR 模型

为了通过基于 Householder 变换的最小二乘法求得多变量 AR 模型的最小二乘解, 首先将 (7.37) 变形为如下包含同一时刻响应的模型形式会比较方便:

$$y_n = B_0 y_n + \sum_{i=1}^m B_i y_{n-i} + w_n, \quad w_n \sim N(0, W) \quad (7.39)$$

其中, 矩阵 B_0 是一个对角线及其以上元素全为 0 的下三角矩阵

$$B_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ b_0(2,1) & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ b_0(k,1) & \cdots & b_0(k,k-1) & 0 \end{bmatrix} \quad (7.40)$$

而方差协方差矩阵 W 是一个对角矩阵

$$W = \begin{bmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \ddots & \\ 0 & & & \sigma_k^2 \end{bmatrix} \quad (7.41)$$

由 (7.39) 可得

$$y_n = (I - B_0)^{-1} \sum_{i=1}^m B_i y_{n-i} + (I - B_0)^{-1} w_n \quad (7.42)$$

因此该模型可通过

$$\begin{aligned} A_i &= (I - B_0)^{-1} B_i \\ V &= (I - B_0)^{-1} W (I - B_0)^{-T} \end{aligned} \quad (7.43)$$

与多变量 AR 模型 (7.35) 建立一一对应关系。

因此, 只要能够估计模型 (7.39) 的系数 B_0, B_1, \dots, B_m 以及 $\sigma_1^2, \dots, \sigma_k^2$, 就可以通过 (7.43) 确定多变量 AR 模型。这种方法的优点在于, 由于方差协方差矩阵 W 是对角矩阵, 因此不需要同时估计全部系数。若将各系数矩阵 B_i 表示为

$$B_i = \begin{bmatrix} b_i(1,1) & \cdots & b_i(1,k) \\ \vdots & \ddots & \vdots \\ b_i(k,1) & \cdots & b_i(k,k) \end{bmatrix} \quad (7.44)$$

则可以将其分解为 k 个模型, 即 $\{b_i(p,q), i=1, \dots, m, q=1, \dots, k, \sigma_p^2\}$ ($p=1, \dots, k$) 独立地进行估计。与一次同时估计所有参数的方法相比, 这种方法效率要高得多。

为了利用 Householder 方法估计该模型, 首先构造 $(N-m) \times (km+k)$ 矩阵

$$X = \begin{bmatrix} y_m^T & \cdots & y_1^T & y_{m+1}^T \\ y_{m+1}^T & \cdots & y_2^T & y_{m+2}^T \\ \vdots & \ddots & \vdots & \vdots \\ y_{N-1}^T & \cdots & y_{N-m}^T & y_N^T \end{bmatrix} \quad (7.45)$$

并通过 Householder 变换将其化为上三角矩阵

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1,km+k} \\ & \ddots & \vdots \\ & & s_{km+k,km+k} \\ & & & \mathbf{O} \end{bmatrix} \quad (7.46)$$

此时,左上角的 $(km+1) \times (km+1)$ 子矩阵包含了估计第 1 个分量模型

$$y_n(1) = \sum_{i=1}^j b_i(1,1)y_{n-i}(1) + \cdots + \sum_{i=1}^j b_i(1,k)y_{n-i}(k) + w_n \quad (7.47)$$

所需的全部信息。也就是说,当 $j \leq m$ 时, j 阶模型的残差方差以及 AIC 可由

$$\hat{\sigma}_j^2(1) = \frac{1}{N-m} \sum_{i=kj+1}^{M+1} s_{i,km+1}^2$$

$$\text{AIC}_j(1) = (N-m)(\log 2\pi \hat{\sigma}_j^2(1) + 1) + 2(kj+1) \quad (7.48)$$

计算得到。此外,若将回归系数按 $c = (b_1(1,1), \dots, b_1(1,k), \dots, b_j(1,1), \dots, b_j(1,k))^T$ 排列,则它们可由线性方程组

$$\begin{bmatrix} s_{11} & \cdots & s_{1,kj} \\ & \ddots & \vdots \\ \mathbf{O} & & s_{kj,kj} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_{kj} \end{bmatrix} = \begin{bmatrix} s_{1,km+1} \\ \vdots \\ s_{kj,km+1} \end{bmatrix} \quad (7.49)$$

的解得到,而该方程组可通过回代法方便地求解。

接下来,为了估计第 2 个分量的模型,通过适当的 Householder 变换,将矩阵变换为

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1,km} & s_{1,km+1} & s_{1,km+2} & \cdots & s_{1,km+k} \\ s_{21} & \cdots & s_{2,km} & & s_{2,km+2} & \cdots & s_{2,km+k} \\ & \ddots & \vdots & & \vdots & & \vdots \\ & & s_{km+1,km} & & s_{km+1,km+2} & \cdots & s_{km+1,km+k} \\ & & & & s_{km+2,km+2} & \cdots & s_{km+2,km+k} \\ & & & & & \ddots & \vdots \\ & & & & & & s_{km+k,km+k} \\ & & & & & & \mathbf{O} \end{bmatrix} \quad (7.50)$$

的形式,则该矩阵左上角的 $(km+2) \times (km+2)$ 子矩阵包含了估计第 2 个分量模型所需的全部信息。当 $j \leq m$ 时, j 阶模型的残差方差及 AIC 可由

$$\hat{\sigma}_j^2(2) = \frac{1}{N-m} \sum_{i=kj+2}^{km+2} s_{i,km+2}^2$$

$$\text{AIC}_j(2) = (N-m)(\log 2\pi \hat{\sigma}_j^2(2) + 1) + 2(kj+2) \quad (7.51)$$

计算得到。此外,为求得回归系数,只需解关于 $kj+1$ 维向量 $c =$

$(b_1(2,1), \dots, b_1(2,k), \dots, b_j(2,1), \dots, b_j(2,k), b_0(2,1))^T$ 的线性方程组

$$\begin{bmatrix} s_{11} & \cdots & s_{1,kj} & s_{1,km+1} \\ s_{21} & \cdots & s_{2,kj} & \\ & \ddots & \vdots & \\ & & s_{kj+1,kj} & \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{kj+1} \end{bmatrix} = \begin{bmatrix} s_{1,km+2} \\ s_{2,km+2} \\ \vdots \\ s_{kj+1,km+2} \end{bmatrix} \quad (7.52)$$

即可。

同样地重复这一过程, 为了估计第 k 个分量的模型, 只需通过适当的 Householder 变换求得

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1,km} & s_{1,km+1} & \cdots & s_{1,km+k-1} & s_{1,km+k} \\ \vdots & & \vdots & & \ddots & \vdots & \vdots \\ s_{k-1,1} & \cdots & s_{k-1,km} & & & s_{k-1,km+k-1} & s_{k-1,km+k} \\ s_{k1} & \cdots & s_{k,km} & & & & s_{k,km+k} \\ & \ddots & \vdots & & & & \vdots \\ & & s_{km+k-1,km} & & & & s_{km+k-1,km+k} \\ & & & 0 & & & s_{km+k,km+k} \end{bmatrix} \quad (7.53)$$

当 $j \leq m$ 时, j 阶模型的残差方差及 AIC 可由

$$\hat{\sigma}_j^2(k) = \frac{1}{N-m} \sum_{i=kj+k}^{km+k} s_{i,km+k}^2$$

$$\text{AIC}_j(k) = (N-m)(\log 2\pi \hat{\sigma}_j^2(k) + 1) + 2(kj+k) \quad (7.54)$$

计算得到。此外, 该模型的回归系数可由关于 $kj+k-1$ 维向量 $c = (b_1(k,1), \dots, b_1(k,k), \dots, b_j(k,1), \dots, b_j(k,k), b_0(k,1), \dots, b_0(k, k-1))^T$ 的线性方程组

$$\begin{bmatrix} s_{11} & \cdots & s_{1,kj} & s_{1,km+1} & \cdots & s_{1,q-1} \\ \vdots & & \vdots & & \ddots & \vdots \\ s_{k-1,1} & \cdots & s_{k-1,kj} & & & s_{k-1,q-1} \\ s_{k1} & \cdots & s_{k,kj} & & & \\ & \ddots & \vdots & & & \\ & & s_{r,kj} & & & \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_{k-1} \\ c_k \\ \vdots \\ c_r \end{bmatrix} = \begin{bmatrix} s_{1,q} \\ \vdots \\ s_{k-1,q} \\ s_{k,q} \\ \vdots \\ s_{r,q} \end{bmatrix} \quad (7.55)$$

的解得到。这里, $q = km+k$, $r = kj+k-1$ 。

在这种基于最小二乘法的估计方法中, 可以对不同变量使用不同的阶数, 因此比 Yule-Walker 方法能进行更灵活的建模。此外, 还可以假定变量之间的响应具有时间滞后, 或者假定某些特定系数为 0 来进行模型估计; 关于这种更复杂模型估计的程序, 可参见参考文献 [2]。

包 TSSS 中的函数 `marlsq` 利用 Householder 方法的最小二乘法来估计多变量 AR 模型。AR 阶数的最大值可以通过参数 `lag` 指定。如果未显式指定, 则使用默认值

$2\sqrt{N}$ 。其中, N 为数据个数。该函数的输出为:

`maice.order`: 使 AIC 最小的阶数。
`aic`: 阶数 $0, \dots, \text{lag}$ 的 AR 模型的 AIC。
`arcoef`: 各阶 AR 模型的 AR 系数矩阵。
`v`: AIC 最优模型的方差协方差矩阵。

```
> y <- as.matrix( HAKUSAN )
> z <- marlsq( y, lag=10 )
> z
> marspc( z$arcoef, v = z$v )
```

例: 下面给出对由方向角速度、横摇、纵摇和舵角组成的 4 变量船舶数据, 在 $\text{lag}=10$ 时拟合多变量 AR 模型的结果。于阶数 $m=10$ 时得到最小 AIC 6575.741。利用估计得到的 AR 模型的系数矩阵 `arcoef` 以及创新项的方差协方差矩阵 `v`, 可以通过函数 `marspc` 计算互谱、相干性以及功率贡献率。图 7.3 给出了利用所估计的多变量 AR 模型计算得到的功率贡献率。从上到下依次表示方向角速度、横摇、纵摇和舵角的功率贡献率。左侧是功率谱的分解, 右侧是相对功率贡献率。在各图中, 自下而上的 4 个区域依次表示方向角速度、横摇、纵摇和舵角的贡献。从这些图中可以读出: 纵摇对方向角速度的贡献、方向角速度和舵角对横摇的贡献、方向角速度对纵摇的贡献, 以及方向角速度对舵角的贡献。

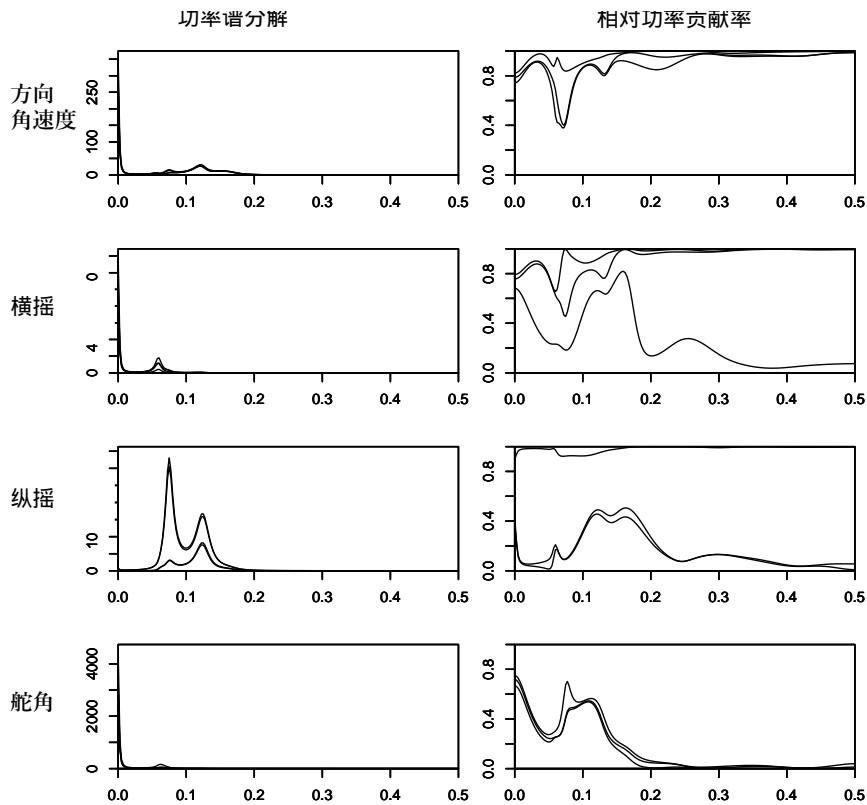


图 7.3 针对 4 变量船舶数据, 由多变量 AR 模型得到的功率贡献率

● 章末习题

- 将 m 阶 AR 模型记为 $y_n = a_1^m y_{n-1} + \cdots + a_m^m y_{n-m} + v_n$, $v_n \sim N(0, \sigma_m^2)$.
 - 利用 $\sigma_m^2 = (1 - (a_m^m)^2) \sigma_{m-1}^2$ 成立这一事实, 说明在什么条件下可以判断 $AR(m)$ 比 $AR(m-1)$ 更好。
 - 设用 100 个数据计算 PARCOR (AR 模型的系数 a_j^j) 时得到 $a_1^1 = 0.9$, $a_2^2 = -0.6$, $a_3^3 = 0.3$, $a_4^4 = -0.1$, $a_5^5 = 0.15$ 。当 $C_0 = 1$ 时, 计算 σ_m^2 , $m = 1, \dots, 5$ 。
 - 对于问 (2) 的情形, 计算 AIC_m , 并确定最优阶数。
- 说明 Yule-Walker 方法、最小二乘法 and PARCOR 方法之间的区别。
- 设想一种利用 AR 模型判断两个时间序列 x_n 和 y_n 是否独立的方法。

第 8 章

局部平稳 AR 模型

现实中的时间序列很多都是非平稳的。对非平稳时间序列进行建模最简单的方法是把时间区间划分为若干适当的小区间,并在每个小区间内假设其为平稳。通过在各区间拟合 AR 模型,就可以得到一种近似表示非平稳时间序列的模型。本章将说明一种自动粗略决定区间数与分割点的方法,以及对分割点进行精确决定的方法。关于系数随时间变化的时变系数 AR 模型将在第 13 章说明。

8.1 局部平稳 AR 模型

假设时间序列 y_1, \dots, y_N 在整体区间上并非平稳,但若将其划分为适当的小区间,则在各小区间内可以视为平稳。这样按区间分段满足平稳性的时间序列称为局部平稳时间序列。为了具体说明,设划分的区间个数为 k , 各区间的数据个数为 N_i , ($N_1 + \dots + N_k = N$)。但在实际建模中, k 与 N_i 的值是未知的。因此,在局部平稳时间序列的分析中,需要估计分割点的个数与位置,以及各区间内的模型。

局部平稳 AR 模型 (locally stationary AR model) 是假设在每个小区间内,时间序列都可以用 AR 模型表示的模型。先将第 i 个小区间记为 $[n_{i0}, n_{i1}]$ 。 n_{i0} 与 n_{i1} 由

$$n_{i0} = \sum_{j=1}^{i-1} N_j + 1, \quad n_{i1} = \sum_{j=1}^i N_j$$

给出。局部平稳 AR 模型假设: 在第 j 个小区间内, 时间序列 y_n 服从 AR 模型

$$y_n = \sum_{i=1}^{m_j} a_{ji} y_{n-i} + v_{nj} \quad (8.1)$$

其中, v_{nj} 为满足 $E[v_{nj}] = 0$, $E[v_{nj}^2] = \sigma_j^2$, $E[v_{nj} y_{n-m}] = 0$ 的白噪声。

局部平稳 AR 模型的似然为

$$L = p(y_1, \dots, y_N) = \prod_{j=1}^k \prod_{n=n_{j0}}^{n_{j1}} p(y_n | y_1, \dots, y_{n-1}) \quad (8.2)$$

因此, 与 AR 模型的最小二乘法情形相同, 忽略最初的 m_1 个数据的分布, 把 N_1 用 $N_1 - m_1$ 代替, 并把 n_{10} 用 $m_1 + 1$ 代替, 则该模型的似然近似为

$$\prod_{j=1}^k \left(\frac{1}{2\pi\sigma_j^2} \right)^{\frac{N_j}{2}} \exp \left\{ -\frac{1}{2\sigma_j^2} \sum_{n=n_{j0}}^{n_{j1}} \left(y_n - \sum_{i=1}^{m_j} a_{ji} y_{n-i} \right)^2 \right\} \quad (8.3)$$

将此似然视为小区间数 k 、小区间长度 N_j 、AR 模型阶数 m_j 、自回归系数 $a_j = (a_{j1}, \dots, a_{jm_j})^T$ 以及白噪声方差 σ_j^2 的函数, 则对数似然为

$$\begin{aligned} & \ell(k, N_j, m_j, a_j, \sigma_j^2; (j=1, \dots, k)) \\ &= -\frac{1}{2} \sum_{j=1}^k \left\{ N_j \log 2\pi\sigma_j^2 + \frac{1}{\sigma_j^2} \sum_{n=n_{j0}}^{n_{j1}} \left(y_n - \sum_{i=1}^{m_j} a_{ji} y_{n-i} \right)^2 \right\} \end{aligned} \quad (8.4)$$

对其关于 σ_j^2 偏微分并令其为 0, 则对任意 a_j , 使对数似然达到最大值的 σ_j^2 为

$$\hat{\sigma}_j^2 = \frac{1}{N_j} \sum_{n=n_{j0}}^{n_{j1}} \left(y_n - \sum_{i=1}^{m_j} a_{ji} y_{n-i} \right)^2 \quad (8.5)$$

进一步把它代入 (8.4), 得到对数似然为

$$\begin{aligned} & \ell(k, N_j, m_j, a_j, \hat{\sigma}_j^2; (j=1, \dots, k)) \\ &= -\frac{1}{2} \sum_{j=1}^k (N_j \log 2\pi\hat{\sigma}_j^2 + N_j) \\ &= -\frac{N-m_1}{2} (\log 2\pi + 1) - \frac{1}{2} \sum_{j=1}^k N_j \log \hat{\sigma}_j^2 \end{aligned} \quad (8.6)$$

因此, a_{j1}, \dots, a_{jm_j} 的极大似然估计值可以用第 5 章的最小二乘法通过最小化 $\hat{\sigma}_j^2$ 得到。

由于第 j 个 AR 模型以 m_j 个 AR 系数与方差为参数, 所以局部平稳 AR 模型的 AIC 为

$$\text{AIC} = (N - m_1)(\log 2\pi + 1) + \sum_{j=1}^k N_j \log \hat{\sigma}_j^2 + 2 \sum_{j=1}^k (m_j + 1) \quad (8.7)$$

小区间数 k 、小区间长度 N_j 、AR 模型阶数 m_j 只需在各种可能性中寻找使 AIC 最小者即可。

8.2 自动划分为任意个区间

如前节所示, 原则上利用最小二乘法与 AIC 即可估计最优的局部平稳 AR 模型。然而, 要对局部平稳模型的区间数 k 以及小区间长度 N_1, \dots, N_k 的所有可能组合都进行尝试并寻找使 AIC 最小者, 需要极其庞大的计算量, 在实时使用时并不现实。因此, 先预先规定划分的最小单位 L , 并仅将 $n_i = iL$ 作为分割点候选。

此时, 可以通过如下步骤自动决定局部平稳 AR 模型的分割点。

1. 决定假设平稳的小区间长度 L , 以及各区间拟合的 AR 模型的最高阶数 m 。
其中, L 应足够长, 以便能够拟合 m 阶 AR 模型。
2. 对 y_1, \dots, y_L 分别拟合 0 阶到 m 阶的 AR 模型, 求 $AIC_0(0), \dots, AIC_0(m)$, 并令 $AIC_0 = \min_j AIC_0(j)$ 。同时置 $k = 1, n_{10} = m + 1, n_{11} = L, N_1 = L - m$ 。
3. 对 $y_{n_{k1}+1}, \dots, y_{n_{k1}+L}$ 分别拟合到 m 阶的 AR 模型, 求 $AIC_1(0), \dots, AIC_1(m)$, 并令 $AIC_1 = \min_j AIC_1(j)$ 。 AIC_1 是假设在时刻 $n_{k1} + 1$ 模型发生变化时的新模型 AIC。将区间 $[n_{k0}, n_{k1} + L]$ 分割为两个小区间 $[n_{k0}, n_{k1}]$ 与 $[n_{k1} + 1, n_{k1} + L]$ 的局部平稳 AR 模型 AIC 为

$$AIC_D = AIC_0 + AIC_1$$

把该模型称为**分割模型** (divided model)。

4. 把 $y_{n_{k0}}, \dots, y_{n_{k1}+L}$ 视为一个区间并拟合到 m 阶的 AR 模型, 计算 $AIC_P(0), \dots, AIC_P(m)$, 并令 $AIC_P = \min_j AIC_P(j)$ 。该模型是不在时刻 $n_{k1} + 1$ 处分割、而将区间 $[n_{k0}, n_{k1} + L]$ 视为一个小区间的模型 AIC, 称为**合并模型** (pooled model)。
5. 比较 AIC 值以判断 (3) 与 (4) 哪个模型更好。
 - (a) 当 $AIC_D < AIC_P$ 时, 判断分割模型更好。因此, $n_{k1} + 1$ 成为新的分割点: 先置 $k \equiv k + 1$, 再置 $n_{k0} \equiv n_{k-1,1} + 1, n_{k1} = n_{k-1,1} + L, N_k = L, AIC_0 = AIC_D$ 。
 - (b) 当 $AIC_D \geq AIC_P$ 时, 采用合并模型: 将新的小区间 $[n_{k1} + 1, n_{k1} + L]$ 并入原小区间, 得到 $[n_{k0}, n_{k1} + L]$ 。因此, 置 $n_{k1} \equiv n_{k1} + L, N_k = N_k + L, AIC_0 = AIC_P$ 。
6. 若还有不少于 L 个数据, 则返回步骤 (3)。否则, k 为分割数, $[1, n_{11}], [n_{20}, n_{21}], \dots, [n_{k0}, N]$ 给出 k 个分段。

实现该步骤时, 每获得新的 L 个数据都需要拟合两类 AR 模型, 但利用 5.4 节的数据追加方法可以高效计算。

首先在步骤 (2) 中, 从 y_1, \dots, y_L 构造 $(L - m) \times (m + 1)$ 矩阵

$$X_0 = \begin{bmatrix} y_m & \cdots & y_1 & y_{m+1} \\ \vdots & \ddots & \vdots & \vdots \\ y_{L-1} & \cdots & y_{L-m} & y_L \end{bmatrix} \quad (8.8)$$

并用豪斯霍尔德变换 H_0 将其缩约为上三角矩阵

$$H_0 X_0 = \begin{bmatrix} S \\ O \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots & s_{1m} & s_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & s_{mm} & s_{m,m+1} \\ 0 & & & s_{m+1,m+1} \end{bmatrix} \quad (8.9)$$

则

$$\hat{\sigma}_0^2(j) = \frac{1}{L-m} \sum_{i=j+1}^{m+1} s_{i,m+1}^2 \quad (8.10)$$

$$\text{AIC}_0(j) = (L-m) \log \hat{\sigma}_0^2(j) + 2(j+1) \quad (8.11)$$

即可计算拟合到 y_1, \dots, y_L 的 j 阶 AR 模型的 AIC。以下本节中, 常数项 $(L-m)(\log 2\pi + 1)$ 与模型选择无关, 因此省略。

为执行步骤 (3), 从 $y_{n_{k1}+1}, \dots, y_{n_{k1}+L}$ 构造 $L \times (m+1)$ 矩阵

$$X_1 = \begin{bmatrix} y_{n_{k1}} & \cdots & y_{n_{k1}-m+1} & y_{n_{k1}+1} \\ \vdots & \ddots & \vdots & \vdots \\ y_{n_{k1}+L-1} & \cdots & y_{n_{k1}+L-m} & y_{n_{k1}+L} \end{bmatrix} \quad (8.12)$$

并用豪斯霍尔德变换 H_1 缩约为上三角矩阵

$$H_1 X_1 = \begin{bmatrix} R \\ O \end{bmatrix} = \begin{bmatrix} r_{11} & \cdots & r_{1m} & r_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & r_{mm} & r_{m,m+1} \\ & & & r_{m+1,m+1} \\ & 0 & & \end{bmatrix} \quad (8.13)$$

则同步骤 (2) 一样,

$$\hat{\sigma}_1^2(j) = \frac{1}{L} \sum_{i=j+1}^{m+1} r_{i,m+1}^2 \quad (8.14)$$

$$\text{AIC}_1(j) = L \log \hat{\sigma}_1^2(j) + 2(j+1) \quad (8.15)$$

即可计算拟合到新的 L 个数据的 j 阶 AR 模型 AIC。此时

$$\text{AIC}_D \equiv \min_j \text{AIC}_0(j) + \min_j \text{AIC}_1(j) \quad (8.16)$$

就是假设在时刻 $n_{k1} + 1$ 发生结构变化时的局部平稳 AR 模型 AIC。

接着, 为在步骤 (4) 中对合并后的数据 $y_{n_{k0}}, \dots, y_{n_{k0}+L}$ 拟合 AR 模型, 把由前半数据 $y_{n_{k0}}, \dots, y_{n_{k1}}$ 得到的上三角矩阵 S 与步骤 (3) 得到的上三角矩阵 R 拼接, 构造 $2(m+1) \times (m+1)$ 矩阵

$$X_2 = \begin{bmatrix} S \\ R \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots & s_{1m} & s_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & s_{mm} & s_{m,m+1} \\ 0 & & & s_{m+1,m+1} \\ r_{11} & \cdots & r_{1m} & r_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & r_{mm} & r_{m,m+1} \\ 0 & & & r_{m+1,m+1} \end{bmatrix} \quad (8.17)$$

并用豪斯霍尔德变换缩约为上三角矩阵

$$H_2 X_2 = \begin{bmatrix} T \\ \mathbf{O} \end{bmatrix} = \begin{bmatrix} t_{11} & \cdots & t_{1m} & t_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & t_{mm} & t_{m,m+1} \\ & & & t_{m+1,m+1} \\ & & & 0 \end{bmatrix} \quad (8.18)$$

则

$$\hat{\sigma}_P^2(j) = \frac{1}{N_k + L} \sum_{i=j+1}^{m+1} t_{i,m+1}^2 \quad (8.19)$$

$$\text{AIC}_P(j) = (N_k + L) \log \hat{\sigma}_P^2(j) + 2(j+1) \quad (8.20)$$

即可计算拟合到 $y_{n_{k0}}, \dots, y_{n_{k1}+L}$ 的 j 阶 AR 模型 AIC。因此定义

$$\text{AIC}_P \equiv \min_j \text{AIC}_P(j) \quad (8.21)$$

就得到在假设时刻 $n_{k1} + 1$ 没有结构变化时拟合的 AR 模型 AIC。

在步骤 (5) 中, 当 $\text{AIC}_D < \text{AIC}_P$ 时把矩阵 T 替换为 S , 当 $\text{AIC}_D \geq \text{AIC}_P$ 时把矩阵 R 替换为 S , 然后返回步骤 (3) 即可。

例: 地震数据的局部平稳建模

TSSS 包的函数 `lsar` 用于估计局部平稳 AR 模型, 并自动把时间序列分割为在各小区间内可以视为平稳的小区间。需要以下参数:

`max.arorder` AR 模型的最大阶数
`ns0` 基本小区间的长度

该函数的输出为:

`model:` = 1: 采用合并模型时
 = 2: 采用切换模型时
`ns:` 局部平稳跨度的数据数
`span:` 局部平稳跨度的起点与终点
`nf:` 计算谱的频率点数
`ms:` 切换模型的阶数
`sds:` 切换模型的创新方差
`aics:` 切换模型的 AIC
`mp:` 合并模型的阶数
`sdp:` 合并模型的创新方差
`aicp:` 合并模型的 AIC
`spec:` 局部平稳谱

```
> data( MYE1F )
> lsar( MYE1F, max.arorder = 10, ns0 = 150 )
```

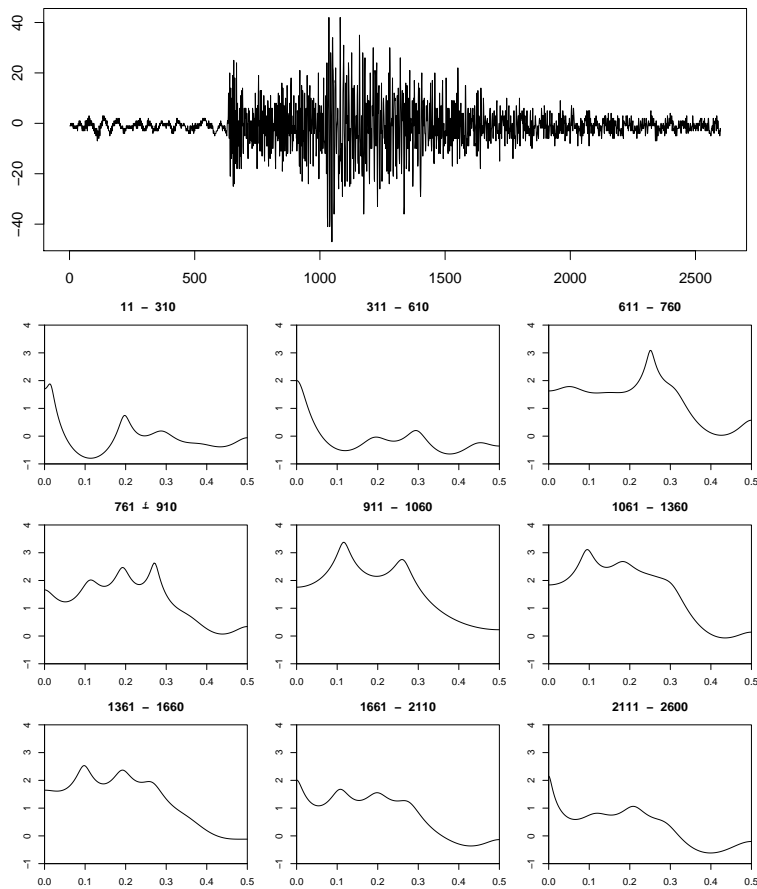


图 8.1 (上) 地震波东西向分量记录与 (下) 由局部平稳 AR 模型估计的局部平稳谱.

图 8.1 给出了对地震东西向分量记录 ($N = 2600$) 在 $L = 150$, $m = 10$ 的设定下拟合局部平稳 AR 模型的结果。记录中包含称为常微动的噪声部分, 以及 P 波与 S 波两种地震波。并且, 下方各图显示了由局部平稳 AR 模型判定为平稳的小区间范围, 以及在该区间估计得到的谱。在 $n = 310, 610, 760, 910, 1060, 1360, 1660, 2110$ 这 8 处检测到了变化。 $n = 610$ 附近的变化对应于 P 波到达所引起的方差与谱变化。 $n = 610 \sim 910$ 部分为 P 波, 其中 $610 \sim 760$ 段单一周期成分较强, 而后半段 $760 \sim 1060$ 则可见多种周期成分混在一起。 $n = 1060$ 之后为 S 波部分, 随着振幅减小, 不仅功率降低, 而且谱主要峰的位置从低频向高频侧移动。 $n = 2110$ 之后被认为没有变化。

8.3 变化时点的精密估计

到上一节为止, 我们给出了将非平稳时间序列自动分割为多个在各区间内可视为平稳的小区间的方法。本节假设已知时间序列 y_n 在某个区间 $[n_0, n_1]$ 内其概率结构发生变化, 讨论如何检测该结构变化的精确时刻。为此, 对满足 $n_0 \leq n \leq n_1$ 的各时刻 n , 都假设在该时刻发生结构变化, 把数据分成两个区间 $[1, n-1]$ 与 $[n, N]$, 分别拟合不同的 AR 模型并计算两者 AIC 之和, 从而评估假设“在时刻 n 变化”的模型优劣。因此, 要用局部平稳 AR 模型精确确定变化时点, 只需对 $n_0 \leq n \leq n_1$ 的所有 n 计算 AIC 并寻找取最小值的时刻即可。若每改变一次 n 都重新计算 AR 模型, 需要大量计算, 但此处同样可利用 5.4 节的数据追加方法高效计算。

首先, 由 y_1, \dots, y_{n_0} 构造 $(n_0 - m) \times (m + 1)$ 矩阵

$$X_0 = \begin{bmatrix} y_m & \cdots & y_1 & y_{m+1} \\ \vdots & \ddots & \vdots & \vdots \\ y_{n_0-1} & \cdots & y_{n_0-m} & y_{n_0} \end{bmatrix} \quad (8.22)$$

通过 Householder 变换 H_0 将其缩约为上三角矩阵

$$H_0 X_0 = \begin{bmatrix} S \\ O \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots & s_{1m} & s_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & s_{mm} & s_{m,m+1} \\ & & & s_{m+1,m+1} \\ & & & 0 \end{bmatrix} \quad (8.23)$$

则可通过

$$\hat{\sigma}_0^2(j) = \frac{1}{n_0 - m} \sum_{i=j+1}^{m+1} s_{i,m+1}^2 \quad (8.24)$$

$$\text{AIC}_0(j) = (n_0 - m) \log \hat{\sigma}_0^2(j) + 2(j + 1) \quad (8.25)$$

计算对 y_1, \dots, y_{n_0} 拟合 j 阶 AR 模型的 AIC。因此定义

$$\text{AIC}_0^1 \equiv \min_j \text{AIC}_0(j) \quad (8.26)$$

即为假设在时刻 $n_0 + 1$ 发生结构变化时前半部分 AR 模型的 AIC。

接下来, 为了对 $n_0 + p$ 个 (p 可以为 1) 数据 y_1, \dots, y_{n_0+p} 拟合 AR 模型, 只需在上面求得的上三角矩阵下方追加新数据

$$X_1 = \begin{bmatrix} s_{11} & \cdots & s_{1m} & s_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & s_{mm} & s_{m,m+1} \\ & & & s_{m+1,m+1} \\ y_{n_0} & \cdots & y_{n_0-m+1} & y_{n_0+1} \\ \vdots & & \vdots & \vdots \\ y_{n_0+p-1} & \cdots & y_{n_0-m+p} & y_{n_0+p} \end{bmatrix} \quad (8.27)$$

再通过 Householder 变换 H_1 缩约为上三角矩阵

$$H_1 X_1 = \begin{bmatrix} R \\ O \end{bmatrix} = \begin{bmatrix} r_{11} & \cdots & r_{1m} & r_{1,m+1} \\ & \ddots & \vdots & \vdots \\ & & r_{mm} & r_{m,m+1} \\ & & & r_{m+1,m+1} \\ & 0 & & \end{bmatrix} \quad (8.28)$$

则可通过

$$\hat{\sigma}_1^2(j) = \frac{1}{n_0 - m + p} \sum_{i=j+1}^{m+1} r_{i,m+1}^2$$

$$\text{AIC}_1(j) = (n_0 - m + p) \log \hat{\sigma}_1^2(j) + 2(j+1) \quad (8.29)$$

计算对 y_1, \dots, y_{n_0+p} 拟合 j 阶 AR 模型的 AIC。因此定义

$$\text{AIC}_1^1 \equiv \min_j \text{AIC}_1(j) \quad (8.30)$$

即为假设在时刻 $n_0 + p + 1$ 发生结构变化时前半部分 AR 模型的 AIC。以下同样反复进行, 即可计算对 $\{y_1, \dots, y_{n_0}\}, \{y_1, \dots, y_{n_0+p}\}, \dots, \{y_1, \dots, y_{n_1}\}$ 拟合的 AR 模型的 AIC, 即 $\text{AIC}_0^1, \text{AIC}_1^1, \dots, \text{AIC}_\ell^1$ 。

同样地, 先对 y_{n_1+1}, \dots, y_N 拟合 AR 模型, 然后依次追加 p 个数据 $\{y_{n_1-p+1}, \dots, y_{n_1}\}, \{y_{n_1-2p+1}, \dots, y_{n_1-p}\}, \dots$, 即可求得后半部分模型的 AIC, 即 $\text{AIC}_\ell^2, \text{AIC}_{\ell-1}^2, \dots, \text{AIC}_0^2$ 。

此时

$$\text{AIC}_j = \text{AIC}_j^1 + \text{AIC}_j^2 \quad (8.31)$$

表示假设在时刻 $n_0 + jp + 1$ 发生结构变化的局部平稳 AR 模型的 AIC。因此, 在 $\text{AIC}_0, \dots, \text{AIC}_\ell$ 中寻找取最小值的时刻, 即可得到最优的划分时点。

TSSS 程序包的函数 `lsar.chgpt` 用于求时间序列结构变化时点的精密估计值。该函数需要如下参数:

- `max.arorder` AR 模型的最大阶数
- `subinterval` 用于建模的数据区间的起点和终点, 以 `c(n0, ne)` 的形式指定
- `candidate` 变化点候选的最小值和最大值, 以 `c(n1, n2)` 的形式指定。其中需满足 $n_0 + 2k < n_1 < n_2 + k < n_e$ (k 为 `max.arorder`)

该函数的输出为:

- `aic`: 在区间 $[n_1, n_2]$ 上拟合 AR 模型的 AIC
- `aicmin`: 最小 AIC
- `change.point`: 估计得到的变化点

```

> data(MYE1F)
> lsar.chgpt(MYE1F, max.arorder = 10, subinterval = c(200,1000), candidate =
c(400,800))
>
> lsar.chgpt(MYE1F, max.arorder = 10, subinterval = c(600,1400), candidate =
c(800,1200))

```

例：P波与S波到达时刻的估计 图 8.2 是对图 8.1 中可见明显变化的 $n = 600$ 和 $n = 1000$ 附近，精密调查其变化时点的结果。图 (a) 是 $n = 400 \sim 800$ 部分的放大图，前半段为常微动，后半段被认为是 P 波。对此，图 (b) 是根据 (8.31) 计算的 AIC 值的绘图。在 $n = 630$ 处取最小值 3366.38，由此可判定 P 波在 $n = 630$ 到达。

图 (c) 是 $n = 800 \sim 1200$ 部分的放大图，前半段为 P 波，后半段为 S 波。由图 (d) 所示的 AIC 值可判定 S 波在 $n = 1026$ 到达。需要注意的是，(a) 的情况下 AIC 变化极其急剧，两端处 AIC 上升了 600~800，表明估计精度很好；而 (c) 的情况下，距离 200 个点的位置 AIC 上升仅约 60，相对平缓，说明 S 波的检测比 P 波更困难。

8.4 变化时点的事后概率

本节考虑一种更充分利用局部平稳 AR 模型 AIC 所包含信息的方法。文献 [3] 指出， $\exp\{-AIC/2\}$ 可以作为在用极大似然法估计参数时模型似然的一个合理定义。对局部平稳 AR 模型而言

$$p(y|j) = \exp\left\{-\frac{1}{2}AIC_j\right\} \quad (8.32)$$

就成为将 $n_0 + pj + 1$ 假定为到达时刻时的局部平稳 AR 模型的似然。

因此，如果给定到达时刻的先验分布，则到达时刻的后验分布可由下式得到：

$$p(j|y) = \frac{p(y|j)p(j)}{\sum_j p(y|j)p(j)} \quad (8.33)$$

下面对到达时刻的候选区间采用均匀分布作为先验分布。在某些情况下，把更大的权重放在候选区间的中央可能更合理。不过，由于似然 $p(y|j)$ 在多数情况下只会在很窄的区间内取较大值，因此只有先验概率的局部形状会影响后验概率。也就是说，只要使用足够平滑的函数，先验概率的选择对当前问题并不那么重要。

对已估计到的到达时刻，一个重要用途是用于地震震源的估计。传统上，震源估计多采用加权最小二乘法；但若使用局部平稳 AR 模型的似然或到达时刻的后验分布进行极大似然法或贝叶斯建模，则有望更精确地估计震源。

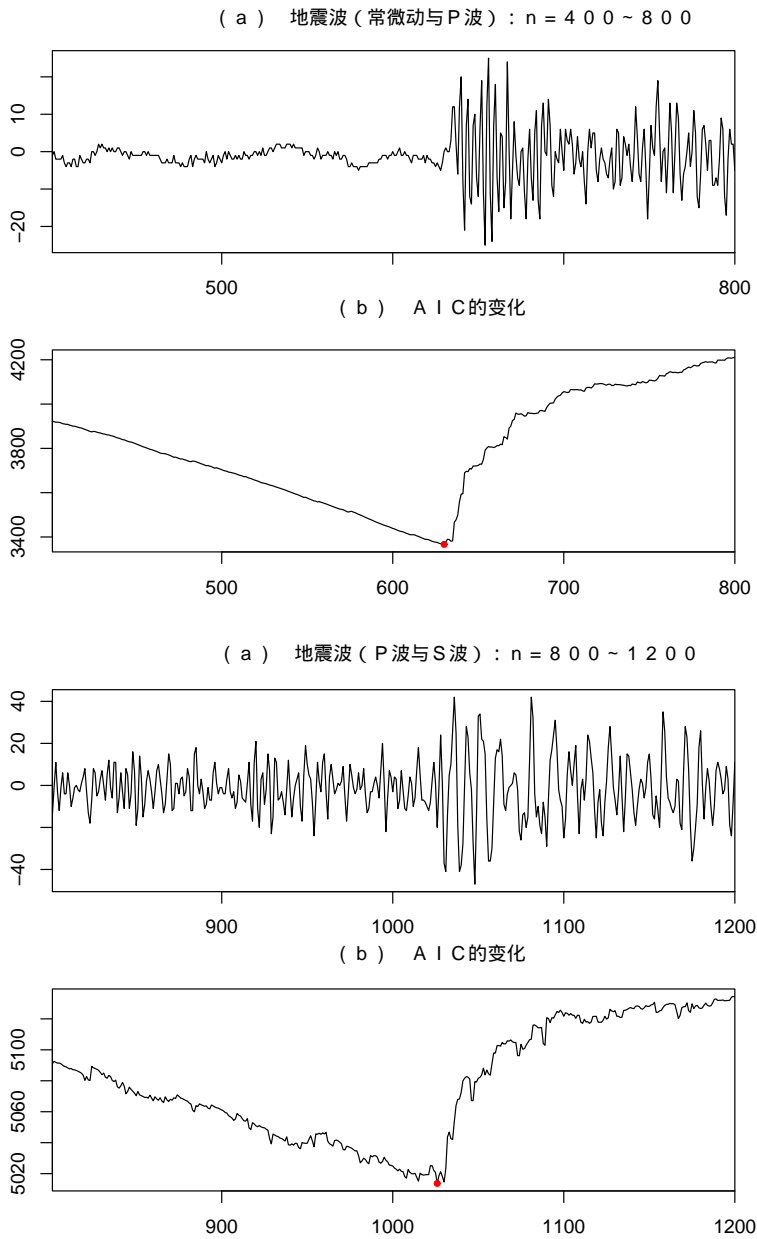


图 8.2 P 波与 S 波到达时刻的估计

使用 TSSS 包的函数 `lsar.chgpt`, 可以如下计算时间序列变化点的后验概率:

```
> x <- lsar.chgpt(MYE1F, max.arorder=10, subinterval = c(400,800), candidate =
c(600,700))
> AICP <- x$aic
> post <- exp( -(AICP-min(AICP))/2 )
> post <- post/sum(post)
> plot( post,type="l", lwd=2 )
```

图 8.3 给出了 MYE1F 数据的 P 波 (左) 与 S 波 (右) 的到达时刻后验分布。P 波

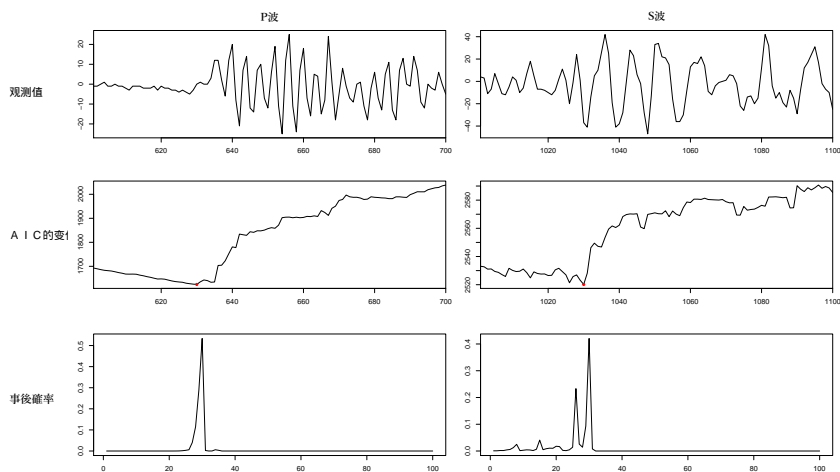


图 8.3 地震波到达时刻的后验分布

到达时, AIC 值在 $n = 630$ 附近取得明显最小值。另一方面, S 波到达时 AIC 变化并不那么明确, 实际上具有若干个极小点。正如 y 轴所示, 左图 AIC 的变化范围约为 400, 而右图约为 60。因此, 最下方图中 P 波的后验分布呈单峰, 而 S 波的情况则能看到多个峰。

● 章末习题

1. 在局部平稳 AR 模型中, 当均值发生变化时, 应考虑怎样的模型? 并讨论此时的 AIC。
2. 对如图 1.2 所示的时间序列, 考虑仅方差发生变化的局部平稳模型。
3. 参考第 11 章的多项式趋势模型, 给出当多项式在数据区间中途发生变化时的模型与 AIC。
4. 在问题 2 中, 若假设趋势的连续性或光滑性, 会怎样? 请加以讨论。
5. 在问题 5 中, 假设 (8.22) 与 (8.27) 的矩阵各自进行豪斯霍尔德变换需要 $\frac{1}{2}n_0m^2$ 、 $\frac{1}{2}(p+1)m^2$ 的计算量, 比较普通 AR 模型与 8.3 节局部平稳 AR 模型的计算量。

第 9 章

利用状态空间模型的时间序列分析

时间序列分析中所使用的各种模型, 都可以通过状态空间模型进行统一处理。时间序列分析中的许多问题都可以表述为状态空间模型中的状态估计问题。本节将给出用于逐次且高效地进行状态估计的卡尔曼滤波与平滑算法, 并说明其在时间序列长期预测、插值以及参数估计中的应用。

9.1 状态空间模型

设 y_n 为 ℓ 维时间序列。此时, 用来表示该时间序列的如下模型称为**状态空间模型** (state-space model)。

$$x_n = F_n x_{n-1} + G_n v_n, \quad (\text{系统模型}) \quad (9.1)$$

$$y_n = H_n x_n + w_n, \quad (\text{观测模型}) \quad (9.2)$$

这里, x_n 是无法直接观测到的 k 维向量, 称为**状态** (state)。 v_n 称为系统噪声或状态噪声, 是服从均值向量为 0、方差协方差矩阵为 Q_n 的 m 维正态白噪声。另一方面, w_n 称为观测噪声, 设其为服从均值向量 0、方差协方差矩阵 R_n 的 ℓ 维正态白噪声。 F_n 、 G_n 、 H_n 分别是 $k \times k$ 、 $k \times m$ 、 $\ell \times k$ 的矩阵。时间序列分析中所使用的许多线性模型都可以表示成这种状态空间模型的形式, 并加以统一处理。

状态空间模型可作如下两种解释。首先, 如果将 (9.2) 中的观测模型看作表示时间序列 y_n 被观测的机制的回归模型, 那么状态 x_n 就是其中的回归系数。此时, (9.1) 中的系统模型就成为描述这些回归系数随时间变化方式的模型。另一方面, 如果将状态向量 x_n 看作需要估计的信号, 那么系统模型表示信号的生成机制, 而观测模型则表示在实际观测该信号时, 信号如何被变换并叠加噪声。

下面给出一个与简单时间序列模型相对应的状态空间模型的例子。设时间序列

y_n 服从 AR 模型

$$y_n = \sum_{i=1}^m a_i y_{n-i} + v_n \quad (9.3)$$

此时, 如果定义状态向量为 $x_n = (y_n, y_{n-1}, \dots, y_{n-m+1})^T$, 则很容易验证, x_n 与 x_{n-1} 之间满足

$$x_n = Fx_{n-1} + Gv_n \quad (9.4)$$

的关系。其中, F 与 G 分别由

$$F = \begin{bmatrix} a_1 & a_2 & \cdots & a_m \\ 1 & & & \\ & \ddots & & \\ & & 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (9.5)$$

所确定的 $m \times m$ 矩阵以及 m 维向量。此外, 由于状态 x_n 的第 1 个分量就是 y_n , 如果令 $H = [1 \ 0 \ \cdots \ 0]$, 则可由

$$y_n = Hx_n \quad (9.6)$$

得到观测模型。进一步地, 如果分别令系统噪声与观测噪声的方差为 $Q = \sigma^2$ 以及 $R = 0$, 就得到了 AR 模型的状态空间模型。由此可见, AR 模型是状态空间模型中的一种特殊情形: 其中状态向量 x_n 由截至时刻 n 的观测值完全确定, 且观测噪声为 0。

这里需要注意的是, 状态空间模型的表示方式并不是唯一的。例如, 当模型 (9.1)、(9.2) 给定时, 若 T 为任意可逆矩阵, 则通过变换

$$z_n = Tx_n, \quad F'_n = TF_nT^{-1}, \quad G'_n = TG_n, \quad H'_n = H_nT^{-1} \quad (9.7)$$

可得到与 (9.1)、(9.2) 等价的状态空间模型

$$\begin{aligned} z_n &= F'_n z_{n-1} + G'_n v_n, \\ y_n &= H'_n z_n + w_n. \end{aligned}$$

这里再尝试如下定义状态:

$$x_n = (y_n, \tilde{y}_{n+1|n-1}, \dots, \tilde{y}_{n+m-1|n-1})^T$$

其中, $\tilde{y}_{n+i|n-1} = \sum_{j=i+1}^m a_j y_{n+i-j}$, 表示 y_{n+i} 的 1 步预测值 $y_{n+i|n-1} = \sum_{j=1}^m a_j y_{n+i-j}$ 中与时刻 $n-1$ 之前的观测值相关的部分。此时, 若令

$$F = \begin{bmatrix} a_1 & 1 & & \\ a_2 & & \ddots & \\ \vdots & & & 1 \\ a_m & & & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad H = [1 \ 0 \ \cdots \ 0] \quad (9.8)$$

便可得到 AR 模型的另一种状态空间模型表示。

一般来说, 本书中所讨论的许多模型, 如 ARMA 模型、趋势成分模型、季节成分模型等, 都可以用

$$F_i = \begin{bmatrix} a_{1i} & 1 & & \\ a_{2i} & & \ddots & \\ \vdots & & & 1 \\ a_{mi} & & & 0 \end{bmatrix}, \quad G_i = \begin{bmatrix} 1 \\ b_{1i} \\ \vdots \\ b_{m-1,i} \end{bmatrix}, \quad H_i = [c_{1i} \ c_{2i} \ \cdots \ c_{m,i}] \quad (9.9)$$

这样的形式来表示。在实际的时间序列分析中, 有时会使用由 p 个成分构成的合成模型, 而这些成分模型就采用上述形式。如果各状态的维数分别为 m_1, \dots, m_p , 并令 $m = m_1 + \dots + m_p$, 则只需将 $m \times m$ 矩阵、 $m \times p$ 矩阵以及 m 维向量分别定义为

$$F = \begin{bmatrix} F_1 & & \\ & \ddots & \\ & & F_p \end{bmatrix}, \quad G = \begin{bmatrix} G_1 & & \\ & \ddots & \\ & & G_p \end{bmatrix}, \quad H = [H_1 \ \cdots \ H_p] \quad (9.10)$$

便可得到时间序列的状态空间模型。本书中将采用这种形式的模型作为标准形式。

9.2 利用卡尔曼滤波进行状态估计

与状态空间模型相关的一个重要问题是, 基于时间序列 y_n 的观测值来估计状态 x_n 。正如以下各节所示, 时间序列的预测、插值、似然计算等都可以通过利用这种状态估计而统一地实现。

以下考虑这样一个问题: 基于观测值 $Y_j = \{y_1, \dots, y_j\}$, 估计时刻 n 的状态 x_n 。特别地, 当 $j < n$ 时, 这是一个估计观测区间之后未来状态的问题, 称为**预测** (prediction, forecast)。当 $j = n$ 时, 这是一个估计观测区间终点即当前状态的问题, 称为**滤波** (滤波, filtering)。当 $j > n$ 时, 则是一个基于到当前时刻为止的观测值来估计过去状态的问题, 称为**平滑化** (smoothing)。

为了回答这些状态估计问题, 需要求得在给定观测值 Y_j 条件下状态 x_n 的条件分布 $p(x_n|Y_j)$ 。不过, 由于状态空间模型 (9.1)、(9.2) 是线性模型, 而且 v_n 、 w_n 以及 x_0 全都服从正态分布, 因此这些条件分布也都是正态分布。因此, 在研究状态空间模型中的状态估计问题时, 只需求出规定条件分布的均值向量和方差协方差矩阵即可。一般而言, 为了求得在给定观测值 $\{y_1, \dots, y_j\}$ 条件下状态 $\{x_1, \dots, x_n\}$ 的条件联合分布, 需要非常巨大的计算量。然而, 对于状态空间模型, 可以通过逐次计算算法极其高效地求得状态 x_n 的条件边缘分布。这就是称为**卡尔曼滤波**^{*1} (Kalman

^{*1} (译注) "カルマンフィルタ" (Kalman filter), 在中文文献中也常称为"Kalman 滤波器"或"卡尔曼滤

filter) 的算法。

以下将状态 x_n 的条件均值和方差协方差矩阵记为

$$\begin{aligned} x_{n|j} &\equiv E[x_n|Y_j] \\ V_{n|j} &\equiv E[(x_n - x_{n|j})(x_n - x_{n|j})^T] \end{aligned} \quad (9.11)$$

不过, 卡尔曼滤波直接处理的是 $j = n - 1$ 的情形 (一步预测) 和 $j = n$ 的情形 (滤波)。正如下述算法所示, 通过交替重复一步预测与滤波这两个步骤, 就可以依次求得这些量。

[一步预测]

$$\begin{aligned} x_{n|n-1} &= F_n x_{n-1|n-1} \\ V_{n|n-1} &= F_n V_{n-1|n-1} F_n^T + G_n Q_n G_n^T \end{aligned} \quad (9.12)$$

[滤波]

$$\begin{aligned} K_n &= V_{n|n-1} H_n^T (H_n V_{n|n-1} H_n^T + R_n)^{-1} \\ x_{n|n} &= x_{n|n-1} + K_n (y_n - H_n x_{n|n-1}) \\ V_{n|n} &= (I - K_n H_n) V_{n|n-1} \end{aligned} \quad (9.13)$$

在一步预测算法中, x_n 的预测值 (均值) 向量 $x_{n|n-1}$ 只需将 x_{n-1} 的滤波值 $x_{n-1|n-1}$ 左乘状态转移矩阵 F_n 即可。此外, 其方差协方差矩阵 $V_{n|n-1}$ 由两部分组成: 第一项表示 F_n 所带来的变换影响, 第二项表示系统噪声 v_n 的影响。

在滤波算法中, 首先求得称为卡尔曼增益的 K_n 。另外, $y_n - H_n x_{n|n-1}$ 是 y_n 的预测误差, 而 $H_n V_{n|n-1} H_n^T + R_n$ 则是该预测误差的方差协方差矩阵。此时, x_n 的滤波均值向量可以表示为预测向量 $x_{n|n-1}$ 与预测误差经卡尔曼增益加权后的和。又由于

$$x_{n|n} = K_n y_n + (I - K_n H_n) x_{n|n-1}$$

也成立, 因此可知, $x_{n|n}$ 是新的观测值 y_n 与预测向量 $x_{n|n-1}$ 的加权和。 $V_{n|n}$ 还可以写成

$$V_{n|n} = V_{n|n-1} - K_n H_n V_{n|n-1}$$

其中右边第二项表示: 通过利用观测值 y_n 所获得的信息, 使得对状态 x_n 的估计精度得到改善的部分。

9.3 平滑算法

平滑问题是指: 在给定时间序列 $Y_N = \{y_1, \dots, y_N\}$ 时, 估计其中间时刻状态 x_n 的问题。对于这种平滑问题, 也存在与卡尔曼滤波类似的称为**固定区间平滑化**

波器”。本书统一译为“卡尔曼滤波”。

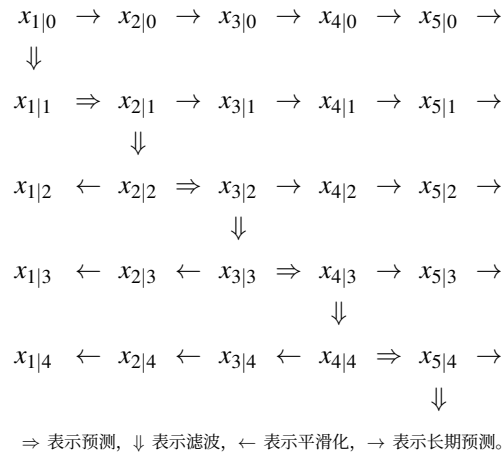


图 9.1 利用卡尔曼滤波与平滑算法进行逐次计算

(fixed-interval smoothing) 的算法。滤波仅利用到时刻 n 为止的观测值来估计 x_n ，而平滑算法则利用所有已经获得的观测值进行估计。因此，进行平滑后，一般可以得到比滤波更高精度的状态估计。

[固定区间平滑化]

$$\begin{aligned}
 A_n &= V_{n|n} F_{n+1}^T V_{n+1|n}^{-1} \\
 x_{n|N} &= x_{n|n} + A_n (x_{n+1|N} - x_{n+1|n}) \\
 V_{n|N} &= V_{n|n} + A_n (V_{n+1|N} - V_{n+1|n}) A_n^T
 \end{aligned} \tag{9.14}$$

如上所示，在平滑算法中，会用到卡尔曼滤波的结果，即 $x_{n|n-1}$ 、 $x_{n|n}$ 、 $V_{n|n-1}$ 、 $V_{n|n}$ 。因此，要进行平滑，首先要通过卡尔曼滤波求出 $\{x_{n|n-1}, x_{n|n}, V_{n|n-1}, V_{n|n}\}$ ($n = 1, \dots, N$)，然后再根据 (9.14) 中的算法，从 $x_{N-1|N}$ 、 $V_{N-1|N}$ 开始，按时间逆向依次求到 $x_{1|N}$ 、 $V_{1|N}$ 。另外需要注意的是，执行平滑算法时需要初值 $x_{N|N}$ 、 $V_{N|N}$ ，而这些量正是由卡尔曼滤波求得的。

9.4 状态的长期预测

卡尔曼滤波算法直接给出的只是一步预测的方法，但通过反复应用它，也可以进行长期预测。下面考虑这样一个长期预测问题：基于时间序列 $Y_n = \{y_1, \dots, y_n\}$ ，估计 j 期之后的状态 x_{n+j} ($j > 1$)。

首先，通过卡尔曼滤波可以得到 x_{n+1} 的一步预测均值 $x_{n+1|n}$ 以及方差协方差矩阵 $V_{n+1|n}$ 。这里注意到观测值 y_{n+1} 尚未获得，因此形式上可以认为 $Y_{n+1} = Y_n$ 成立，并继续进行计算。在这种情况下，显然有 $x_{n+1|n+1} = x_{n+1|n}$ ， $V_{n+1|n+1} = V_{n+1|n}$ 。因此，

由卡尔曼滤波在 $n+1$ 时期的一步预测算法可得

$$\begin{aligned} x_{n+2|n} &= F_{n+2}x_{n+1|n} \\ V_{n+2|n} &= F_{n+2}V_{n+1|n}F_{n+2}^T + G_{n+2}Q_{n+2}G_{n+2}^T \end{aligned} \quad (9.15)$$

这表明, 为了进行两步预测, 只需省略针对 y_{n+1} 的滤波步骤, 而只执行预测步骤即可。

一般地, 为了基于 Y_n 进行到 j 期之后的长期预测, 由于 $Y_n = Y_{n+1} = \dots = Y_{n+j}$ 成立, 因此只需重复执行预测步骤 k 次即可。综上所述, 基于时刻 n 为止的观测值 Y_n 来预测 x_{n+1}, \dots, x_{n+j} 的算法如下:

[长期预测] 对于 $i = 1, \dots, j$

$$\begin{aligned} x_{n+i|n} &= F_{n+i}x_{n+i-1|n} \\ V_{n+i|n} &= F_{n+i}V_{n+i-1|n}F_{n+i}^T + G_{n+i}Q_{n+i}G_{n+i}^T \end{aligned} \quad (9.16)$$

9.5 时间序列的预测

利用以上方法求得的状态 x_n 的预测, 便可以直接实现时间序列本身的预测。也就是说, 由于状态 x_n 与时间序列 y_n 的关系由观测模型 (9.2) 给出, 若将给定 Y_n 条件下 y_{n+j} 的均值与方差协方差矩阵分别记为 $y_{n+j|n} \equiv E[y_{n+j}|Y_n]$, $d_{n+j|n} \equiv \text{Cov}(y_{n+j}|Y_n)$, 则有

$$\begin{aligned} y_{n+j|n} &= E[H_{n+j}x_{n+j} + w_{n+j}|Y_n] \\ &= H_{n+j}x_{n+j|n} \end{aligned} \quad (9.17)$$

$$\begin{aligned} d_{n+j|n} &= \text{Cov}(H_{n+j}x_{n+j} + w_{n+j}|Y_n) \\ &= H_{n+j}\text{Cov}(x_{n+j}|Y_n)H_{n+j}^T + H_{n+j}\text{Cov}(x_{n+j}, w_{n+j}|Y_n) \\ &\quad + \text{Cov}(w_{n+j}, x_{n+j}|Y_n)H_{n+j}^T + \text{Cov}(w_{n+j}|Y_n) \\ &= H_{n+j}V_{n+j|n}H_{n+j}^T + R_{n+j} \end{aligned} \quad (9.18)$$

由此可得。这样一来, 基于时间序列观测值 Y_n 的 y_{n+j} 的预测分布是一个均值为 $y_{n+j|n}$ 、方差协方差矩阵为 $d_{n+j|n}$ 的正态分布, 并且它们可以由 (9.17)、(9.18) 方便地求得。此时, y_{n+j} 的预测值为 $y_{n+j|n}$, 而其标准误差则由 $(d_{n+j|n})^{1/2}$ 给出。另需注意的是, 时间序列的一步预测 $y_{n|n-1}$ 、 $d_{n|n-1}$ 已经在卡尔曼滤波 (9.13) 中求出, 并且在滤波算法中被加以利用。

例: 食品产业从业人员数据的长期预测

TSSS 包中的函数 `tsmooth` 基于时间序列模型的状态空间表示来进行长期预测。需要如下参数:

f:	状态转移矩阵 F_n
g:	矩阵 G_n
h:	矩阵 H_n
q:	系统噪声的方差协方差矩阵 Q_n
r:	观测噪声的方差协方差矩阵 R
x0:	初始状态向量 $x_{0 0}$
v0:	初始状态的方差协方差矩阵 $V_{0 0}$
filter.end:	滤波结束时点
predict.end:	预测结束时点
minmax:	观测值的上限和下限 (除此之外视为异常值)

该函数的输出:

mean.smooth:	平滑均值
cov.smooth:	平滑方差
esterr:	状态估计误差
likelihood:	模型的对数似然
aic:	模型的 AIC

函数 `tsmooth` 需要时间序列的状态空间模型。在下面的 R 代码示例中, 先用函数 `arfit` 估计 AR 模型, 再利用该函数的输出, 也就是使 AIC 最小的 AR 阶数、AR 系数以及创新项方差, 来定义状态空间模型。如果希望使用并非通过 AIC 最小化自动确定的阶数 `z1$maice.order` 来进行滤波与长期预测, 则只需像 `m1 <- 5` 这样直接指定参数 `m1` 即可。

```

> data(BLSALLFOOD)
> # AR 模型的估计
> BLS120 <- BLSALLFOOD[1:120]
> z1 <- arfit(BLS120, plot = FALSE)
> tau2 <- z1$sigma2
>
> # 状态空间模型的构建
> # m = maice.order, k=1
> m1 <- z1$maice.order
> arcoef <- z1$arcoef[[m1]]
> f <- matrix(0.0e0, m1, m1)
> f[1, ] <- arcoef
> if (m1 != 1)
> for (i in 2:m1) f[i, i-1] <- 1
> g <- c(1, rep(0.0e0, m1-1))
> h <- c(1, rep(0.0e0, m1-1))
> q <- tau2[m1+1]
> r <- 0.0e0
> x0 <- rep(0.0e0, m1)
> v0 <- NULL
>
> # 状态空间模型下的平滑
> s1 <- tsmooth(BLSALLFOOD, f, g, h, q, r, x0, v0, filter.end = 120, predict.end =
156)
> s1
>
> plot(s1, BLSALLFOOD)

```

图 9.2 给出了对 BLSALLFOOD 数据 ($N = 156$) 中前 120 个数据拟合 AR 模型, 并利用该模型对后 36 个时点进行长期预测的结果。在 AR 模型的估计中, 首先计算时间序列的样本均值 \bar{y} , 并通过

$$y_n^* = y_n - \bar{y}$$

将其转换为均值为 0 的时间序列, 然后对 y_1^*, \dots, y_N^* 应用 Yule-Walker 方法进行参数估计。在预测中, 则利用 AR 模型的状态空间模型, 借助卡尔曼滤波求得长期预测值 $y_{n+j|n}^*$, 再通过

$$y_{n+j|n} = y_{n+j|n}^* + \bar{y}$$

得到时间序列的长期预测值。图 9.2 展示了以这种方式得到的预测分布均值 $y_{120+j|120}$ ($j = 1, \dots, 36$) 以及标准预测误差区间 $y_{120+j|120} \pm \sqrt{\hat{d}_{120+j|120}}$ 。实际时间序列在 $n \leq 120$ 时用实线表示, 在 $n > 120$ 时用 \circ 表示。(a)、(b)、(c)、(d) 分别表示假定 AR 阶数为 $m = 1, 5, 10, 15$ 时的结果。1 阶 AR 模型的长期预测值呈指数型迅速衰减, 完全没有利用该数据中周期波动所包含的信息。当 $m = 5$ 时, 预测值在最初约 1 年内能够在一定程度上预测出振荡的样子, 但其振幅衰减得相当快。当 $m = 10$

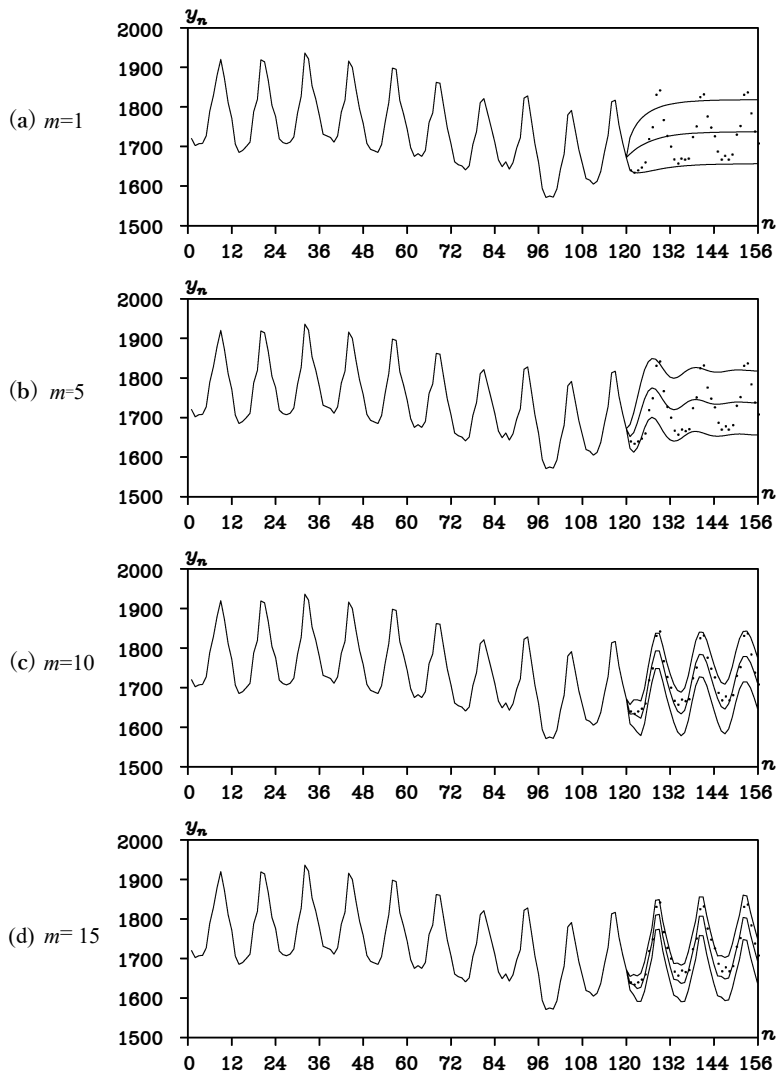


图 9.2 长期预测分布 (粗线: 均值, 细线: \pm (标准差), \circ : 实测值)

时, 预测值已经能够较好地预测实际数据的波动。而当 $m = 15$ 时, 甚至连实际的波动模式也能较好地预测出来。像一步预测这样预测区间较短的情况下, 无论采用哪种模型, 预测结果都相差不大; 但在长期预测中, 不同模型之间会产生显著差异。因此可以看出, 不合适的模型所做出的预测有可能导致严重偏差的结果, 因此选择适当的模型是极其重要的。

9.6 时间序列模型的似然计算与参数估计

设存在一个由参数 θ 所规定的时间序列模型, 并且其状态空间模型已知。给定长度为 N 的时间序列 y_1, \dots, y_N 时, 记由该时间序列模型所决定的 y_1, \dots, y_N 的 N 维

联合密度函数为 $f_N(y_1, \dots, y_N | \theta)$ 。此时, 该模型的似然定义为

$$L(\theta) = f_N(y_1, \dots, y_N | \theta) \quad (9.1)$$

这里, 对 $n = N, N-1, \dots, 2$ 而言,

$$f_n(y_1, \dots, y_n | \theta) = f_{n-1}(y_1, \dots, y_{n-1} | \theta) g_n(y_n | y_1, \dots, y_{n-1}, \theta)$$

成立。反复应用这一关系可得, 时间序列模型的似然可以表示为条件密度函数的乘积, 即

$$L(\theta) = \prod_{n=1}^N g_n(y_n | y_1, \dots, y_{n-1}, \theta) = \prod_{n=1}^N g_n(y_n | Y_{n-1}, \theta) \quad (9.19)$$

这里, 为简便起见, 令 $Y_0 = \phi$, 并记 $f_1(y_1 | \theta) \equiv g_1(y_1 | Y_0, \theta)$ 。此时, 对数似然为

$$\ell(\theta) = \log L(\theta) = \sum_{n=1}^N \log g_n(y_n | Y_{n-1}, \theta). \quad (9.20)$$

如 (9.17)、(9.18) 所示, $g_n(y_n | Y_{n-1}, \theta)$ 是在给定观测值 Y_{n-1} 时 y_n 的预测分布, 它服从均值为 $y_{n|n-1}$ 、方差协方差矩阵为 $d_{n|n-1}$ 的正态分布, 因此具体可写为

$$g_n(y_n | Y_{n-1}, \theta) = \left(\frac{1}{\sqrt{2\pi}} \right)^\ell |d_{n|n-1}|^{-\frac{1}{2}} \times \exp \left\{ -\frac{1}{2} (y_n - y_{n|n-1})^T d_{n|n-1}^{-1} (y_n - y_{n|n-1}) \right\} \quad (9.21)$$

因此, 将其代入 (9.20) 可知, 该时间序列模型的对数似然为

$$\ell(\theta) = -\frac{1}{2} \left\{ \ell N \log 2\pi + \sum_{n=1}^N \log |d_{n|n-1}| + \sum_{n=1}^N (y_n - y_{n|n-1})^T d_{n|n-1}^{-1} (y_n - y_{n|n-1}) \right\} \quad (9.22)$$

由此给出。

AR 模型、ARMA 模型等平稳模型, 以及趋势模型、季节调整模型等许多非平稳模型, 都可以表示为线性高斯型状态空间模型的形式。因此, 对于这类时间序列模型, 就得到了一个利用卡尔曼滤波和 (9.23) 计算对数似然的统一算法。为了求得时间序列模型参数的极大似然估计值, 只需将以上得到的对数似然作为目标函数, 通过数值优化方法 (附录 C) 求出使其最大的参数即可。有关状态空间模型参数估计的具体例子, 请参见第 10~15 章。

这样, 状态空间模型中所包含的参数可以通过对 (9.22) 进行数值极大化来求得, 但这种方法通常需要相当大的计算量。因此, 如果可以解析地得到极大似然估计值或其较好的近似, 那么最好利用这些结果。例如, 在没有缺测值的情况下, 对于

AR 模型, 与其使用上述方法, 不如采用第 7 章中的 Yule-Walker 方法、最小二乘法、PARCOR 方法等。此外, 即使需要严格的极大似然估计值, 也应先用这些方法求出近似解, 再以其作为初值应用数值优化。

即使在没有这种简便方法、必须对对数似然进行极大化的情况下, 也有时可以降低直接进行数值优化时所处理参数的维数。在状态空间模型 (9.1)、(9.2) 中, 设时间序列的维数为 $\ell = 1$, 并且 w_n 的方差恒为 $R_n = \sigma^2$ 。此时, 若由

$$\begin{aligned} V_{n|n-1} &= \sigma^2 \tilde{V}_{n|n-1}, & V_{n|n} &= \sigma^2 \tilde{V}_{n|n}, \\ Q_n &= \sigma^2 \tilde{Q}_n, & \tilde{R} &= 1 \end{aligned} \quad (9.23)$$

重新定义 $\tilde{V}_{n|n}$ 、 $\tilde{V}_{n|n-1}$ 、 \tilde{Q}_n 以及 \tilde{R} , 则可以看出, 即使使用这些参数, 卡尔曼滤波 (9.12)、(9.13) 仍会给出相同的结果。首先, 在一步预测的步骤中, 用 $\tilde{V}_{n-1|n-1}$ 、 $\tilde{V}_{n|n-1}$ 代替 $V_{n-1|n-1}$ 、 $V_{n|n-1}$ 显然可以得到等价的结果。此外, 在滤波步骤中,

$$\begin{aligned} K_n &= V_{n|n-1} H_n^T (H_n V_{n|n-1} H_n^T + R_n)^{-1} \\ &= \sigma^2 \tilde{V}_{n|n-1} H_n^T \sigma^{-2} (H_n \tilde{V}_{n|n-1} H_n^T + 1)^{-1} \\ &= \tilde{V}_{n|n-1} H_n^T (H_n \tilde{V}_{n|n-1} H_n^T + \tilde{R})^{-1} \\ &= \tilde{K}_n \end{aligned} \quad (9.24)$$

因此, 当取 $\tilde{R} = 1$ 时, 卡尔曼增益 \tilde{K}_n 与 K_n 相同, 于是可知在滤波步骤中, 也可以用 $\tilde{V}_{n|n}$ 、 $\tilde{V}_{n|n-1}$ 来代替 $V_{n|n}$ 、 $V_{n|n-1}$ 。另外, 在上述计算中, 状态的均值 $x_{n|n-1}$ 、 $x_{n|n}$ 并不会发生任何变化。

由以上可知, 当 R_n 不随时间变化时, 即便 $R = \sigma^2$ 是未知数, 也可以令 $R = 1$ 来应用卡尔曼滤波。此时, 由 (9.18) 可知 $d_{n|n-1} = \sigma^2 \tilde{d}_{n|n-1}$, 因此

$$\ell(\theta) = -\frac{1}{2} \left\{ N \log 2\pi\sigma^2 + \sum_{n=1}^N \log \tilde{d}_{n|n-1} + \frac{1}{\sigma^2} \sum_{n=1}^N \frac{(y_n - y_{n|n-1})^2}{\tilde{d}_{n|n-1}} \right\} \quad (9.25)$$

于是, 由

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{1}{2} \left\{ \frac{N}{\sigma^2} - \frac{1}{(\sigma^2)^2} \sum_{n=1}^N \frac{(y_n - y_{n|n-1})^2}{\tilde{d}_{n|n-1}} \right\} = 0 \quad (9.26)$$

可知, σ^2 的极大似然估计值为

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N \frac{(y_n - y_{n|n-1})^2}{\tilde{d}_{n|n-1}} \quad (9.27)$$

进一步地, 若将参数向量 θ 中除 σ^2 以外的部分记为 θ^* , 则将 (9.27) 代入 (9.25) 后得到

$$\ell(\theta^*) = -\frac{1}{2} \left\{ N \log 2\pi \hat{\sigma}^2 + \sum_{n=1}^N \log \tilde{d}_{n|n-1} + N \right\} \quad (9.28)$$

这样一来, 通过这种方法, 就可以将需要数值优化算法求解的参数维数减少一个。

将上述方法总结如下:

1. 令 $R = 1$, 并应用卡尔曼滤波。
2. 由 (9.27) 求得 $\hat{\sigma}^2$ 。
3. 由 (9.28) 求得对数似然 $\ell(\theta^*)$ 。
4. 在重复执行 (1)~(3) 步骤的同时, 通过数值优化使对数似然 $\ell(\theta^*)$ 最大化, 从而求得极大似然估计值 $\hat{\theta}^*$ 。

9.7 缺测值的插补

在进行时间序列观测时, 由于观测设备故障等偶然因素, 或者由于观测对象及观测系统的物理限制等必然原因, 有时会出现时间序列的一部分无法被观测到的情况。在这种情况下, 实际上未能观测到的数据称为**缺测值** (missing observation)。即便只有百分之几的缺测值, 如果它们零散分布, 那么提取出连续观测到的部分后, 实际可利用的数据长度也可能变得非常短。为避免这种情况, 有时会将缺测值替换为 0 或数据的均值, 或者先通过线性插值进行估计, 然后再将其视为所有观测值都已获得而进行计算。正如后面将看到的, 这种处理实际上等价于对时间序列任意假定了某种模型, 因而可能给后续分析带来很大的偏差。本节将说明在时间序列存在缺测值时的似然计算方法与插补方法。

利用时间序列的状态空间模型, 即使对于包含缺测值的数据, 也可以计算严格的似然, 因此可以求得参数的极大似然估计值。设 $I(n)$ 为到时刻 n 为止, 时间序列实际上被观测到的时点所构成的集合。若不存在缺测, 则 $I(n) = \{1, \dots, n\}$ 。此时, 令 $Y_n \equiv \{y_i | i \in I(n)\}$, 则与上一节相同, 时间序列的对数似然为

$$\begin{aligned} \ell(\theta) &= \log p(Y_N | \theta) \\ &= \sum_{n \in I(N)} \log p(y_n | Y_{n-1}, \theta) \end{aligned} \quad (9.29)$$

这里, 与长期预测的情形相同, 当 y_n 为缺测值时, 有 $Y_n \equiv Y_{n-1}$ 成立, 因此在卡尔曼滤波中, 只需省略滤波部分即可。也就是说, 对所有的 n 都执行预测步骤, 而滤波步骤只对满足 $n \in I(N)$ 的 n 执行, 这样即使存在缺测, 也可以依次求得状态的预测分布 $p(x_n | Y_{n-1})$, 即 $x_{n|n-1}$ 与 $V_{n|n-1}$ 。因此, 当用 (9.17) 与 (9.18) 求得 $y_{n|n-1}$ 与

$d_{n|n-1}$ 之后, 与上一节一样, 时间序列的对数似然可以定义为

$$\begin{aligned} \ell(\theta) = & -\frac{1}{2} \sum_{n \in I(N)} \left\{ \ell \log 2\pi + \log |d_{n|n-1}| \right. \\ & \left. + (y_n - y_{n|n-1})^T d_{n|n-1}^{-1} (y_n - y_{n|n-1}) \right\}. \end{aligned} \quad (9.30)$$

当时间序列模型已经给定时, 可以利用该模型来进行**缺测值的插补**。首先, 与似然计算时相同, 对缺测值部分省略滤波步骤, 并利用卡尔曼滤波求得预测分布 $\{x_{n|n-1}, V_{n|n-1}\}$ 与滤波分布 $\{x_{n|n}, V_{n|n}\}$ 。之后, 再应用平滑算法 (9.14), 即使在存在缺测值的情况下, 也可以求得状态的平滑值 $x_{n|N}$ 。此时, $y_{n|N} = H_n x_{n|N}$ 给出了缺测值 y_n 的估计值。另外, 通过 $d_{n|N} = H_n V_{n|N} H_n^T + R_n$ 可以得到其估计误差方差。

```
> data(BLSALLFOOD)
> # AR 模型的估计
> z2 <- arfit(BLSALLFOOD, plot = FALSE)
> tau2 <- z2$sigma2
>
> # m = maice.order, k=1
> m2 <- z2$maice.order
> arcoef <- z2$arcoef[[m2]]
> f <- matrix(0.0e0, m2, m2)
> f[1, ] <- arcoef
> if (m2 != 1)
> for (i in 2:m2) f[i, i-1] <- 1
> g <- c(1, rep(0.0e0, m2-1))
> h <- c(1, rep(0.0e0, m2-1))
> q <- tau2[m2+1]
> r <- 0.0e0
> x0 <- rep(0.0e0, m2)
> v0 <- NULL
>
> tsmooth(BLSALLFOOD, f, g, h, q, r, x0, v0, missed = c(41, 101), np = c(30, 20))
```

TSSS 包中的函数 `tsmooth` 可以通过时间序列的状态空间表示, 在时间序列包含缺测值的情况下, 也能够进行对数似然的计算和缺测值的插补。除了长期预测所需的状态空间模型参数之外, 还需要有关缺测值位置的信息。为此, 用向量 `missed` 指定缺测值的起始位置, 用向量 `np` 指定连续缺测值的个数。

在下面的例子中, 由于取 `missed=c(41,101)`, `np=c(30,20)`, 因此时间序列的 $n = 41, \dots, 70$ 与 $n = 101, \dots, 120$ 被视为缺测值。与长期预测的情况相同, 也可以通过 `m2` 来指定 AR 模型的阶数。

例: 食品产业从业人数数据的缺测值插补

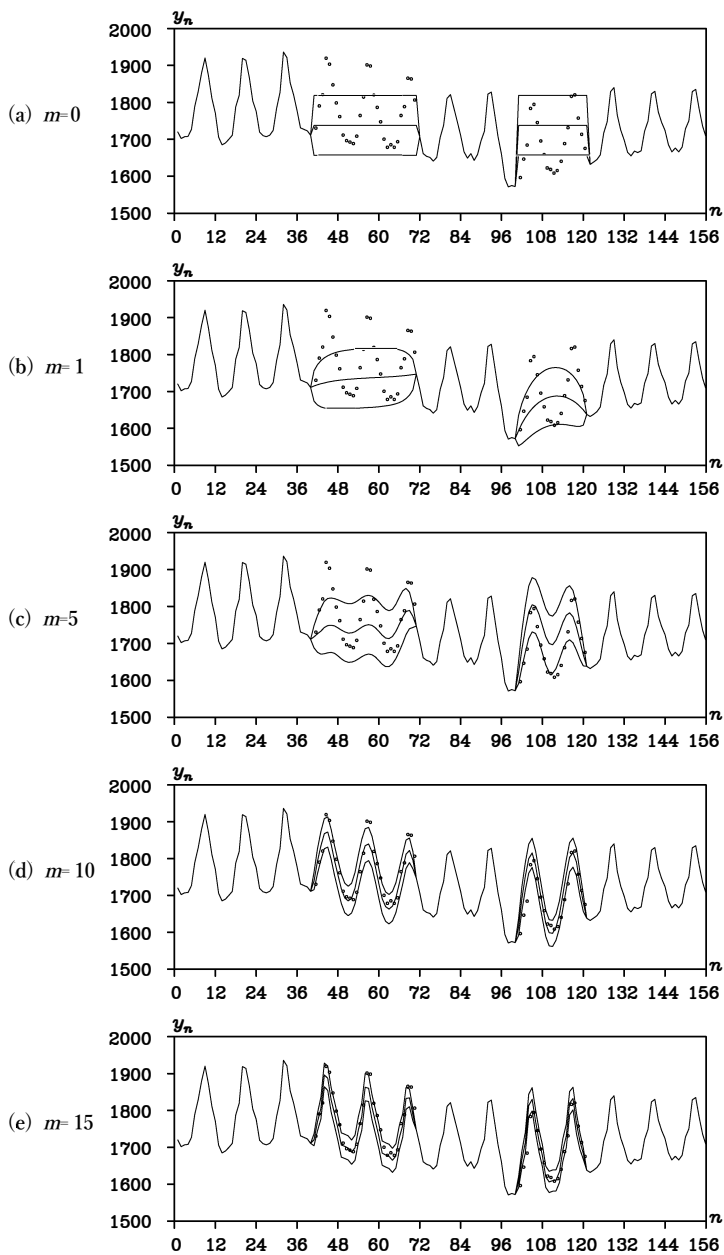


图 9.3 缺测值的插补 (给出了预测分布的均值与 \pm (标准差), \circ 为实测值)

图 9.3 给出了对 BLSALLFOOD 数据中 156 个观测值里的 $y_{41} \sim y_{70}$ 以及 $y_{101} \sim y_{120}$ 共 50 个观测值假定为缺测值时, 分别利用 0 阶、1 阶、5 阶、10 阶以及由 AIC 选择的 15 阶 AR 模型进行插补的结果。AR 系数采用的是通过最大化 (9.30) 所得到的极大似然估计量。用 0 阶模型进行插补, 相当于用整体均值进行替换。用 1 阶或 5 阶 AR 模型进行插补时, 由于不能充分捕捉时间序列的变化模式, 因此无法很好地重现实测值 (以 \circ 表示)。另一方面, 在 10 阶 AR 模型中, 周期性变化得到了较好的再现。进一步地, 在由 AIC 最小化所选择的最优 15 阶 AR 模型所得到的 (e) 中, 实际

数据的变化被惊人地再现出来。由此可见, 由最优模型得到的插补结果, 与用均值替换或进行线性插值等传统插补方法有显著不同。正如从 (a) 可以看出那样, 进行这类插补实际上等价于假定了一种特殊的时间序列模型, 而除非时间序列实际上服从该模型, 否则这类插补有可能对后续分析产生严重的不良影响。

● 章末习题

1. 证明通过 (9.7) 的变换, 可以得到与 (9.1)、(9.2) 等价的状态空间模型。
2. 设给定状态 x_n 维数为 1 的状态空间模型 $x_n = x_{n-1} + v_n$, $y_n = x_n + w_n$ 。其中, $v_n \sim N(0, \tau^2)$, $w_n \sim N(0, 1)$, $x_0 \sim N(0, 10^2)$ 。
 - (1) 写出该模型的卡尔曼滤波算法。
 - (2) 推导 $V_{n+1|n}$ 与 $V_{n|n-1}$ 的关系式。
 - (3) 设当 $n \rightarrow \infty$ 时, $V_{n|n-1} \rightarrow V$ 。证明此时 V 满足方程 $V^2 - \tau^2 V - \tau^2 = 0$ 。
 - (4) 当时刻 $n \rightarrow \infty$ 时, 卡尔曼滤波变成什么形式? (定常卡尔曼滤波)
3. 当 $V^2 - \tau^2 V - \tau^2 = 0$ 的解记为 V 时, 求定常状态下预测、滤波和平滑化的方差。进一步地, 对于 $\tau^2 = 1, 0.1, 0.01, 0.001$ 的情形, 求出它们的数值。

第 10 章

ARMA 模型的估计

本章将说明利用状态空间表示和卡尔曼滤波高效计算 ARMA 模型对数似然的方法。通过第 4 章介绍的方法对该对数似然进行数值最大化, 可以求得 ARMA 模型参数的最大似然估计值。此外, 还将介绍利用相邻阶数的 ARMA 模型参数估计值, 自动估计指定范围内所有 ARMA 模型的方法。

10.1 ARMA 模型的状态空间表示

假设给定 (m, ℓ) 阶平稳 ARMA 模型 (自回归移动平均模型):

$$y_n = \sum_{j=1}^m a_j y_{n-j} + v_n - \sum_{j=1}^{\ell} b_j v_{n-j} \quad (10.1)$$

其中, v_n 为均值为 0、方差为 σ^2 的正态白噪声。在此, 定义 $\tilde{y}_{n+i|n-1}$ 如下:

$$\tilde{y}_{n+i|n-1} = \sum_{j=i+1}^m a_j y_{n+i-j} - \sum_{j=i}^{\ell} b_j v_{n+i-j} \quad (10.2)$$

$\tilde{y}_{n+i|n-1}$ 表示在 y_{n+i} 中, 可以用时刻 $n-1$ 之前的时间序列观测值 y_{n-1}, y_{n-2}, \dots 以及时刻 n 之前的噪声 v_n, v_{n-1}, \dots 直接表达的部分。此时, 以下关系成立:

$$\begin{aligned} y_n &= a_1 y_{n-1} + \tilde{y}_{n|n-1} + v_n \\ \tilde{y}_{n+i|n-1} &= a_{i+1} y_{n-1} + \tilde{y}_{n+i|n-2} - b_i v_n \\ \tilde{y}_{n+k-1|n-1} &= a_k y_{n-1} - b_{k-1} v_n \end{aligned} \quad (10.3)$$

因此, 令 $k = \max(m, \ell + 1)$, 并将 k 维状态向量定义为:

$$x_n = (y_n, \tilde{y}_{n+1|n-1}, \dots, \tilde{y}_{n+k-1|n-1})^T \quad (10.4)$$

则可以表达为如下状态空间模型的形式:

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n \\ y_n &= Hx_n \end{aligned} \quad (10.5)$$

其中,

$$\begin{aligned} F &= \begin{bmatrix} a_1 & 1 & & \\ & a_2 & \ddots & \\ & \vdots & & 1 \\ & a_k & & \end{bmatrix}, & G &= \begin{bmatrix} 1 \\ -b_1 \\ \vdots \\ -b_{k-1} \end{bmatrix} \\ H &= [1 \ 0 \ \cdots \ 0] \end{aligned} \quad (10.6)$$

在此预先定义: 当 $i > m$ 时 $a_i = 0$, 当 $i > \ell$ 时 $b_i = 0$ 。也就是说, ARMA 模型可以表示为系数矩阵 F 、 G 、 H 不随时间变化且观测噪声为 0 的状态空间模型。

10.2 AR 模型的初始状态分布

为了对 ARMA 模型的状态空间表示应用卡尔曼滤波, 需要确定初始状态向量的均值 $x_{0|0}$ 以及方差协方差矩阵 $V_{0|0}$ 。由于 $x_{0|0}$ 和 $V_{0|0}$ 对应于在尚未获得时间序列 y_n 的任何观测值时的滤波分布, 因此可以通过计算处于平稳状态的 x_0 的均值和方差协方差矩阵来求得。

由于 ARMA 模型的初始状态方差协方差矩阵形式较复杂, 本节首先考虑 AR 模型的情况。这里考虑 AR 模型的两种状态空间表示。第一种是如 (9.3) 所示的自然状态空间表示, 在这种情况下, 方差协方差矩阵形式非常简单。由 (9.8) 式给出的第二种表示是 ARMA 模型状态空间表示的一个特例, 这种情况下的初始方差协方差矩阵可以扩展到 ARMA 模型的情形。

如第 9 章所示, AR 模型

$$y_n = \sum_{j=1}^m a_j y_{n-j} + v_n, \quad v_n \sim N(0, \sigma^2) \quad (10.7)$$

可以表示为状态空间模型的形式。

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n \\ y_n &= Hx_n \end{aligned} \quad (10.8)$$

其中, 矩阵 F 、 G 、 H 以及状态向量 x_n 定义如下:

$$\begin{aligned} F &= \begin{bmatrix} a_1 & a_2 & \cdots & a_m \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}, & G &= \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, & x_n &= \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-m+1} \end{bmatrix}, \\ H &= [1 \ 0 \ \cdots \ 0] \end{aligned} \quad (10.9)$$

对于这种状态空间表示, 初始状态向量定义为 $x_0 = [y_0, y_{-1}, \cdots, y_{1-m}]^T$, 其均值和方

差协方差矩阵给出如下。

$$\begin{aligned}
 x_{0|0} &= E[x_0] = [0, 0, \dots, 0]^T \\
 V_{0|0} &= \text{Cov}(x_0, x_0) = E[x_0 x_0^T] \\
 &= E \begin{bmatrix} y_0 y_0 & y_0 y_{-1} & \cdots & y_0 y_{1-m} \\ y_{-1} y_0 & y_{-1} y_{-1} \cdots & y_{-1} y_{1-m} & \\ \vdots & \ddots & \vdots & \\ y_{1-m} y_0 & y_{1-m} y_{-1} \cdots & y_{1-m} y_{1-m} & \end{bmatrix} \\
 &= E \begin{bmatrix} C_0 & C_1 & \cdots & C_{1-m} \\ C_1 & C_0 \cdots & C_{2-m} & \\ \vdots & \ddots & \vdots & \\ C_{1-m} & C_{1-m} \cdots & C_0 & \end{bmatrix} \tag{10.10}
 \end{aligned}$$

如上所述, 在 AR 模型的情况下, 使用 (10.9) 的状态空间表示可以很方便地计算初始状态的方差协方差矩阵, 但遗憾的是, 这种表示方法无法简单地扩展到 ARMA 模型。如 (9.8) 所示, AR 模型也可以表示如下:

$$\begin{aligned}
 F &= \begin{bmatrix} a_1 & 1 & & \\ a_2 & & \ddots & \\ \vdots & & & 1 \\ a_m & & & \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad x_n = \begin{bmatrix} y_n \\ \tilde{y}_{n+1|n-1} \\ \vdots \\ \tilde{y}_{n+m-1|n-1} \end{bmatrix}, \\
 H &= [1 \ 0 \ \cdots \ 0] \tag{10.11}
 \end{aligned}$$

其中, $\tilde{y}_{n+1|n-1}$ 定义为 $\tilde{y}_{n+1|n-1} = \sum_{j=i+1}^m a_j y_{n+i-j}$ 。这种状态空间表示是 ARMA 模型状态空间表示的一个特例, 以下所示的 AR 模型初始状态方差协方差矩阵的计算方法也可以扩展到 AR 模型的应用中。

在这种状态空间表示下, 初始状态向量给出为 $x_0 = [y_n, \tilde{y}_{1|n-1}, \dots, \tilde{y}_{m-1|n-1}]^T$, 因此方差协方差矩阵的各元素如下:

$$\begin{aligned}
 V_{0|0} &= \text{Cov}(x_0, x_0) = (V_{ij}) \\
 V_{11} &= E[y_0, y_0] = C_0 \\
 V_{1i} &= V_{i1} = E[y_0 \tilde{y}_{i-1|n-1}] = E \left[y_0 \sum_{j=i}^m a_j y_{i-1-j} \right] = \sum_{j=i}^m a_j C_{j+1-i} \\
 V_{ij} &= E[\tilde{y}_{i-1|n-1} \tilde{y}_{j-1|n-1}] \\
 &= \sum_{p=i}^m \sum_{q=j}^m a_p a_q E[y_{i-1-p} y_{j-1-q}] = \sum_{p=i}^m \sum_{q=j}^m a_p a_q C_{q-j-p+i}
 \end{aligned}$$

因此, 初始状态 $x_{0|0}$ 的方差协方差矩阵给出如下。

$$V_{0|0} = \begin{bmatrix} C_0 & \sum_{j=2}^m a_j C_{j-1} & \cdots & a_m C_1 \\ \sum_{j=2}^m a_j C_{j-1} & \sum_{p=2}^m \sum_{q=2}^m a_p a_q C_{q-p} \cdots & \sum_{p=2}^m a_p a_m C_{m-p} \\ \vdots & \ddots & \vdots \\ a_m C_1 & \sum_{q=2}^m a_m a_q C_{q-m} \cdots & a_m a_m C_0 \end{bmatrix} \quad (10.12)$$

10.3 ARMA 模型的初始状态

为了对 ARMA 模型的状态空间表示应用卡尔曼滤波, 需要给定初始状态的均值 $x_{0|0}$ 以及方差协方差矩阵 $V_{0|0}$ 。由于 $x_{0|0}$ 和 $V_{0|0}$ 表示在没有 y_n 观测值时的滤波分布, 因此与 AR 模型的情况一样, 只需计算 x_0 在平稳状态下的均值和方差协方差矩阵即可。首先, 由 $E[v_n] = 0$ 显然可以得出 $E[y_n] = 0$ 和 $E[\tilde{y}_{n+i|n-1}] = 0$ 。因此 x_n 的期望值为 0, 只需设 $x_{0|0} = (0, \dots, 0)^T$ 即可。接下来, 若将 $V_{0|0}$ 的 (i, j) 元素记为 V_{ij} , 则可以通过以下公式求得:

$$\begin{aligned} V_{11} &= E[y_0 y_0] = C_0 \\ V_{1i} &= V_{i1} = E[y_0 \tilde{y}_{i-1|1}] \\ &= E\left[y_0 \left(\sum_{j=i}^m a_j y_{i-1-j} - \sum_{j=i-1}^{\ell} b_j v_{i-1-j} \right)\right] \\ &= \sum_{j=i}^m a_j C_{j+1-i} - \sum_{j=i-1}^{\ell} b_j g_{j+1-i} \\ V_{ij} &= E[\tilde{y}_{i-1|1} \tilde{y}_{j-1|1}] \\ &= \sum_{p=i}^m \sum_{q=j}^m a_p a_q E[y_{i-1-p} y_{j-1-q}] - \sum_{p=i}^m \sum_{q=j-1}^{\ell} a_p b_q E[y_{i-1-p} v_{j-1-q}] \\ &\quad - \sum_{p=i-1}^{\ell} \sum_{q=j}^m b_p a_q E[v_{i-1-p} y_{j-1-q}] + \sum_{p=i-1}^{\ell} \sum_{q=j-1}^{\ell} b_p b_q E[v_{i-1-p} v_{j-1-q}] \\ &= \sum_{p=i}^m \sum_{q=j}^m a_p a_q C_{q-j-p+i} - \sum_{p=i}^m \sum_{q=j-1}^{\ell} a_p b_q g_{q-j-p+i} \\ &\quad - \sum_{p=i-1}^{\ell} \sum_{q=j}^m b_p a_q g_{p-i-q+j} + \sum_{p=i-1}^{\ell} b_p b_{p+j-i} \sigma^2 \end{aligned} \quad (10.13)$$

其中, C_i 和 g_i 分别是 ARMA 模型的自协方差函数和脉冲响应函数, 可以通过第 6 章的方法求得。

10.4 ARMA 模型的最大似然估计

利用以上结果, 可以求得关于 ARMA 模型参数 $\theta = (\sigma^2, a_1, \dots, a_m, b_1, \dots, b_\ell)^T$ 的对数似然。首先, 通过前一节的方法确定初始状态 $x_{0|0}$ 和 $V_{0|0}$ 。接着通过卡尔曼滤波求出 $x_{n|n-1}, V_{n|n-1}$ ($n = 1, \dots, N$), 令 $d_{n|n-1} = HV_{n|n-1}H^T$, $y_{n|n-1} = Hx_{n|n-1}$, 则根据 (9.23), 对数似然可以按如下公式计算:

$$\ell(\theta) = -\frac{N}{2} \log 2\pi - \frac{1}{2} \sum_{n=1}^N \log d_{n|n-1} - \frac{1}{2} \sum_{n=1}^N \frac{(y_n - y_{n|n-1})^2}{d_{n|n-1}} \quad (10.14)$$

通过数值优化的方法使这样求得的对数似然 $\ell(\theta)$ 最大化, 即可求得 θ 的最大似然估计值。不过, 与 9.6 节的方法类似, 按照以下步骤, 仅对 σ^2 而言, 无需数值优化即可简单求得。

1. 设定 $\sigma^2 = 1$ 并应用卡尔曼滤波, 对 $n = 1, \dots, N$ 计算 $d_{n|n-1}$ 和 $y_{n|n-1}$ 。
2. 通过 $\hat{\sigma}^2 = N^{-1} \sum_{n=1}^N (y_n - y_{n|n-1})^2 / d_{n|n-1}$ 求出 σ^2 的最大似然估计值。
3. 通过 $\ell'(a_1, \dots, a_m, b_1, \dots, b_\ell) \equiv \ell(\hat{\sigma}^2, a_1, \dots, a_m, b_1, \dots, b_\ell)$ 定义仅关于 AR 系数 a_1, \dots, a_m 和 MA 系数 b_1, \dots, b_ℓ 的对数似然。

通过对上述程序定义的对数似然函数应用数值优化方法, 可以求得 $\theta' = (a_1, \dots, a_m, b_1, \dots, b_\ell)^T$ 的最大似然估计值。但在实际操作中, 为了始终满足平稳性和可逆性的条件, 通常会进行如下转换。首先, 从 AR 系数 a_1, \dots, a_m 求得 PARCOR c_1, \dots, c_m , 则平稳性条件变为 $-1 < c_i < 1$, ($i = 1, \dots, m$)。此时, 若设

$$\alpha_i = \log \left(\frac{1+c_i}{1-c_i} \right), \quad i = 1, \dots, m \quad (10.15)$$

则 α_i 满足 $-\infty < \alpha_i < \infty$ 。反之, 对于任意的 $(\alpha_1, \dots, \alpha_m)^T \in \mathbf{R}^m$, 若进行逆转换:

$$c_i = \frac{e^{\alpha_i} - 1}{e^{\alpha_i} + 1} \quad (10.16)$$

则 c_i 始终满足 $|c_i| < 1$, 对应的 AR 系数是平稳的。另一方面, 为了保证对于任意选取的 $(\beta_1, \dots, \beta_\ell)^T \in \mathbf{R}^\ell$ 具有可逆性, 只需进行如下转换:

$$d_i = \frac{e^{\beta_i} - 1}{e^{\beta_i} + 1} \quad (10.17)$$

并将 d_1, \dots, d_ℓ 形式上视为 PARCOR, 求出对应的 MA 系数 b_1, \dots, b_ℓ 即可。此时, 若令 $\theta'' = (\alpha_1, \dots, \alpha_m, \beta_1, \dots, \beta_\ell)^T$, 则对应于任意 $\theta'' \in \mathbf{R}^{m+\ell}$ 的 ARMA 模型将始终

表 10.1 对太阳黑子数数据的对数序列拟合各种阶数 ARMA(m, ℓ) 模型时的 AIC

		ℓ					
		0	1	2	3	4	5
m	0	317.12	141.38	73.50	58.75	55.22	54.29
	1	105.43	64.72	139.05	55.96	62.69	58.24
	2	43.26	39.44	41.21	36.57	23.82	25.23
	3	41.50	41.30	110.15	15.01	25.56	21.26
	4	40.70	42.13	27.16	16.95	14.20	23.11
	5	41.14	42.99	18.78	14.42	14.02	23.84

满足平稳性和可逆性条件。若系数需要在某个 $0 < C < 1$ 范围内满足 $|c_i| < C$, 则只需用以下转换代替 (10.16) 即可:

$$c_i = \frac{e^{\alpha_i} - 1}{e^{\alpha_i} + 1} C \quad (10.18)$$

例：太阳黑子数数据

TSSS 包中的函数 `armafit` 用于求 ARMA 模型参数的最大似然估计值。需要给出参数 `ar.order` 和 `ma.order`。 `ar` 和 `ma` 分别是 AR 系数向量和 MA 系数向量的初始值。如果没有给出这些值, 则估计时将使用默认值。

该函数的输出如下:

```
sigma2: 新息的方差
llkhood: 模型的对数似然
aic: 模型的 AIC
arcoef: AR 系数向量
macoef: MA 系数向量
```

```
> data(Sunspot) # 太阳黑子数数据
> y <- log10(Sunspot) # y = 太阳黑子数数据的对数
>
> # ARMA(3,3) 模型
> z1 <- armafit(y, ar.order=3, ma.order=3)
```

在太阳黑子数数据的情况下, 当 ARMA 阶数为 (3,3) 时, 函数 `armafit` 的输出如下:

表 10.2 对太阳黑子数数据的对数序列拟合各种阶数 ARMA(m, ℓ) 模型时的对数似然

		ℓ					
		0	1	2	3	4	5
m	0	-157.56	-68.69	-33.75	-25.38	-22.61	-21.14
	1	-50.72	-29.36	-65.52	-22.98	-25.34	-22.12
	2	-18.63	-15.72	-15.61	-12.29	-4.91	-4.62
	3	-16.64	-15.72	-48.68	-1.95	-4.69	-1.54
	4	-15.35	-15.06	-6.58	-0.47	1.90	-1.56
	5	-14.57	-14.50	-1.39	1.79	2.99	-0.92

σ^2	5.80310e-02		
对数似然	-0.506		
AIC	15.013		
AR 系数	2.541	-2.367	0.804
MA 系数	1.581	-0.511	-0.177

表 10.1 总结了使用函数 `armafit` 对太阳黑子数数据的对数值在 $0 \leq m \leq 5, 0 \leq \ell \leq 5$ 范围内独立拟合 36 个 ARMA 模型时的 AIC 值。根据此表判断, 使 AIC 最小的 $m = 5, \ell = 4$ 的模型是最好的。另一方面, 表 10.2 总结了估计模型的对数似然值。看这个表可以发现, (1,2) 阶、(3,2) 阶、(1,4) 阶等的值与周围的值相比过小, 这是因为它们没有从默认初始值收敛到对数似然的最大点。例如, 如果在 (3,2) 阶 ARMA 模型中限制 $a_3 = 0$, 则变为 (2,2) 阶 ARMA 模型。因此, (3,2) 阶 ARMA 模型的对数似然最大值应该大于 (2,2) 阶模型的对数似然最大值。通常, (m, ℓ) 阶 ARMA 模型的对数似然应该比 ($m-1, \ell$) 阶和 ($m, \ell-1$) 阶模型的大。与此相对应, 在表 10.1 的 AIC 情况下, 不应该增加 2 以上。因此, 如果对数似然比左侧或上方的模型减少, 或者 AIC 增加 2 以上, 则表明它没有收敛到对数似然的全局最大值。在表 10.2 中, 这种情况用粗体表示。

10.5 关于参数的初始值

为缓解上一节中所指出的 ARMA 模型在最大似然估计过程中容易收敛到局部解的问题, 这里将讨论 AR 系数与 MA 系数初始值的设定方法。

在 $\ell = 0$ 的 AR 模型情形下, AR 系数可以相对容易地求得。此外, 如有需要, 还可以通过 Yule-Walker 方法或最小二乘法得到较好的近似值, 并将其作为初始值加以利用。

另一方面, 对于一般的 ARMA 模型, 当 $\ell > 0$ 时, 对数似然函数往往具有多个极大点, 因此根据初始值的不同, 可能会得到不同的估计结果。因此, 即便使用时间序列分析软件中自动设定的默认初始值, 并对各个阶数的模型分别进行估计, 也并不总能得到令人满意的结果。

在这种情况下, 若将 ARMA(m, ℓ) 模型的初始值设定为 ARMA($m-1, \ell$) 与 ARMA($m, \ell-1$) 两个模型中对数似然值较大的那个模型的最大似然估计结果, 则可以保证所得到的新模型其对数似然值必然不小于该初始模型。

例: 太阳黑子数数据 (续)

TSSS 软件包中的函数 `armafit2` 采用上述 AR 系数与 MA 系数的初始值设定方法, 对不超过指定阶数的所有 ARMA 模型进行估计。在使用时, 需要给定 AR 阶数的最大值 (`ar.order`) 以及 MA 阶数的最大值 (`ma.order`)。AR 系数向量和 MA 系数向量的初始值均由程序自动设定。

该函数的输出如下:

<code>aicmin:</code>	所考察的模型中 AIC 的最小值
<code>maice.order:</code>	使 AIC 最小的 ARMA 阶数
<code>sigma2:</code>	所有 ARMA 模型的方差
<code>llkhood:</code>	所有 ARMA 模型的对数似然
<code>aic:</code>	所有 ARMA 模型的 AIC
<code>coef:</code>	所有 ARMA 模型的 AR 系数向量及 MA 系数向量

```
> data(Sunspot) # 太阳黑子数数据
> y <- log10(Sunspot) # y = log( 太阳黑子数 )
> armafit2(y, ar.order = 5, ma.order = 5)
```

表 10.3 和表 10.4 显示了通过函数 `armafit2` 求得的 ARMA 模型的对数似然和 AIC。结果表明, 对于所有的 i 和 j , 对数似然满足 $\ell(i, j) \geq \ell(i-1, j)$ 和 $\ell(i, j) \geq \ell(i, j-1)$ 。然而, 需要注意的是, 即使满足上述条件, 也不能保证它收敛到了全局最大值。ARMA 模型是一种表现力极其丰富的模型, 因此当拟合 AR 阶数或 MA 阶数较大的 ARMA 模型时, 它可能会对频谱中的微小波峰或波谷产生反应。因此, 在估计 ARMA 模型后, 通常建议绘制对应的频谱图, 确认是否出现了不自然的波谷。

表 10.3 通过函数 `armafit2` 得到的 ARMA 模型的对数似然

		ℓ					
		0	1	2	3	4	5
m	0	-157.56	-68.69	-33.75	-25.38	-22.61	-21.14
	1	-50.72	-29.36	-23.88	-22.98	-22.11	-19.26
	2	-18.63	-15.72	-15.61	-12.29	-4.91	-4.62
	3	-16.75	-15.65	-15.50	-2.33	-1.89	-0.18
	4	-15.35	-15.06	-6.58	-2.33	-0.45	-0.05
	5	-14.57	-14.50	-1.39	0.76	3.62	3.86

表 10.4 通过函数 `armafit2` 得到的 ARMA 模型的 AIC

		ℓ					
		0	1	2	3	4	5
m	0	317.12	141.38	73.50	58.75	55.22	54.29
	1	105.43	64.72	55.75	55.96	56.21	52.52
	2	43.25	39.44	41.21	36.57	23.82	25.23
	3	41.50	41.30	43.00	18.67	19.78	18.75
	4	40.70	42.13	27.16	20.66	18.91	20.37
	5	41.14	42.99	18.78	16.48	12.77	14.27

● 章末习题

1. 使用 (9.5) 式的表示方法求 m 阶 AR 模型的最大似然估计值时, 求其初始方差协方差矩阵。
2. 请陈述在 ARMA 模型参数估计中使用 (10.9) 式进行变量转换时的注意事项。

第 11 章

趋势的估计

在经济时间序列等数据中，常常可以观察到长期持续增加或减少的倾向。本章前半部分将介绍多项式回归模型，作为包含这种变动倾向的时间序列的分析方法。接着，为了估计多项式回归模型等简单参数模型无法表达的复杂趋势，引入趋势模型。表达参数随机变化的趋势分量模型，是后续各章介绍的各种非平稳时间序列模型的基础。

11.1 多项式回归模型

图 1.1(e) 的 WHARD 数据在整个区间内呈现增长趋势。这种在经济时间序列等中经常出现的长期趋势变动称为**趋势** (trend)。不言而喻，为了把握时间序列的走势和进行预测，估计趋势是必要的。即使目的是分析更短期的变动分量，也不宜将 AR 模型等平稳模型直接应用于原序列，而是通常先用某种方法去除趋势后再对序列进行分析。

本节将介绍作为估计趋势最简单方法的**多项式回归模型** (polynomial trend model)。多项式回归模型假设时间序列 y_n 可以表示为多项式与残差之和

$$y_n = t_n + w_n \quad (11.1)$$

其中， w_n 为均值 0、方差 σ^2 的正态白噪声， t_n 为 x_n 的多项式

$$t_n = a_0 + a_1 x_n + \cdots + a_m x_n^m \quad (11.2)$$

由于该模型是回归模型的特殊情形，可以用第 5 章的方法方便地求解。为了求多项式趋势模型，在 (5.1) 中令第 j 个解释变量为 $x_{nj} = x_n^{j-1}$ ，构造

$$X = \begin{bmatrix} 1 & x_1 & \cdots & x_1^m & y_1 \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & x_N & \cdots & x_N^m & y_N \end{bmatrix} \quad (11.3)$$

即可。此时通过 Householder 变换 U ，缩约为上三角矩阵

$$UX = \begin{bmatrix} S \\ O \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots & s_{1,m+2} \\ & \ddots & \vdots \\ & & s_{m+2,m+2} \\ & & & 0 \end{bmatrix} \quad (11.4)$$

则 j 阶多项式趋势模型的残差方差 $\hat{\sigma}_j^2$ 可由

$$\hat{\sigma}_j^2 = \frac{1}{N} \sum_{i=j+2}^{m+2} s_{i,m+2}^2 \quad (11.5)$$

求得。此外， j 阶模型具有 $j+1$ 个回归系数和残差方差作为参数，其 AIC 为

$$AIC_j = N \log 2\pi \hat{\sigma}_j^2 + N + 2(j+2) \quad (11.6)$$

使其最小的 j 即为最优阶数。对于这样确定的阶数 j ，通过求解一次方程

$$\begin{bmatrix} s_{11} & \cdots & s_{1,j+1} \\ & \ddots & \vdots \\ & & s_{j+1,j+1} \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_j \end{bmatrix} = \begin{bmatrix} s_{1,m+2} \\ \vdots \\ s_{j+1,m+2} \end{bmatrix} \quad (11.7)$$

可求得回归系数的最大似然估计值 $\hat{a}_0, \dots, \hat{a}_j$ 。由于矩阵 S 为上三角矩阵，该一次方程可通过后向代入简单地求解。

例：最高气温数据

R 的 TSSS 程序包中的函数 `polreg` 用于对时间序列拟合多项式回归模型。通过参数 `order` 指定多项式回归的最高阶数，即可用 AIC 确定最优阶数。函数的输出如下：

```
order.maice: AIC 最小的趋势阶数
sigma2:     阶数为  $0 \leq m \leq M$  的模型的残差方差
aic:       阶数为  $0 \leq m \leq M$  的模型的 AIC
daic:     AIC - 最小 AIC
coef:     回归系数  $a(i,m)$ ,  $i = 0, \dots, m$ ,  $m = 1, \dots, M$ 
trend:    AIC 最小模型的趋势估计值
```

利用函数 `polreg` 的输出，例如要绘制 AIC 的变化图，可以将 `polreg` 的输出保存到 `z` 中，AIC 的信息通过 `z$aic` 获取。

```

> # 东京最高气温数据
> data( Temperature )
> polreg( Temperature, order = 13 )
>
> # 显示多项式回归模型 AIC 的变化
> z <- polreg( Temperature, order = 13 )
> plot( z$aic,type="b",lwd=2 )
>
> # 批发额数据
> data( WHARD )
> y <- log10( WHARD )
> polreg( y, order = 15 )

```

表 11.1 是对图 1.1(c) 所示的最高气温数据拟合多项式回归模型时的残差方差 $\hat{\sigma}_j^2$ 和 AIC_j 的汇总。残差方差 $\hat{\sigma}_j^2$ 单调递减，而 AIC 在 6 阶时取得最小值。

表 11.1 最高气温数据

j	$\hat{\sigma}_j^2$	AIC_j	j	$\hat{\sigma}_j^2$	AIC_j
0	60.09	3373.76	7	8.96	2463.09
1	58.89	3366.02	8	8.96	2465.02
2	33.61	3095.35	9	8.91	2464.24
3	23.74	2928.47	10	8.89	2465.29
4	10.18	2518.95	11	8.89	2467.27
5	9.65	2494.72	12	8.86	2467.22
6	8.97	2461.42	13	8.86	2469.18

同样，表 11.2 是 WHARD 数据对数值的情况。在此情况下，10 阶时 AIC 取得最小值。

表 11.2 WHARD 数据

j	$\hat{\sigma}_j^2$	AIC_j	j	$\hat{\sigma}_j^2$	AIC_j
0	0.02752	-113.04	7	0.00122	-582.11
1	0.00163	-548.73	8	0.00115	-589.46
2	0.00150	-560.34	9	0.00112	-591.45
3	0.00149	-559.11	10	0.00107	-595.98
4	0.00147	-559.56	11	0.00107	-594.18
5	0.00142	-562.75	12	0.00106	-594.38
6	0.00123	-583.37	13	0.00105	-593.59

图 11.1 的 (a) 和 (b) 分别展示了各时间序列及使用 AIC 最小阶数的多项式回归模型估计的趋势。(a) 的最高气温数据的情况下，捕捉到了气温的缓慢变化。(b) 的

WHARD 数据的情况下, 整体上获得了平滑的估计值, 但在数据的开始部分趋势呈现急剧变化, 且在对应于 1974~75 年的 $n = 95$ 附近的急剧下降未能被很好地捕捉。对于这种问题的改进趋势估计方法将在 11.3 节以后说明。

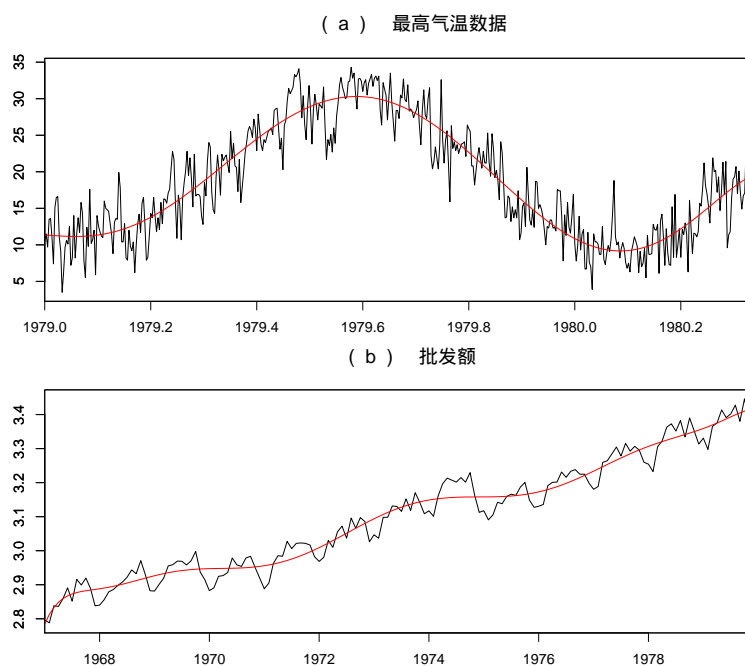


图 11.1 最高气温数据和 WHARD 数据的多项式趋势

11.2 趋势分量模型 – 结构随机变化的模型

前节的多项式趋势模型可以表示为

$$y_n \sim N(t_n, \sigma^2) \quad (11.8)$$

因此该模型可以说是假设方差恒定、但均值按多项式 t_n 变化的模型。这种参数模型在真实趋势确实遵循多项式、且 w_n 可视为白噪声的情况下能给出非常好的估计值，但在其他情况下，可能无法很好地捕捉趋势的变化，或者相反地过度追随数据而拾取偶然变动。这种现象尤其在数据区间的两端经常出现。下面我们考虑为了改善这些问题，将多项式回归模型扩展为更灵活的模型的方法。作为具体例子，考虑表示 1 次多项式（直线）的方程

$$t_n = an + b \quad (11.9)$$

此时，将 $\Delta \equiv 1 - B$ 定义为由 $\Delta t_n = t_n - t_{n-1}$ 给出的时间差分算子则

$$\Delta t_n = a, \quad \Delta^2 t_n = 0 \quad (11.10)$$

成立。这表明 1 次多项式是 2 阶差分方程初值问题

$$\Delta^2 t_n = 0, \quad \Delta t_0 = a, \quad t_0 = b \quad (11.11)$$

的解。一般地, $k-1$ 次多项式可以看作 k 阶差分方程

$$\Delta^k t_n = 0 \quad (11.12)$$

的解。

作为将多项式扩展为更灵活函数的方法, 可以考虑用 t_n 的 k 阶差分近似为 0 来代替差分方程, 即要求 $\Delta^k t_n \approx 0$ 成立。这可以通过引入如下 k 阶随机差分方程

$$\Delta^k t_n = v_n \quad (11.13)$$

来实现。其中, v_n 为均值 0、方差 τ^2 的正态分布 $N(0, \tau^2)$ 的白噪声。以下将 (11.13) 称为**趋势分量模型** (trend component model)。

由于 $\Delta^k t_n = 0$ 的解为 $k-1$ 次多项式, k 阶趋势分量模型可以看作 $k-1$ 次多项式的扩展。当噪声方差 τ^2 较小时, 趋势分量模型的实现值在局部上是类似多项式的非常光滑的函数。但是, 与多项式的显著区别在于, 从整体上看可以表达非常灵活的函数。

例 当 $k=1$ 时, 该模型变为**随机游走模型** (random walk model)

$$t_n = t_{n-1} + v_n, \quad v_n \sim N(0, \tau^2) \quad (11.14)$$

表达了趋势在局部近似取恒定值 $t_n \approx t_{n-1}$ 的模型。此外, 当 $k=2$ 时

$$t_n = 2t_{n-1} - t_{n-2} + v_n \quad (11.15)$$

表达了趋势在局部近似线性变化 $t_n - 2t_{n-1} + t_{n-2} \approx 0$ 的模型。

一般地, 趋势分量模型 (11.13) 由于

$$\Delta^k = (1-B)^k = \sum_{i=0}^k {}_k C_i (-B)^i \quad (11.16)$$

通过 $c_i = (-1)^{i+1} {}_k C_i$ 定义系数 c_i , 可以表示为

$$t_n = \sum_{i=1}^k c_i t_{n-i} + v_n \quad (11.17)$$

当 $k=1$ 时 $c_1=1$, 当 $k=2$ 时 $c_1=2, c_2=-1$ 。该模型不满足平稳性, 但形式上可以看作 k 阶 AR 模型, 因此令

$$\begin{aligned} x_n &= \begin{bmatrix} t_n \\ t_{n-1} \\ \vdots \\ t_{n-k+1} \end{bmatrix}, \quad F = \begin{bmatrix} c_1 & c_2 & \cdots & c_k \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ H &= [1 \quad 0 \quad \cdots \quad 0] \end{aligned} \quad (11.18)$$

可得趋势分量模型的状态空间表示

$$\begin{aligned}x_n &= Fx_{n-1} + Gv_n \\t_n &= Hx_n\end{aligned}\quad (11.19)$$

例 当 $k=1$ 时

$$x_n = t_n, \quad F = G = H = 1 \quad (11.20)$$

即可。此外, 当 $k=2$ 时

$$x_n = \begin{bmatrix} t_n \\ t_{n-1} \end{bmatrix}, \quad F = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad H = [1 \ 0] \quad (11.21)$$

或者

$$x_n = \begin{bmatrix} t_n \\ -t_{n-1} \end{bmatrix}, \quad F = \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad H = [1 \ 0] \quad (11.22)$$

即可。(11.22) 给出了第 9 章所示的标准形。

此外, 当 $k=2$ 时, 将状态向量定义为 $x_n = (t_n, \delta t_n)^T$, 令

$$F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad H = [1 \ 0] \quad (11.23)$$

也可以。在该模型中 $\delta t_n \equiv \Delta t_n$ 成立, 可以简单地验证这与 (11.21) 是等价的表示。

但是, 使用该表示可以将趋势分量模型自然地进一步扩展。即令

$$v_n = \begin{bmatrix} v_{n1} \\ v_{n2} \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad H = [1 \ 0] \quad (11.24)$$

则趋势分量满足

$$\begin{aligned}\delta t_n &= \delta t_{n-1} + v_{n2} \\t_n &= t_{n-1} + \delta t_{n-1} + v_{n1} + v_{n2} \\&= t_{n-1} + \delta t_n + v_{n1}\end{aligned}\quad (11.25)$$

这一模型。在这种情况下 $\Delta t_n = \delta t_n + v_{n1}$, 与之前使用的趋势分量模型具有一维系统噪声不同, 这是一个在趋势的水平 and 斜率两者上都加入独立噪声的更灵活的模型。

11.3 趋势模型

趋势表示时间序列的中长期变动倾向。反过来说, 实际观测到的时间序列是趋势上叠加了各种变动分量的结果。其中最简单的情况是以 w_n 为白噪声

$$y_n = t_n + w_n \quad (11.26)$$

可以这样表示的情况。这是表示获得观测值机制的最简单模型，可以看作第 9 章观测模型的一个例子。为了从时间序列 y_n 中估计趋势 t_n ，可以考虑将趋势分量模型和观测模型组合而成的如下模型。将其称为**趋势模型** (trend model)。

$$\Delta^k t_n = v_n \quad (11.27)$$

$$y_n = t_n + w_n \quad (11.28)$$

其中， v_n 与 (11.13) 相同，为服从均值 0、方差 τ^2 正态分布的白噪声， w_n 为服从均值 0、方差 σ^2 正态分布的白噪声。(11.28) 是将时间序列 y_n 看作在趋势上叠加了独立噪声后的观测结果的观测模型。另一方面，(11.27) 是对趋势变化方式进行建模的趋势分量模型。实际上，通常不具备这样简单的结构，在那种情况下需要如下章所示的更复杂的建模。

利用前节的状态分量模型的状态空间表示，趋势模型可以用如下状态空间模型表示。

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n \\ y_n &= Hx_n + w_n \end{aligned} \quad (11.29)$$

其中，状态向量为适当定义的 k 维向量， F 、 G 、 H 分别为从 (11.27)、(11.28) 适当确定的 $k \times k$ 矩阵、 k 维列向量、 k 维行向量。与趋势分量模型的区别仅在于包含了观测噪声。例如，当 $k = 2$ 时

$$\begin{aligned} x_n &= \begin{bmatrix} t_n \\ t_{n-1} \end{bmatrix}, \quad F = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ H &= [1 \quad 0] \end{aligned} \quad (11.30)$$

即可。

趋势模型的阶数 k 以及方差 τ^2 、 σ^2 确定后，通过卡尔曼滤波和平滑算法可以求得状态向量的平滑值 $x_{1|N}, \dots, x_{N|N}$ 。此时，由于状态向量的第 1 个分量是 t_n ， $x_{n|N}$ 的第 1 个分量即 $Hx_{n|N}$ 就是趋势的平滑值 $t_{n|N}$ 。

例：最高气温数据的趋势

TSSS 程序包的函数 `trend` 用于对时间序列拟合趋势模型并估计趋势。需要趋势的阶数 `trend.order`、系统噪声方差的初始值 `tau2.ini` 以及方差 τ^2 的最优化搜索步长。当 `tau2=NULL` 时，即未给定初始值的情况下，该函数在 $\tau^2 = 2^{-k}$ ， $k = 1, 2, \dots$ 的范围内比较 AIC 值，确定 τ^2 的粗略估计值。以该估计值为初始值反复计算，即可求得高精度的估计值。

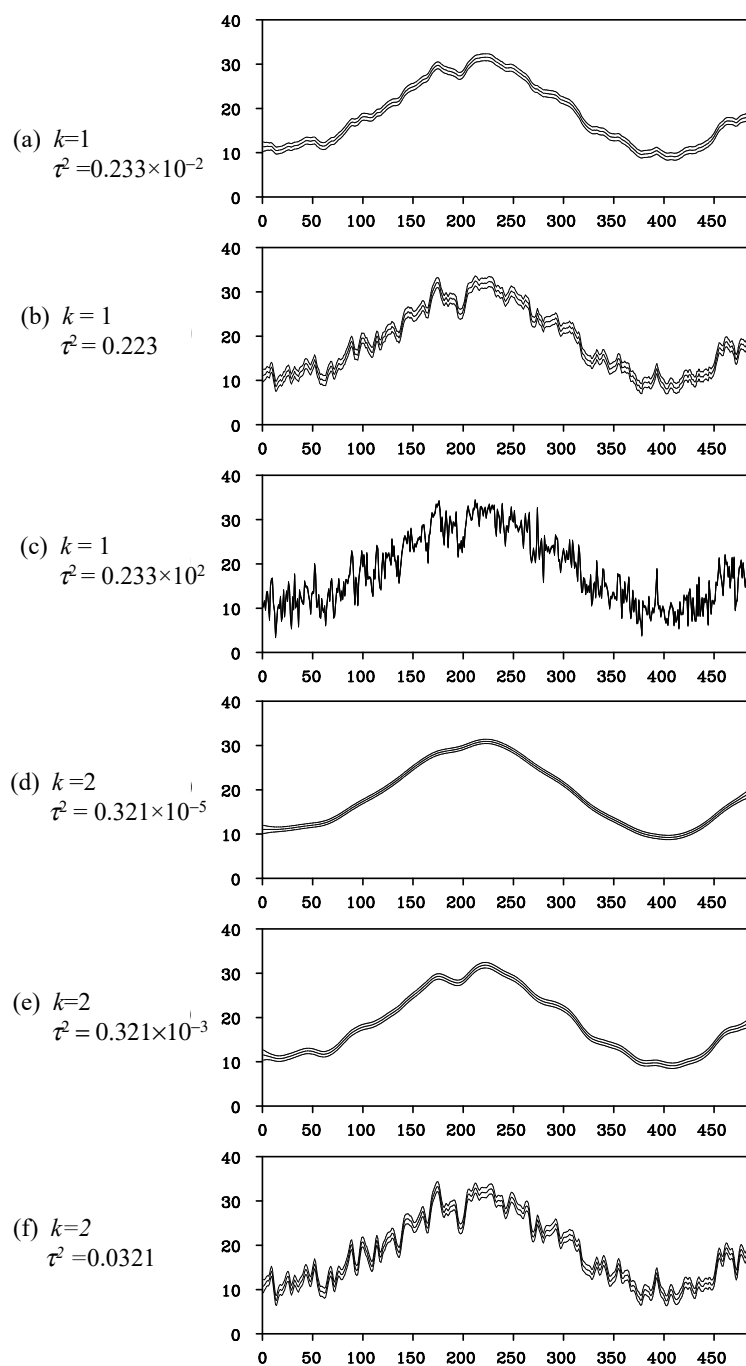
图 11.2 使用不同 k 和 τ^2 的趋势模型估计的气温数据趋势

表 11.3 趋势模型的 AIC

$k = 1$		$k = 2$	
τ^2	AIC	τ^2	AIC
0.223×10^{-2}	2690	0.321×10^{-5}	2556
0.223	2448	0.321×10^{-3}	2506
0.223×10^2	2528	0.0321	2562

该函数的输出包括趋势的估计值 (trend)、残差 (residual)、系统噪声方差 τ^2 (tau2)、观测噪声方差 σ^2 (sigma2)、模型的对数似然 (llhood) 以及 AIC (aic)。

```

> # 东京日最高气温
> data( Temperature )
> # 1 阶趋势模型
> trend( Temperature, trend.order = 1, tau2.ini = 0.223, delta = 0.001 )
>
> # 2 阶趋势模型
> trend( Temperature, trend.order = 2 )

```

图 11.2 展示了对气温数据使用不同阶数 k 和系统噪声方差 τ^2 的 6 个模型求得趋势估计值, 其中 σ^2 使用了最大似然法估计的值。(a) 是 $k = 1$ 、 $\tau^2 = 0.223 \times 10^{-2}$ 的情况。估计的趋势基本捕捉到了气温的年周期变化, 与多项式趋势模型的估计值类似。当 $k = 1$ 、 $\tau^2 = 0.223$ 时的 (b) 中, 趋势估计值再现了更详细的变动。(c) 是 $k = 1$ 、 $\tau^2 = 0.223 \times 10^2$ 的情况。趋势只是在追随数据的变动。

另一方面, 当 $k = 2$ 时, $\tau^2 = 0.321 \times 10^{-5}$ 的 (d) 过于简单, $\tau^2 = 0.0321$ 的 (f) 过于复杂, 而 $\tau^2 = 0.321 \times 10^{-3}$ 的 (e) 则获得了恰到好处的估计值。进一步比较 (b) 和 (e), (e) 获得了更平滑的估计值。

趋势模型中包含阶数 k 和方差 σ^2 与 τ^2 作为参数。如上例所示, 根据这些值的不同, 可以获得各种不同的趋势估计值。因此, 为了获得好的趋势估计值, 需要将这些值设定为适当的值。 σ^2 和 τ^2 可以通过最大似然法求出最优值, 利用 9.6 节的方法, 实际上只需选择 $\lambda = \frac{\tau^2}{\sigma^2}$ 的值, σ^2 的值就会自动确定。由此可知, 决定趋势估计值的是系统噪声与观测噪声的方差比, 称为折中参数 (trade-off parameter)。

趋势模型的阶数可以通过 AIC 来确定, 通常多使用 $k = 2$ 。但是, 当趋势变化幅度较大时, $k = 1$ 可能更好。

表 11.3 汇总了图 11.2 所示模型的 AIC 值。在 $k = 1$ 中 (b) 最小, 在 $k = 2$ 中 (e) 最

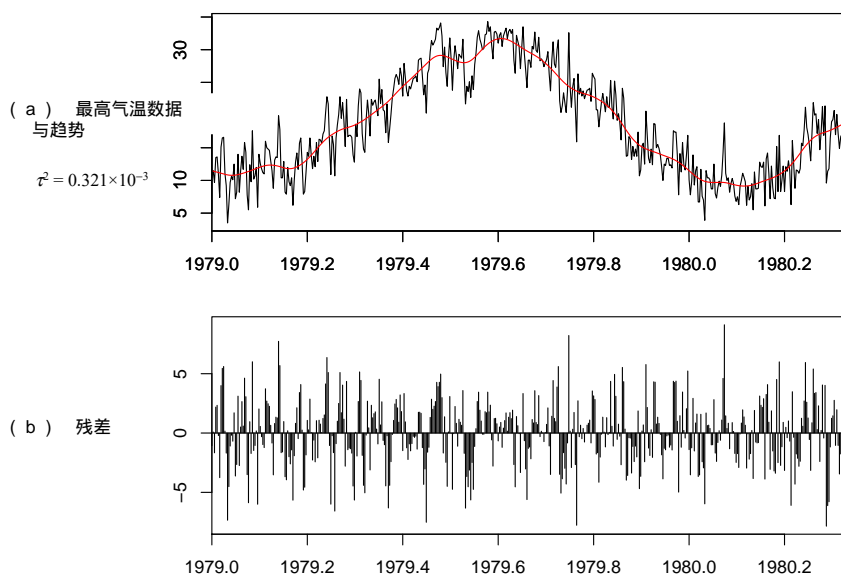


图 11.3 函数 trend 的输出示例

小。此外, (b) 和 (e) 相比, (b) 明显更小, 表明 (b) 是所有模型中最好的估计值。注意, (b) 和 (e) 的 τ^2 值是通过最大似然法求得的最优值。

根据 AIC, 比起 $k=2$ 所求得平滑趋势, 选择了 $k=1$ 的变动较大的趋势。这表明在趋势模型中 $k=1$ 更好, 但需要注意的是, 假设 w_n 为白噪声的趋势模型本身可能存在问题。图 11.3 展示了对最高气温数据使用 $k=2$ 时函数 trend 的输出。得到了 $\hat{\tau}^2 = 0.321 \times 10^{-3}$ 。在这种情况下, 残差序列呈现序列相关, 难以认为是白噪声。

如下一章所示, 在这种情况下, 通过使用季节调整模型将时间序列分解为趋势、AR 分量和观测噪声三个分量, 可以获得既具有平滑趋势又对数据有良好拟合的模型。这里附带说明, 用于估计季节调整模型的函数在不包含季节分量的情况下也可以使用。

● 章末习题

1. 根据股价的随机游走假说, 股价 y_n 服从随机游走模型 $y_n = y_{n-1} + v_n$, ($v_n \sim N(0, \sigma^2)$)。
 - (1) 当某日 n 的日经平均股价为 $y_n = 17,000$ 且 $\sigma^2 = 40,000$ 时, 求 k 日后 ($k = 1, \dots, 5$) 的股价预测值及其预测误差的方差。

- (2) 求 4 日后达到 17,400 日元以上的概率。
- (3) 实际股价是否满足随机游走假说? 如果存在问题, 应如何改进模型?
- (4) 尝试估计实际股价数据的趋势。

第 12 章

季节调整模型

本章将说明作为趋势模型扩展的一个例子——季节调整模型。经济时间序列中存在每年同一时期呈现相同变动模式的特有模式。对这类时间序列进行走势判断或预测时,如果不考虑该特征,就有可能得出错误的结论。季节调整模型是用于分析以固定时间间隔重复相同模式的时间序列的模型。通过季节调整模型,可以将时间序列 y_n 分解为趋势 t_n 、季节分量 s_n 和白噪声 w_n 三个分量,表示为 $y_n = t_n + s_n + w_n$ 。同时也将说明包含平稳 AR 分量和星期效应项的扩展模型。

12.1 季节分量模型

时间序列变动中每年重复出现的变动模式部分 s_n 称为**季节分量**(seasonal component)。以下考虑一个周期内有 p 个观测值的情况。月度数据的情况下 $p = 12$, 季度数据的情况下 $p = 4$ 。此时,将季节分量记为 s_n , 则 s_n

$$s_n = s_{n-p} \quad (12.1)$$

近似成立。利用时间延迟算子 B 来表示,由 $B^p s_n = s_{n-p}$ 可知

$$(1 - B^p)s_n = 0 \quad (12.2)$$

近似成立。

与第 11 章的随机趋势分量模型的情况类似,作为随时间缓慢变化的季节分量的模型,可以考虑

$$(1 - B^p)^\ell s_n = v_{n2}, \quad v_{n2} \sim N(0, \tau_2^2) \quad (12.3)$$

特别地,令 $\ell = 1$ 时,得到假设 $\{s_{pn+i}\}$ ($n = 1, 2, \dots$) 为随机游走的模型

$$s_n = s_{n-p} + v_{n2} \quad (12.4)$$

因此,将前章的模型作为趋势分量 t_n 的模型,由 (11.15)、(12.4) 以及 $y_n = t_n + s_n + w_n$, 即可得到季节调整的基本模型。

然而, 作为季节分量模型最自然的 (12.4) 在实际中并不能很好地适用。其原因在于, 比较趋势分量模型和季节分量模型的算子表示 (11.18) 和 (12.3), 可以发现

$$(1 - B^p)^\ell = (1 - B)^\ell (1 + B + \cdots + B^{p-1})^\ell$$

因此包含 $(1 - B)^q$ ($q \geq 1$) 作为公共因子。此时令 e_n 为差分方程

$$(1 - B)^q e_n = 0 \quad (12.5)$$

的任意解。当 $q = 1$ 时 e_n 取任意常数即可。这里定义

$$\begin{aligned} t'_n &= t_n + e_n \\ s'_n &= s_n - e_n \end{aligned}$$

则它们分别满足 (11.15) 和 (12.3), 而且

$$y_n = t'_n + s'_n + w_n \quad (12.6)$$

成立。因此, 对于完全相同的 v_{n1} 、 v_{n2} 、 w_n , 存在多种分解。而且, 由于与这些分解对应的模型的似然由 v_{n1} 、 v_{n2} 、 w_n 确定, 因此无法判断这些分解的优劣。这样, 在时间序列分解中使用具有公共因子的分量时, 表示的唯一性就会丧失。

保证分解唯一性的一个简单方法是, 使各分量模型不具有公共因子。因此, 注意到要使 $(1 - B^p)^\ell = 0$ 成立, 只需

$$(1 + B + \cdots + B^{p-1})^\ell = 0 \quad (12.7)$$

成立即可。为了使 $S_n \approx S_{n-p}$, 只需

$$\sum_{i=0}^{p-1} B^i s_n \approx 0 \quad (12.8)$$

成立即可。因此, 作为允许时间变化的季节变动分量的随机模型, 可以考虑

$$\left(\sum_{i=0}^{p-1} B^i \right)^\ell s_n = v_{n2}, \quad v_{n2} \sim N(0, \tau_2^2) \quad (12.9)$$

本书将该模型称为阶数 ℓ 、周期 p 的**季节分量模型** (seasonal component model)。但是, 除了季节分量呈现显著趋势性变化的情况外, 通常使用 $\ell = 1$ 的一阶模型

$$\sum_{i=0}^{p-1} s_{n-i} = v_{n2}, \quad v_{n2} \sim N(0, \tau_2^2) \quad (12.10)$$

为了求得季节分量模型的状态空间表示, 首先将

$$\left(\sum_{i=0}^{p-1} B^i \right)^\ell = 1 - \sum_{i=1}^{\ell(p-1)} d_i B^i \quad (12.11)$$

展开。当 $\ell = 1$ 时 $d_i = -1$, 当 $\ell = 2$ 时 $d_i = -i - 1$, ($i \leq p - 1$), $d_i = i + 1 - 2p$, ($p \leq i \leq 2(p - 1)$)。在这种情况下, 季节分量模型形式上也是 AR 模型, 因此其状态空间表示为

$$x_n = \begin{bmatrix} s_n \\ s_{n-1} \\ \vdots \\ s_{n-\ell(p-1)+1} \end{bmatrix}, \quad F = \begin{bmatrix} d_1 & d_2 \cdots & d_{\ell(p-1)} \\ 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$H = [1, 0, \cdots, 0] \quad (12.12)$$

12.2 标准季节调整模型

在标准的季节调整中, 将时间序列 y_n

$$y_n = t_n + s_n + w_n \quad (12.13)$$

分解为趋势分量 t_n 、季节分量 s_n 、观测噪声 w_n 三个分量。利用前面推导的趋势分量模型和季节分量模型, 标准季节调整模型为

$$y_n = t_n + s_n + w_n \quad (\text{观测模型}) \quad (12.14)$$

$$\Delta^k t_n = v_{n1} \quad (\text{趋势分量模型}) \quad (12.15)$$

$$\left(\sum_{i=0}^{p-1} B^i \right)^\ell s_n = v_{n2} \quad (\text{季节分量模型}) \quad (12.16)$$

其中 $w_n \sim N(0, \sigma^2)$, $v_{n1} \sim N(0, \tau_1^2)$, $v_{n2} \sim N(0, \tau_2^2)$ 。将该模型称为标准的**季节调整模型** (seasonal adjustment model)。

趋势分量模型的阶数为 k , 周期为 p 时, 定义 $k + (p - 1)\ell$ 维状态向量 $x_n = (t_n, \cdots, t_{n-k+1}, s_n, s_{n-1}, \cdots, s_{n-(p-1)\ell+1})^T$, 噪声向量 $v_n = (v_{n1}, v_{n2})^T$, 令

$$F = \begin{bmatrix} F_1 & \mathbf{O} \\ \mathbf{O} & F_2 \end{bmatrix}, \quad G = \begin{bmatrix} G_1 & \mathbf{O} \\ \mathbf{O} & G_2 \end{bmatrix}, \quad H = [H_1 \ H_2] \quad (12.17)$$

则标准季节调整模型可由

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n \\ y_n &= Hx_n + w_n \end{aligned} \quad (12.18)$$

这一状态空间模型表示。其中, F_1 、 G_1 和 H_1 是趋势分量模型状态空间表示中使用的矩阵和向量, F_2 、 G_2 和 H_2 是季节分量模型状态空间表示中使用的矩阵和向量。

例如, 当 $k = 2$ 、 $\ell = 1$ 、 $p = 4$ 时

$$F = \begin{bmatrix} 2 & -1 & & & & \\ 1 & 0 & & & & \\ & & -1 & -1 & -1 & \\ & & 1 & & & \\ & & & & & 1 \end{bmatrix}, \quad G = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (12.19)$$

$$H = [1 \ 0 \ 1 \ 0 \ 0 \ 0]$$

即可。

例：WHARD 批发额数据的季节调整

R 程序包 TSSS 中的函数 `season` 用于拟合季节调整模型，估计趋势分量和季节分量。通过指定以下参数，可以求得各种季节调整模型：

<code>trend.order:</code>	趋势模型的阶数 (0, 1, 2, 3)
<code>seasonal.order:</code>	季节阶数 (0, 1, 2)
<code>period:</code>	当时间序列 y 的属性为 NULL 时，需要指定季节周期的长度
<code>tau2.ini:</code>	系统噪声方差 τ^2 的初始估计值
<code>filter:</code>	指定滤波起始时点和终点的向量 $c(x1, x2)$
<code>predict:</code>	长期预测的最终时点 ($\geq x2$)。
<code>log:</code>	逻辑变量。为 TRUE 时使用数据 y 的对数变换
<code>minmax:</code>	观测值的最小值和最大值。其他值视为异常值
<code>plot:</code>	逻辑变量。为 TRUE (默认) 时，绘制趋势和季节分量图

季节调整法不仅适用于月度数据的年周期，还可以应用于各种周期。例如，季度数据设 `period=4`，月度数据设=12，日数据的周周期设=5 或 7，小时数据的日周期设=24 即可。函数 `season` 给出以下输出：

```
tau2:      系统噪声的方差
sigma2:    观测噪声的方差
llkhood:   模型的对数似然
aic:       模型的 AIC
trend:     趋势分量 (trend.order > 0 时)
seasonal:  季节分量 (seasonal.order > 0 时)
noise:     噪声分量
cov:       平滑估计值的方差协方差矩阵
```

```
> # 批发额数据
> data( WHARD )
> season( WHARD, trend.order =2, seasonal.order =1, log =TRUE )
```

图 12.1 是对 WHARD 数据的对数值应用标准季节调整模型，分解为趋势、季节分量和噪声的结果。最大似然法估计的参数为 $\hat{\tau}_1^2 = 0.0248$ ， $\hat{\tau}_2^2 = 0.11 \times 10^{-7}$ ， $\hat{\sigma}^2 = 0.156 \times 10^{-3}$ ， $AIC = -728.50$ 。趋势整体呈现稳定的增长倾向，但在 1974~75 年前后出现了急剧下降。此外，季节分量在整个期间内非常稳定。

函数 `season` 还可以利用估计的季节调整模型进行长期预测。为此需要通过参数

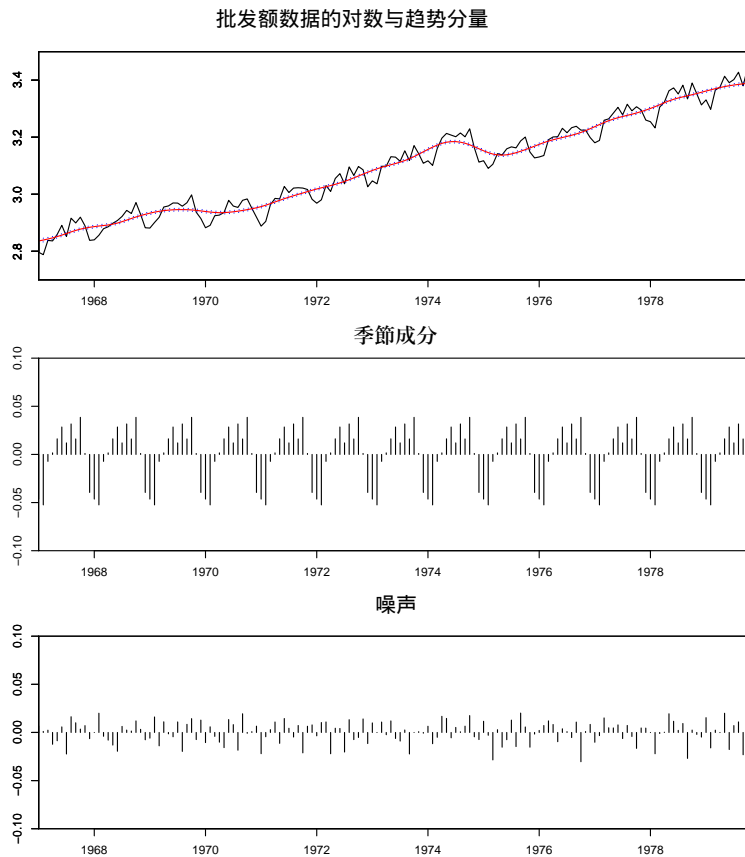


图 12.1 标准季节调整模型对 WHARD 数据的季节调整

`filter` 指定用于模型参数估计的数据区间。在以下示例中, 指定 `filter=c(1,132)`, 季节调整模型使用时间序列数据 y_1, \dots, y_{132} 进行估计, 其余数据 y_{133}, \dots, y_N 由估计的模型进行预测。

```
> # 批发额数据
> data( WHARD )
> season( WHARD, trend.order = 2, seasonal.order = 1, filter = c(1,132), log =
TRUE )
```

图 12.2 展示了 156 个数据中仅使用前 132 个观测值, 对剩余 2 年 (24 个) 时间序列用第 9 章的方法进行长期预测的结果。图中绘制了预测分布的均值 $y_{132+j|132}$ 以及 ± 1 个标准差的预测误差范围, 即 $y_{132+j|132} \pm \sqrt{d_{132+j|132}}$ ($j = 1, \dots, 24$)。同时也显示了实际的时间序列值, 由于该数据的趋势和季节分量都很稳定, 长期预测的均值与实际值重叠, 表明获得了极好的预测值。

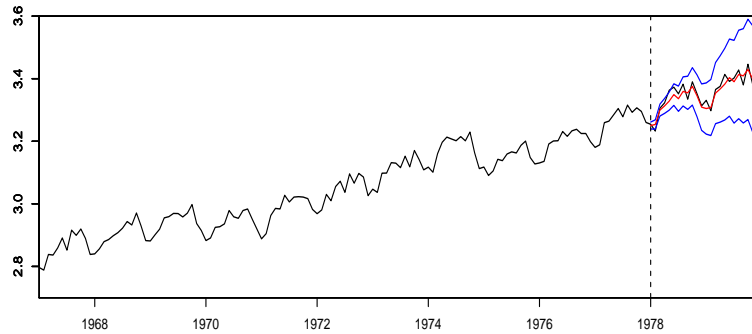


图 12.2 季节调整模型的长期预测

12.3 包含平稳 AR 分量的分解

这里考虑标准季节调整法的扩展。在标准季节调整法中，时间序列被分解为趋势分量、季节分量和观测噪声三个分量，各分量分别服从 (12.15)、(12.16) 等模型，且假设观测噪声为白噪声。因此，当严重偏离这些假设时，标准季节调整法得到的分解可能是不恰当的。图 12.3 是对图 1.1(d) 所示的食品产业从业人数数据，取 $k=2$ 、 $\ell=1$ 、 $p=12$ 进行季节调整的结果。在这种情况下，估计的趋势与图 12.1 不同，在数据的中间部分和末端呈现波动。

```
> data( BLSALLFOOD )
> season( BLSALLFOOD, trend.order = 2, seasonal.order = 1 )
```

```
> data( BLSALLFOOD )
> season( BLSALLFOOD, trend.order = 2, seasonal.order = 1, filter = c(1,132) )
```

首先考虑得到这样的趋势时的问题。图 12.4 是食品产业从业人数数据 156 个数据中，与图 12.2 类似地仅使用前 132 个观测值对剩余 2 年（24 个）时间序列进行长期预测的结果。在这种情况下，预测值 $y_{132+j|132}$ 看似很好地预测了实际时间序列 y_{132+j} ，但从预测误差范围爆发性扩大可以推断，实际上该数据的情况下，利用该模型进行预测几乎没有可信度。

图 12.5 是以 $n=132$ 为中心，前后各半年即 $n=126 \sim 138$ 的各点为预测起始点，对 $n=156$ 进行长期预测的 13 种预测值的叠加图。在图 12.3 中趋势估计值下降的 $n=130$ 以前为起始时点进行长期预测时，预测值将大幅向下方偏移。另一方

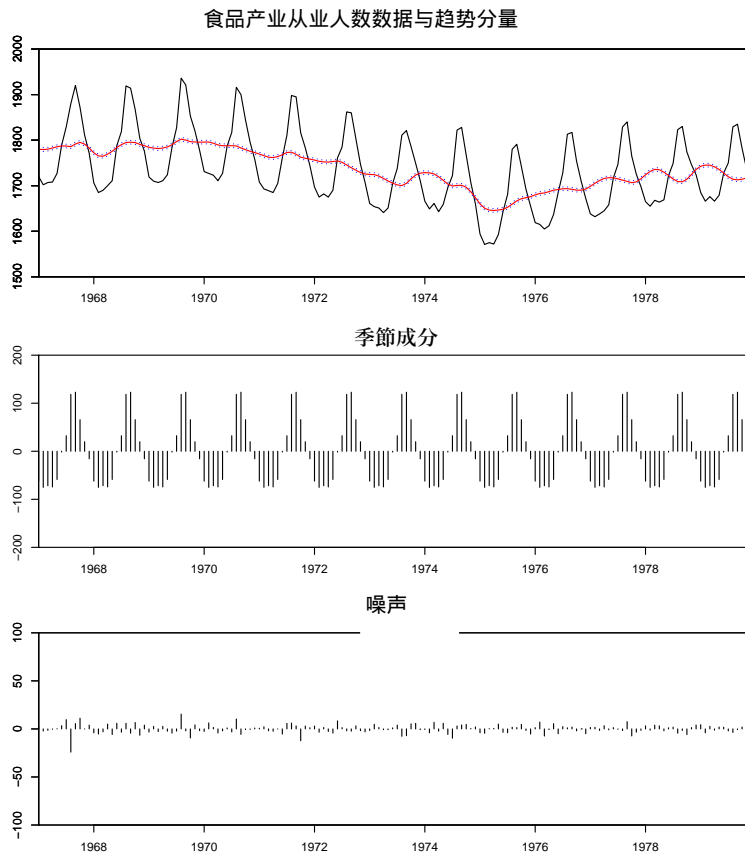
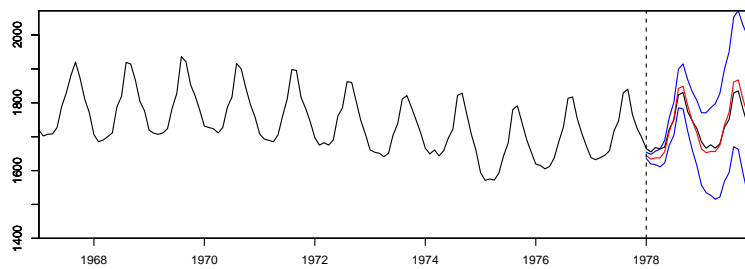


图 12.3 标准季节调整模型对食品产业从业人数数据的季节调整

图 12.4 食品产业从业人数数据的长期预测 (预测起始时点: $n = 132$)

面, 从趋势估计值呈上升趋势的 $n = 135$ 以后开始长期预测时, 预测值反而大幅向上方偏移。预测误差范围迅速扩大反映了这一情况。季节调整模型中含有噪声项的趋势分量模型虽能灵活表现复杂趋势, 但在长期预测中其均值 $t_{n+j|n}$ 与简单按差分方程

$$\Delta^k t_{n+j|n} = 0 \quad (12.20)$$

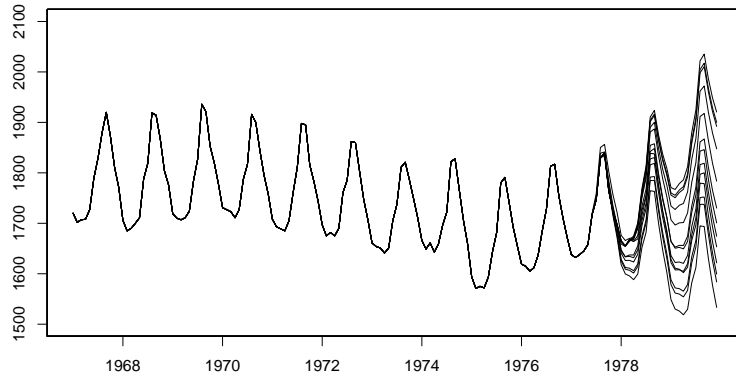


图 12.5 移动预测起始点时食品产业从业人数数据的长期预测 (预测起始点: $n=126\sim 138$)

计算的结果一致。因此, 根据趋势最终阶段的倾向, 预测值会向下或向上偏移。这样, 当估计的趋势呈现波动时, 利用标准季节调整法求得的模型来预测未来的走势是不恰当的。

如果仅以 1 年或 2 年以上的长期预测为目的, 通过将 τ_1^2 即趋势分量系统噪声的方差设定为比最大似然法估计的最优值更小的值, 可以得到更平滑的趋势, 从而可能获得较好的预测值。但是, 这种方法不仅难以合理确定 τ_1^2 的值, 而且长期预测改善的同时, 1 期前预测等短期预测能力会显著下降。

为了获得在短期预测和长期预测两方面都表现优异的模型, 需要在标准季节调整模型考虑的三个分量之外, 加入新的分量 p_n

$$y_n = t_n + s_n + p_n + w_n \quad (12.21)$$

考虑四分量的分解。其中 p_n 称为**平稳 AR 分量** (stationary AR component), 假设服从自回归模型

$$p_n = \sum_{i=1}^{m_3} a_i p_{n-i} + v_{n3} \quad (12.22)$$

其中, v_{n3} 为均值 0、方差 τ_3^2 的正态白噪声。该模型表示的不是趋势分量那样的长期倾向, 而是循环变动等短期变动分量。在 (12.21) 的模型中, 标准季节调整中用一个趋势分量表示的部分, 被自动分解为长期趋势曲线 t_n 和短期变动 p_n 两部分。如第 9 章所示, AR 模型令 $x_n = (p_n, p_{n-1}, \dots, p_{n-m_3+1})^T$,

$$F_3 = \begin{bmatrix} a_1 & a_2 & \cdots & a_{m_3} \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}, \quad G_3 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (12.23)$$

$$H_3 = [1 \ 0 \ \cdots \ 0], \quad Q_3 = \tau_3^2$$

可得状态空间模型。因此，利用第9章的模型合成方法，要进行(12.21)的分解，可使用由

$$F = \begin{bmatrix} F_1 & & \\ & F_2 & \\ & & F_3 \end{bmatrix}, \quad G = \begin{bmatrix} G_1 & 0 & 0 \\ 0 & G_2 & 0 \\ 0 & 0 & G_3 \end{bmatrix}$$

$$H = [H_1 \ H_2 \ H_3], \quad Q = \begin{bmatrix} \tau_1^2 & 0 & 0 \\ 0 & \tau_2^2 & 0 \\ 0 & 0 & \tau_3^2 \end{bmatrix} \quad (12.24)$$

定义的状态空间模型即可。

例：包含AR分量的模型的季节调整法 函数 `season` 中，要用包含AR分量的模型进行季节调整，只需如下通过参数 `ar.order` 指定AR阶数即可。

```
> data(BLSALLFOOD)
> season(BLSALLFOOD, trend.order = 2, seasonal.order = 1, ar.order = 2)
```

图12.6是将食品产业从业人数数据用该方法分解为趋势分量、季节分量、平稳AR分量和观测噪声的结果。估计的趋势与图12.1的情况类似，是非常平滑的曲线。另一方面，短期变动以平稳AR分量的形式表示，(趋势分量) + (平稳AR分量)与图12.3的趋势分量非常相似。该模型的AIC为1336.54。标准季节调整模型的AIC为1369.30，因此该模型明显更好。

```
> data(BLSALLFOOD)
> season(BLSALLFOOD, trend.order = 2, seasonal.order = 1, ar.order = 2, filter =
c(1,132))
```

图12.7是使用该模型在 $n = 132$ 时点进行长期预测的结果。在这种情况下，无论短期还是长期都获得了良好的预测结果。

12.4 包含星期效应项的分解

在记录百货商店每月销售额的月度数据等中，由于一个月中包含的各星期天数的差异，可能导致销售额出现显著差异。这是因为周日和周六顾客较多，而周一等特定星期设有定休日。这种现象不仅出现在经济时间序列中，在观测 NO_x 和 CO 等环境数据中也经常出现。

为了消除这种星期差异的影响而提出的方法是**星期调整** (trading-day adjustment)。为了用时间序列模型实现星期调整，需要引入服从适当模型的星期效应

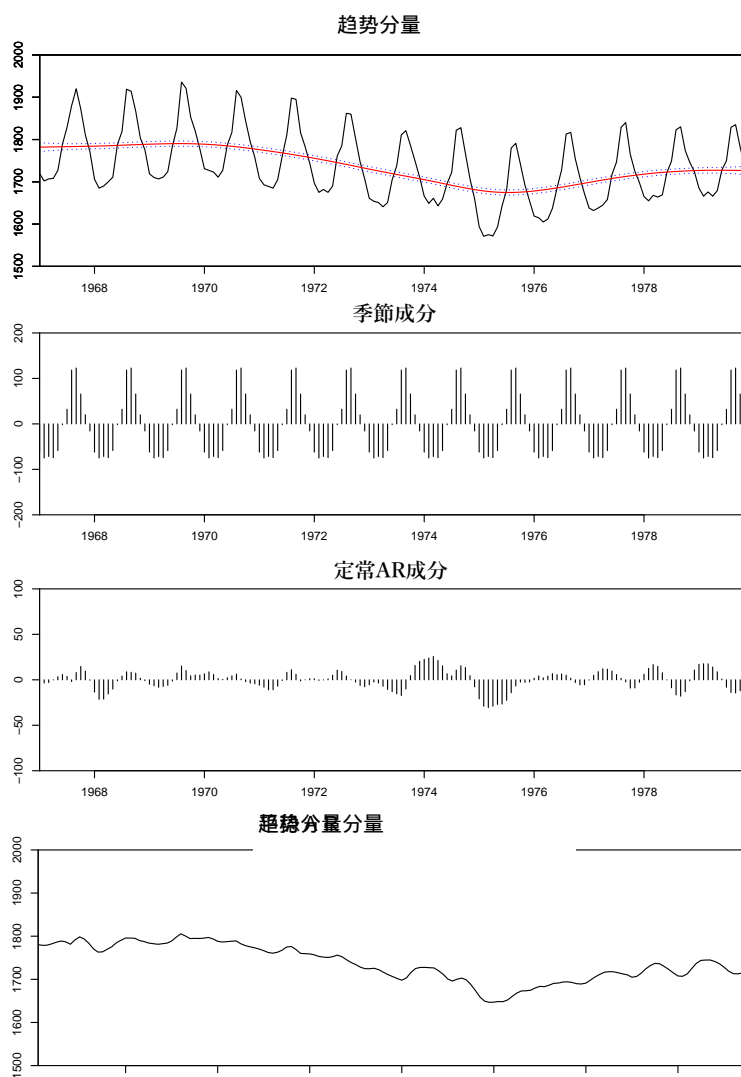


图 12.6 包含平稳 AR 分量的模型对 BLSALLFOOD 数据的季节调整

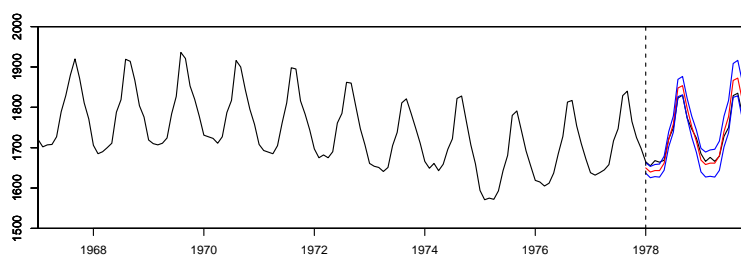


图 12.7 包含平稳 AR 分量的模型对食品产业从业人数数据的长期预测

项。将与时间序列 y_n 对应的第 n 个月中周日到周六的天数记为 $d_{n1}^*, \dots, d_{n7}^*$ 。各 d_{ni}^* 取 4 或 5 的值。此时假设星期效应为

$$td_n = \sum_{i=1}^7 \beta_{ni} d_{ni}^* \quad (12.25)$$

β_{ni} 是第 i 个星期日的增减对 y_n 的影响系数。但为了保证时间序列分解的唯一性, 需要附加所有系数之和为 0 的条件, 即

$$\beta_{n1} + \dots + \beta_{n7} = 0 \quad (12.26)$$

此时, 由于 $\beta_{n7} = -(\beta_{n1} + \dots + \beta_{n6})$, 星期效应实际上仅用 $\beta_{n1}, \dots, \beta_{n6}$ 即可表示为

$$\begin{aligned} td_n &= \sum_{i=1}^6 \beta_{ni} (d_{ni}^* - d_{n7}^*) \\ &\equiv \sum_{i=1}^6 \beta_{ni} d_{ni} \end{aligned} \quad (12.27)$$

其中 $d_{ni} \equiv d_{ni}^* - d_{n7}^*$ 是各星期天数减去周六天数。

作为星期效应项的模型, 为简单起见假设 β_{ni} 本身随时刻 n 按一阶趋势分量模型

$$\Delta \beta_{ni} = v_{n4}^{(i)}, \quad v_{n4}^{(i)} \sim N(0, \tau_4^2), \quad i = 1, \dots, 6 \quad (12.28)$$

变化时, 表示星期效应项的状态空间模型为

$$\begin{aligned} F_{n4} &= G_{n4} = I_6, \quad H_{n4} = [d_{n1}, \dots, d_{n6}] \\ x_{n4} &= \begin{bmatrix} \beta_{n1} \\ \vdots \\ \beta_{n6} \end{bmatrix}, \quad Q = \begin{bmatrix} \tau_4^2 & & \\ & \ddots & \\ & & \tau_4^2 \end{bmatrix} \end{aligned} \quad (12.29)$$

在这种情况下, H_{n4} 为依赖于时刻 n 的向量。此外, 为简单起见假设 Q 为所有对角元素相等的对角矩阵。但通常使用星期效应系数不随时间变化而恒定, 即

$$\beta_{ni} \equiv \beta_i \quad (12.30)$$

的模型。此时, 令 $\tau_4^2 = 0$ 或 $G = 0$ 即可。此时,

$$\begin{aligned} x_n &= \begin{bmatrix} x_{n1} \\ x_{n2} \\ x_{n3} \\ x_{n4} \end{bmatrix}, \quad F = \begin{bmatrix} F_1 & & & \\ & F_2 & & \\ & & F_3 & \\ & & & F_4 \end{bmatrix}, \quad G = \begin{bmatrix} G_1 & 0 & 0 \\ 0 & G_2 & 0 \\ 0 & 0 & G_3 \\ 0 & 0 & 0 \end{bmatrix} \\ H &= [H_1 \quad H_2 \quad H_3 \quad H_{n4}], \quad Q = \begin{bmatrix} \tau_1^2 & 0 & 0 \\ 0 & \tau_2^2 & 0 \\ 0 & 0 & \tau_3^2 \end{bmatrix} \end{aligned} \quad (12.31)$$

则可以构成将时间序列

$$y_n = t_n + s_n + p_n + td_n + w_n \quad (12.32)$$

分解为 5 个分量的模型。

例：包含星期效应项的季节调整

R 函数 `season` 中，要进行星期调整，只需如下设定 `trade=TRUE` 即可。

```
> season( WHARD, trend.order = 2, seasonal.order = 1, ar.order = 0, trade = TRUE,
log = TRUE )
```

图 12.8 是对 WHARD 批发额数据的对数值应用包含星期效应项的季节调整模型，分解为趋势、季节分量、星期效应项和噪声的结果。趋势和季节分量的估计值与图 12.1 的标准分解情况几乎相同。星期效应项看起来不太大，但从图 12.9 中（季节分量）+（星期效应项）的图来看，可以发现原始数据的细微变动也被很好地再现。顺便提一下，标准模型的 AIC 为 -728.50 ，包含星期效应项时为 -778.18 。当星期效应项的系数 β_i 被估计后， d_{n1}, \dots, d_{n6} 对于将来的 n 也是确定的，因此可以实现高精度的预测。

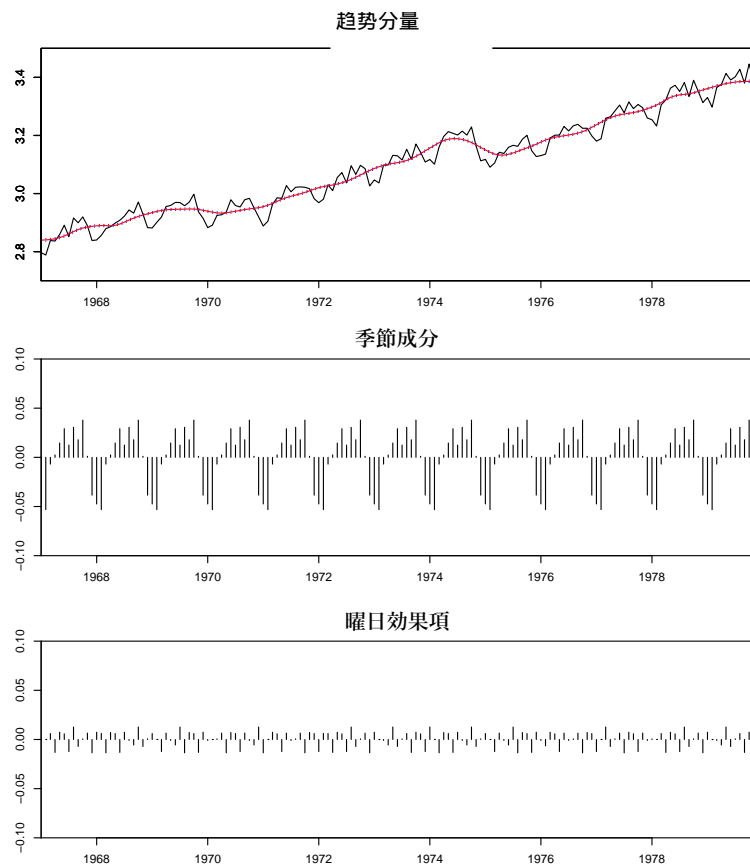


图 12.8 WHARD 数据的星期调整

函数 `season` 可以将时间序列分解为趋势、季节分量和星期效应项。在以下 R 代码中，这些分量分别由参数 `z$trend`、`z$season`、`z$day.effect` 给出。图 12.9 展示了

WHARD 数据对数的季节分量、星期效应项、季节分量 + 星期效应项以及 WHARD 数据对数减去趋势后的结果。虽然星期效应项的大小较小，但季节分量和星期效应项之和与数据减去趋势后的结果非常相似。这表明标准季节调整模型的噪声分量中相当大的部分实际上是星期效应项。

```

> data(WHARD)
> z <- season(WHARD, trend.order = 2, seasonal.order = 1, ar.order = 0, trade =
TRUE, log = TRUE)
> a1 <- z$seasonal + z$day.effect
> a2 <- log10( as.numeric(WHARD) )
> a3 <- a2 - z$trend
> par(mar=c(2,2,3,1)+0.1)
> par( mfrow=c(4,1) )
> plot( z$seasonal,type="l", ylim=c(-0.1,0.1), main="seasonal component" )
> plot( z$day.effect,type="l", ylim=c(-0.1,0.1), main="trading-day effect" )
> plot( a1,type="l", ylim=c(-0.1,0.1), main="seasonal + trading-day effect" )
> plot( a3,type="l", ylim=c(-0.1,0.1), main="data - trend" )

```

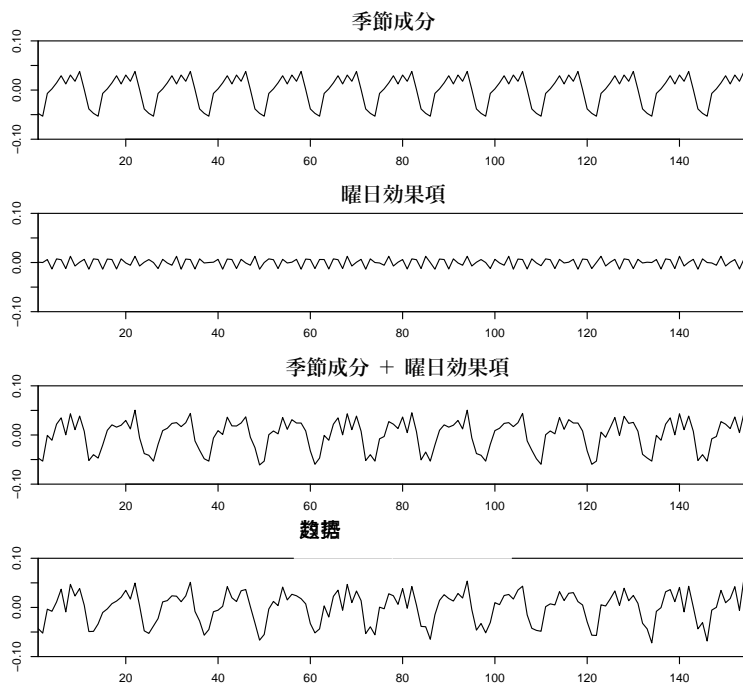


图 12.9 批发额数据的星期调整

1. 说明在包含平稳 AR 分量的分解中需要注意的事项。
2. 考虑与 12.1 节不同的季节分量模型。
3. 在星期调整中, 给出以下假设下的模型。
 - (1) 仅考虑周末和其他星期两种效应的情况
 - (2) 周日、周六和其他星期三种情况效应不同的情况
4. 在季节调整中, 说明本书未提及的其他可能因素。

第 13 章

时变系数 AR 模型

典型的非平稳时间序列中，有一类是平均值随时间变化的，另一类则是围绕平均值变动的方式发生变化的。在后一种情况下，时间序列的方差、自协方差函数以及功率谱会随时间变化。本章将针对此类时间序列，说明随时间变化的方差估计方法，以及拟合系数随时间变化的时变系数 AR 模型的方法。

13.1 时变方差模型

设 y_n 是均值为 0、方差为 σ_n^2 的正态白噪声。此时，假设 $\sigma_{2m-1}^2 = \sigma_{2m}^2$ ，通过

$$s_m = y_{2m-1}^2 + y_{2m}^2 \quad (13.1)$$

生成长度减半的时间序列 $s_1, \dots, s_{N/2}$ 。由于 s_m 服从自由度为 2 的 χ^2 分布（即指数分布）， s_m 的密度函数为：

$$f(s) = \frac{1}{2\sigma^2} e^{-s/2\sigma^2} \quad (13.2)$$

因此，进一步进行变换：

$$z_m = \log\left(\frac{s_m}{2}\right) \quad (13.3)$$

则 z_m 的密度函数变为：

$$g(z) = \frac{1}{\sigma^2} \exp\left\{z - \frac{e^z}{\sigma^2}\right\} = \exp\left\{(z - \log \sigma^2) - e^{(z - \log \sigma^2)}\right\} \quad (13.4)$$

这表明可以表示为：

$$z_m = \log \sigma^2 + w_m \quad (13.5)$$

其中， w_m 是双指数分布，其密度函数由下式给出：

$$h(w) = \exp\{w - e^w\} \quad (13.6)$$

因此, 通过状态空间模型

$$\begin{aligned}\Delta^k t_m &= v_m \\ z_m &= t_m + w_m\end{aligned}\quad (13.7)$$

求得 z_m 的趋势项, 即可估计原序列 y_n 方差的对数值。问题在于 w_m 的分布并非正态分布, 但由于双指数分布的均值为 $-\gamma$ (其中 $\gamma=0.57722$ 为欧拉常数), 方差为 $\frac{\pi^2}{6}$, 因此通过进行如下正态分布近似:

$$w_m \sim N\left(-\gamma, \frac{\pi^2}{6}\right)\quad (13.8)$$

即可利用卡尔曼滤波进行趋势估计。此时, $t_{m|M} + \gamma$ ($M = \frac{N}{2}$) 即为 $\log \sigma_m^2$ 的估计值。此外, 如果在第 14 章的非高斯型滤波器中直接使用式 (13.6) 的密度函数, 则可以在不使用正态近似的情况下求得趋势。

实际上, 在式 (13.1) 中使用相邻两个项的和并非硬性要求, 主要是为了使 w_n 的分布更接近正态分布。因此, 在使用非高斯型滤波器时, 也可以取 $s_n = y_n^2$, 这种情况下则不需要 $\sigma_{2m-1}^2 = \sigma_{2m}^2$ 的假设。

例: 地震数据的时变方差

R 语言包 TSSS 中的函数 `tvvar` 用于估计时变方差。参数 `trend.order` 是趋势成分模型的阶数, `tau2.ini` 和 `delta` 分别指定系统噪声方差 τ^2 最大似然估计的初始估计值和搜索步长。如果未明确给出这些参数 (即 `tau2.ini = NULL` 时), 则会在 $\tau^2 = 2^{-k}, k = 1, 2, \dots$ 的范围内选择最优值。如果希望更精确地估计 τ^2 , 可以将该输出作为初始值重复计算。

函数 `tvvar` 返回以下输出:

```
tvv:      时变方差
sm:      按式 (13.1) 变换后的数据
trend:   变换后数据 sm 的趋势项
noise:   残差
nordata: 标准化后的数据
tau2:    系统噪声的方差
sigma2:  观测噪声的方差
llkhood: 模型的对数似然
aic:     模型的 AIC
tsname:  时间序列  $y_n$  的名称
```

```
> data( MYE1F )
> tvvar( MYE1F, trend.order = 2, tau2.ini = 6.6e-06, delta = 1.0e-06 )
```

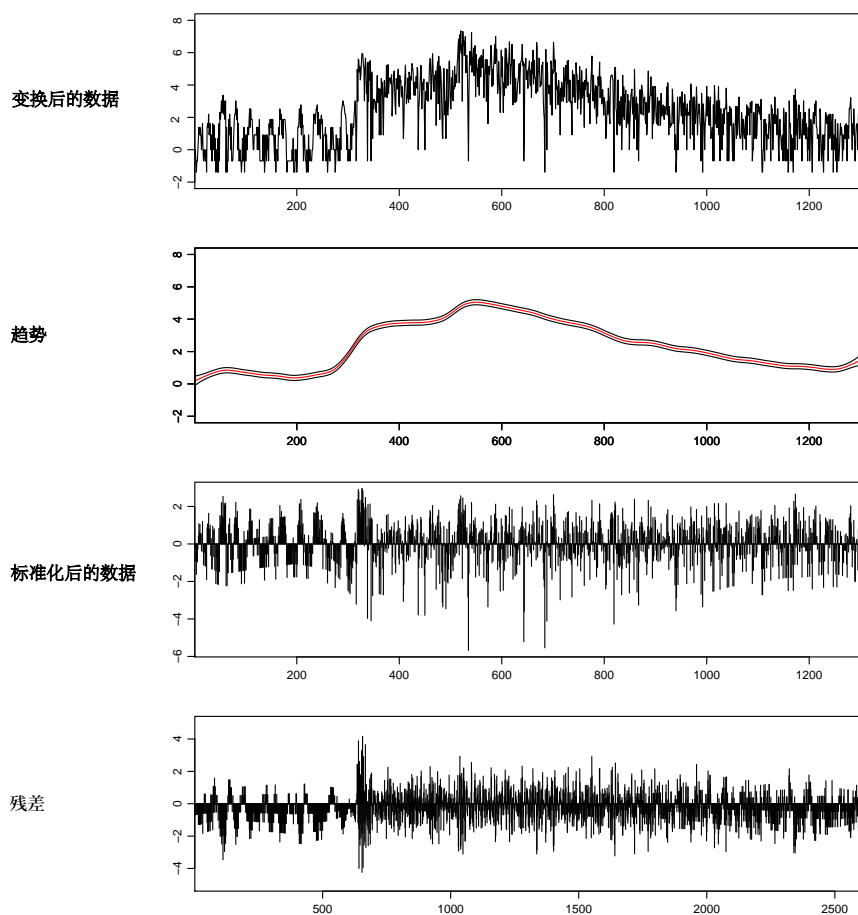


图 13.1 地震波数据的时变方差估计与时间序列标准化

图 13.1 显示了对图 2.1 所示的地震波数据应用函数 `tvvar` 的结果，其中趋势阶数为 2，初始方差估计值为 $\tau^2 = 0.66 \times 10^{-5}$ 。由上至下依次为：按式 (13.1) 变换后的数据 s_m 、 s_m 的趋势、方差对数的估计值 $\log \hat{\sigma}^2$ 、通过 $\tilde{y}_n = \frac{y_n}{\sqrt{\hat{\sigma}^2}}$ 标准化后的数据以及残差。地震波数据虽非白噪声，但通过此方法可以变换为方差近似为 1 的时间序列。

13.2 时变系数 AR 模型

平稳时间序列的变动特征由自协方差函数或功率谱刻画。因此，对于变动特征本身随时间变化的非平稳时间序列，其自协方差函数或功率谱也会随时间变化。在平稳时间序列中，通过适当选择 AR 模型或 ARMA 模型等时间序列模型的阶数和系数，可以表达各种自协方差函数和功率谱。因此，当变动方式随时间变化时，时间序列模型本身也会发生变化。本节针对非平稳时间序列 y_n ，考虑系数随时间

变化的 AR 模型:

$$y_n = \sum_{j=1}^m a_{nj} y_{n-j} + w_n \quad (13.9)$$

其中, w_n 是均值为 0、方差为 σ^2 的正态白噪声。该模型称为 m 阶时变系数 AR 模型 (time-varying coefficients AR model), 而 a_{nj} 是时刻 n 处滞后为 j 的自回归系数, 称为时变 AR 系数。

对于给定的时间序列 y_1, \dots, y_N , 即使仅考虑自回归系数, 时变系数 AR 模型也包含 mN 个未知数, 简单地对模型 (13.9) 应用最大似然法或最小二乘法无法获得有意义的结果。因此, 与趋势模型和季节调整模型类似, 我们为参数的时间变化引入随机趋势成分模型。在趋势模型中, 我们将趋势成分 t_n 视为未知数并为其时间变化引入模型; 而在时变系数 AR 模型中, 由于 AR 系数 a_{nj} 随时刻 n 变化, 我们引入以下约束:

$$\Delta^k a_{nj} = v_{nj}, \quad j = 1, \dots, m \quad (13.10)$$

其中, Δ 是关于时刻 n 的差分算子, 满足 $\Delta a_{nj} = a_{nj} - a_{n-1,j}$ 。 k 通常取 1 或 2。 $v_n = (v_{n1}, \dots, v_{nm})^T$ 是均值向量为 0、方差协方差矩阵为 Q 的 m 维正态白噪声。通常假设当 $i \neq j$ 时 v_{ni} 与 v_{nj} 相互独立, 因此 Q 是以 $\tau_{11}^2, \dots, \tau_{mm}^2$ 为对角元的对角矩阵, 可写为 $Q = \text{diag}\{\tau_{11}^2, \dots, \tau_{mm}^2\}$ 。这里进一步假设所有方差相等, 即 $\tau_{11}^2 = \dots = \tau_{mm}^2 = \tau^2$ 。关于该假设的含义和合理性将在第 13.4 节探讨。

接下来, 为了在约束模型 (13.10) 下估计时变系数 AR 模型 (13.9), 我们将这两个模型表示为状态空间模型。首先, 当 $k=1$ 时定义 $x_{nj} = a_{nj}$; 当 $k=2$ 时定义 $x_{nj} = (a_{nj}, a_{n-1,j})^T$ 为 k 维向量。此时, 式 (13.10) 可以表示为系统模型形式:

$$x_{nj} = F^{(k)} x_{n-1,j} + G^{(k)} v_{nj} \quad (13.11)$$

其中,

$$F^{(1)} = G^{(1)} = 1$$

$$F^{(2)} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, \quad G^{(2)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (13.12)$$

如果将式 (13.9) 右侧的第 j 项视为该模型的第 j 个成分, 则可表示为:

$$a_{nj} y_{n-j} = H_n^{(kj)} x_{nj} \quad (13.13)$$

其中, $H_n^{(1j)} = y_{n-j}$, $H_n^{(2j)} = (y_{n-j}, 0)$ 。此时, 时变系数 AR 模型的第 j 个成分利用 $F^{(k)}$ 、 $G^{(k)}$ 和 $H_n^{(kj)}$ 可表示为状态空间模型:

$$\begin{aligned} x_{nj} &= F^{(k)}x_{n-1,j} + G^{(k)}v_{nj} \\ a_{nj}y_{n-j} &= H_n^{(kj)}x_{nj} \end{aligned} \quad (13.14)$$

此外, 利用 $H^{(1)} = 1$ 、 $H^{(2)} = (1, 0)$, 可以写成:

$$H_n^{(kj)} = y_{n-j}H^{(k)} \quad (13.15)$$

利用上述成分模型, 可得到时变系数 AR 模型的状态空间表示:

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n \\ y_n &= H_n x_n + w_n \end{aligned} \quad (13.16)$$

其中, F 和 G 分别是如下 $km \times km$ 和 $km \times m$ 矩阵, H_n 和 x_n 则是如下 km 维向量:

$$\begin{aligned} F &= \begin{bmatrix} F^{(k)} & & \\ & \ddots & \\ & & F^{(k)} \end{bmatrix} = I_m \otimes F^{(k)} \\ G &= \begin{bmatrix} G^{(k)} & & \\ & \ddots & \\ & & G^{(k)} \end{bmatrix} = I_m \otimes G^{(k)} \\ H_n &= [H_n^{(k1)}, \dots, H_n^{(km)}] = (y_{n-1}, \dots, y_{n-m}) \otimes H^{(k)} \\ x_n &= \begin{cases} (a_{n1}, \dots, a_{nm})^T, & k=1 \text{ 时} \\ (a_{n1}, a_{n-1,1}, \dots, a_{nm}, a_{n-1,m})^T, & k=2 \text{ 时} \end{cases} \\ Q &= \begin{bmatrix} \tau^2 & & \\ & \ddots & \\ & & \tau^2 \end{bmatrix}, \quad R = \sigma^2 \end{aligned} \quad (13.17)$$

在这里, I_m 是 $m \times m$ 单位矩阵, \otimes 表示矩阵的 Kronecker 积。若 A 是 $k \times \ell$ 矩阵, B 是 $m \times n$ 矩阵, 则 $A \otimes B$ 是 $km \times \ell n$ 矩阵。

综上所述, 时变系数 AR 模型 (13.9) 及其关于系数时间变化的约束模型 (13.10)

可以表示为状态空间模型。例如当 $m=2$ 、 $k=2$ 时, 状态空间模型如下:

$$\begin{aligned} \begin{bmatrix} a_{n1} \\ a_{n-1,1} \\ a_{n2} \\ a_{n-1,2} \end{bmatrix} &= \begin{bmatrix} 2 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{n-1,1} \\ a_{n-2,1} \\ a_{n-1,2} \\ a_{n-2,2} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{n1} \\ v_{n2} \end{bmatrix} \\ y_n &= (y_{n-1}, 0, y_{n-2}, 0) \begin{bmatrix} a_{n1} \\ a_{n-1,1} \\ a_{n2} \\ a_{n-1,2} \end{bmatrix} + w_n \\ \begin{bmatrix} v_{n,1} \\ v_{n,2} \end{bmatrix} &\sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 & 0 \\ 0 & \tau^2 \end{bmatrix}\right), \quad w_n \sim N(0, \sigma^2) \end{aligned} \quad (13.18)$$

上述状态空间模型中包含若干待估参数。其中, 观测噪声 w_n 的方差 σ^2 和系统噪声 v_n 的方差 τ^2 可以利用第 9 章的方法计算对数似然, 从而通过最大似然法进行估计。此外, AR 阶数 m 和约束模型的阶数 k 可选取使 AIC 最小的值。一旦估计出模型参数 m 、 k 、 σ^2 和 τ^2 , 即可应用平滑算法获得状态向量的平滑值 $x_{n|N}$ 。此时, 该向量的第 $(j-1)k+1$ 个分量即为 $a_{n|N}$, 即 a_{nj} 的估计值。

时变系数 AR 模型的平滑计算需要 $km \times km \times N$ 的数组。因此, 当 AR 阶数 m 或数据量 N 非常大时, 可能会出现内存不足的情况。最简单的解决方法是假设系数每隔预设的 r 步变化一次。若取 r 为 20, 所需内存将减至 $\frac{1}{20}$ 。如果系数变化平滑且缓慢, 这种假设对结果影响不大。即使没有内存问题, 为了在下一节说明的时变谱图中便于展示, 也最好使用较大的 r 。要在 $r > 1$ 的情况下执行卡尔曼滤波, 只需每计算一次预测步骤, 重复执行 r 次滤波步骤即可。

例: 地震数据的时变系数 AR 模型

R 语言的 TSSS 包中的函数 `tvar` 用于估计时变系数 AR 模型。该函数的输入包括:

<code>y</code> :	单变量时间序列
<code>trend.order</code> :	趋势阶数 (1 或 2)
<code>ar.order</code> :	AR 阶数
<code>span</code> :	局部平稳跨度 (r)
<code>outlier</code> :	异常值发生的时间点
<code>tau2.ini</code> :	系统噪声 τ^2 的初始估计值。当 <code>tau2.ini=NULL</code> 时, 会在 $\tau^2 = 2^{-k}, k = 1, 2, \dots$ 范围内寻找最优值
<code>delta</code> :	搜索步长
<code>plot</code> :	逻辑变量。为 TRUE (默认) 时绘制 PARCOR 图

函数 `tvar` 的输出如下:

<code>arcoef</code> :	时变 AR 系数
<code>sigma2</code> :	观测噪声方差 σ^2
<code>tau2</code> :	系统噪声方差 τ^2
<code>llkhood</code> :	模型的对数似然
<code>aic</code> :	模型的 AIC
<code>parcor</code> :	时变偏自相关系数 (时变 PARCOR)

```

> data(MYE1F)
> # 阶数 k=1
> z <- tvar( MYE1F, trend.order = 1, ar.order = 8, span = 20,
tau2.ini = 1.0e-03, delta = 1.0e-04 )
> # 阶数 k=2
> z <- tvar( MYE1F, trend.order = 2, ar.order = 8, span = 20,
tau2.ini = 6.6e-06, delta = 1.0e-06 )

```

表 13.1 地震波数据的时变系数 AR 模型的 AIC

m	$k=1$	$k=2$	m	$k=1$	$k=2$
1	6536.6	6520.4	6	4884.1	4873.8
2	5568.6	5643.2	7	4874.8	4878.7
3	5113.7	5134.5	8	4859.5	4866.9
4	4866.1	4853.9	9	4868.3	4884.9
5	4894.2	4886.0	10	4881.2	4911.9

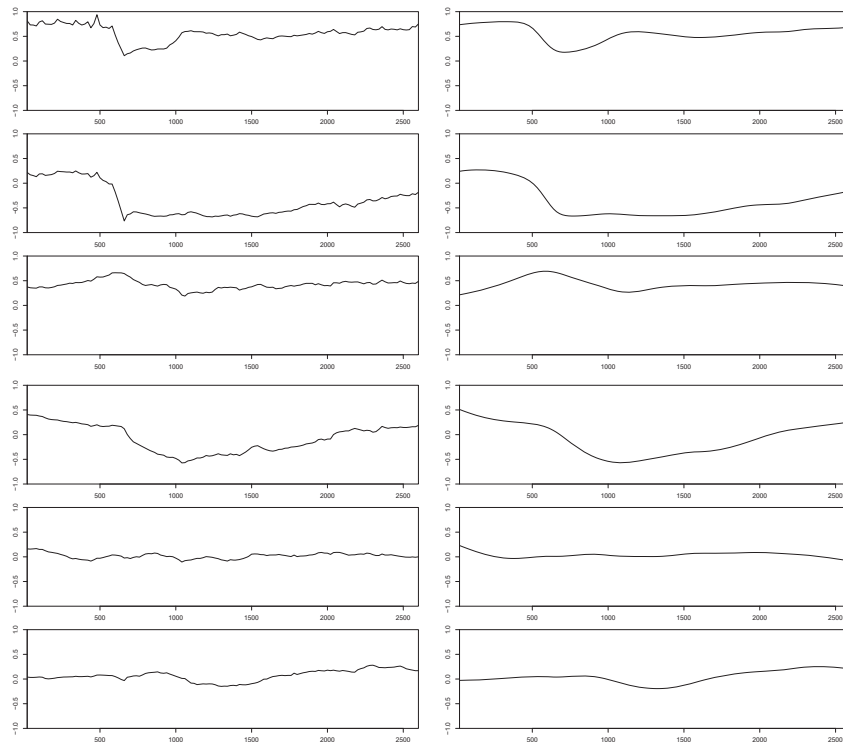


图 13.2 标准化地震波数据的时变 PARCOR 估计值。仅显示前 6 个。左图: $k=1$, 右图: $k=2$ 。横轴: 时刻, 纵轴: 时变 PARCOR

表 13.1 汇总了将 $r=20$ 的各种阶数时变系数 AR 模型拟合至图 13.1 标准化地震波数据时的 AIC 值。由于时变系数 AR 模型假设方差 σ^2 是恒定的, 因此当方差

显著变化 (如图 1.1(f) 所示) 时, 最好使用如图 13.1 所示变换为方差近似恒定的数据。\$m\$ 是 AR 阶数, \$k\$ 是差分阶数。无论 \$k=1\$ 还是 \$k=2\$, 都在 \$m=8\$ 时 AIC 最小。

图 13.2 展示了在 \$k=1\$ 和 \$k=2\$ 两种情况下, 采用 AIC 取得最小值时对应阶数的时变系数 AR 模型, 对各系数随时间变化的估计结果。需要指出的是, 这里并未直接给出 AR 系数本身, 而是仅显示了时变 PARCOR \$b_{ni}\$ 中 \$i=1\sim 6\$ 的变化情况。图的左侧对应 \$k=1\$ 的情形, 右侧对应 \$k=2\$ 的情形。基于局部平稳 AR 模型的分析结果表明, 在 \$n=630\$ 时刻 P 波到达, 在 \$n=1026\$ 时刻 S 波到达, 可以看到与此相对应, 各系数也发生了变化。由 \$k=1\$ 得到的估计值波动较为明显。另一方面, 由 \$k=2\$ 得到的估计值较为平滑,乍看之下似乎更为理想, 但实际上并未能充分反映地震波到达所引起的急剧变化。根据 AIC 的判别结果, 可以认为 \$k=1\$ 的模型更为合适。关于既能够刻画系数的急剧变化, 又能获得变动较小的估计值的方法, 将在 13.5 节中加以说明。

13.3 时变谱的估计

对于平稳 AR 模型, 其功率谱由下式给出:

$$p(f) = \frac{\sigma^2}{|1 - \sum_{j=1}^m a_j e^{-2\pi i j f}|^2}, \quad -\frac{1}{2} \leq f \leq \frac{1}{2} \quad (13.19)$$

因此, 对于 AR 系数作为时刻 \$n\$ 的函数 \$a_{nj}\$ 给出的时变系数 AR 模型 (13.9), 可以将时刻 \$n\$ 的瞬时谱定义为:

$$p_n(f) = \frac{\sigma^2}{|1 - \sum_{j=1}^m a_{nj} e^{-2\pi i j f}|^2}, \quad -\frac{1}{2} \leq f \leq \frac{1}{2} \quad (13.20)$$

利用前述方法, 可以将 AR 系数 \$a_{nj}\$ 估计为时间的函数, 因此功率谱 \$\{p_n(f)\}\$ (\$n=1, \dots, N\$) 也可以视为时间的函数。这被称为**时变谱** (time-varying spectrum)。

然而, 在实际现象中, 方差的变化和波形的变化往往同时发生。因此, 在时变谱估计中, 首先估计时变方差 \$\hat{\sigma}_n^2\$, 并通过

$$\tilde{y}_n = \frac{y_n}{\sqrt{\hat{\sigma}_n^2}} \quad (13.21)$$

将数据转换为标准化数据, 再估计时变系数 AR 模型。此时, 原始时间序列的时变谱计算公式为:

$$p_n(f) = \frac{\hat{\sigma}_n^2 \hat{\sigma}_n^2}{|1 - \sum_{j=1}^m \hat{a}_{nj} e^{-2\pi i j f}|^2}, \quad -\frac{1}{2} \leq f \leq \frac{1}{2} \quad (13.22)$$

例: 地震数据的时变谱

R 语言包 TSSS 中的函数 `tvspec` 利用 `tvar` 的输出计算时变谱。该函数的输入包括：

`arcoef`: 时变 AR 系数
`sigma2`: 观测噪声方差
`span`: 局部平稳跨度
`nf`: 计算频谱的频率点数

```
> data( MYE1F )
> par( mar=c(2,2,1,1) )
> # k=1 的情况
> z <- tvar( vnordata,trend.order = 1,ar.order = 8,span = 20,
tau2.ini = 5.0e-03,delta = 1.0e-04,plot = FALSE)
> spec <- -tvspc(zarcoef, zsigma2,span = 20,nf = 400)
> plot(spec,tvv = vtvv, dx = 2, dy = 0.10 )
>
> # k=2 的情况
> z <- tvar(vnordata,trend.order = 2,ar.order = 8,span = 20,
tau2.ini = 4.4e-05,delta = 1.0e-07,plot = FALSE)
> spec <- -tvspc(zarcoef, zsigma2,span = 20,nf = 400)
> plot(spec,tvv = vtvv, dx = 2, dy = 0.10)
```

图 13.3 给出了根据式 (13.22) 计算得到的时变谱。左侧为 $k=1$ 模型的估计结果，右侧为 $k=2$ 模型的估计结果。横轴表示频率，纵轴表示谱的对数，斜向表示时间的推移。可以看到，随着 P 波的到达， $f=0.25$ 附近的功率逐渐增强；随后 S 波到达时， $f=0.1$ 附近的功率进一步增强。此后，谱峰位置在逐渐向右移动的同时，逐步回到原先常微动的谱形态。

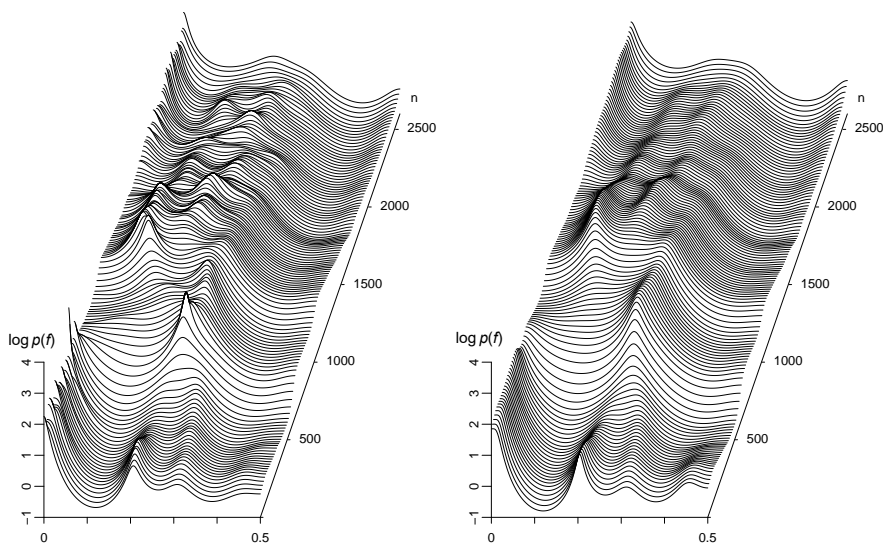


图 13.3 地震数据的时变谱。左图： $k=1$ ，右图： $k=2$

13.4 时变系数 AR 模型的系统噪声假设

在第 13.2 节中, 我们展示了通过趋势成分模型表达 AR 系数的时间变化, 并利用状态空间模型估计时变系数 AR 模型。在式 (13.17) 的状态空间模型中, 假设系统噪声的方差协方差矩阵为 $Q = \text{diag} \{ \tau^2, \dots, \tau^2 \}$ 形式的对角矩阵。虽然这看起来是非常强的假设, 但本节将通过考虑 AR 算子的频率响应函数的平滑性, 来说明该假设可以自然地导出。首先, 考虑时变 AR 系数的傅里叶变换:

$$A(f, n) = 1 - \sum_{j=1}^m a_{nj} e^{-2\pi i j f}, \quad -\frac{1}{2} \leq f \leq \frac{1}{2} \quad (13.23)$$

这相当于将 AR 模型视为白化滤波器时的频率响应函数。此时, 式 (13.20) 的时变谱可表示为:

$$p_n(f) = \frac{\sigma^2}{|A(f, n)|^2} \quad (13.24)$$

频谱的特性由 $A(f, n)$ 决定, 因此通过适当控制 $A(f, n)$ 的平滑度, 可以获得良好的时变谱估计值。在此, 通过考虑 $A(f, n)$ 随时刻 n 变化的平滑性, 可以引入如下约束模型。为了评价 $A(f, n)$ 随时间的阶跃变化, 取关于时刻 n 的 k 阶差分:

$$\Delta^k A(f, n) = \sum_{j=1}^m \Delta^k a_{nj} e^{-2\pi i j f} \quad (13.25)$$

考虑平方积分:

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} |\Delta^k A(f, n)|^2 df = \sum_{j=1}^m (\Delta^k a_{jn})^2 \quad (13.26)$$

因此, 通过减小 AR 系数 k 阶差分的平方和, 可以减小频谱随时间的变化。由于式 (13.26) 对各项赋予了相等的权重, 这相当于假设了使用共同的 τ^2 的模型:

$$\Delta_n^k a_{jn} = v_{nj}, \quad v_{nj} \sim N(0, \tau^2), \quad j = 1, \dots, m \quad (13.27)$$

即在式 (13.10) 中取 $\tau_{11}^2 = \dots = \tau_{mm}^2 = \tau^2$ 的假设是自然导出的。

13.5 关于系数的剧烈变化

如图 13.1 所示的经过标准化处理的地震数据那样, 当在某一时刻有新的信号加入并导致波形发生急剧变化时, 如图 13.2 所示, 所估计得到的时变系数往往会呈现较大的离散, 或者无法充分对应这种急剧变化。在这种情况下, 若采用第 8 章介

绍的局部平稳 AR 模型, 即可得到变化点的估计值, 利用这些信息有时能够获得更为理想的时变系数估计结果。

假设在时刻 $n = n_1, \dots, n_p$ 处存在急剧变化。此时, 式 (13.10) 中的误差项 v_{nj} 的绝对值将取非常大的值, 因此有必要仅在 $n = n_1, \dots, n_p$ 这些时刻增大系统噪声的方差 τ^2 。严格来说, 通过增大 τ^2 , 会使得 a_{nj} 的 k 阶差分变化幅度增大。因此, 在 $k = 1$ 的情况下, a_{nj} 将呈现阶梯状变化, 并形成不连续点; 而在 $k = 2$ 的情况下, 则表现为斜率发生急剧变化的折点。若要在 $k \geq 2$ 的情形下产生不连续点, 则需要对整个状态向量引入噪声。为此, 可以对 $x_{n|n-1}$ 与 $V_{n|n-1}$ 进行初始化, 或在 $V_{n|n-1}$ 的所有对角元上加上较大的值。该方法不仅适用于时变系数的估计, 同样也可以应用于趋势等的估计问题。

例: 伴随突发结构变化的时变谱估计

在函数 `tvspec` 中处理突发结构变化时, 需要利用参数 `outlier` 指定变点位置:

```
> data(MYE1F)
> z <- tvar( vnordata, trend.order = 2, ar.order = 8, span = 20,
  outlier = c(630, 1026), tau2.ini = 6.6e-06, delta = 1.0e-06,
  plot = FALSE)
> spec <- tvspec(zarcoef, zsigma2, span = 20, nf = 400)
> plot(spec, tvv = tvvv, dx = 2, dy = 0.10 )
```

图 13.4 给出了在 $k = 2$ 、 $m = 8$ 的设定下, 并假定在 $n = 630$ 和 $n = 1026$ 处存在急剧变化时, 所估计得到的时变系数及其对应的时变谱。可以看到, 系数的急剧变化仅在与 P 波和 S 波到达时刻相对应的这两个位置发生显著变化, 而在其他时刻则呈现出平滑的变化趋势。此外, 在 S 波对应的区间内, 一方面主导频率随时间逐渐减弱, 另一方面直流分量 ($f = 0$) 逐步增强, 清楚地刻画了谱形逐渐向常时微动谱过渡的过程。

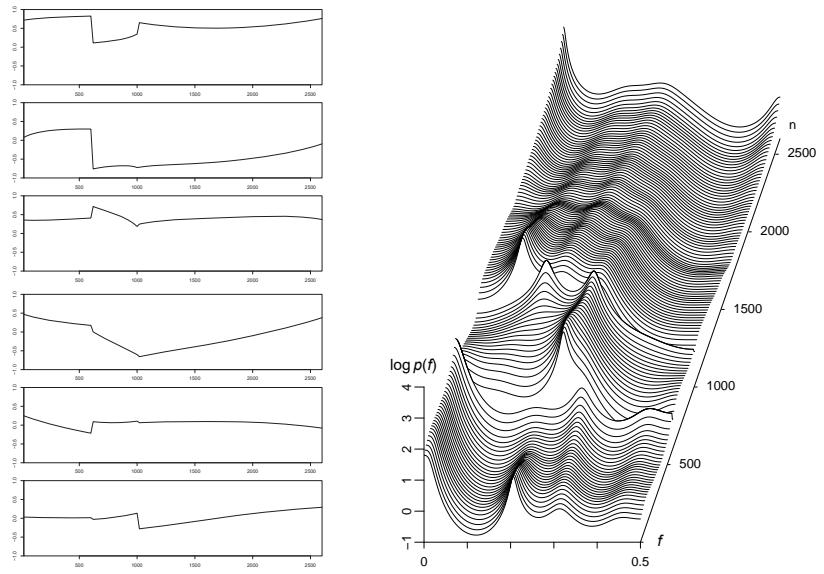


图 13.4 在参数变化中加入不连续点估计出的时变 PARCOR (左) 与时变谱 (右)

● 章末习题

1. 请思考除地震波外, 方差随时间变化的序列示例。
2. 请思考估计系数随时间变化的回归模型的方法。

第 14 章

非高斯型模型

本章将状态空间模型扩展到系统噪声或观测噪声为非高斯型的情形。非高斯型模型适用于处理由于系统结构变化而引起的参数急剧变化，以及异常值的处理等问题。在非高斯型模型的情况下，卡尔曼滤波和平滑算法往往不能得到好的结果。然而，对于非高斯型状态空间模型，也可以像卡尔曼滤波那样导出递推公式，并借助数值计算实现滤波和平滑算法。本章还将介绍非高斯型状态空间模型的各种应用，以及向非线性状态空间模型的扩展。

14.1 非高斯型模型的必要性

正如前章中的例子所示，许多时间序列模型都可以利用状态向量 x_n ，表示为线性高斯型状态空间模型

$$\begin{aligned} x_n &= Fx_{n-1} + Gv_n \\ y_n &= Hx_n + w_n \end{aligned} \quad (14.1)$$

的形式。该状态空间模型作为时间序列模型非常自然，而且可以利用卡尔曼滤波和平滑算法等高效的计算方法，因此可以灵活地构造时间序列模型。不过，尽管基于状态空间表示的时间序列建模极为方便，但并不是所有时间序列都能用线性高斯型状态空间模型来建模。例如，在概率结构随时间变化的非平稳时间序列中，这种变化方式往往同时包含缓慢平滑的变化和急剧的变化。在这种情况下，通常的线性高斯型状态空间模型虽然能够较好地表示缓慢平滑的变化，但若恰当地表示急剧变化，则必须构造更加复杂的模型。此外，为了消除数据中经常出现的异常值的影响，还需要引入异常值的自动检测方法或稳健估计方法。再者，包含非线性的系统或离散过程，也都是通常状态空间模型难以妥善处理的例子。

下面考虑如何解决这些问题。在状态空间模型中，概率结构的变化往往对应于

状态的变化。如果对系统噪声 v_n 假设如柯西分布那样的重尾分布, 就可以用一个分布同时表示高概率发生的平滑变化以及极小概率发生的急剧变化。同样地, 为了处理异常值, 也可以对观测噪声 w_n 使用重尾分布。此外, 当系统存在非线性, 或者观测值服从离散分布时, 状态的分布必然成为非高斯型。像这样, 为了解决传统状态空间模型难以处理的各种问题, 就有必要引入非高斯性。以下各节将说明针对非高斯型状态空间模型的滤波和平滑方法及其应用。

14.2 非高斯型状态空间模型与状态估计

考虑如下的状态空间模型:

$$x_n = Fx_{n-1} + Gv_n, \quad (\text{系统模型}) \quad (14.2)$$

$$y_n = Hx_n + w_n, \quad (\text{观测模型}) \quad (14.3)$$

系统噪声 v_n 和观测噪声 w_n 分别是服从密度函数 $q(v)$ 与 $r(w)$ 的白噪声, 但与前章中的状态空间模型不同, 它们不一定服从正态分布。在这种情况下, 状态向量 x_n 一般也将成为非高斯型。像这样, 不再假定高斯分布时, 模型 (14.2)、(14.3) 称为**非高斯型状态空间模型** (non-Gaussian state-space model)。显然, 这种非高斯型模型是通常状态空间模型的扩展。

与第 9 章相同, 记截至时刻 j 为止所观测到的时间序列信息为 $Y_j \equiv \{y_1, \dots, y_j\}$ 。此外, 将截至时刻 j 的状态 x_n 的实现值集合记为 $X_j \equiv \{x_1, \dots, x_j\}$ 。再进一步, 设初始向量 x_0 服从密度函数 $p(x_0|Y_0)$ 。此时, 状态估计问题就是求 $p(x_n|Y_m)$, 也就是在信息 Y_m 给定时状态 x_n 的条件分布。特别地, 当 $n > m$ 、 $n = m$ 、 $n < m$ 时, 分别称为预测、滤波和平滑化。

卡尔曼滤波给出的是状态的均值和方差协方差矩阵, 但对于非高斯型状态空间模型, 则需要直接计算分布。然而, 即使在这种情况下, 对于状态空间模型 (14.2)、(14.3), 利用 $p(x_n|x_{n-1}, Y_{n-1}) = p(x_n|x_{n-1})$, $p(y_n|x_n, Y_{n-1}) = p(y_n|x_n)$ 成立这一事实, 仍可按如下方式递推地求得预测分布与滤波分布。

[一步预测]

$$\begin{aligned} p(x_n|Y_{n-1}) &= \int_{-\infty}^{\infty} p(x_n, x_{n-1}|Y_{n-1}) dx_{n-1} \\ &= \int_{-\infty}^{\infty} p(x_n|x_{n-1}, Y_{n-1}) p(x_{n-1}|Y_{n-1}) dx_{n-1} \\ &= \int_{-\infty}^{\infty} p(x_n|x_{n-1}) p(x_{n-1}|Y_{n-1}) dx_{n-1} \end{aligned} \quad (14.4)$$

[滤波]

$$\begin{aligned}
 p(x_n|Y_n) &= p(x_n|y_n, Y_{n-1}) \\
 &= \frac{p(y_n|x_n, Y_{n-1})p(x_n|Y_{n-1})}{p(y_n|Y_{n-1})} \\
 &= \frac{p(y_n|x_n)p(x_n|Y_{n-1})}{p(y_n|Y_{n-1})}
 \end{aligned} \tag{14.5}$$

其中, $p(y_n|Y_{n-1})$ 由 $\int p(y_n|x_n)p(x_n|Y_{n-1})dx_n$ 求得。一步预测公式 (14.4) 是卡尔曼滤波中一步预测的扩展。 $p(x_n|x_{n-1})$ 是在状态 x_{n-1} 给定时 x_n 的密度函数, 它由系统模型 (14.2) 决定。因此, 只要给定 x_{n-1} 的滤波分布 $p(x_{n-1}|Y_{n-1})$, 就可以计算 $p(x_n|Y_{n-1})$ 。另一方面, 滤波公式 (14.5) 是卡尔曼滤波中滤波步骤的扩展。 $p(y_n|x_n)$ 是在状态 x_n 给定时 y_n 的密度函数, 由观测模型 (14.3) 决定。因此, 只要给定 x_n 的预测分布 $p(x_n|Y_{n-1})$, 就可以计算 $p(x_n|Y_n)$ 。

接下来考虑平滑化问题。对于状态空间模型 (14.2)、(14.3), 利用 $p(x_n|x_{n+1}, Y_N) = p(x_n|x_{n+1}, Y_n)$ 成立这一事实, 可得

$$\begin{aligned}
 p(x_n, x_{n+1}|Y_N) &= p(x_{n+1}|Y_N)p(x_n|x_{n+1}, Y_N) \\
 &= p(x_{n+1}|Y_N)p(x_n|x_{n+1}, Y_n) \\
 &= p(x_{n+1}|Y_N) \frac{p(x_n|Y_n)p(x_{n+1}|x_n, Y_n)}{p(x_{n+1}|Y_n)} \\
 &= p(x_{n+1}|Y_N) \frac{p(x_n|Y_n)p(x_{n+1}|x_n)}{p(x_{n+1}|Y_n)}
 \end{aligned} \tag{14.6}$$

由此, 关于平滑化问题也可以得到如下的递推公式。

[平滑化公式]

$$\begin{aligned}
 p(x_n|Y_N) &= \int_{-\infty}^{\infty} p(x_n, x_{n+1}|Y_N)dx_{n+1} \\
 &= p(x_n|Y_n) \int_{-\infty}^{\infty} \frac{p(x_{n+1}|Y_N)p(x_{n+1}|x_n)}{p(x_{n+1}|Y_n)} dx_{n+1}
 \end{aligned} \tag{14.7}$$

在 (14.7) 的右边中, $p(x_{n+1}|x_n)$ 由系统模型 (14.2) 决定。另外, $p(x_n|Y_n)$ 和 $p(x_{n+1}|Y_n)$ 则由 (14.4)、(14.5) 求得。因此, (14.7) 表明: 只要给定 $p(x_{n+1}|Y_N)$, 就可以计算 $p(x_n|Y_N)$ 。又考虑到 $p(x_N|Y_N)$ 可以通过 (14.5) 计算, 因此与第 9 章中的平滑化相同, 通过依次应用 (14.7), 就能求得平滑分布 $p(x_{N-1}|Y_N), \dots, p(x_1|Y_N)$ 。

14.3 状态估计公式的数值实现

如上所述, 即使对于非高斯型状态空间模型, 也可以导出扩展卡尔曼滤波形式的递推估计公式。利用这些公式, 就能够统一处理广泛的时间序列模型。但问题在于, 滤波和平滑化的公式在实际中应如何计算。

对于线性高斯型状态空间模型, 条件分布 $p(x_n|Y_{n-1})$ 、 $p(x_n|Y_n)$ 以及 $p(x_n|Y_N)$ 全都是正态分布, 因此只需评价其均值和方差协方差矩阵即可, 而 (14.4)、(14.5) 以及 (14.7) 就与通常的卡尔曼滤波和平滑算法等价。然而, 一般情况下, 状态的条件分布 $p(x_n|Y_j)$ 是非高斯型分布, 因此仅靠均值和方差协方差矩阵无法确定该分布。另一方面, 将这些分布用适当调节均值向量和方差协方差矩阵的高斯分布来近似的方法, 如扩展卡尔曼滤波和二次滤波等, 虽已为人所知, 但并不总是能得到好的结果。

本节将说明一种通过直接对非高斯型分布进行数值近似, 从而导出非高斯型滤波与非高斯型平滑算法的方法。近似的方法可以考虑阶梯函数近似、折线近似、样条近似等。此时, (14.4)~(14.7) 就可以借助数值计算来实现。这类数值方法过去被认为由于需要大规模数值计算而不切实际, 但随着计算机速度的提高, 如今至少对于低维问题而言, 它已成为一种极其有力的方法。

本节中, 将采用最简单的阶梯函数来近似出现在 (14.4)、(14.5)、(14.7) 中的密度函数。具体地, 设待近似的密度函数为 $f(t)$ ($-\infty < t < \infty$)。为了用阶梯函数来近似该函数, 首先将定义域划分为 d 个小区间: $t_0 < t_1 < \dots < t_d$ 。其中, t_0 取为足够小的数, t_d 取为足够大的数; 并且为简便起见, 假设分点等间隔, 即有 $t_i = t_0 + i\Delta t$ 。另外, 在实际计算程序中, 通常会随着时刻 n 的变化, 对 t_0 和 t_i 的值进行自适应平移, 但为了简便, 这里假设所有分点都是固定不变的。记如此定义的 d 个区间 $(t_{i-1}, t_i]$ 上 $f(t)$ 的值为 f_i 。当 $f(t)$ 是阶梯函数时, $f_i = f(t_i)$ 。一般情况下, f_i 可由

$$f_i = \Delta t \int_{t_{i-1}}^{t_i} f(t) dt \quad (14.8)$$

求得。利用这些值, $f(t)$ 的近似函数可由 $\{d; t_0, \dots, t_d; f_1, \dots, f_d\}$ 来表示。以下将这样得到的近似函数记为 $\tilde{f}(t)$ 。

为了数值实现非高斯型滤波与平滑公式, 如表 14.1 所示, 需要对 $p(x_n|Y_{n-1})$ 、 $p(x_n|Y_n)$ 、 $p(x_n|Y_N)$ 以及系统噪声的密度函数 $q(v)$ 进行近似。观测噪声的密度函数 $r(v)$ 则最好不离散化, 而是直接用于计算。

以下为简便起见, 以一维趋势模型

$$\begin{aligned} x_n &= x_{n-1} + v_n \\ y_n &= x_n + w_n \end{aligned} \quad (14.9)$$

为例说明数值计算方法。

表 14.1 密度函数的近似

密度函数	近似	记号
$p(x_n Y_{n-1})$	$\{d; t_0, \dots, t_d; p_1, \dots, p_d\}$	$\tilde{p}(t)$
$p(x_n Y_n)$	$\{d; t_0, \dots, t_d; f_1, \dots, f_d\}$	$\tilde{f}(t)$
$p(x_n Y_N)$	$\{d; t_0, \dots, t_d; s_1, \dots, s_d\}$	$\tilde{s}(t)$
$q(v)$	$\{2d+1; t_{-d}, \dots, t_d; q_{-d}, \dots, q_d\}$	$\tilde{q}(v)$

[一步预测] 对 $i = 1, \dots, d$

$$\begin{aligned}
 p_i &= \tilde{p}(t_i) = \int_{t_0}^{t_d} \tilde{q}(t_i - s) \tilde{f}(s) ds \\
 &= \sum_{j=1}^d \int_{t_{j-1}}^{t_j} \tilde{q}(t_i - s) \tilde{f}(s) ds \\
 &= \Delta t \sum_{j=1}^d q_{i-j} f_j
 \end{aligned} \tag{14.10}$$

[滤波] 对 $i = 1, \dots, d$

$$f_i = \tilde{f}(t_i) = \frac{r(y_n - t_i) \tilde{p}(t_i)}{C} = \frac{r(y_n - t_i) p_i}{C} \tag{14.11}$$

其中, C 可由

$$\begin{aligned}
 C &= \int_{t_0}^{t_d} r(y_n - t) \tilde{p}(t) dt = \sum_{j=1}^d \int_{t_{j-1}}^{t_j} r(y_n - t) \tilde{p}(t) dt \\
 &= \Delta t \sum_{j=1}^d r(y_n - t_j) p_j
 \end{aligned} \tag{14.12}$$

计算得到。

[平滑化] 对 $i = 1, \dots, d$

$$\begin{aligned}
 s_i &= \tilde{s}(t_i) = \tilde{f}(t_i) \int_{t_0}^{t_d} \frac{\tilde{q}(t_i - u) \tilde{s}(u)}{\tilde{p}(u)} du \\
 &= \tilde{f}(t_i) \sum_{j=1}^d \int_{t_{j-1}}^{t_j} \frac{\tilde{q}(t_i - u) \tilde{s}(u)}{\tilde{p}(u)} du \\
 &= \Delta t \cdot f(t_i) \sum_{j=1}^d \frac{q_{i-j} s_j}{p_j}
 \end{aligned} \tag{14.13}$$

不过, 在实际计算中, 预测和平滑化步骤之后, 还需要对近似得到的密度函数进行归一化, 使其在整个区间上的积分值为 1。例如, 对于 f_i , 需要计算

$$I(f) = \int_{t_0}^{t_d} f(t) dt = \Delta t (f_1 + \dots + f_d) \tag{14.14}$$

并将其修正为 $f_i \equiv \frac{f_i}{I(f)}$ 。

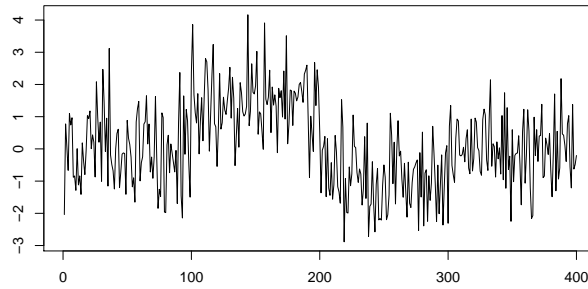


图 14.1 由 (14.15) 生成的数据

14.4 非高斯型趋势模型

在第 11 章的趋势模型中, 如果假定系统噪声 v_n 或观测噪声 w_n 服从非高斯型分布, 就可以得到非高斯型趋势模型 (non-Gaussian trend model)。本节将结合数值例子说明非高斯型模型的特点。

例: 非高斯型趋势估计

R 包 TSSS 中的函数 `ngsmth`, 针对具有非高斯型噪声分布的一维状态空间模型, 执行非高斯型滤波和平滑化。该函数的输入为

y: 一元时间序列。
noisev: 系统噪声密度函数的类型
 1: 高斯 (正态) 分布
 2: Pearson VII 型分布族
 3: 双边指数分布
tau2 系统分布的散布 (方差) 参数
bv: 系统噪声分布的形状参数
noisew: 观测噪声的分布类型
 1: 高斯 (正态) 分布
 2: Pearson VII 型分布族
 3: 双边指数分布
 4: 双重指数分布
sigma2: 观测噪声的散布 (方差) 参数
bw: 观测噪声的形状参数
initd: 初始状态的分布类型
 1: 高斯 (正态) 分布
 2: 均匀分布
 3: 双边指数分布
k: 数值积分的分点数
plot: 逻辑变量。若为 TRUE (默认值), 则将 **trend** 和 **smt** 作图显示

该函数的输出为:

trend: 趋势平滑后验分布的 7 个百分位点
smt: 时刻 $n = 1, \dots, N$ 的平滑密度函数

```

> y <- data( PfilterSample )
> par(mar=c(3,3,1,1)+0.1)
> #
> # 系统噪声: 高斯型 (正态分布)
> s1 <- ngsmth( y, noisev = 1, tau2 = 1.4e-02, noisew = 1, sigma2 = 1.048 )
> # 3D plot
> plot( s1, "smt", theta = 25, phi = 30, expand = 0.25, col="white" )
> #
> # 系统噪声: Pearson 分布族 (b=0.6)
> s2 <- ngsmth( y, noisev = 2, tau2 = 2.11e-10, bv = 0.6,
noisew = 1, sigma2 = 1.042 )
> plot( s2, "smt", theta = 25, phi = 30, expand = 0.25,col="white" )
> #
> # 系统噪声: 柯西分布 (Pearson 分布族 b=1)
> s2 <- ngsmth( y, noisev = 2, tau2 = 3.53e-5, bv = 1.0,
noisew = 1,sigma2 = 1.045 )

```

图 14.1 给出了按如下模型生成的数据。

$$y_n \sim N(\mu_n, 1), \quad \mu_n = \begin{cases} 0, & 1 \leq n \leq 100 \\ 1, & 101 \leq n \leq 200 \\ -1, & 201 \leq n \leq 300 \\ 0, & 301 \leq n \leq 400 \end{cases} \quad (14.15)$$

在中途 3 处平均值发生了变化。为了估计这种伴随结构变化而发生变化的数据均值函数，这里考虑如下 1 阶趋势模型。

$$\begin{aligned} t_n &= t_{n-1} + v_n \\ y_n &= t_n + w_n \end{aligned} \quad (14.16)$$

其中，对观测噪声 w_n 假定为正态分布，而对系统噪声 v_n ，作为非高斯分布的例子，考虑 Pearson VII 型分布族

$$q(v_n) = \frac{C}{(\tau^2 + v_n^2)^b}, \quad \frac{1}{2} < b < \infty \quad (14.17)$$

这里 C 是使 $q(v)$ 在全区间上的积分为 1 的归一化常数，其值为 $C = \tau^{2b-1} \Gamma(b) / \Gamma(\frac{1}{2}) \Gamma(b - \frac{1}{2})$ 。Pearson 分布族在 $b = 1$ 时为柯西分布，在 $b = (k+1)/2$ 时为自由度为 k 的 t 分布，而在 $b \rightarrow \infty$ 时则趋于正态分布。因此，通过改变 b 的取值，可以表示各种左右对称且比正态分布尾部更重的密度函数。

表 14.2 汇总了当 $b = \frac{3}{k}$ ($k = 0, \dots, 5$) 所表示的 6 个 b 的取值对应的 Pearson 分布族下，参数的极大似然估计值、AIC 值等结果。当 $b = 0.60$ 时，AIC 最小，而正态分布模型 ($b = \infty$) 时 AIC 最大。

表 14.2 非高斯型模型

b	τ^2	σ^2	对数似然	AIC
0.60	0.211×10^{-9}	1.042	-597.19	1198.38
0.75	0.299×10^{-7}	1.043	-597.39	1198.78
1.00	0.353×10^{-4}	1.045	-597.99	1198.98
1.50	0.303×10^{-2}	1.045	-599.13	1202.26
3.00	0.406×10^{-1}	1.046	-600.40	1204.80
∞	0.140×10^{-1}	1.048	-600.69	1205.38

图 14.2 画出了高斯型模型 ($b = \infty$) 和非高斯型模型 ($b = 0.6$) 得到的趋势 t_n 的平滑分布 $p(t_n|Y_N)$ 随时间变化的情形。对于高斯型模型，随着时刻 n 的推移，分布缓慢地向左右变动；而对于非高斯型模型，分布极其稳定，仅在 3 处发生了位置变化。图 14.3 左侧给出了在各时刻采用高斯型模型所估计的分布均值以及 $\pm 1, 2, 3 \times$ (标准差)。可以看到趋势出现了较大的起伏，并且无法很好地表示图 14.1 中明显可见

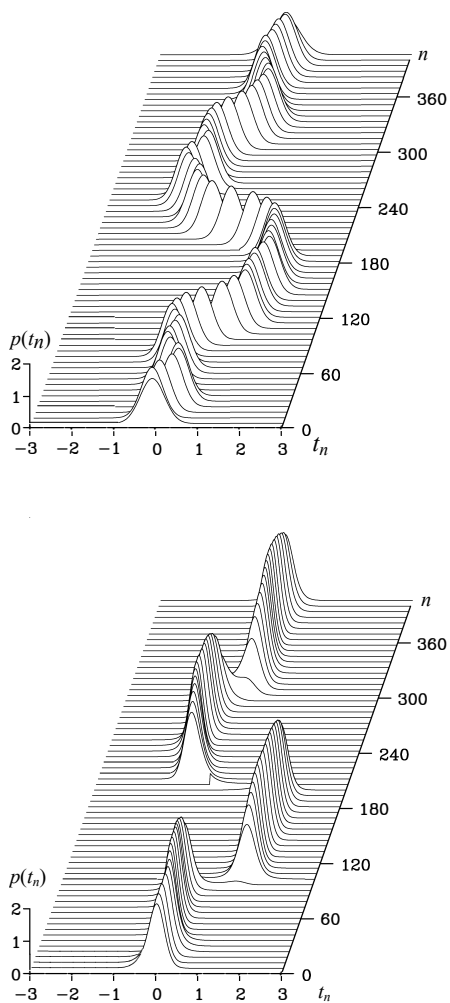


图 14.2 趋势平滑后验分布随时间的变化

的跳变。另一方面，图 14.3 右侧的 7 条曲线是在非高斯型模型下，对应于高斯型分布中位数以及 $\pm 1 \sim \pm 3 \times (\text{标准差})$ 的 0.13, 2.27, 15.87, 50.0, 84.13, 97.73, 99.87 百分位点。比较左右两图可以看出，采用非高斯型模型得到的估计值要平滑得多，而且对跳变的检测效果也很好。

14.5 非对称分布——时变方差模型

现实中得到的数据中，有些由于其生成机制的特性或某种非线性变换的结果，会呈现非对称分布。对于这种数据，如果假定通常的模型并应用最小二乘法等，可能会得到严重偏倚的结果。当观测噪声的分布已知时，直接使用其分布形式 $r(w)$ 进行平滑化，就能得到较好的结果。

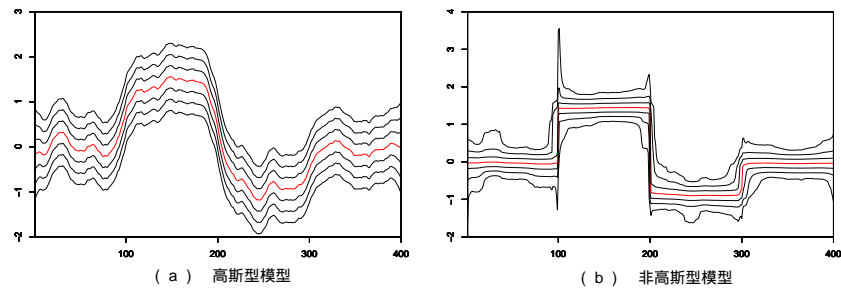


图 14.3 利用高斯型模型与非高斯型模型进行趋势估计

本节考虑 13.1 节中用于估计时变方差的模型

$$\begin{aligned} t_m &= t_{m-1} + v_m \\ y_m &= t_m + w_m \end{aligned} \quad (14.18)$$

在第 13 章中, 曾用正态分布 $N(-\zeta, \frac{\pi^2}{6})$ 近似观测噪声 w_n 的双重指数分布; 本节将比较两种情形: 一种是用正态分布近似, 另一种是利用精确密度函数进行数值计算。

图 14.4 是对图 1.1(f) 中的地震波数据做了滤波变换 $\tilde{y}_n = y_n - 0.5y_{n-1}$ 之后, 每隔一点取一个值, 并进一步按照 13.1 节的方法进行变换所得的数据。进行这样的预处理, 是因为本节的方法假定白噪声, 并且是为了减弱数据前半部分所见到的强自相关。

对该数据, 将 $\sigma^2 = \frac{\pi^2}{6}$ 固定, 并用极大似然法估计高斯型模型, 可得 $\hat{\tau}^2 = 0.04908$, $AIC = 2541.36$ 。图 14.5 中给出了用这一模型估计的趋势, 可以看到其上下起伏波动。这是因为图 14.4 中所见到的向下较大的

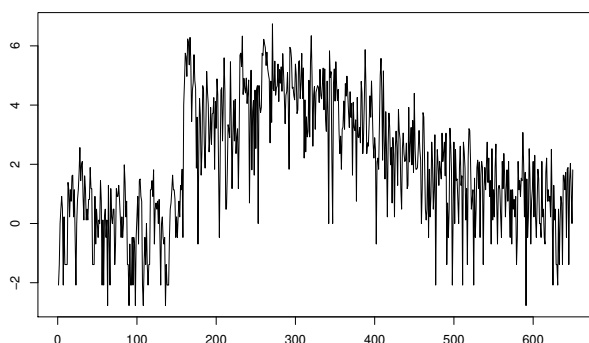


图 14.4 变换后的地震波数据

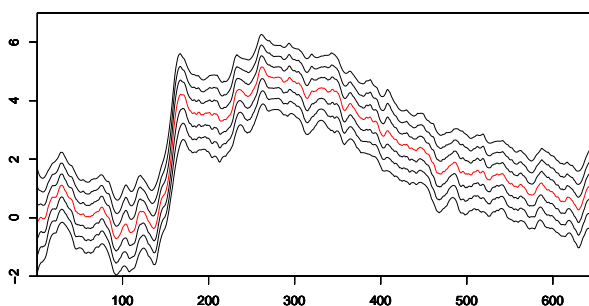


图 14.5 利用高斯型模型对趋势（方差的对数）的估计

```

> ## 地震波数据
> data(MYE1F)
> # 数据变换
> n <- length( MYE1F )
> yy <- rep(0, n)
> for (i in 2:n) yy[i] <- MYE1F[i] - 0.5 * MYE1F[i-1]
> m <- seq(1, n, by = 2)
> y <- yy[m]
> z <- tvvar( y, trend.order = 2, tau2.ini = 4.909e-02, delta = 1.0e-06)
>
> # 系统噪声：正态分布，观测噪声：正态分布
> s3 <- ngsnth( z$sm, noisev = 1, tau2 = z$tau2, noisew = 2, sigma2 = pi*pi/6, k =
190 )
>
> # 系统噪声：柯西分布，观测噪声：双重指数分布
> s3 <- ngsnth( z$sm, noisev = 2, tau2 = 0.00026, noisew = 4,
sigma2 = pi*pi/6, k = 190 )

```

偏差作为异常值起作用，并且趋势对其敏感地作出了响应，从而产生了波动。

另一方面，当把状态空间模型 (14.18) 的系统噪声与观测噪声的密度函数分别设

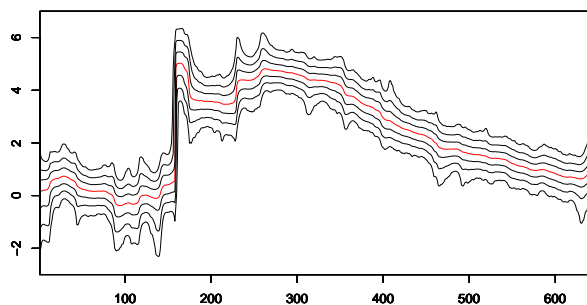


图 14.6 利用非高斯型模型对趋势（方差的对数）的估计

为柯西分布和双重指数分布

$$q(x) = \frac{1}{\pi} \frac{\tau}{\tau^2 + x^2} \quad (14.19)$$

$$r(x) = e^{x-e^x}$$

时，非高斯型模型的结果为 $\hat{\tau}^2 = 0.00017$ ， $AIC=2517.00$ 。由 AIC 的差异可以看出，非高斯型模型比高斯型模型拟合得更好。实际上，图 14.6 中的趋势比高斯型模型下平滑得多，并且很好地表示了伴随 P 波与 S 波到达而产生的方差急剧增大。另外，在该模型中，即使得到明显向下偏离的数据，也只是最下方的 0.13 百分位曲线发生波动，而 50 百分位点并未受到很大影响。由此可见，如果采用适当的模型进行非高斯型平滑化，就能够自动检测参数的跳变，并且得到一种不受分布偏斜影响的估计方法。

例：日经 225 平均股价数据

```

> data( Nikkei225 )
> y <- diff( log( Nikkei225 ) )
> # 数据变换与参数估计
> z <- tvvar( y, trend.order = 2, tau2.ini = 4.909e-02, delta = 1.0e-06)
>
> # 非高斯型平滑化
> s3 <- ngsnth( z$sm, noisev = 1, tau2 = 0.07, noisew = 4, sigma2 = pi*pi/2, k =
190)
> tt <- s3$trend
>
> # 结果显示
> par( mfrow=c(3,1), mar=c(2,2,1,1) )
> plot( z$sm,type="l", ylim=c(-14,-4) )
> plot( tt[,4],type="l", ylim=c(-14,-4),lwd=2 )
> for (j in 1:7){
> lines( tt[,j],type="l", ylim=c(-14,-4) )
> }
> plot( exp(tt[,4]/2),type="l" )

```

时变方差的估计方法可用于金融时间序列中随机波动率的估计。图 14.7(a) 是对日经 225 平均股价数据的对数差分序列，按照 13.1 节的方法进行变换后的结果。对于该数据，使用非高斯型模型 (14.18)，其趋势所对应的 $\log \sigma_n^2$ 的后验分布如 (b) 所示。进一步地，由 (b) 中间的曲线 (50% 点) 取一半再取指数，即

$$\exp(\log \hat{\sigma}_n^2 / 2) = \sqrt{\hat{\sigma}_n^2} = \hat{\sigma}_n$$

便可得到波动率的估计值。

这种时变方差的估计方法也可以立即应用于周期图的平滑。由于已知周期图服从自由度为 2 的 χ^2 分布，因此取其的对数后得到的序列就服从与 (14.19) 相同的分布。因此，本节的方法也适用于对数周期图的平滑化。

另外，利用非高斯型模型，还可以从经过

$$z_n = \log y_n^2 \quad (14.20)$$

变换后的数据中估计时变方差。在这种情况下，只需将由自由度为 1 的 χ^2 分布取对数变换后得到的密度函数

$$r(w) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{w - e^w}{2}\right\} \quad (14.21)$$

作为观测噪声的密度函数即可。在这种情况下，无需作出 $\log \sigma_{2m-1}^2 = \log \sigma_{2m}^2$ 这一假设，也能估计时变方差。

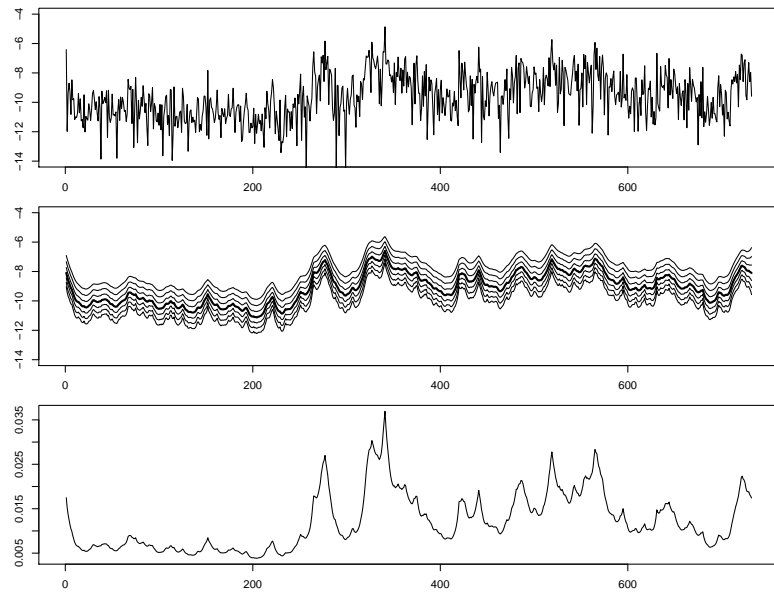


图 14.7 (a) 日经 225 数据的变换值, (b) $\log \sigma_n^2$ 的后验分布, (c) 波动率的估计值。

14.6 非高斯型状态空间模型的应用

本章的非高斯型状态空间模型具有极其广泛的应用范围。本节将针对若干情形说明非高斯型状态空间模型的使用方法。

14.6.1 利用混合高斯分布处理异常值

可以认为异常值服从与通常观测噪声分布不同的另一种分布。为了处理时间序列中的异常值, 可以假定异常值以概率 $\alpha \ll 1$ 出现, 并将观测噪声 w_n 的分布设为混合分布

$$r(w) = (1 - \alpha)\varphi(w; \mu, \sigma^2) + \alpha\varphi(w; \xi, \delta^2) \quad (14.22)$$

其中, $\varphi(x; \mu, \sigma^2)$ 表示均值为 μ 、方差为 σ^2 的正态分布的密度函数。右边第 1 项对应正常观测值的分布, 第 2 项对应异常值的分布。特别地, 当 $\xi = \mu = 0$ 、 $\delta^2 \gg \sigma^2$ 时, 异常值可以看作均值为 0 但具有很大方差噪声的观测值。柯西分布等可以视为对此类混合分布的近似。因此, 也可以通过将柯西分布或 Pearson 分布族等用作观测噪声的密度函数来处理异常值。

另一方面, 当像图 1.1(h) 中的地下水位数据那样, 异常总是向上出现时, 可以令 $\xi \gg \mu$ 、 $\delta^2 = \sigma^2$, 从而在忽略这些异常值的同时估计趋势。

14.6.2 非平稳离散序列

非高斯型模型也可以应用于非平稳二项分布和泊松分布等离散过程的平滑化。

下面考虑取整数值的时间序列 $\{m_1, \dots, m_N\}$ 。设该时间序列在第 n 期取值 m_n 的概率服从泊松分布

$$P(m_n|p_n) = \frac{e^{-p_n} p_n^{m_n}}{m_n!} \quad (14.23)$$

其中, p_n 表示某一事件在单位时间内发生的平均次数。这里, 泊松分布的均值并不是常数, 而是随时间逐渐变化。这样的模型称为**非平稳泊松过程模型**(nonstationary Poisson process model)。如果将趋势成分模型用作 p_n 的时间变化模型, 就可以得到 $p_n = p_{n-1} + v_n$ 这样的模型。不过, 由于 p_n 始终必须满足 $0 < p_n < \infty$, 因此可以通过 $\theta_n = \log p_n$ 定义新的变量, 并把它作为状态来引入趋势成分模型。因此, 用于平滑非平稳泊松过程的模型为

$$\theta_n = \theta_{n-1} + v_n \quad (14.24)$$

$$P(m_n|\theta_n) = \frac{e^{-p_n} p_n^{m_n}}{m_n!} \quad (14.25)$$

其中 $p_n = e^{\theta_n}$ 。这里, 第一式对应系统模型, 第二式对应观测模型, 它们各自给出了条件分布。因此, 对于这种离散序列, 也可以应用本章的方法, 并从数据中估计随时间变化的泊松概率。

完全类似地, 非高斯型状态空间模型也可以应用于非平稳二项分布的平滑化。

设在第 n 期的 ℓ_n 次观测中, 某一事件发生 m_n 次的概率服从二项分布

$$P(m_n|p_n, \ell_n) = \binom{\ell_n}{m_n} p_n^{m_n} (1-p_n)^{\ell_n - m_n} \quad (14.26)$$

这里, p_n 称为二项概率, 表示某一事件发生的平均概率。在这种情况下, p_n 随时间变化的模型称为**非平稳二项过程模型**(nonstationary binary process model), 并可利用如下模型来估计 p_n :

$$\begin{aligned} \theta_n &= \theta_{n-1} + v_n \\ P(m_n|\theta_n, \ell_n) &= \binom{\ell_n}{m_n} p_n^{m_n} (1-p_n)^{\ell_n - m_n} \end{aligned} \quad (14.27)$$

其中, $p_n = e^{\theta_n} / (1 + e^{\theta_n})$ 。

14.6.3 直接估计时变方差的方法

在 14.5 节中, 我们通过对时间序列平方值的对数进行平滑化来估计时变方差; 而利用如下模型, 也可以进行更为直接的估计:

$$\begin{aligned} t_n &= t_{n-1} + v_n \\ y_n &\sim N(0, e^{t_n}) \end{aligned} \quad (14.28)$$

在该模型中, 方差始终为正, 因此将其对数记为 t_n , 并把它作为状态。此时, e^{t_n} 就是时刻 n 的方差。

如果利用这一方法从日经 225 平均股价数据的对数差分序列中估计 t_n , 则可得与图 4.8 类似的结果。估计时得到的对数似然值在两种方法之间完全不同, 但这是因为用于模型估计的数据不同。若对 4.6 节中的数据变换进行应用, 并对 AIC 作修正, 则可以确认这两个模型实际上是相同的。

14.6.4 非线性状态空间模型

下面考虑由以下非线性状态空间模型表示的系统:

$$\begin{aligned} x_n &= f(x_{n-1}) + w_n \\ y_n &= h(x_n) + \varepsilon_n, \end{aligned} \quad (14.29)$$

其中, y_n 和 x_n 分别是 1 维时间序列和 m 维状态向量, $f(x)$ 和 $h(x)$ 是状态 x 的非线性函数。另外, w_n 与 ε_n 分别是具有密度函数 $q(w)$ 、 $r(\varepsilon)$ 的 m 维和 1 维白噪声, 并与 x_n 和 y_n 的过去独立。初始状态向量 x_0 服从密度函数 $p(x_0)$ 。

对于非线性状态空间模型 (14.29), 条件分布 $p(x_n|x_{n-1})$ 和 $p(y_n|x_n)$ 为

$$\begin{aligned} p(x_n|x_{n-1}) &= q(x_n - f(x_{n-1})) \\ p(y_n|x_n) &= r(y_n - h(x_n)). \end{aligned} \quad (14.30)$$

此外, 还可验证以下关系成立:

$$\begin{aligned} p(x_n|x_{n-1}, Y_{n-1}) &= p(x_n|x_{n-1}) \\ p(y_n|x_n, Y_{n-1}) &= p(y_n|x_n) \\ p(x_n|x_{n+1}, Y_N) &= p(x_n|x_{n+1}, Y_n). \end{aligned} \quad (14.31)$$

由于一步预测、滤波和平滑化公式仅依赖于 (14.31) 中的关系推导而来, 因此, 非高斯型滤波与平滑化公式同样可以应用于非线性状态空间模型。关于这一方法的详细内容, 请参见 [23] 和 [26]。

● 章末习题

1. 说明柯西分布和双边指数分布在建模中的作用。
2. 当 y_n 服从均值为 0、方差为 1 的正态分布时, 验证 $v_n = \log y_n^2$ 的分布由 (14.21) 给出。
3. 考虑如下模型: 趋势以概率 α 按正态分布发生变化, 其余情况下保持不变。

第 15 章

粒子滤波与平滑化

本章介绍作为高维非线性、非高斯型状态空间模型的滤波与平滑方法而开发的粒子滤波^{*1}（也称为蒙特卡罗滤波、Bootstrap 滤波）。这种方法不同于传统的近似法，它用大量粒子来近似一般的非高斯型分布。借此，只需通过各粒子的时间演化和重采样（有放回抽样）这两个简单操作，就能够实现复杂系统的滤波与平滑化。

15.1 非线性 · 非高斯型状态空间模型与分布的近似

本章考虑如下形式的非线性、非高斯型状态空间模型

$$x_n = F(x_{n-1}, v_n) \quad (15.1)$$

$$y_n = H(x_n, w_n), \quad (15.2)$$

其中， y_n 是时间序列， x_n 是 k 维状态向量，(15.1) 称为系统模型或状态模型，(15.2) 称为观测模型。它是对前章所使用的状态空间模型的推广。系统噪声 v_n 与观测噪声 w_n 分别服从密度函数 $q(v)$ 与 $r(w)$ ，是 ℓ 维与 1 维白噪声。另外，初始状态向量 x_0 服从密度函数 $p_0(x)$ 。函数 F 与 H 一般为非线性函数。不过，对于 $y = H(x, w)$ ，假定在给定 x 与 y 时， w 可唯一确定，并且可用关于 y 可微的函数 G 表示为 $w = G(y, x)$ 。满足这一条件的例子包括 $H(x, w) = H_1(x) + w$ 和 $H(x, w) = e^x w$ ，此时分别有 $G(y, x) = y - H_1(x)$ 、 $G(y, x) = e^{-x} y$ 。为简便起见，本章中假定 y_n 与 w_n 为一维，但也可以扩展到多维情形。

本章将针对这一非线性、非高斯型状态空间模型，讨论状态估计问题。与前章相同，记到时刻 t 为止得到的观测值为 $Y_t \equiv \{y_1, \dots, y_t\}$ ，并说明如何通过粒子近似来计算预测分布 $p(x_n | Y_{n-1})$ 、滤波分布 $p(x_n | Y_n)$ 以及平滑化分布 $p(x_n | Y_N)$ 。

^{*1}（译注）原文为“粒子フィルタ”（particle filter）。该方法也称为蒙特卡罗滤波（Monte Carlo filter）或 Bootstrap 滤波，在中文文献中也常称为“粒子滤波器”。

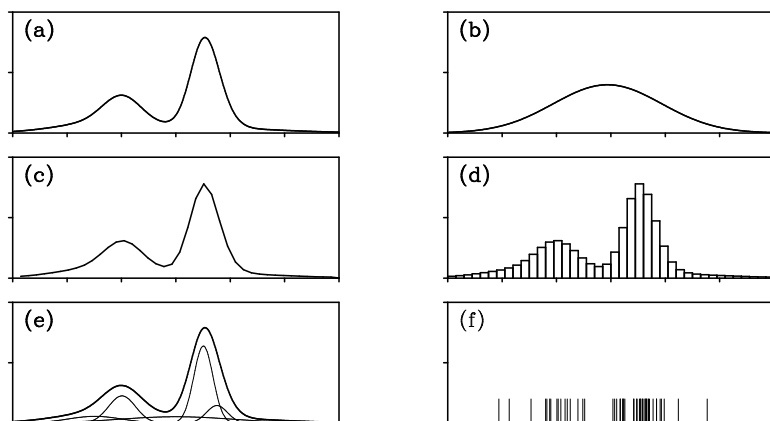


图 15.1 非高斯型分布的近似方法：(a) 假定的真实分布，(b) 正态近似，(c) 折线近似，(d) 阶梯函数近似，(e) 混合正态近似，(f) 用实现值进行近似

图 15.1 展示了近似非高斯型分布的方法。假定 (a) 中双峰分布为真实分布。(b) 表示用单个正态分布对其进行近似的情况，相当于扩展卡尔曼滤波或将模型作线性/正态近似的情形。当真实分布为双峰以上或左右不对称时，这种方法无法得到好的近似。(c) 表示用 40 个分点的折线进行近似，(d) 表示用 40 个分点的阶梯函数进行近似。前章中的非高斯型滤波与平滑，在数值计算时使用的就是这些近似；实际计算中，分点数往往取更大，通常可达数百。这些方法能够以相当高的精度近似任意分布。(e) 表示用 5 个正态分布的混合进行近似的情形。细线表示各个正态分布的贡献部分，粗线表示它们相加后得到的混合正态分布。高斯和滤波/平滑化所使用的正是这种近似。

本章介绍的粒子滤波方法不同于这些近似方法，它用大量粒子来表示分布。每个粒子都是从待表示分布中独立生成的一个实现值。图 15.1(f) 表示用 100 个实现值对 (a) 中的分布进行近似的情形。横轴上的短竖线表示实现值的位置。在 (a) 中密度函数较高的区域，粒子分布得更多。图 15.2 比较了由这些实现值构成的经验分布函数与真实分布函数。(a)、(b)、(c) 分别对应 $m=10, 100, 1000$ 的情形。粗线表示真实分布函数，细线表示经验分布函数。当 $m=10$ 时，与真实分布函数偏差较大；但随着 $m=100, 1000$ 的增加，可以看出能够得到非常好的近似。

在粒子滤波中，预测、滤波、平滑化的分布都用 m 个“粒子”按表 15.1 所示进行近似。粒子数 m 会根据模型复杂度和所需精度，通常取 1000 个到 10 万个左右。这相当于用由大量“粒子”所定义的经验分布函数，来近似各个分布函数。

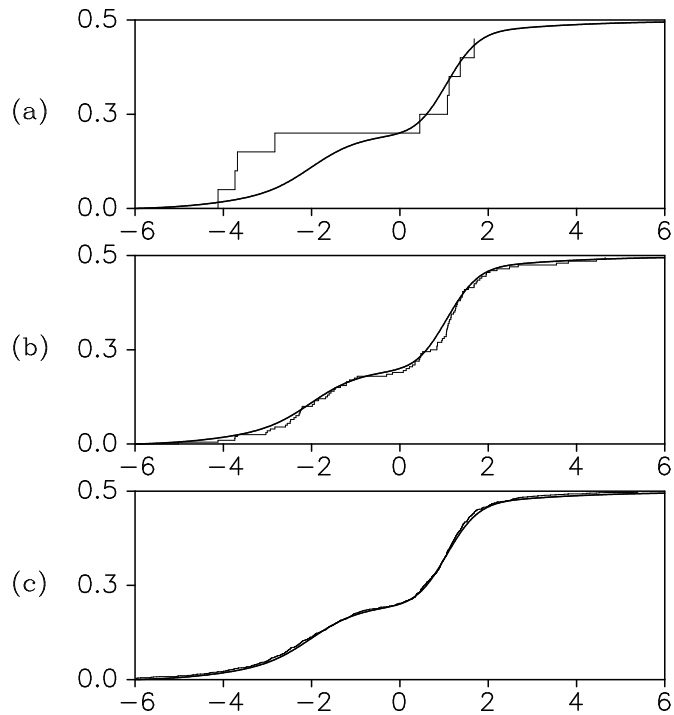


图 15.2 改变粒子数时经验分布函数与真实分布的比较：(a) $m = 10$, (b) $m = 100$, (c) $m = 1000$

15.2 粒子滤波

本节给出一种粒子滤波算法，用于逐次生成近似预测分布与滤波分布的粒子。根据粒子滤波，服从预测分布的粒子 $\{p_n^{(1)}, \dots, p_n^{(m)}\}$ 可以由前一期滤波实现值 $\{f_{n-1}^{(1)}, \dots, f_{n-1}^{(m)}\}$ 生成，而滤波实现值 $\{f_n^{(1)}, \dots, f_n^{(m)}\}$ 则可以通过对预测分布的实现值 $\{p_n^{(1)}, \dots, p_n^{(m)}\}$ 进行重采样来生成。

15.2.1 一步预测

在预测步骤中，假定已经给定可视为状态 x_{n-1} 的条件分布 $p(x_{n-1}|Y_{n-1})$ 的实现值的 m 个粒子 $\{f_{n-1}^{(1)}, \dots, f_{n-1}^{(m)}\}$ ，以及可视为系统噪声 v_n 的独立实现值的粒子

表 15.1 粒子滤波中的近似

分布	密度函数	由粒子进行的近似
预测分布	$p(x_n Y_{n-1})$	$\{p_n^{(1)}, \dots, p_n^{(m)}\}$
滤波分布	$p(x_n Y_n)$	$\{f_n^{(1)}, \dots, f_n^{(m)}\}$
平滑化分布	$p(x_n Y_N)$	$\{s_{n N}^{(1)}, \dots, s_{n N}^{(m)}\}$
系统噪声的分布	$p(v_n)$	$\{v_n^{(1)}, \dots, v_n^{(m)}\}$

$\{v_n^{(1)}, \dots, v_n^{(m)}\}$ 。也就是说, 对 $i = 1, \dots, m$ 有

$$f_{n-1}^{(i)} \sim p(x_{n-1}|Y_{n-1}), \quad v_n^{(i)} \sim q(v) \quad (15.3)$$

成立。此时,

$$p_n^{(j)} = F(f_{n-1}^{(j)}, v_n^{(j)}) \quad (15.4)$$

所定义的 $p_n^{(j)}$, 可视为从 x_n 的一步预测分布中生成的粒子。(参见附录 D)

15.2.2 滤波

接下来, 在滤波步骤中, 首先计算基于观测值 y_n 得到的粒子 $p_n^{(j)}$ 的“似然”(或者贝叶斯系数) $\alpha_n^{(j)}$ 。即, 对 $j = 1, \dots, m$ 定义

$$\alpha_n^{(j)} = p(y_n | p_n^{(j)}) = r(G(y_n, p_n^{(j)})) \left| \frac{\partial G}{\partial y_n} \right|$$

其中, G 是函数 H 的逆函数, r 是观测噪声 w 的密度函数。 $\alpha_n^{(j)}$ 可以看作表示粒子 $p_n^{(j)}$ 重要程度的权重系数。

接着, 对 $p_n^{(1)}, \dots, p_n^{(m)}$ 按其“似然” $\alpha_n^{(1)}, \dots, \alpha_n^{(m)}$ 成比例的概率进行重采样, 从而得到 m 个粒子 $f_n^{(1)}, \dots, f_n^{(m)}$ 。也就是说, 新的粒子 $f_n^{(j)}, (j = 1, \dots, m)$ 由

$$f_n^{(j)} = \begin{cases} p_n^{(1)} & \text{概率 } \alpha_n^{(1)} / (\alpha_n^{(1)} + \dots + \alpha_n^{(m)}) \\ \vdots & \vdots \\ p_n^{(m)} & \text{概率 } \alpha_n^{(m)} / (\alpha_n^{(1)} + \dots + \alpha_n^{(m)}) \end{cases} \quad (15.5)$$

进行有放回抽样得到。此时 $\{f_n^{(1)}, \dots, f_n^{(m)}\}$ 可视为来自滤波分布 $p(x_n | Y_n)$ 的实现值。

(附录 D)

15.2.3 粒子滤波算法

将以上结果归纳起来, 可得如下粒子滤波算法。

1. 对 $j = 1, \dots, m$, 生成 k 维随机数 $f_0^{(j)} \sim p_0(x)$ 。
2. 对 $n = 1, \dots, N$ 执行以下步骤。
 - (i) 对 $j = 1, \dots, m$:
 - 生成 ℓ 维随机数 $v_n^{(j)} \sim q(v)$ 。
 - 计算 $p_n^{(j)} = F(f_{n-1}^{(j)}, v_n^{(j)})$ 。
 - 计算 $\alpha_n^{(j)} = r(G(y_n, p_n^{(j)})) \left| \frac{\partial G}{\partial y_n} \right|$ 。
 - (ii) 从 $\{p_n^{(1)}, \dots, p_n^{(m)}\}$ 中按 $\{\alpha_n^{(1)}, \dots, \alpha_n^{(m)}\}$ 成比例的概率进行 m 次重采样(有放回抽样), 生成 $\{f_n^{(1)}, \dots, f_n^{(m)}\}$ 。

15.2.4 模型的似然

状态空间模型 (15.1)、(15.2) 通常包含噪声方差以及 F 、 H 的系数等未知参数 θ 。给定观测值 y_1, \dots, y_N 时, 由 (9.21) 可知参数 θ 的对数似然为

$$\ell(\theta) = \sum_{n=1}^N \log p(y_n | Y_{n-1}), \quad (15.6)$$

其中, $p(y_1 | Y_0) = p_0(y_1)$ 。这里, 利用预测分布的粒子近似

$$\begin{aligned} p(y_n | Y_{n-1}) &= \int p(y_n | x_n) p(x_n | Y_{n-1}) dx_n \\ &\cong \frac{1}{m} \sum_{j=1}^m p(y_n | p_n^{(j)}) = \frac{1}{m} \sum_{j=1}^m \alpha_n^{(j)}, \end{aligned} \quad (15.7)$$

则对数似然可近似为

$$\ell(\theta) = \sum_{n=1}^N \log p(y_n | Y_{n-1}) \cong \sum_{n=1}^N \log \left(\sum_{j=1}^m \alpha_n^{(j)} \right) - N \log m \quad (15.8)$$

参数的极大似然估计值 $\hat{\theta}$ 可以通过数值方式极大化这一对数似然来求得。

不过, 需要注意的是, 利用粒子滤波得到的对数似然中存在由蒙特卡罗方法带来的近似误差。解决这一问题的一种方法是, 将参数 θ 也视为状态向量的一部分, 从而进行联合估计的自组织建模。

15.2.5 关于重采样方法

本节讨论滤波步骤中所需的重采样方法。基于随机抽样的基本算法如下。

- 对 $j = 1, \dots, m$, 重复执行以下 (a)~(c)。

(a) 生成均匀随机数 $u_n^{(j)} \in U[0, 1]$ 。

(b) 寻找满足 $\frac{1}{C} \sum_{\ell=1}^{i-1} \alpha_n^{(\ell)} < u_n^{(j)} \leq \frac{1}{C} \sum_{\ell=1}^i \alpha_n^{(\ell)}$ 的 i 。

其中, $C = \sum_{\ell=1}^m \alpha_n^{(\ell)}$ 。

(c) 令滤波粒子为 $f_n^{(j)} = p_n^{(i)}$ 。

不过, 上述重采样的目的, 是将由带权重 $\{\alpha_n^{(1)}, \dots, \alpha_n^{(m)}\}$ 的粒子 $\{p_n^{(1)}, \dots, p_n^{(m)}\}$ 所定义的分函数, 重新表示为等权经验分布函数的形式, 因此并不一定必须进行严格的随机抽样。考虑到这一点, 在重采样方法中, 关于排序和随机数的生成方式可以有各种变形。一种改进方法如下, 即采用分层抽样。也就是说, 将区间 $(0, 1]$

划分为 m 个等长小区间, 并在每个区间内求一个 $u_n^{(j)}$ 。在这种情况下, 步骤 (a) 可由以下两种方法之一替代:

(a-S) 生成均匀随机数 $u_n^{(j)} \sim U\left(\frac{j-1}{m}, \frac{j}{m}\right]$ 。

(a-D) 对固定的 $\alpha \in (0, 1]$, 令 $u_n^{(j)} = \frac{j-\alpha}{m}$ 。

对于 α , 可以取 $1/2$, 也可以用随机数来确定。严格地说, 要进行分层抽样, 首先需要将 $p_n^{(1)}, \dots, p_n^{(m)}$ 按大小顺序重新排列, 并相应地重新排列 $\alpha_n^{(1)}, \dots, \alpha_n^{(m)}$, 然后再进行上述抽样。然而, 排序在粒子滤波中, 尤其当粒子数 m 很大时, 会成为最耗时的运算; 并且, 如果将分层抽样的目的理解为: 从权重总和为 $1/m$ 的粒子组中恰好抽取 1 个粒子, 那么实际上并不一定需要按顺序重新排序。

根据比较实验的结果, 与严格随机抽样相比, 采用分层的方法能够得到好得多的近似; 而且在两种分层方法中, (a-D) 的确定性方法比 (a-S) 的分层抽样法还要略好一些。

15.2.6 数值例

为了说明一步预测分布和滤波分布是如何用“粒子”近似的, 这里给出仅一步的粒子滤波结果 ($n=1$)。考虑如下 1 阶线性、非高斯型模型。

$$\begin{aligned} x_n &= x_{n-1} + v_n \\ y_n &= x_n + w_n \end{aligned} \quad (15.9)$$

其中, v_n 与 w_n 分别服从密度函数为 $q(v) = \sqrt{0.1}\pi^{-1}(v^2 + 0.1)^{-1}$ 的柯西分布以及标准正态分布 $N(0, 1)$ 的白噪声。另外, 假定初始分布 $p_0(x_0)$ 也服从标准正态分布。不用说, 在粒子滤波中, 正态性的假设并不是本质性的。在上述条件下, 求得了预测分布 $p(x_1|Y_0)$ 以及滤波分布 $p(x_1|Y_1)$ 。粒子数取 $m=100$ 。之所以采用较少的粒子数 $m=100$, 是为了使图更易于观察; 在通常计算中, 会使用更多的粒子。

图 15.3(a) 中的实线表示所假定的初始分布 $p_0(x_0)$ 。竖线表示从该分布生成的 100 个实现值的位置, 条形函数表示它们的直方图。另一方面, (b) 中的粗线表示真实分布函数, 细线表示由 (a) 中的实现值得到的经验分布函数。同样地, (c)、(d) 给出了系统噪声 v_1 的密度函数、实现值等。

(e)、(f) 与预测分布 $p(x_1|Y_0)$ 有关。(e) 中的曲线是由 (a) 与 (c) 的密度函数通过数值积分求得的“真实”密度函数。(f) 中的实线则是对其进一步积分得到的“真实”

分布函数。另一方面, (e) 中的竖线表示将 (a)、(c) 中的实现值代入式 (15.4) 后得到的 $p_1^{(j)}$ 的位置。其直方图构成了对真实函数的近似。经验分布函数与真实分布函数之间的关系则示于 (f)。

(g) 中的曲线是在假定得到观测值 $y_1 = 2$ 时的滤波分布, 它由式 (14.5) 的非高斯型滤波求得。(g) 中的粒子位置与 (e) 相同, 但竖线高度与似然 $\alpha_n^{(j)}$ 成正比。因此, (h) 中的累积分布函数其阶梯宽度与 $\alpha_n^{(j)}$ 成正比, 虽然阶梯位置与 (f) 相同, 但由于阶梯高度不同, 因此与等步长的 (f) 完全不同, 并近似了滤波分布。(i) 表示重采样后的直方图、粒子位置以及精确的滤波分布。可以看出, (i) 中的密度函数与 (j) 中的经验分布函数, 都分别很好地近似了 (g)、(h)。

15.3 粒子平滑化

上一节中的粒子滤波方法, 只需保存过去的粒子, 就可以扩展到平滑化的情形。以下约定, $(s_{1|n}^{(j)}, \dots, s_{n|n}^{(j)})^T$ 表示联合分布 $p(x_1, \dots, x_n | Y_n)$ 的第 j 个实现值。此时, 为了进行平滑化, 只需将 15.2.3 节算法步骤 2 中的 (ii) 改为如下形式即可。

(ii-S) 对于 $j = 1, \dots, m$, 通过对 $(s_{1|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_n^{(j)})^T$ 进行重采样, 生成 $(s_{1|n}^{(j)}, \dots, s_{n-1|n}^{(j)}, s_{n|n}^{(j)})^T$ 。

在这一修改中, 只需将 $\{(s_{1|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_n^{(j)})^T, j = 1, \dots, m\}$ 按照 15.2.3 节步骤 2(ii) 中的权重进行重采样, 就能够实现非线性、非高斯型模型的固定区间平滑化。但是, 如下所示, 由于实际计算是基于有限个 (m 个) 样本进行的, 实际上不同粒子的数目会单调不减; 随着不断重复重采样, 粒子会集中到少数几个粒子上, 最终将损害分布的形状。

因此, 在平滑化算法中, 可以考虑将 (ii-S) 修改为如下形式。

(ii-L) 对于 $j = 1, \dots, m$, 通过对 $(s_{n-L|n-1}^{(j)}, \dots, s_{n-1|n-1}^{(j)}, p_n^{(j)})$ 进行重采样, 生成 $(s_{n-L|n}^{(j)}, \dots, s_{n-1|n}^{(j)}, s_{n|n}^{(j)})$ 。其中, 令 $f_n^{(j)} = s_{n|n}^{(j)}$, 并将 L 取为固定值。

有趣的是, 这一修正后的算法, 等价于进行 L 滞后的固定滞后平滑化。当 L 取较大值时, $p(x_n | Y_{n+L})$ 能够较好地近似 $p(x_n | Y_N)$, 但是, $\{x_{n|n+L}^{(1)}, \dots, x_{n|n+L}^{(m)}\}$ 的分布反而会偏离 $p(x_n | Y_{n+L})$ 。因此, L 不宜取得过大。通常取 20 以下较为合适。

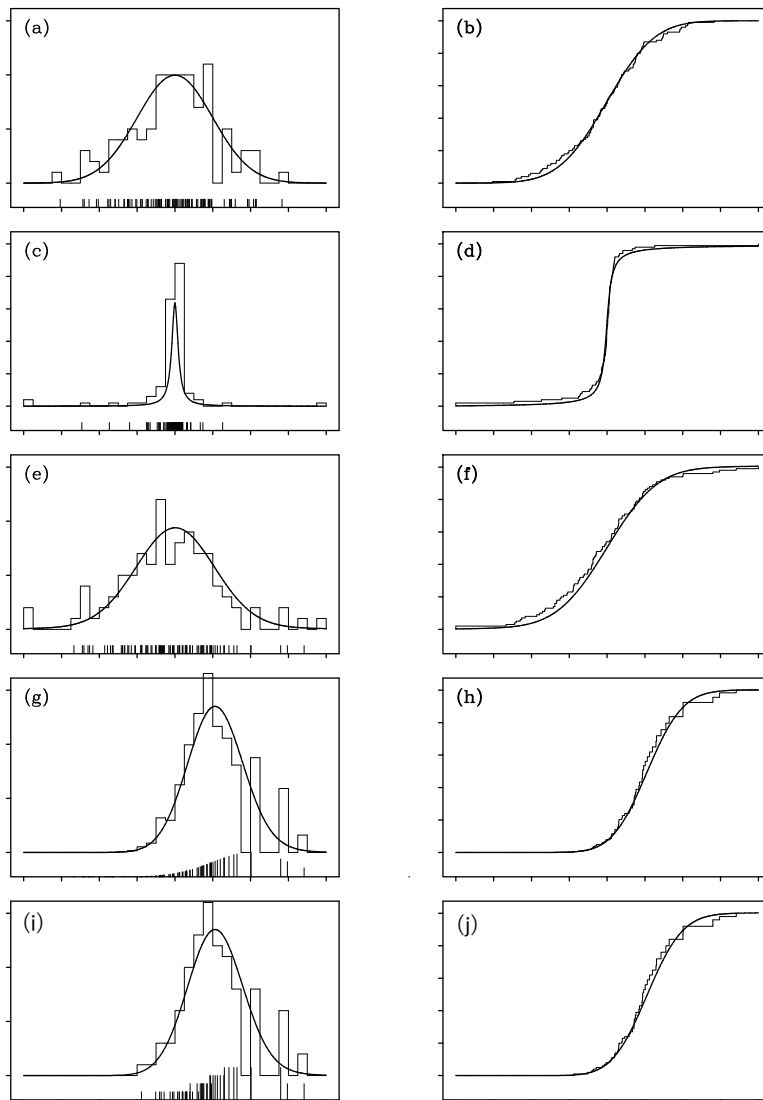


图 15.3 一步粒子滤波：(a) 初始分布的密度函数、实现值与直方图，(b) 初始分布的分布函数与经验分布函数，(c) 系统噪声的密度函数、实现值与直方图，(d) 系统噪声的分布函数与经验分布函数，(e) 预测分布的密度函数、实现值与直方图，(f) 预测分布的分布函数与经验分布函数，(g) 滤波分布的密度函数、带权实现值与直方图，(h) 滤波分布的分布函数与经验分布函数，(i) 重采样后的实现值与直方图，(j) 重采样后的经验分布函数

下面的例子是针对图 14.1 中人工生成的数据，假定使用如下的一阶趋势模型

$$\begin{aligned} x_n &= x_{n-1} + v_n \\ y_n &= x_n + w_n \end{aligned} \quad (15.10)$$

并用粒子滤波进行状态估计时得到的结果。这里假定 w_n 服从均值为 0、方差为 σ^2 的正态分布，而对系统噪声则考虑正态分布和柯西分布两种情形。

TSSS 包中的函数 `pfilter` 针对具有非高斯型系统噪声和观测噪声的线性状态空间模型，执行粒子滤波与平滑化。该函数需要如下输入：

y: 一元时间序列
m: 粒子数
model: 系统噪声的模型:
 = 0: 正态分布
 = 1: 柯西分布
 = 2: 混合高斯分布
lag: 固定滞后平滑化的滞后数
initd: 状态的初始分布:
 = 0: $N(0,1)$
 = 1: $U(-4,4)$
 = 2: $\text{Cauchy}(0,1)$
 = 3: 以 $x=0$ 为中心的 Delta 函数
sigma2: 观测噪声的方差 σ^2
tau2: 系统噪声的方差 τ^2
plot: 逻辑变量。若为 TRUE (默认值), 则显示后验分布图

该函数的输出为:

smooth.dist: 后验分布
llkhood: 模型的对数似然

图 15.4(a) 给出了系统噪声为正态分布时, 通过卡尔曼滤波得到的精确滤波分布。中央的粗线表示分布的均值, 三个灰色区域分别表示 $\pm 1\sigma$ 、 $\pm 2\sigma$ 、 $\pm 3\sigma$ 的范围。另一方面, (b)、(c)、(d) 中取 $L=20$, 并分别对应 $m=100$ 、1,000、10,000 的粒子滤波结果。图中的粗线表示中位数, 深灰色区域表示 $\pm 1\sigma$ 范围, 浅灰色区域表示 $\pm 2\sigma$ 范围。在 (d) 中, 还进一步给出了 $\pm 3\sigma$ 范围。(b) 中 $m=100$ 时, 虽然抓住了 (a) 的大致特征, 但波动较大, 不能算是很好的近似。然而, 随着粒子数的增加, 可以得到更精确的近似; 在 (c) 的 $m=1,000$ 情况下, 虽然 $\pm 3\sigma$ 点的波动仍稍大, 但其余范围已经得到了较好的近似。在 (d) 的 $m=10,000$ 情况下, 则成为对 (a) 的非常好的近似。

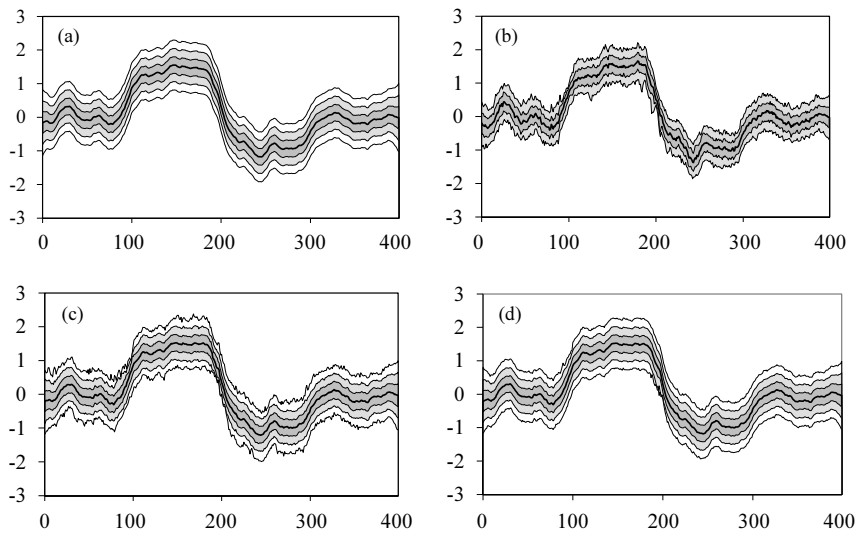


图 15.4 粒子滤波的结果：(a) 卡尔曼滤波得到的精确值，(b)~(d) 固定滞后 ($L=20$) 粒子平滑化的结果 ((b) $m=1,000$, (c) $m=10,000$, (d) $m=100,000$)

```

> # 输入测试数据：在 N=101, 201, 301 处有水平跳变
> y <- data( pfilter_sample )
>
> # 卡尔曼滤波
> # trend( y, trend.order = 1, tau2.ini = 1.4e-2, delta = 0.0 )
>
> # 粒子滤波
> # 系统噪声：正态分布，观测噪声：正态分布
> # 粒子数 m = 1,000:
> pfilter(y, m = 1000 , model = 0, lag = 20 , initd = 0, sigma2 = 1.048, tau2 =
1.40e-2)
>
> # 粒子数 m = 10,000:
> pfilter(y, m = 10000, model = 0, lag = 20 , initd = 0, sigma2 = 1.048, tau2 =
1.40e-2)
>
> # 粒子数 m = 100,000:
> pfilter(y, m = 100000, model = 0, lag = 20 , initd = 0, sigma2 = 1.048, tau2 =
1.40e-2)

```

```

> # 非高斯型平滑化
> # 系统噪声：柯西分布
> ngsmth( y, noisev = 2, bv = 1.0, tau2 = 1.40e-2, noisew = 1, sigma2 = 1.048 )
> #
> # 粒子滤波·平滑化
> # 系统噪声：柯西分布
> pfilter( y, m = 100000, model = 0, lag = 20 , initd = 0, sigma2 = 1.045, tau2 =
3.53e-5)

```

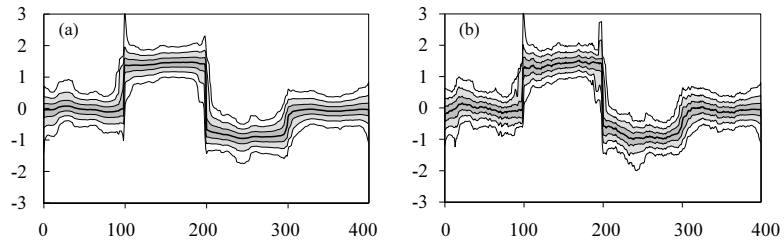


图 15.5 利用柯西分布进行平滑化: (a) 用非高斯型平滑化算法计算得到的精确分布. (b) 粒子平滑化结果 ($m = 100,000$)。

图 15.5 给出了系统噪声服从柯西分布时的结果。左图是第 14 章中的非高斯型平滑化得到的精确结果，右图是取 $m = 10000$ 、 $L = 20$ 时粒子滤波得到的平滑化结果。与左图相比，虽然波动稍大一些，但在 $n = 100$ 、 200 、 300 处的变化仍被很好地捕捉到了。另外， $\pm 3\sigma$ 的误差范围也被非常好地近似了。

15.3.1 非线性平滑化

利用粒子滤波，可以对非线性状态空间模型执行滤波与平滑化。图 15.6 的 (a) 和 (b) 给出了由如下非线性模型

$$\begin{aligned} x_n &= \frac{1}{2}x_{n-1} + \frac{25x_{n-1}}{x_{n-1}^2 + 1} + 8\cos(1.2n) + v_n \\ y_n &= \frac{x_n^2}{10} + w_n \end{aligned} \quad (15.11)$$

生成的 x_n 与 y_n 。其中， $v_n \sim N(0, 1)$ ， $w_n \sim N(0, 10)$ ， $v_0 \sim N(0, 5)$ 。这里考虑基于 100 个观测值 y_n ，估计未知状态 x_n 的问题。 x_n 会不时在正负两个区域之间切换，但在观测模型中，观测到的是平方后的 x_n^2 ，因此几乎无法判断其符号。已知扩展卡尔曼滤波在这种情况下常常会发散。图 15.6(c) 给出了用粒子滤波求得的 x_n 的后验分布 $p(x_n|Y_n)$ 。可以看出，即使在非线性模型中，也能够得到未知状态 x_n 的良好估计。

TSSS 包中的函数 `pfilterNL` 用于对上述非线性状态空间模型执行粒子滤波和平滑化。该函数需要如下输入：

- y: 一元时间序列
- m: 粒子数
- lag: 固定滞后平滑化的滞后数
- sigma2: 观测噪声的方差 σ^2
- tau2: 系统噪声的方差 τ^2
- plot: 逻辑变量。若为 TRUE (默认值)，则显示后验分布图

该函数的输出为：

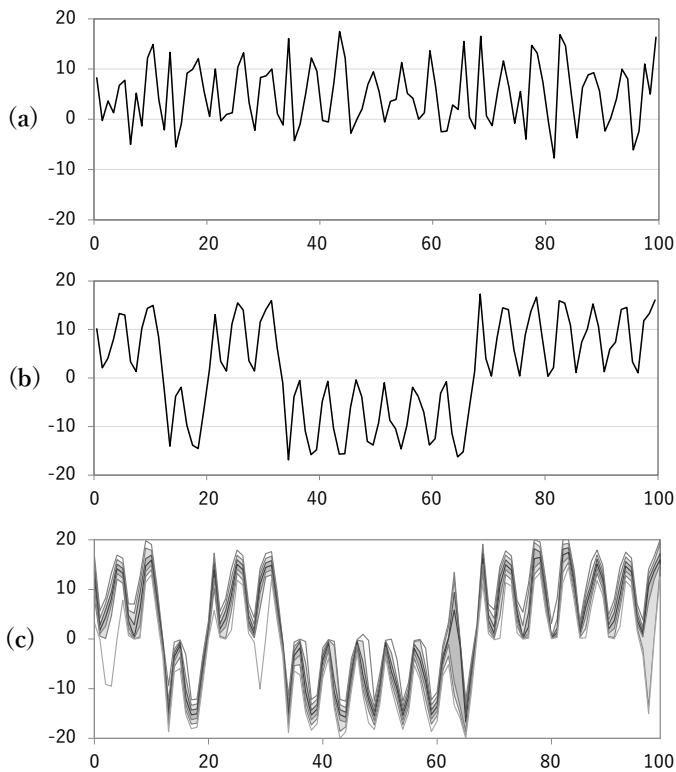


图 15.6 非线性平滑化: (a) 数据 y_n , (b) 未知状态 x_n , (c) 通过粒子平滑化得到的状态 x_n 的平滑分布

`smooth.dist`: 后验分布。但其中, `smooth.dist[,4]` 表示中位数,
`smooth.dist[,j]`, $j=1,2,3,5,6,7$: 表示 $-3, -2, -1, 1, 2, 3$ 个标准差
`llkhood`: 模型的对数似然

```
> data(NLmodel)
> x <- NLmodel[, 1]
> pfilterNL(x, m = 100000, lag = 20, sigma2 = 10.0, tau2 = 1.0, xrange = c(-20,
20), seed = 2019071117)
```

对于这种非线性模型, 已知扩展卡尔曼滤波有时会发散 ([23])。图 15.6(c) 给出了通过粒子滤波和平滑化所得到的平滑化后验分布。可以看出, 利用这种方法能够得到状态 x_n 非常好的估计值。

为了使粒子滤波和平滑化成功, 关键在于能够恰当地表示状态的非高斯型分布。图 15.7(a) 与 (b) 分别给出了在 $t = 30$ 和 48 时的状态边缘后验分布 $p(x_t|y_{1:t+j})$, $j = -1, 0, 1, 2$ 。无论是在 $t = 30$ 还是在 48 的情形中, 预测分布 $p(x_t|y_{1:t-1})$ 和滤波分布 $p(x_t|y_{1:t})$ 都是双峰型的。但是, 在 $t = 30$ 时的 2-滞后平滑化 $p(x_{30}|y_{1:32})$ 中, 分布左半部分消失了, 分布变成了单峰型。在 $t = 48$ 时也能看到同样的现象。这种情况

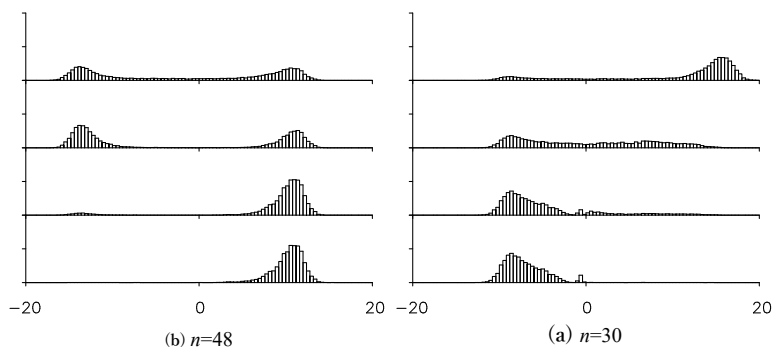


图 15.7 $t = 30$ (左) 和 $t = 48$ (右) 时的固定滞后平滑化。从上到下依次为: 预测分布、滤波分布、1-滞后以及 2-滞后平滑化。

下, 在预测分布中右侧峰值低于左侧峰值。然而, 在 1-滞后和 2-滞后平滑化的结果中, 右侧区域的峰值消失了。在这种情况下, 扩展卡尔曼滤波很可能得到符号相反的结果。

● 章末习题

1. 针对图 1(g) 中的日经 225 数据, 考虑一个同时描述趋势与波动率变化的模型。
2. 在蒙特卡罗滤波中, 若希望将估计精度提高一个数量级, 需要将粒子数增加到什么程度?
3. 在蒙特卡罗滤波中, 如果不将粒子限制为等概率, 而是直接利用其权重, 是否就可以不进行重采样?
4. 比较固定区间平滑化和固定滞后平滑化所需保存数据量的差异。

第 16 章

仿真

通过生成服从各种分布的随机数, 可以获得白噪声的实现值。时间序列模型将时间序列表示为以白噪声为输入的系统输出。因此, 给定时间序列模型后, 可以利用随机数生成白噪声的实现值并投入模型, 从而生成时间序列。本章将介绍服从各种分布的随机数的生成方法、服从 ARMA 模型的时间序列的生成方法, 以及利用状态空间模型统一仿真方法。

16.1 均匀随机数的生成

服从某种分布独立生成的值的序列称为**随机数** (random number)。换言之, 随机数就是白噪声的实现值。时间序列模型将时间序列表示为以白噪声为输入的系统输出。因此, 以随机数为输入, 可以生成服从各种模型的时间序列。这种时间序列的生成方法称为**仿真** (simulation)。

在计算机中使用随机数时, 通常使用按适当算法生成的伪随机数。进行时间序列仿真所需的服从正态分布等分布的随机数, 是先生成服从均匀分布的**均匀随机数** (uniform random number), 再通过适当变换求得。

长期以来广泛用于生成均匀随机数的方法是**同余法** (congruence method)。该方法以适当的整数 I_0 为初始值, 通过

$$I_k \equiv aI_{k-1} + c \pmod{m} \quad (16.1)$$

依次生成 I_1, I_2, \dots 。当 $c = 0$ 时, 特别称为乘法型同余法。此时, I_k 取 $[0, m)$ 上的值, 因此 $\frac{I_k}{m}$ 为 $[0, 1)$ 上的分布。

但是, 该方法生成的数列的周期最多为 m , 因此无法生成长度超过 m 的随机数。 a 、 c 、 m 选择适当的整数即可, 例如

$$\begin{aligned} a=1229 & \quad c=351750 & m=1664501 \\ a=1103515245 & \quad c=12345 & m=2^{31} \end{aligned}$$

等组合据说可以生成较好的均匀随机数 [27]。后者的组合在 1990 年代之前被 C 语言所使用。当 $m=2^p$ 、 $a=5 \pmod{m}$ 且 c 为奇数时，该算法生成的序列可以达到最大可能周期 m ，已知 0 到 $m-1$ 的所有数恰好各出现一次。

在带滞后的 Fibonacci 法中，整数序列 I_n 由以下递推关系生成。

$$I_{n+p} = I_{n+q} + I_n \pmod{m} \quad (16.2)$$

其中 m 通常设为 2^k (k 为比特长度)。此时，生成的序列最大周期为 $2^{k-1}(2^p-1)$ 。当前 C 语言中，为实现约 2^{60} 的周期，使用 $p=31$ 、 $q=3$ 、 $m=2^{31}$ 。

GFSR (generalized feedback shift register) 法 [28] 中， k 维二进制数列由以下递推关系生成。

$$I_{n+p} = I_{n+q} \oplus I_n \quad (16.3)$$

其中 $p > q$ ， \oplus 表示异或 (exclusive OR) 运算。整数 p 和 q 的选取要使多项式 $X^p + X^q + 1$ 不可约。此时，生成序列的最大周期为 $2^p - 1$ 。满足此条件的 p 和 q 的例子有以下组合。

$$\begin{aligned} p=521 & \quad q=158 & \text{周期 } 2^{521}-1 \\ p=521 & \quad q=32 & \text{周期 } 2^{521}-1 \\ p=607 & \quad q=273 & \text{周期 } 2^{607}-1 \end{aligned}$$

Twisted GFSR 法 [29] 中， k 维二进制数列由以下递推关系生成。

$$I_{n+p} = I_{n+q} \oplus I_n A \quad (16.4)$$

其中 A 为伴随矩阵等 $k \times k$ 正则矩阵。常用的 TT800 算法使用 $p=25$ 、 $q=7$ ，周期为 $2^{pk}-1$ 。进一步，Mersenne Twister 由

$$I_{n+p} = I_{n+q} \oplus I_{n+1} B \oplus I_n A \quad (16.5)$$

生成，当 $p=624$ 时，适当选择 A 和 B ，生成的序列周期为 $2^{19937}-1$ ，已知在 623 维空间中均匀分布 [30]。

与基于某种算法由软件生成的这些伪随机数不同，也有由硬件生成的物理随机数。物理随机数原理上可以生成无周期、无相关的序列，因此在仿真中需要更严格的随机数时被使用。

R 中均匀随机数的生成

```
> # 生成均匀随机数
> y <- runif( 100 )
>
> y <- runif( 100, min = -1, max = 1)
```

在 R 环境中, 可以通过函数 `runif` 生成基于 Mersenne Twister 的均匀随机数。调用该函数时需要以下输入:

`n`: 随机数的长度
`min, max`: 均匀分布的上限和下限

当未指定 `min` 或 `max` 时, 使用默认值 0 和 1。

16.2 白噪声的生成

为了生成正态白噪声的实现值, 只需生成服从正态分布的随机数序列即 **正态随机数** (normal random number)。正态随机数的生成方法中, Box-Muller 法广为人知。该方法是, 设 2 个独立的 $[0, 1)$ 上的均匀随机数为 U_1 、 U_2 ,

$$\begin{aligned} X_1 &= \sqrt{-2\log U_1} \cos 2\pi U_2 \\ X_2 &= \sqrt{-2\log U_1} \sin 2\pi U_2 \end{aligned} \quad (16.6)$$

由此定义的 X_1 和 X_2 互相独立地服从标准正态分布 $N(0, 1)$ 。但是, 实际上使用如下改进的不含 `cos`、`sin` 计算的 Marsaglia 算法, 可以更快速地生成正态随机数。

1. 生成均匀随机数 U_1 、 U_2 。
2. 令 $V_1 = 2U_1 - 1$ 、 $V_2 = 2U_2 - 1$ 。
3. 令 $S^2 = V_1^2 + V_2^2$ 。
4. 若 $S^2 \geq 1$ 则返回第 1 步。
5. $X_1 = V_1 \sqrt{\frac{-2\log S}{S}}$, $X_2 = V_2 \sqrt{\frac{-2\log S}{S}}$

R 中正态随机数的生成

```
> # 生成标准正态随机数
> y <- rnorm( 100 )
>
> # 生成均值 0、标准差 2 的正态随机数
> y <- rnorm( 100, mean = 0, sd = 2)
```

在 R 环境中，正态随机数可以用函数 `rnorm` 生成。所需输入如下：

`n`: 随机数的长度
`mean`: 正态分布的均值
`sd`: 正态分布的标准差

当未指定 `mean` 或 `sd` 时，分别使用默认值 0 和 1。

```
> # 生成均匀随机数
> plot( runif(200), type = "l", ylim = c(0,1) )
>
> # 标准正态随机数
> y <- rnorm( 200 )
> plot( y, type = "l", ylim = c(-3,3) )
>
> hist( y ) # 直方图
> acf( y ) # 自相关函数
```

图 16.1 的 (a) 是用 R 的函数 `runif` 生成的 200 个均匀随机数。相对地，(b) 是用函数 `rnorm` 生成的正态随机数（白噪声）。(c) 和 (d) 分别展示了这 200 个正态随机数的直方图和样本自相关函数。同样，(e) 和 (f) 是将数据数增加到 2000 个时的直方图和样本自相关函数。两个直方图都比较接近正态分布的密度函数，样本自相关函数在 $\pm 1/\sqrt{n}$ 的误差范围内，随着数据数的增加趋向于 0。

为了进行非高斯型状态空间模型的仿真，需要求服从任意密度函数 $p(x)$ 的白噪声实现值的方法。当非高斯型分布为柯西分布、 χ^2 分布、双指数分布等特殊分布时，有时可以通过适当变换均匀随机数或正态随机数来求得。首先介绍一些这样的例子。

16.2.1 χ^2 分布

自由度 j 的 χ^2 分布是用独立的 j 个正态随机变量 X_1, \dots, X_j

$$\chi_j^2 = X_1^2 + \dots + X_j^2 \quad (16.7)$$

表示的随机变量的分布。因此，要生成服从 χ^2 分布的随机数，只需先生成 j 个正态随机数，通过 (16.7) 求 χ_j^2 即可。特别地，令 $j=2$ 可得到服从指数分布的随机数。进一步令 $Z = \log \chi_2^2$ 可得到服从双指数分布的随机数。但是，设 u 为均匀随机数，指数随机数可由

$$v = -\log u \quad (16.8)$$

求得，双指数随机数可由

$$v = \log(-\log u) \quad (16.9)$$

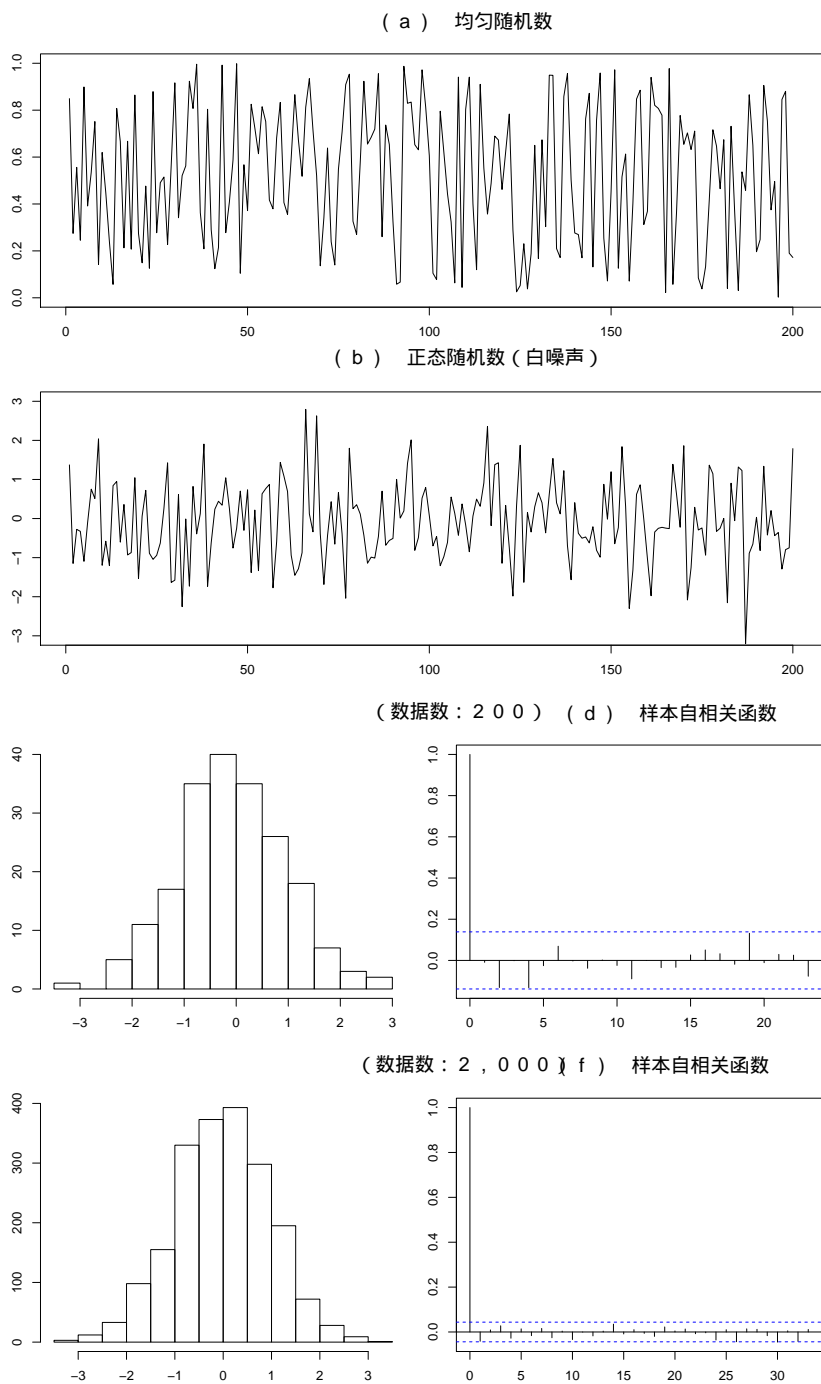


图 16.1 均匀随机数和白噪声的实现值、直方图和自相关函数。

直接从均匀随机数求得。

16.2.2 柯西分布

第 14 章非高斯型模型中使用的柯西分布 (Cauchy distribution) 的密度函数为

$$p(x) = \frac{1}{\pi(x^2 + 1)} \quad (16.10)$$

要生成服从柯西分布的随机数, 首先生成 $[0, 1)$ 上的均匀随机数 u , 当 $u \neq 0.5$ 时令

$$v = \tan \pi u \quad (16.11)$$

即可。

16.2.3 服从任意分布的随机数的生成

一般来说, 要生成服从任意密度函数 $f(x)$ 的随机数, 可以使用分布函数的逆函数方法。分布函数由

$$F(x) = \int_{-\infty}^x f(t) dt \quad (16.12)$$

定义的单调递增函数, 满足 $0 \leq F(x) \leq 1$ 、 $F(-\infty) = 0$ 、 $F(\infty) = 1$ 。求满足 $F(G(y)) = y$ 的逆函数 $G(y) \equiv F^{-1}(y)$ ($0 \leq y \leq 1$), 则 G 为从 $(0, 1)$ 到 $(-\infty, \infty)$ 的函数。此时, 令 u 为 $(0, 1)$ 上的均匀随机数, 则 $v = G(u)$ 为服从密度函数 $f(v)$ 的随机数。利用该方法和第 4 章定义的密度函数, 可以生成服从各种分布的随机数。

在 R 环境中, 可以用 `rXXX` 形式的函数生成服从以下各种连续分布的随机数。实际计算中将 `xxx` 替换为适当的名称即可:

```
rchisq:   $\chi^2$  分布
rt:      自由度  $df$  的  $t$  分布
rcauchy: 柯西分布
rexp:    指数分布
```

```
> # 生成服从各种分布的随机数
> y <- rnorm( 100 ) # 正态分布
>
> y <- rcauchy( 100, location = 0, scale = 1 ) # 柯西分布
>
> y <- rt( 100, df = 3, ncp = 0 ) #  $t$  分布
>
> y <- rchisq( 100, df = 2, ncp = 0 ) #  $\chi^2$  分布
```

16.3 ARMA 模型的仿真

假设已给定正态白噪声 v_1, \dots, v_N 。此时 ARMA 模型

$$y_n = \sum_{j=1}^m a_j y_{n-j} + v_n - \sum_{j=1}^{\ell} b_j v_{n-j}, \quad (16.13)$$

的仿真在获得过去观测值 y_1, \dots, y_m 的情况下，以它们为初始值，噪声 v_n 利用生成的随机数即可进行仿真。

另一方面，当未给定初始值时，在 $k = \max(m, \ell)$ 时，可以对 $j = 1, \dots, k$ 使用 $y_j = 0$ 作为初始值。此时，对 $n = k + 1, \dots, N$ ，将 y_{n-1}, \dots, y_{n-m} 和 $v_n, v_{n-1}, \dots, v_{n-\ell}$ 代入 (16.13) 即可生成 y_n 。

当使用 0 作为初始值时，为了消除任意设定的初始值的影响，丢弃最初的 L 个数据，使用剩余的 $n = N - L$ 个是很重要的。为此， L 需要取足够大的值，使得第 6 章定义的脉冲响应函数可以视为已衰减到 0。更简单的方法是使用 $L = 1000$ 这样足够大的值。

例：AR 模型的仿真

以下 R 代码的示例基于给定的初始值执行 AR 模型的仿真。

```
> n <- 200
> m <- 2
> ii <- m+1:n
> mm <- 1:m
> a <- c( 1.5, -0.8 )
> par( mfrow=c(2,1), mar=c(2,2,1,1) )
>
> # 生成正态随机数
> v <- rnorm( n )
> plot( v, type="l" )
>
> # 生成 AR 过程
> y <- rep(0, n)
> y[1] <- 0
> y[2] <- 0.5
>
> for( i in ii ){
>   y[i] <- v[i]
>   for( j in mm ) {
>     y[i] <- y[i] + a[j]*y[i-j]
>   }
> }
> plot( v, type="l" )
```

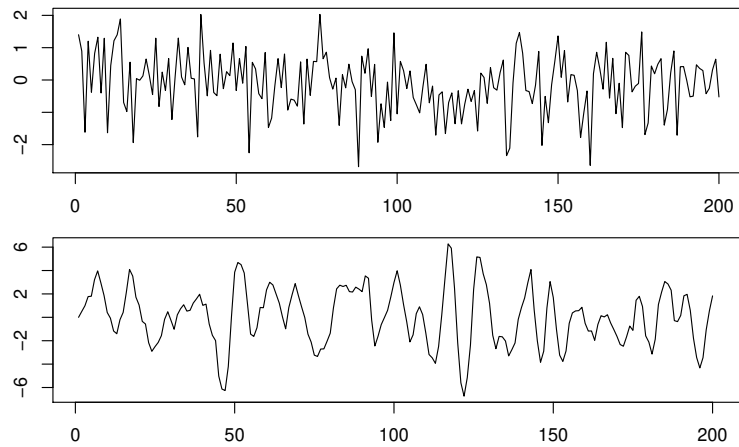


图 16.2 正态白噪声（上图）与 AR 模型的仿真（下图）。

图 16.3 展示了 2 阶 AR 模型的仿真结果。初始状态为 $y_1 = 0$ 、 $y_2 = 0.5$ ，使用正态白噪声 v_n 生成了包含 2 个初始值在内的 200 个观测值。上图展示了生成的正态白噪声 v_n 。而下图展示了使用该白噪声生成的 AR 过程。

16.4 利用状态空间模型的仿真

利用前节方法生成的白噪声实现值，可以生成服从给定模型的时间序列。这称为时间序列的**仿真** (simulation)。假设时间序列模型以状态空间模型

$$x_n = Fx_{n-1} + Gv_n \quad (16.14)$$

$$y_n = Hx_n + w_n \quad (16.15)$$

的形式表示。这里，给定状态向量的初始值 x_0 和均值 0、方差协方差矩阵 Q 的 k 维白噪声实现值 v_1, \dots, v_N ，通过反复应用 (16.14)，可以简单地求得状态向量的实现值 x_1, \dots, x_N 。进一步，给定均值 0、方差协方差矩阵 R 的 ℓ 维白噪声实现值 w_1, \dots, w_N ，通过 (16.15) 即可生成时间序列 y_1, \dots, y_N 。

服从方差协方差矩阵 Q 的 k 维正态白噪声，可以用前节方法生成的一维白噪声来生成。将 k 个一维正态白噪声实现值排列为 $u_n = (u_n^{(1)}, \dots, u_n^{(k)})^T$ ，则 u_n 显然为均值向量 0、方差协方差矩阵为单位矩阵 I 的 k 维正态白噪声。这里，通过正定对称矩阵 Q 的 **Cholesky 分解** (Cholesky decomposition) 求得满足 $Q = LL^T$ 的 $k \times k$ 下三角矩阵 L 。此时，

$$v_n = Lu_n \quad (16.16)$$

由此定义 v_n ，则

$$E[v_n v_n^T] = E[Lu_n u_n^T L^T] = LE[u_n u_n^T]L^T = LI_k L^T = Q \quad (16.17)$$

因此, v_n 为服从均值向量 0、方差协方差矩阵 Q 的 k 维正态白噪声。

利用 R 进行高斯型状态空间模型的仿真

TSSS 程序包的函数 `simssm` 用于进行线性状态空间模型的仿真。需要如下设定状态空间模型的参数:

<code>n</code> :	仿真生成的数据数
<code>trend.order</code> :	趋势分量模型的阶数 (0 或 1)
<code>trend</code> :	<code>trend.order</code> >0 时, 趋势分量的初始值:
<code>seasonal.order</code> :	季节分量模型的阶数 (0,1,2)
<code>seasonal</code> :	<code>seasonal.order</code> >0 时, 季节分量长度 $p-1$ 的初始值:
<code>p</code> :	<code>seasonal.order</code> >0 时, 一个季节的长度 (通常为 12)
<code>ar.order</code> :	AR 分量模型的阶数 (0 或正整数)
<code>ar</code> :	<code>ar.order</code> >0 时, 长度 <code>ar.order</code> 的 AR 分量初始估计值:
<code>sigma2</code> :	观测噪声的方差 σ^2
<code>tau1</code> :	<code>trend.order</code> >0 时, 趋势分量的方差 τ_1^2
<code>tau2</code> :	<code>seasonal.order</code> >0 时, 季节分量模型的方差 τ_2^2
<code>tau3</code> :	<code>ar.order</code> >0 时, AR 分量模型的方差 τ_3^2
<code>plot</code> :	逻辑变量。为 TRUE (默认值) 时, 显示仿真生成的数据

函数 `simssm` 的输出对象 "simulate" 给出高斯型状态空间模型的仿真数据。

例: 随机游走模型和 AR 模型的仿真

函数 `simssm` 可以按如下方式进行带观测噪声的随机游走模型和 AR 模型的仿真:

```
> # 利用状态空间模型进行仿真
> # 带观测噪声的随机游走模型
> simssm(n = 200, seasonal.order = 0, arcoef=0.5, ar=0, tau3=pi*pi/6, sigma2 = 1,
seed = 333)
>
> # 2 阶 AR 模型
> simssm(n = 200, seasonal.order = 0, arcoef=c(1.5,-0.8), ar=c(0,0.1), tau3=1,
sigma2 = 0, seed = 567)
```

图 16.3 展示了随机游走模型

$$\begin{aligned}x_n &= x_{n-1} + v_n \\y_n &= x_n + w_n\end{aligned}\quad (16.18)$$

的仿真结果。其中, 系统噪声 v_n 为 $N(0, \frac{\pi^2}{6})$, 观测噪声 w_n 为 $N(0, 1)$ 的正态白噪声, 初始值为 $x_0 = 0$ 。

例：季节调整模型的仿真

函数 `simssm` 可以按如下方式进行季节调整模型的仿真：

```

> # BLSALLFOOD 数据的仿真
> data(BLSALLFOOD)
> # 季节调整模型的估计> m1 <- 2; m2 <- 1; m3 <- 2
> z <- season(BLSALLFOOD, trend.order = m1, seasonal.order = m2, ar.order = m3)
>
> # 状态空间模型参数的设定
> n1 <- length(BLSALLFOOD)
> trend <- z$trend[m1:1]
> arcoef <- z$arcoef
> period <- 12
> seasonal <- z$seasonal[(period-1):1]
> ar <- z$ar[m3:1]
> tau1 <- z$tau2[1]
> tau2 <- z$tau2[2]
> tau3 <- z$tau2[3]
>
> # 利用状态空间模型进行仿真
> simssm(n = n1, trend, seasonal.order = m2, seasonal, arcoef, ar, tau1, tau2,
tau3, sigma2 = z$sigma2, seed = 333)

```

图 16.4 是利用图 12.6 所示的包含平稳 AR 分量的季节调整模型 ($m_1 = 2$ 、 $m_2 = 1$ 、 $m_3 = 2$ 、 $p = 12$) 的季节调整结果进行仿真的结果。其中，初始值 x_0 使用了从数据得到的估计值 $x_{0|N}$ 。图 1.1 (d) 的时间序列与季节变动很相似，但趋势的变动完全不同。

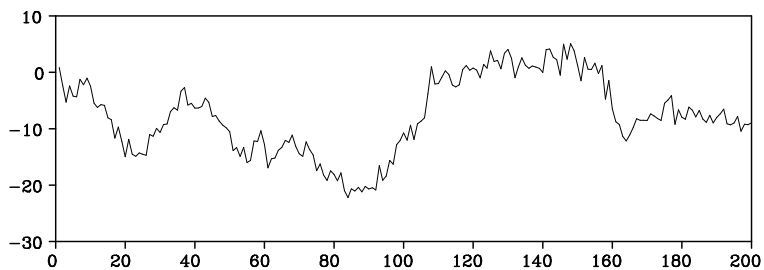


图 16.3 随机游走模型的仿真

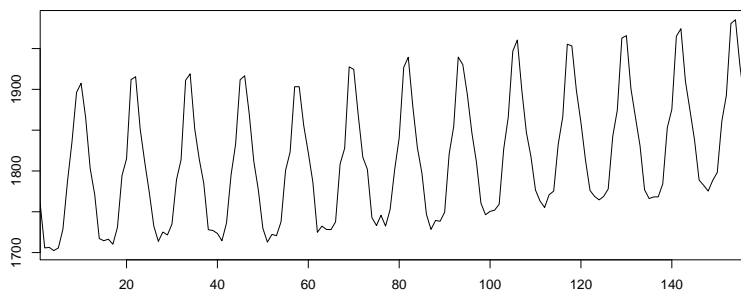


图 16.4 季节调整模型的仿真

16.5 利用非高斯型状态空间模型的仿真

前节的方法可以简单地扩展到非高斯型时间序列模型的仿真。时间序列的非高斯型状态空间模型

$$x_n = Fx_{n-1} + Gv_n \quad (16.19)$$

$$y_n = Hx_n + w_n \quad (16.20)$$

中, v_n 和 w_n 分别为服从由密度函数 $q(v)$ 和 $r(w)$ 定义的一般分布的白噪声。为简单起见, 以下假设 v_n 的 k 个分量和 w_n 的 ℓ 个分量互相独立。为了进行上述非高斯型状态空间模型的仿真, 需要生成服从给定非高斯型分布 $p(x)$ 的随机数。

程序包 TSSS 的函数 `ngsim` 用于进行非高斯型状态空间模型的仿真。需要以下输入:

<code>n</code> :	仿真生成的数据数
<code>trend.order</code> :	趋势分量模型的阶数 (0 或 1)
<code>trend</code> :	<code>trend.order > 0</code> 时, 趋势分量模型的初始值
<code>seasonal.order</code> :	季节分量模型的阶数 (0,1,2)
<code>seasonal</code> :	<code>seasonal.order > 0</code> 时, 季节分量模型长度 $p-1$ 的初始值
<code>p</code> :	<code>seasonal.order > 0</code> 时, 季节分量一个周期的长度 (通常为 12)
<code>ar.order</code> :	AR 分量模型的阶数 (0 或正整数)
<code>ar</code> :	<code>ar.order > 0</code> 时, AR 分量模型长度 <code>ar.order</code> 的初始值
<code>noisew</code> :	观测噪声的分布类型 -1: 柯西分布 -2: 指数分布 -3: 双指数分布 0: 双指数分布 + 欧拉常数 1: 正态分布 2: Pearson 分布族 3: 双指数分布
<code>wminmax</code> :	观测噪声的上限值和下限值
<code>paramw</code> :	观测噪声分布的参数 <code>noisew = 1</code> 时, 方差 σ^2 <code>noisew = 2</code> 时, 散布参数 (σ^2) 和形状参数 (bw)
<code>noisev</code> :	系统噪声的分布类型 -1: 柯西分布 -2: 指数分布 -3: 双指数分布 0: 双指数分布 + 欧拉常数 1: 正态分布 2: Pearson 分布族 3: 双指数分布
<code>vminmax</code> :	系统噪声的上限值和下限值
<code>paramv</code> :	系统噪声分布的参数 <code>noisev = 1</code> 时, 方差 τ^2 <code>noisev = 2</code> 时, 散布参数 (τ^2) 和形状参数 (bv)
<code>plot</code> :	逻辑变量。TRUE (默认值) 时, 绘制仿真数据

函数 `ngsim` 的输出对象 "simulate" 给出非高斯型状态空间模型的仿真数据。

例: 具有双指数分布噪声的趋势模型

```

> # 以双指数分布为系统噪声的趋势模型仿真
> ar3 <- ngsim(n = 400, arcoef = 1.00, noisew = 1, paramw = 1, noisev = 3, paramv
= 1, seed = 555)
> plot(ar3, use = c(201, 400))

```

图 16.5 是在图 16.3 的随机游走模型中, 将系统噪声 v_n 改为双指数随机数 $+\zeta$ (ζ 为欧拉常数, 0.57722) 进行仿真的结果。为了比较, 图 16.5 的 (a) 和 (b) 展示了图 16.2 中使用的正态分布和双指数分布的密度函数。两个密度函数的均值和方差相同, 但仿真结果有很大差异。在与 (a) 对应的图 16.2 的情况下, 趋势周围的变动上下对称, 而在与 (b) 对应的 (c) 的情况下, 变动方式因上下方向的不同而有很大差异, 呈现出在股价变动等中常见的特征。

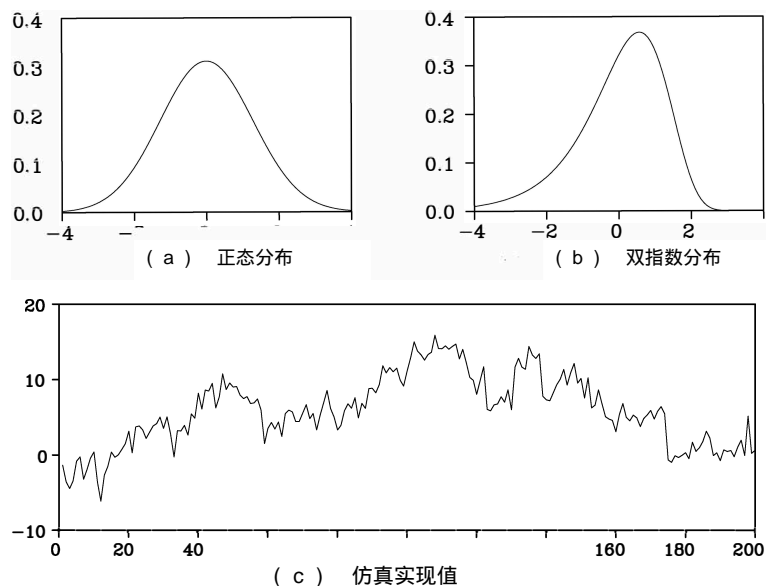


图 16.5 系统噪声的密度函数和仿真结果

例: AR 模型: 高斯型噪声和非高斯型噪声

图 16.6 是对 1 阶 AR 模型

$$y_n = 0.9y_{n-1} + v_n \quad (16.21)$$

进行仿真的结果。其中, (a) 是 v_n 使用正态分布 $N(0,1)$ 的情况, (b) 是使用 t 分布的情况。即使阶数和系数相同的 AR 模型, 当噪声性质不同时, 生成的时间序列看起来有很大差异。特别是在 (b) 中偶尔出现大的跳跃, 整体的变动幅度超过 2 倍。

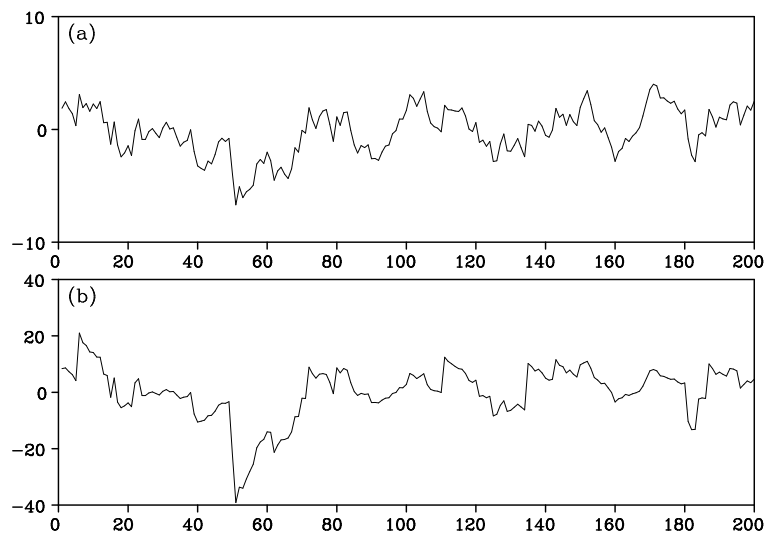


图 16.6 使用不同噪声分布的 AR 模型仿真。(a) 正态分布, (b) t 分布

```
> # AR 模型的仿真
> # 正态噪声
> ar1 <- ngsim(n = 400, arcoef = 0.90, noisew = 1, paramw = 1, noisev = 1, paramv
= 1, seed = 555)
> plot(ar1, use = c(201, 400))
>
> # 非高斯型噪声 ( $t$  分布)
> ar1 <- ngsim(n = 400, arcoef = 0.95, noisew = 1, paramw = 1, noisev = 2, paramv
= c(1,1), seed = 555)
> plot(ar1, use = c(201, 400))
```

● 章末习题

1. 生成 $[0,1]$ 上的均匀随机数 U 和 V 共 2 个, 令 $W = U + V$ 时, W 是什么样的随机数?
2. 在 AR 模型 $y_n = ay_{n-1} + v_n$, $v_n \sim N(0,1)$ 的仿真中, 通常生成 $m+n$ 个, 仅使用后半部分的 n 个。 m 应设定为多大的值?
3. 给出基于当前已有的观测值, 对未来值进行仿真的方法。

附录 A

非线性最优化的算法

为简单起见, 考虑目标函数 $f(x)$ 可以通过二阶 Taylor 展开表示为

$$f(x) = f(x^*) + (x - x^*)^T g(x^*) + \frac{1}{2}(x - x^*)^T H(x - x^*) \quad (\text{A.1})$$

的情况。其中, $x = (x_1, \dots, x_m)^T$,

$$g(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_m} \end{bmatrix}, \quad H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_m \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_m \partial x_m} \end{bmatrix} \quad (\text{A.2})$$

由 (A.1) 可得

$$g(x) = g(x^*) + H(x - x^*) \quad (\text{A.3})$$

设 x^* 为使目标函数 $f(x)$ 最小的点, 则 $g(x^*) = 0$ 成立, 因此

$$g(x) = H(x - x^*) \quad (\text{A.4})$$

因此, 设当前近似值为 x , 若 H 和 $g(x)$ 已知, 则

$$x^* = x - H^{-1}g(x) \quad (\text{A.5})$$

即可求得解 x^* 。

在实际问题中, $f(x)$ 能像 (A.1) 那样以二次形式表示的情况较少, 仅计算一次 (A.5) 往往无法得到解 x^* 。此外, 矩阵 H 通常不是常数而是随 x 变化, 因此从适当的初始值 x_0 出发, 通过

$$x_k = x_{k-1} - H_{k-1}^{-1}g(x_{k-1}) \quad (\text{A.6})$$

反复迭代。这就是**牛顿法** (Newton-Raphson method)。

但是, 时间序列模型的似然函数是参数的复杂函数, 解析地求 H_k 往往很困难, 因此使用称为**拟牛顿法**的如下算法。

1. 令 $k = 0$, 适当确定初始值 x_0 和 H_0^{-1} 。通常取 $H_0 = I$ 。

2. 计算 $g(x_{k-1})$ 。
3. 通过 $h_k = H_{k-1}^{-1}g(x_{k-1})$ 求方向向量。
4. 令 $x_k = x_{k-1} - \lambda_k h_k$, 通过线性搜索确定使 $f(x_k)$ 最小的 λ_k 。
5. 通过下列公式之一求 H_k^{-1} 的估计值。(DFP 公式, BFGS 公式)

$$H_k^{-1} = H_{k-1}^{-1} + \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta g_k} - \frac{H_{k-1}^{-1} \Delta g_k \Delta g_k^T H_{k-1}^{-1}}{\Delta g_k^T H_{k-1}^{-1} \Delta g_k} \quad (\text{A.7})$$

$$H_k^{-1} = H_{k-1}^{-1} - \frac{\Delta x_k \Delta g_k^T H_{k-1}^{-1}}{\Delta x_k^T \Delta g_k} - \frac{H_{k-1}^{-1} \Delta g_k \Delta x_k^T}{\Delta x_k^T \Delta g_k} + \left\{ 1 + \frac{\Delta g_k^T H_{k-1}^{-1} \Delta g_k}{\Delta x_k^T \Delta g_k} \right\} \frac{\Delta x_k \Delta x_k^T}{\Delta x_k^T \Delta g_k} \quad (\text{A.8})$$

其中, $\Delta x_k = x_k - x_{k-1}$, $\Delta g_k = g(x_k) - g(x_{k-1})$ 。

6. 若未满足收敛条件, 则令 $k = k + 1$ 返回第 (2) 步。

在这种拟牛顿法中, 无需计算 H_k 或 H_k^{-1} , 而且通过线性搜索自动调节步长 λ_k , 因此即使实际的目标函数在当前点 x_k 附近与二次曲面有较大偏离时, 也能在一定程度上应对。

附录 B

Levinson 算法的推导

假设已给定自协方差函数 C_k , ($k = 0, 1, \dots$), 且 $m-1$ 阶 AR 模型的 Yule-Walker 解已经求得。为了明确表示这是 $m-1$ 阶模型的系数, 将其记为 $\hat{a}_1^{m-1}, \dots, \hat{a}_{m-1}^{m-1}$, 预测误差记为 v_n^{m-1} , 预测误差方差记为 $\hat{\sigma}_{m-1}^2$ 。接下来考虑利用这些估计值, 高效计算 m 阶 AR 模型参数的 Yule-Walker 估计值 $\hat{a}_1^m, \dots, \hat{a}_m^m$ 及 $\hat{\sigma}_m^2$ 的方法。

系数 $\hat{a}_1^{m-1}, \dots, \hat{a}_{m-1}^{m-1}$ 满足 $m-1$ 阶的 Yule-Walker 方程, 因此

$$C_k = \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_{j-k} \quad (k = 1, \dots, m-1) \quad (\text{B.1})$$

成立。这意味着关于 v_n^{m-1} ,

$$E\{v_n^{m-1} y_{n-k}\} = E\left\{\left(y_n - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-j}\right) y_{n-k}\right\} = 0 \quad (\text{B.2})$$

在 $k = 1, \dots, m-1$ 的范围内成立。

这里, 将时间方向反转, 考虑用未来数据表示当前值的 $m-1$ 阶后向 AR 模型 (backward AR model)

$$y_n = \sum_{i=1}^{m-1} a_i y_{n+i} + w_n^{m-1} \quad (\text{B.3})$$

由于 $C_{-k} = C_k$ 成立, 该后向 AR 模型也满足相同的 Yule-Walker 方程 (B.1)。因此, 关于后向预测误差 w_n^{m-1} 也有

$$E\{w_{n-m}^{m-1} y_{n-k}\} = E\left\{\left(y_{n-m} - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-m+j}\right) y_{n-k}\right\} = 0 \quad (\text{B.4})$$

对 $k = 1, \dots, m-1$ 成立。

这里, 定义前向预测误差 v_n^{m-1} 和后向预测误差 w_{n-m}^{m-1} 的加权和为

$$\begin{aligned} z_n &\equiv v_n^{m-1} - \beta w_{n-m}^{m-1} \\ &\equiv y_n - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-j} - \beta \left(y_{n-m} - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-m+j} \right) \\ &= y_n - \sum_{j=1}^{m-1} \left(\hat{a}_j^{m-1} - \beta \hat{a}_{m-j}^{m-1} \right) y_{n-j} - \beta y_{n-m} \end{aligned} \quad (\text{B.5})$$

z_n 是 $y_n, y_{n-1}, \dots, y_{n-m}$ 的线性组合, 而且由 (B.2)、(B.4) 可知, 对 $k=1, \dots, m-1$,

$$E\{z_n y_{n-k}\} = 0 \quad (\text{B.6})$$

始终成立。因此, 若确定 β 的值使得对 $k=m$ 也有 $E(z_n y_{n-m}) = 0$ 成立, 则 m 阶 Yule-Walker 方程就得到满足。为此,

$$\begin{aligned} E(z_n y_{n-m}) &= E\left\{ y_n - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-j} - \beta \left(y_{n-m} - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} y_{n-m+j} \right) \right\} y_{n-m} \\ &= C_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_{m-j} - \beta \left(C_0 - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_j \right) = 0 \end{aligned} \quad (\text{B.7})$$

由此可知, 令

$$\begin{aligned} \beta &= \left(C_0 - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_j \right)^{-1} \left(C_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_{m-j} \right) \\ &= \left(\hat{\sigma}_{m-1}^2 \right)^{-1} \left(C_m - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_{m-j} \right) \end{aligned} \quad (\text{B.8})$$

即可。因此, 将该 β 记为 \hat{a}_m^m , 令

$$\hat{a}_j^m \equiv \hat{a}_j^{m-1} - \hat{a}_m^m \hat{a}_{m-j}^{m-1} \quad (\text{B.9})$$

则由 (B.5)~(B.7) 可知

$$E\left\{ \left(y_n - \sum_{j=1}^m \hat{a}_j^m y_{n-j} \right) y_{n-k} \right\} = C_k - \sum_{j=1}^m \hat{a}_j^m C_{k-j} = 0 \quad (\text{B.10})$$

对 $k=1, \dots, m$ 成立, 即 $\hat{a}_1^m, \dots, \hat{a}_m^m$ 为 m 阶 Yule-Walker 方程的解。此外

$$\begin{aligned} \hat{\sigma}_m^2 &= C_0 - \sum_{j=1}^m \hat{a}_j^m C_j \\ &= C_0 - \sum_{j=1}^{m-1} \left(\hat{a}_j^{m-1} - \hat{a}_m^m \hat{a}_{m-j}^{m-1} \right) C_j - \hat{a}_m^m C_m \\ &= C_0 - \sum_{j=1}^{m-1} \hat{a}_j^{m-1} C_j - \hat{a}_m^m \left(C_m - \sum_{j=1}^{m-1} \hat{a}_{m-j}^{m-1} C_j \right) \\ &= \hat{\sigma}_{m-1}^2 - \hat{a}_m^m \left(C_m - \sum_{j=1}^{m-1} \hat{a}_{m-j}^{m-1} C_j \right) \end{aligned} \quad (\text{B.11})$$

由 (B.8) 可知 $C_m - \sum_{j=1}^{m-1} \hat{a}_{m-j}^{m-1} C_j = \hat{a}_m^m \hat{\sigma}_{m-1}^2$, 利用此关系可得

$$\hat{\sigma}_m^2 = \hat{\sigma}_{m-1}^2 (1 - (\hat{a}_m^m)^2) \quad (\text{B.12})$$

由此即可计算 $\hat{\sigma}_m^2$ 。

附录 C

卡尔曼滤波和平滑算法的推导

C.1 卡尔曼滤波

[预测] 由 $x_n = F_n x_{n-1} + G_n v_n$ 可得

$$\begin{aligned} x_{n|n-1} &= \mathbf{E}(x_n | Y_{n-1}) \\ &= \mathbf{E}(F_n x_{n-1} + G_n v_n | Y_{n-1}) \\ &= F_n \mathbf{E}(x_{n-1} | Y_{n-1}) \\ &= F_n x_{n-1|n-1} \end{aligned} \quad (\text{C.1})$$

$$\begin{aligned} V_{n|n-1} &= \mathbf{E}(x_n - x_{n|n-1})^2 \\ &= \mathbf{E}(F_n(x_{n-1} - x_{n-1|n-1}) + G_n v_n)^2 \\ &= F_n \mathbf{E}(x_{n-1} - x_{n-1|n-1})^2 F_n^T + G_n \mathbf{E}(v_n^2) G_n^T \\ &= F_n V_{n-1|n-1} F_n^T + G_n Q_n G_n^T \end{aligned} \quad (\text{C.2})$$

其中, 为简单起见将 xx^T 简记为 x^2 。

[滤波] 将 y_n 的预测误差记为 ε_n , 由 $y_n = H_n x_n + w_n$

可得

$$\begin{aligned} \varepsilon_n &\equiv y_n - \mathbf{E}(y_n | Y_{n-1}) \\ &= H_n x_n + w_n - \mathbf{E}(H_n x_n + w_n | Y_{n-1}) \\ &= H_n x_n + w_n - H_n \mathbf{E}(x_n | Y_{n-1}) \\ &= H_n(x_n - x_{n|n-1}) + w_n \end{aligned} \quad (\text{C.3})$$

因此

$$\text{Var}(\varepsilon_n) = H_n V_{n|n-1} H_n^T + R_n \quad (\text{C.4})$$

$$\begin{aligned} \text{Cov}(x_n, \varepsilon_n) &= \text{Cov}(x_n, H_n(x_n - x_{n|n-1}) + w_n) \\ &= \text{Var}(x_n - x_{n|n-1}) H_n^T \\ &= V_{n|n-1} H_n^T \end{aligned} \quad (\text{C.5})$$

此时, 利用 $Y_n = \{Y_{n-1}, y_n\} = Y_{n-1} \oplus \varepsilon_n$ 以及在正态分布情况下 x_n 关于 Y_n 的条件期望

可以用正交投影表示这一性质, 可得

$$\begin{aligned} x_{n|n} &= \mathbf{E}(x_n|Y_n) = \text{Proj}(x_n|Y_n) \\ &= \text{Proj}(x_n|Y_{n-1}, \boldsymbol{\varepsilon}_n) \\ &= \text{Proj}(x_n|Y_{n-1}) + \text{Proj}(x_n|\boldsymbol{\varepsilon}_n) \end{aligned} \quad (\text{C.6})$$

$\text{Proj}(x_n|\boldsymbol{\varepsilon}_n)$ 可通过将 x_n 对 $\boldsymbol{\varepsilon}_n$ 进行回归得到, 由 (C.4)、(C.5) 可得

$$\begin{aligned} \text{Proj}(x_n|\boldsymbol{\varepsilon}_n) &= \text{Cov}(x_n, \boldsymbol{\varepsilon}_n) \text{Var}(\boldsymbol{\varepsilon}_n)^{-1} \boldsymbol{\varepsilon}_n \\ &= V_{n|n-1} H_n^T (H_n V_{n|n-1} H_n^T + R_n)^{-1} \boldsymbol{\varepsilon}_n \\ &\equiv K_n \boldsymbol{\varepsilon}_n \end{aligned} \quad (\text{C.7})$$

因此

$$x_{n|n} = x_{n|n-1} + K_n \boldsymbol{\varepsilon}_n \quad (\text{C.8})$$

进一步

$$\begin{aligned} V_{n|n-1} &= \mathbf{E}(x_n - x_{n|n-1})^2 \\ &= \mathbf{E}(x_n - x_{n|n} + K_n \boldsymbol{\varepsilon}_n)^2 \\ &= V_{n|n} + K_n \text{Var}(\boldsymbol{\varepsilon}_n) K_n^T \end{aligned} \quad (\text{C.9})$$

由此可得

$$\begin{aligned} V_{n|n} &= V_{n|n-1} - K_n H_n V_{n|n-1} \\ &= (I - K_n H_n) V_{n|n-1} \end{aligned} \quad (\text{C.10})$$

C.2 平滑

设 $\boldsymbol{\delta}_{n+1} \equiv x_{n+1} - x_{n+1|n}$ 为 x_{n+1} 的预测误差, 令

$$Z_n \equiv Y_n \oplus \boldsymbol{\delta}_{n+1} \oplus \{v_{n+1}, \dots, v_N, w_{n+1}, \dots, w_N\}$$

此时

$$\begin{aligned} z_n &\equiv \text{Proj}(x_n|Z_n) \\ &= \text{Proj}(x_n|Y_n) + \text{Proj}(x_n|\boldsymbol{\delta}_{n+1}) \\ &\quad + \text{Proj}(x_n|v_{n+1}, \dots, v_N, w_{n+1}, \dots, w_N) \end{aligned} \quad (\text{C.11})$$

可以分解为上式, 且

$$\begin{aligned} \text{Proj}(x_n|Y_n) &= x_{n|n} \\ \text{Proj}(x_n|\boldsymbol{\delta}_{n+1}) &= \text{Cov}(x_n, \boldsymbol{\delta}_{n+1}) \text{Var}(\boldsymbol{\delta}_{n+1})^{-1} \boldsymbol{\delta}_{n+1} \\ \text{Proj}(x_n|v_{n+1}, \dots, v_N, w_{n+1}, \dots, w_N) &= 0 \end{aligned} \quad (\text{C.12})$$

成立。进一步

$$\text{Var}(\delta_{n+1}) = V_{n+1|n} \quad (\text{C.13})$$

$$\begin{aligned} \text{Cov}(x_n, \delta_{n+1}) &= \text{Cov}(x_n, F_{n+1}(x_n - x_{n|n}) + G_{n+1}v_{n+1}) \\ &= \text{E}(x_n - x_{n|n})^2 F_{n+1}^T \\ &= V_{n|n} F_{n+1}^T \end{aligned} \quad (\text{C.14})$$

因此, 令 $A_n = V_{n|n} F_{n+1}^T V_{n+1|n}^{-1}$, 则

$$z_n = x_{n|n} + A_n(x_{n+1} - x_{n+1|n}) \quad (\text{C.15})$$

这里, 考虑到 Z_N 生成 Y_N , 可得

$$\begin{aligned} x_{n|N} &= \text{Proj}(x_n | Y_N) \\ &= \text{Proj}(\text{Proj}(x_n | Z_N) | Y_N) \\ &= \text{Proj}(z_n | Y_N) \\ &= x_{n|n} + A_n(x_{n+1|N} - x_{n+1|n}) \end{aligned} \quad (\text{C.16})$$

进一步, 由

$$x_n - x_{n|N} + A_n x_{n+1|N} = x_n - x_{n|n} + A_n x_{n+1|n} \quad (\text{C.17})$$

利用 $\text{E}\{(x_n - x_{n|N})x_{n+1|N}^T\} = \text{E}\{(x_n - x_{n|n})x_{n+1|n}^T\} = 0$, 可得

$$V_{n|N} + A_n \text{E}\{x_{n+1|N}x_{n+1|N}^T\}A_n^T = V_{n|n} + A_n \text{E}\{x_{n+1|n}x_{n+1|n}^T\}A_n^T \quad (\text{C.18})$$

这里, 进一步利用

$$\begin{aligned} \text{E}\{(x_{n+1} - x_{n+1|N})x_{n+1|N}^T\} &= 0 \\ \text{E}\{(x_n - x_{n|n})x_{n+1|n}^T\} &= 0 \end{aligned}$$

可得

$$\begin{aligned} &\text{E}\{x_{n+1|N}x_{n+1|N}^T\} \\ &= \text{E}\{(x_{n+1|N} - x_{n+1} + x_{n+1})(x_{n+1|N} - x_{n+1} + x_{n+1})^T\} \\ &= V_{n+1|N} + \text{E}\{x_{n+1}x_{n+1}^T\} + 2\text{E}\{(x_{n+1|N} - x_{n+1})x_{n+1}^T\} \\ &= V_{n+1|N} + \text{E}\{x_{n+1}x_{n+1}^T\} - 2\text{E}\{(x_{n+1|N} - x_{n+1})(x_{n+1|N} - x_{n+1})^T\} \\ &= \text{E}\{x_{n+1}x_{n+1}^T\} - V_{n+1|N} \end{aligned} \quad (\text{C.19})$$

$$\text{E}\{x_{n+1|n}x_{n+1|n}^T\} = \text{E}\{x_{n+1}x_{n+1}^T\} - V_{n+1|n} \quad (\text{C.20})$$

将此代入 (C.18) 可得

$$V_{n|N} = V_{n|n} + A_n(V_{n+1|N} - V_{n+1|n})A_n^T \quad (\text{C.21})$$

付録 D

粒子滤波的算法

(a) 一步预测

假设已给定可视为状态 x_{n-1} 的条件分布 $p(x_{n-1}|Y_{n-1})$ 实现值的 m 个粒子 $\{f_{n-1}^{(1)}, \dots, f_{n-1}^{(m)}\}$, 以及可视为系统噪声 v_n 独立实现值的粒子 $\{v_n^{(1)}, \dots, v_n^{(m)}\}$ 。即假设对 $i = 1, \dots, m$,

$$f_{n-1}^{(i)} \sim p(x_{n-1}|Y_{n-1}), \quad v_n^{(i)} \sim q(v) \quad (\text{D.1})$$

成立。此时, x_n 的一步预测分布为

$$\begin{aligned} p(x_n|Y_{n-1}) &= \int \int p(x_n, x_{n-1}, v_n|Y_{n-1}) dx_{n-1} dv_n \\ &= \int \int p(x_n|x_{n-1}, v_n, Y_{n-1}) p(v_n|x_{n-1}, Y_{n-1}) p(x_{n-1}|Y_{n-1}) dx_{n-1} dv_n \end{aligned} \quad (\text{D.2})$$

由于系统噪声与之前的状态和观测值独立, 有 $p(v_n|x_{n-1}, Y_{n-1}) = p(v_n)$, 另外, 由于 x_n 仅依赖于 x_{n-1} 和 v_n , 有 $p(x_n|x_{n-1}, v_n, Y_{n-1}) = p(x_n|x_{n-1}, v_n) = \delta(x_n - F(x_{n-1}, v_n))$, 因此

$$p(x_n|Y_{n-1}) = \int \int \delta(x_n - F(x_{n-1}, v_n)) p(v_n) p(x_{n-1}|Y_{n-1}) dx_{n-1} dv_n \quad (\text{D.3})$$

因此, 当 $p(v_n)$ 和 $p(x_{n-1}|Y_{n-1})$ 的实现值 $\{v_n^{(j)}\}$ 和 $\{f_{n-1}^{(j)}\}$ 已知时, $p(x_n|Y_{n-1})$ 的实现值 $\{p_n^{(j)}\}$ 可由

$$p_n^{(j)} = F(f_{n-1}^{(j)}, v_n^{(j)}) \quad (\text{D.4})$$

求得。

(b) 滤波

当分布 $p(x_n|Y_{n-1})$ 的 m 个独立实现值 $p_n^{(1)}, \dots, p_n^{(m)}$ 已知时, 相当于分布 $p(x_n|Y_{n-1})$ 由经验分布函数

$$P_n(x) = \frac{1}{m} \sum_{i=1}^m I(x, p_n^{(i)}), \quad (\text{D.5})$$

近似。其中, $I(x, a)$ 是当 $x < a$ 时 $I(x, a) = 0$, 其他情况 $I(x, a) = 1$ 的函数。这意味着预测分布 $p(x_n|Y_{n-1})$ 由概率函数

$$\Pr(x_n = p_n^{(j)}|Y_{n-1}) = \frac{1}{m}, \quad j = 1, \dots, m \quad (\text{D.6})$$

近似。此时, 给定观测值 y_n 后, 后验分布可由

$$\begin{aligned} \Pr(x_n = p_n^{(j)}|Y_n) &= \Pr(x_n = p_n^{(j)}|Y_{n-1}, y_n) \\ &= \lim_{\Delta y \rightarrow 0} \frac{\Pr(x_n = p_n^{(j)}, y_n \leq y \leq y_n + \Delta y|Y_{n-1})}{\Pr(y_n \leq y \leq y_n + \Delta y|Y_{n-1})} \\ &= \frac{p(y_n|p_n^{(j)})\Pr(x_n = p_n^{(j)}|Y_{n-1})}{\sum_{i=1}^m p(y_n|p_n^{(i)})\Pr(x_n = p_n^{(i)}|Y_{n-1})} \\ &= \frac{\alpha_n^{(j)} \cdot \frac{1}{m}}{\sum_{i=1}^m \alpha_n^{(i)} \cdot \frac{1}{m}} = \frac{\alpha_n^{(j)}}{\sum_{i=1}^m \alpha_n^{(i)}}. \end{aligned} \quad (\text{D.7})$$

求得。与 $\Pr(x_n = p_n^{(j)}|Y_n)$ 对应的累积分布函数

$$\frac{1}{\sum_{i=1}^m \alpha_n^{(i)}} \sum_{i=1}^m \alpha_n^{(i)} I(x, p_n^{(i)}) \quad (\text{D.8})$$

仅在 m 个点 $p_n^{(1)}, \dots, p_n^{(m)}$ 处有与 $\alpha_n^{(1)}, \dots, \alpha_n^{(m)}$ 成比例的量 (即 (15.8) 式右边) 的跳跃。

由 (D.8) 式, 已经得到了滤波分布的近似, 但为了在下一时点进行预测步骤的计算 (D.1), 需要用 m 个粒子 $f_n^{(1)}, \dots, f_n^{(m)}$ 重新表示。这相当于将 (D.8) 的分布用经验分布函数

$$\frac{1}{m} \sum_{i=1}^m I(x, f_n^{(i)}) \quad (\text{D.9})$$

表示, 通过以概率

$$\Pr(f_n^{(j)} = p_n^{(i)}|Y_n) = \frac{\alpha_n^{(i)}}{\alpha_n^{(1)} + \dots + \alpha_n^{(m)}}, \quad j = 1, \dots, m \quad (\text{D.10})$$

对 m 个实现值 $\{p_n^{(1)}, \dots, p_n^{(m)}\}$ 进行重采样, 即可求得 $\{f_n^{(1)}, \dots, f_n^{(m)}\}$ 。

假设 $\Pr(x_1 = s_{1|n-1}^{(j)}, \dots, x_{n-1} = s_{n-1|n-1}^{(j)}|Y_{n-1}) = 1/m$ 以及 $v_n^{(j)} \sim q(v)$ 成立, 定义 $(p_{1|n-1}^{(j)}, \dots, p_{n|n-1}^{(j)})$ 如下。

$$p_{i|n-1}^{(j)} = \begin{cases} s_{i|n-1}^{(j)}, & i = 1, \dots, n-1 \\ F(s_{n-1|n-1}^{(j)}, v_n^{(j)}), & i = n \end{cases} \quad (\text{D.11})$$

此时, $(p_{1|n-1}^{(j)}, \dots, p_{n|n-1}^{(j)})$ 可视为给定观测值 Y_{n-1} 时 (x_1, \dots, x_n) 的联合分布的实现值。

接下来, 当观测值 y_n 给定时, 分布 $\Pr(x_1 \leq p_{1|n-1}^{(j)}, \dots, x_n \leq p_{n|n-1}^{(j)} | Y_{n-1})$ 按以下方式更新。

$$\begin{aligned}
 & \Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_n) \\
 &= \Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1}, y_n) \\
 &= \frac{p(y_n | x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)})}{p(y_n | Y_{n-1})} \\
 &\times \Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1}) \\
 &= \frac{p(y_n | p_{n|n-1}^{(j)}) \Pr(x_1 = p_{1|n-1}^{(j)}, \dots, x_n = p_{n|n-1}^{(j)} | Y_{n-1})}{p(y_n | Y_{n-1})}. \tag{D.12}
 \end{aligned}$$

由于 $p_{n|n-1}^{(j)}$ 与滤波算法 (D.5) 中的 $p_n^{(j)}$ 相同, 平滑分布 $p(x_1, \dots, x_n | Y_n)$ 只需对 m 个 n 维向量 $(p_{1|n-1}^{(j)}, \dots, p_{n|n-1}^{(j)})^T, j = 1, \dots, m$, 以与滤波相同的权重进行重采样即可。

参考文献

- [1] 赤池弘次, 北川源四郎編, 時系列解析の実際 I, II, 朝倉書店, 1994, 1995.
- [2] 赤池弘次, 中川東一郎, ダイナミックシステムの統計的解析と制御, サイエンス社, 1972.
- [3] Akaike, H., “A Bayesian extension of the minimum AIC procedure of autoregressive model fitting”. *Biometrika*, **66**, 237-24, 1979.
- [4] Akaike, H. ほか, TIMSAC-78, Computer Science Monographs, No.11, 1979.
- [5] Bartlett, M.S., On the Theoretical Specification and Sampling Properties of Autocorrelated Time-Series, *Supplement to the Journal of the Royal Statistical Society*, **8**, 1946.
- [6] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.
- [7] Brockwell, P. J., Davis, R. A., and Fienberg, S. E. (1991). *Time Series: Theory and Methods*. Springer Science & Business Media.
- [8] Doucet, A., de Freitas, and Gordon, N., *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001.
- [9] 尾崎統, 北川源四郎編, 時系列解析の方法, 朝倉書店, 1990,
- [10] 片山徹, 応用カルマンフィルタ, 朝倉書店, 1983.
- [11] Kitagawa, G. and Gersch, W., *Smoothness Priors Analysis of Time Series*, Lecture Notes in Statistics, No.106, Springer Verlag, 1996.
- [12] 坂元慶行, 石黒真木夫, 北川源四郎, 情報量統計学, 共立出版, 1983.
- [13] Sakamoto, Y., Ishiguro, M., Kitagawa, G., *Akaike Information Criterion Statistics*, D. Reidel Publishing Company, 1986.
- [14] 日野幹雄, スペクトル解析, 朝倉書店, 1977.
- [15] 広松毅, 浪花貞夫, 経済時系列分析, 朝倉書店, 1990.

-
- [16] 得丸英勝, 添田喬, 中溝高好, 秋月影雄, 計数・測定=ランダムデータ処理の理論と応用, 培風館, 1982.
- [17] Harvey, A.C., Time Series Models, Philip Allan Publishers, 1981.
(邦訳) 国友直人, 山本拓, 時系列モデル入門, 東京大学出版会, 1985.
- [18] 今野浩, 山下浩, 非線形計画法, 日科技連, 1978.
- [19] 伏見正則, 乱数, 東京大学出版会, 1989.
- [20] 小西貞則, 北川源四郎, 情報量規準, 朝倉書店, 2004.
- [21] Anderson, B.D.O. and Moore, J., *Optimal Filtering*, Prentice-Hall, Inc., 1979.
- [22] Kitagawa, G., “Non-Gaussian state-space modeling of nonstationary time series (with discussion),” *J. Amer. Statist. Assoc.*, 82, 1987, 1032–1063.
- [23] Kitagawa, G., “A nonlinear smoothing method for time series analysis”, *Statistica Sinica*, 1, 371–388, 1991.
- [24] Kitagawa, G., “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,” *Journal of Computational and Graphical Statistics*, Vol.5, No.1, 1996 1–25.
- [25] Kitagawa, G. and Gersch, W., “A smoothness priors time varying AR coefficient modeling of nonstationary time series”, *IEEE Trans. on Auto. Control*, **AC-30**, 48–56, 1985.
- [26] Kitagawa, G. and Gersch, W., *Smoothness Priors Analysis of Time Series*, (Vol. **116**, Springer, New York, 1996.
- [27] Knuth, D.E, *The Art of Computer Programming* (日本語版), ASCII Addison Wesley Programming Series, 2004.
- [28] Lewis, T.G. and Payne, W.H, “Generalized feedback shift register pseudorandom number algorithm,” *JACM*, **20** 456–468. 1973.
- [29] Matsumoto, M. and Kurita, Y., “Twisted GFSR generators II,” *ACM Trans. on Modeling and Computer Simulations*, **4**, 254–266, 1994.
- [30] Matsumoto, M. and Nishimura, T., “Mersenne twister, A 623-dimensionally equidistributed uniform pseudorandom number generator,” *ACM Trans. on Modeling and Computer Simulations*, 1998.
- [31] Shumway, R. H. and Stoffer, D. S. *Time Series Analysis and its Applications:*

with R Examples. Springer, 2017.

- [32] West, M. and Harrison, J., *Bayesian Forecasting and Dynamic Models*, Springer, 1996.

章末习题解答

第1章

- 适当选择采样间隔是很重要的。间隔过大则无法捕捉时间序列的特征，而过小则用于建模的参数可能被消耗在非本质的部分，反而无法表达关键特征。主要周期分量的 $1/5 \sim 1/10$ 左右是一个参考标准。
- (解答示例) 记录某商店每天销售的速溶咖啡个数的时间序列。由于是正值时间序列，分布呈上下不对称。此外，由于打折或非价格促销策略，销量有时会有显著增加。
- (1) 将 (1.1) 式关于 y 求解得 $y = e^z / (1 + e^z)$.
(2) $z = \log\{(y-a)/(b-y)\}$, 逆变换为 $y = (a + be^z) / (1 + e^z)$.
- 当原始时间序列包含噪声并可表示为 $y_n = a + bn + \varepsilon_n$ 时，差分后 $\Delta y_n = b + \varepsilon_n - \varepsilon_{n-1}$ ，虽然趋势被去除，但噪声的结构反而变得更复杂。
- 前一年的趋势变化会影响今年的修正值。例如，如果上一年度中途趋势的斜率增加，即使今年度趋势斜率保持不变，修正后看起来斜率也会从中途开始减小。
- (1) 当 $y_n = T_n + w_n$, $T_n = a + bn$, $w_n \sim N(0, \sigma^2)$ 为白噪声时

$$\begin{aligned}\hat{T}_n &= \frac{1}{3}(y_{n-1} + y_n + y_{n+1}) \\ &= \frac{1}{3}(T_{n-1} + T_n + T_{n+1}) + \frac{1}{3}(w_{n-1} + w_n + w_{n+1})\end{aligned}$$

其中，右边第一项等于 T_n 。而右边第2项的均值为0，由 $E(w_{n-1} + w_n + w_{n+1})^2 = 3\sigma^2$ 可知方差为 $\sigma^2/3$ 。

- 移动平均可以通过增大项数获得平滑的估计值，但具有无法检测急剧结构变化以及对异常值敏感的缺点。移动中位数具有相反的性质。

第2章

1. 高斯 (正态) 分布的形状完全由均值和方差决定。因此, 如果弱平稳性保证均值和方差不变, 则分布本身也不变, 强平稳性成立。
2. 将每个时刻服从标准柯西分布的序列视为时间序列时, 其概率分布恒定且满足 (2.4), 因此是强平稳的。但由于柯西分布不存在均值和方差, 不能说弱平稳性成立。
3. $C_k = E(y_n - \mu)(y_{n+k} - \mu) = E(y_{n+k} - \mu)(y_n - \mu) = E(y_n - \mu)(y_{n-k} - \mu) = C_k$. 第三个等式仅在 y_n 平稳时成立。
4. $C_0 = Ey_n^2 = Ev_n^2 - 2cEv_nv_{n-1} + c_2E_{v_{n-1}}^2 = 1 + c^2$, $C_1 = E(v_n - cv_{n-1})(v_{n-1} - cv_{n-2}) = -c$, $E_k = 0 (k \leq 2)$ 。
5. 设 $\alpha = (\alpha_1, \dots, \alpha_k)$ 为不全为 0 的任意向量, 则

$$\begin{aligned} \alpha C \alpha^T &= \sum_{i=1}^k \sum_{j=1}^k \alpha_i \alpha_j C_{i-j} = \sum_{i=1}^k \sum_{j=1}^k \alpha_i \alpha_j E(y_{n-i} y_{n-j}) \\ &= E \left(\sum_{i=1}^k \alpha_i y_{n-i} \right)^2 \geq 0 \end{aligned}$$

6. (1) 对 (2.21) 式两边取期望得

$$E[\hat{C}_k(i, j)] = \frac{1}{N} \sum_{n=k+1}^N C_k(i, j) = \frac{N-k}{N} C_k(i, j)$$

- (2) 由 (2.21) 式定义的样本互协方差是非负定的, 因此在 AR 模型的估计等中能保证平稳性, 这是其优点。将分母替换为 N 可以得到 $C_k(i-j)$ 的无偏估计量, 但会失去这个优点。

7. (1) 当数据数为 n 时 $\hat{C}_0 \sim N(0, 2\sigma^2/n)$, $\hat{C}_k \sim N(0, \sigma^2/n)$, $\hat{R}_k \sim N(0, n^{-1})$
(2) 检验对所有 k 是否成立 $|\hat{R}_k| < \frac{\alpha}{\sqrt{n}}$ 。 α 例如取 2。

第 3 章

1. 由 $e^{-2\pi i k f} = \cos(2\pi k f) - i \sin(2\pi k f)$ 可得 $p(f) = \sum_{k=-\infty}^{\infty} C_k \cos(2\pi k f) - i \sum_{k=-\infty}^{\infty} C_k \sin(2\pi k f)$, 但由于 C_k 是偶函数, $\sin(2\pi k f)$ 是奇函数, $C_k \sin(2\pi k f)$ 为奇函数, 上式右边第二项为 0。
2. $p(f) = 1 + C^2 - 2C \cos(2\pi f)$

3.

$$\begin{aligned}
 p(f) &= \sum_{k=-\infty}^{\infty} C_k e^{-2\pi i k f} = \frac{\sigma^2}{1-a^2} \sum_{k=-\infty}^{\infty} a^{|k|} e^{-2\pi i k f} \\
 &= \frac{\sigma^2}{1-a^2} \left(\frac{1}{1-ae^{-2\pi i f}} + \frac{1}{1-ae^{2\pi i f}} - 1 \right) \\
 &= \frac{\sigma^2}{1-a^2} \frac{1-a^2}{|1-ae^{-2\pi i f}|^2} = \frac{\sigma^2}{|1-ae^{-2\pi i f}|^2}
 \end{aligned}$$

4.

$$\begin{aligned}
 & \frac{1}{N} \left| \sum_{n=1}^N y_n \exp(-2\pi i(n-1)j/N) \right|^2 \\
 &= \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^N y_n y_m \exp(-2\pi i(n-1)j/N) \exp(2\pi i(m-1)j/N) \\
 &= \frac{1}{N} \left\{ \sum_{n=1}^N y_n^2 + 2 \sum_{k=1}^{N-1} \sum_{n=k+1}^N y_n y_{n-k} \exp(-2\pi i k j/N) \right\} \\
 &= \hat{C}_0 + 2 \sum_{k=1}^{N-1} \hat{C}_k \exp(-2\pi i k j/N)
 \end{aligned}$$

5. 对 (3.9) 式两边取期望得

$$\begin{aligned}
 E[\hat{p}(f)] &= \sum_{k=1-N}^{N-1} E[\hat{C}_k] e^{-2\pi i k f} = \sum_{k=1-N}^{N-1} \frac{N-|k|}{N} C_k e^{-2\pi i k f} \\
 &= \sum_{k=1-N}^{N-1} C_k e^{-2\pi i k f} - \frac{1}{N} \sum_{k=1-N}^N |k| C_k e^{-2\pi i k f}
 \end{aligned}$$

第 4 章

1. 对数似然为 $\ell = \sum_{i=1}^n \log f(m_i|\lambda) = \sum_{i=1}^n \{-\lambda + m_i \log \lambda - \log m_i!\}$ 。因此

$$\frac{\partial \ell}{\partial \lambda} = \sum_{i=1}^n \left\{ -1 + \frac{m_i}{\lambda} \right\} = -n + \frac{1}{\lambda} \sum_{i=1}^n m_i = 0$$

可得 $\hat{\lambda} = n^{-1} \sum_{i=1}^n m_i$ 。

2. (1) $\hat{\mu}_0 = \frac{1}{n+m} \left(\sum_{i=1}^n x_i + \sum_{i=1}^m y_i \right)$, $\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^n x_i$, $\hat{\mu}_2 = \frac{1}{m} \sum_{i=1}^m y_i$, $\hat{\sigma}_0^2 =$
 $\frac{1}{n+m} \left(\sum_{i=1}^n (x_i - \hat{\mu}_0)^2 + \sum_{i=1}^m (y_i - \hat{\mu}_0)^2 \right)$,
 $\hat{\tau}_0^2 = \frac{1}{n+m} \left(\sum_{i=1}^n (x_i - \hat{\mu}_1)^2 + \sum_{i=1}^m (y_i - \hat{\mu}_2)^2 \right)$ 此时

$$\text{AIC}_0 = (n+m)(\log 2\pi \hat{\sigma}_0^2 + 1) + 2 \times 2$$

$$\text{AIC}_1 = (n+m)(\log 2\pi \hat{\tau}_0^2 + 1) + 2 \times 3$$

进行比较, 若 $\text{AIC}_0 < \text{AIC}_1$, 则视为均值相等更好。

(2) $\hat{\sigma}_1^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_0)^2$, $\hat{\sigma}_2^2 = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{\mu}_0)^2$, $AIC_2 = n(\log 2\pi\hat{\sigma}_1^2 + 1) + m(\log 2\pi\hat{\sigma}_2^2 + 1) + 2 \times 3$ 。若 $AIC_0 < AIC_2$, 则视为方差相等更好。这里, 由 $\sum_{k=-\infty}^{\infty} |C_k| < \infty$ 可知, 对任意 $\varepsilon > 0$, 存在满足 $\sum_{k=m+1}^{\infty} |C_k| < \varepsilon/2$ 的 m 。此时

$$\begin{aligned} \left| \frac{1}{N} \sum_{k=-\infty}^{\infty} |k| C_k e^{-2\pi i k f} \right| &\leq \frac{2}{N} \sum_{k=1}^{\infty} k |C_k| \leq \frac{2}{N} \sum_{k=1}^m k C_0 + 2 \sum_{k=m+1}^{\infty} \frac{k}{N} |C_k| \\ &\leq \frac{2}{N} \frac{m(m+1)}{2} C_0 + \varepsilon \end{aligned}$$

当 N 增大时右边第 1 项可以任意小, 因此 $E[\hat{p}(f)] \rightarrow p(f)$ 。

3. 正态分布模型的 AIC 为: 令 $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n y_i$, $\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{\mu})^2$, 则 $AIC_0 = n \log 2\pi\hat{\sigma}^2 + n + 2 \times 2$ 。

柯西分布模型的 AIC 为: 将 (4.36) 式最大化求最大似然估计值 $\hat{\mu}$ 、 $\hat{\tau}^2$ 时, $AIC_1 = n \log \hat{\tau}^2 - 2n \log \pi - \sum_{i=1}^n \log \{(y_i - \hat{\mu})^2 + \hat{\tau}^2\} + 2 \times 2$ 。若 $AIC_0 < AIC_1$, 则判断正态分布模型更好。

4. (1) 由二项分布模型 $f(m|n, p) = {}_n C_m p^m (1-p)^{n-m}$ 可得对数似然 $\ell(p) = \log {}_n C_m + m \log p + (n-m) \log(1-p)$ 。因此由 $\frac{d\ell}{dp} = 0$ 得 $\frac{m}{p} - \frac{n-m}{1-p} = 0$, 解得 $\hat{p} = \frac{m}{n}$ 。

(2) (1) 的模型 AIC 为 $AIC_1 = -2 \log {}_n C_m - 2m \log \hat{p} - 2(n-m) \log(1-\hat{p}) + 2$ 。另一方面, 假设等概率时 $p = \frac{1}{2}$, 故 $AIC_0 = -2 \log {}_n C_m + 2n \log 2$ 。若 $AIC_0 < AIC_1$, 则视为等概率更好。

5. 由 $\frac{d}{d\theta} \log f(y|\theta) = f(y|\theta)^{-1} \frac{df(y|\theta)}{d\theta}$ 可得

$$\begin{aligned} \frac{d^2}{d\theta^2} \log f(y|\theta) &= \frac{d}{d\theta} \left(\frac{d}{d\theta} \log f(y|\theta) \right) \\ &= \frac{d}{d\theta} \left(f(y|\theta)^{-1} \frac{df}{d\theta} \right) = -f(y|\theta)^{-2} \left(\frac{df}{d\theta} \right)^2 + f(y|\theta)^{-1} \frac{d^2 f}{d\theta^2} \\ &= - \left(\frac{d}{d\theta} \log f(y|\theta) \right)^2 + f(y|\theta)^{-1} \frac{d^2 f}{d\theta^2} \end{aligned}$$

因此,

$$\begin{aligned} J &= -E \left[\frac{d^2}{d\theta^2} \log f(y|\theta) \right] \\ &= E \left(\frac{d}{d\theta} \log f(y|\theta) \right)^2 - E \left[f(y|\theta)^{-1} \frac{d^2 f}{d\theta^2} \right] \\ &= I - \int_{-\infty}^{\infty} \frac{d^2 f}{d\theta^2} dy = I - \frac{d^2}{d\theta^2} \int_{-\infty}^{\infty} f(y|\theta) dy = I \end{aligned}$$

6. 由 $z = k(y) = \log y$, $f(z) = (2\pi\sigma^2)^{-1} \exp\{-(z-\mu)^2/2\sigma^2\}$ 可得 $\frac{dk}{dy} = y^{-1}$ 。因此

$$g(y) = \left| \frac{dk}{dy} \right| f(k(y)) = \frac{1}{y} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\log y - \mu)^2}{2\sigma^2}}$$

第 5 章

1. 令 μ 的最大似然估计值为 $\hat{\mu}$, 参数个数为 1, 因此

$$\text{AIC} = n \log 2\pi\sigma^2 - \frac{1}{\sigma^2} \sum_{i=1}^n (y_i - \hat{\mu})^2 + 2$$

2. (1) 由 $S = \sum_{n=1}^N (y_n - ax_n^2 - bx_n)^2$ 可得

$$\begin{aligned} \frac{dS}{da} &= -2 \sum_{n=1}^N x_n^2 (y_n - ax_n^2 - bx_n) \\ &= -2 \left(\sum_{n=1}^N x_n^2 y_n - a \sum_{n=1}^N x_n^4 - b \sum_{n=1}^N x_n^3 \right) = 0 \end{aligned}$$

$$\begin{aligned} \frac{dS}{db} &= -2 \sum_{n=1}^N x_n (y_n - ax_n^2 - bx_n) \\ &= -2 \left(\sum_{n=1}^N x_n y_n - a \sum_{n=1}^N x_n^3 - b \sum_{n=1}^N x_n^2 \right) = 0 \end{aligned}$$

解之得

$$\begin{aligned} \hat{a} &= \frac{\sum x_n^2 y_n \sum x_n^2 - \sum x_n y_n \sum x_n^3}{\sum x_n^4 \sum x_n^2 - (\sum x_n^3)^2} \\ \hat{b} &= \frac{\sum x_n y_n \sum x_n^4 - \sum x_n^2 y_n \sum x_n^3}{\sum x_n^4 \sum x_n^2 - (\sum x_n^3)^2} \end{aligned}$$

(2) $y_n = ax_n(x_n - b) + \varepsilon_n$ 。通过数值最优化, 对两个变量 a, b 最小化 $S = \sum_{n=1}^N (y_n - ax_n^2 - abx_n)^2$ 可求得最小二乘估计值。

第 6 章

1. (1) 在 (6.21) 式中令 $m = 1$, $1 - a_1 B = 0$ 的根在单位圆外为条件。因此 $|B| =$

$$|a_1^{-1}| > 1 \quad \text{即} \quad |a_1| < 1.$$

(2) 由 (6.21) 式, $m = 2$ 时根为 $B = (a_1 \pm \sqrt{a_1^2 + 4a_2})/2$ 。因此平稳性条件为:

当 $a_1^2 + 4a_2 \geq 0$ 时 $a_2 < 1 - a_1$ 且 $a_2 < 1 + a_1$, 当 $a_1^2 + 4a_2 < 0$ 时 $a_2 > -1$ 。综上, 满足平稳性的区域为由三条直线 $a_2 < 1 - a_1$ 、 $a_2 < 1 + a_1$ 、 $a_2 > -1$ 围成的三角形。

2. (1) σ^2

(2) 由 $y_{n+2} = ay_{n+1} + v_{n+2} = a^2 y_n + av_{n+1} + v_{n+2}$ 可得 $y_{n+2|n} = a^2 y_n$ 。因此 $E(y_{n+2} - y_{n+2|n})^2 = a^2 E v_{n+1}^2 + E v_{n+2}^2 = (1 + a^2) \sigma^2$

(3) 可以表示为 $y_{n+k} = v_{n+k} + g_1 v_{n+k-1} + g_2 v_{n+k-2} + \dots$ 。其中, 由 AR(1) 的脉冲响应函数得 $g_i = a^i$ 。此时 $y_{n+k|n} = g_k v_n + g_{k+1} v_{n+1} + \dots$, 因此

$$\begin{aligned} E(y_{n+k} - y_{n+k|n})^2 &= E(v_{n+k} + g_1 v_{n+k-1} + \dots + g_{k-1} v_{n+1})^2 \\ &= (1 + g_1^2 + \dots + g_{k-1}^2) \sigma^2 \\ &= (1 + a^2 + \dots + a^{2(k-1)}) \sigma^2 = \frac{1 - a^{2k}}{1 - a^2} \sigma^2 \end{aligned}$$

3. (1) 由 $C_0 = -0.9C_1 + 1$, $C_1 = -0.9C_0$ 得 $C_0 = 0.81C_0 + 1$, $C_0 = 1/0.19 = 5.26$,

$$C_k = 5.26 \times (-0.9)^k$$

(2) $C_0 = 25/7$, $C_1 = 75/28$, $C_3 = 15/14$, $C_4 = -9/28$

(3) $C_0 = 1 + b^2$, $C_1 = -b$, $C_k = 0$ ($k > 1$)

(4) 令 $E y_n v_{n-k} = g_k$, 则 $g_0 = 1$, $g_1 = a - b$ 。因此 $C_0 = aC_1 + g_0 - bg_1$, $C_1 = aC_0 - bg_0$ 可得

$$C_0 = \frac{1 - 2ab + b^2}{1 - a^2}, \quad C_k = \frac{(1 - ab)(a - b)}{1 - a^2} a^{k-1}$$

4. (1) 将 $v_n = y_n - ay_{n-1}$ 代入 $v_n = bv_{n-1} + w_n$ 的两边得

$$y_n = (a + b)y_{n-1} - aby_{n-2} + w_n$$

即得到 2 阶 AR 模型。

(2) y_n 的自协方差函数为 $a^k(1 - a^2)^{-1}$ ($k = 0, 1, 2, 3$)。因此 $C_0 = (1 - a^2)^{-1} + 0.01$, $C_k = a^k(1 - a^2)^{-1}$

5. (1) 将 $C_0 = 1 + b^2$, $C_1 = -b$, $C_k = 0$ ($k > 1$) 代入 $p(f) = \sum_{k=-\infty}^{\infty} C_k e^{-2\pi i k f}$ 得

$$p(f) = (1 + b^2) - be^{-2\pi i f} - be^{2\pi i f} = |1 - be^{-2\pi i f}|^2$$

(2) 由 $dp/df = -4\pi a \sin(2\pi f) / \{1 - 2a \cos(2\pi f) + a^2\}^2 = 0$ 得 $\sin(2\pi f) = 0$, 即 $f = 0$ 或 $1/2$ 时 $p(f)$ 取极值。当 $a > 0$ 时在 $f = 0$ 取最大值, 在 $f = 1/2$ 取最小值。 $a < 0$ 时相反。

6. 在 $y_n = v_n - bv_{n-1} = (1 - bB)v_n$ 中, 由 $(1 - bB)^{-1} = 1 + bB + b^2B^2 + \dots$ 得 $y_n = -by_{n-1} - b^2y_{n-2} - \dots + v_n$, 即为无穷阶 AR 模型。

7. (1) $y_{n+1|n} = -bv_n$, $y_{n+k|n} = 0$, 因此 $E\varepsilon_{n+1|n}^2 = 1$, $E(\varepsilon_{n+k|n}^2) = 1 + b^2$ ($k \geq 1$)

(2) $y_n = v_n a v_{n-1} + a^2 v_{n-2} + \dots$, $E(\varepsilon_{n+k|n}^2) = 1 + a^2 + \dots + a^{2(k-1)}$

(3) 由 $y_n = v_n + v_{n-1} + v_{n-2} + \dots$ 得 $E(\varepsilon_{n+k|n}^2) = k$ 。

1. (1) $AIC_m = N(\log 2\pi \hat{\sigma}_{n-1}^2 + \log(1 - (a_m^m)^2) + 2(m+1)) = AIC_{m-1} + N\log(1 - (a_m^m)^2) + 2$ 。因此, 若 $N\log(1 - (a_m^m)^2) + 2 < 0$, 即 $(a_m^m)^2 > 1 - e^{-2/N}$, 则判断 $AR(m)$ 比 $AR(m-1)$ 更好。
 - (2) $\sigma_1^2 = (1 - 0.9^2) \times 1 = 0.19$, $\sigma_2^2 = 0.1216$, $\sigma_3^2 = 0.1107$, $\sigma_4^2 = 0.1095$, $\sigma_5^2 = 0.1071$ (显示到小数点后 4 位)
 - (3) $AIC_0 = 100(\log 2\pi \times 1 + 2) = 285.78$
 $AIC_1 = 100(\log 2\pi \times 0.19 + 2) = 121.71$
 $AIC_2 = 79.09$, $AIC_3 = 71.65$, $AIC_4 = 72.65$, $AIC_5 = 72.37$ 。因此 3 阶为最优。
2. Yule-Walker 法的优点是始终保证平稳性, 但特别在数据较少时偏差较大。这是因为相当于假设观测数据区间前后为 0。最小二乘法的优点是可以获得偏差较小的估计值, 且数据的合并等操作比较容易, 但不能保证平稳性, 且最初几个点用于条件化而不能直接用于参数估计。PARCOR 法如果限于 AR 模型的估计, 可以非常高效地获得接近最大似然估计值的估计值。
3. 分别对时间序列 x_n 和 y_n 拟合一元 AR 模型, 其 AIC 值分别记为 AIC_x 、 AIC_y 。另一方面, 对 $(x_n, y_n)^T$ 拟合二元 AR 模型, 求 AIC_0 。若 $AIC_x + AIC_y < AIC_0$, 则视 x_n 和 y_n 为独立更好。

第 8 章

1. 考虑 $y_n = \mu + a_1 y_{n-1} + \cdots + a_m y_{n-m} + v_n$ 所表示的增加 1 个参数的局部平稳模型即可。
2. 将 y_n 在 $n = 1, \dots, k$ 时设为 $N(0, \sigma_1^2)$, $n = k+1, \dots, N$ 时设为 $N(0, \sigma_2^2)$ 的模型与 $y_n \sim N(0, \sigma_0^2)$, $n = 1, \dots, N$ 进行比较即可。AIC 分别为

$$AIC_{12} = N \log 2\pi + k \log \hat{\sigma}^2 + (N-k) \log \hat{\sigma}_2^2 + N + 2 \times 2$$

$$AIC_0 = N \log 2\pi + N \log \hat{\sigma}_0^2 + N + 2$$

若 $AIC_{12} < AIC_0$, 则认为方差发生了变化。

3. 当 $n = 1, \dots, k$ 时 $y_n = a_0 + a_1 x_n + \cdots + a_p x_n^p + \varepsilon_n$, $n = k+1, \dots, N$ 时 $y_n = b_0 + b_1 x_n + \cdots + a_q x_n^q + \delta_n$ 。其中 $\varepsilon_n \sim N(0, \sigma_1^2)$, $\delta_n \sim N(0, \sigma_2^2)$ 。

$$AIC = N \log 2\pi + k \log \hat{\sigma}_1^2 + (N-k) \log \hat{\sigma}_2^2 + N + 2 \times (p+q+4)$$

第 9 章

1. 对 (9.1) 两边左乘 T 得 $Tx_n = TF_nx_n + TG_nv_n = TF_nT^{-1}Tx_n + TG_nv_n$. 这里令 $z_n = Tx_n$, 利用 $y_n = H_nx_n + w_n = H_nT^{-1}Tx_n + w_n$, 可以验证得到等价的表示。

2. (1) 初始值: $x_{0|0} = 0, V_{0|0} = 100$

$$\text{预测: } x_{n|n-1} = x_{n-1|n-1}, V_{n|n-1} = V_{n-1|n-1} + \tau^2$$

$$\text{滤波: } G_n = V_{n|n-1}/(V_{n|n-1} + 1),$$

$$x_{n|n} = x_{n|n-1} + G_n(y_n - x_{n|n-1}),$$

$$V_{n|n} = (1 - G_n)V_{n|n-1} = G_n$$

$$(2) V_{n+1|n} = V_{n|n-1}/(V_{n|n-1} + 1) + \tau^2$$

$$(3) \text{ 在上式中令 } V_{n+1|n} = V_{n|n-1} = V.$$

$$(4) \text{ 设第 (3) 题的解为 } V, \text{ 则滤波的协方差为 } V' = V/(V + 1),$$

$$x_{n+1|n} = x_{n|n} = x_{n|n-1} + \frac{V}{V+1}(y_n - x_{n|n-1})$$

3. $V_P = V, V_F = G = V/(V + 1), A = V/(V + 1)/V = 1/(V + 1)$, 由 $V_S = V_F + A^2(V_S - V_P)$ 得 $V_S = V/(V + 2)$

τ^2	1	0.1	0.01	0.001
V_P	1.1618	0.370	0.105	0.032
V_F	0.618	0.270	0.095	0.031
V_S	0.447	0.156	0.050	0.016

第 10 章

1. 由 $x_n = (y_n, y_{n-1}, \dots, y_{n-m+1})^T$ 可知 $Ex_n x_n^T$ 的 (i, j) 分量为 $Ey_{n-i+1}y_{n-j+1} = C_{i-j}$. 因此, 计算 m 阶 AR 模型的自协方差函数 $C_k (k = 0, 1, \dots)$, 按如下方式确定即可。

$$V_{0|0} = \begin{bmatrix} C_0 & C_1 & \cdots & C_{m-1} \\ C_1 & C_0 & \cdots & C_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m-1} & C_{m-2} & \cdots & C_0 \end{bmatrix}$$

2. 在 (10.9) 式的变换后变量中, 绝对值较大的区域内对数似然的变化很小, 使用数值差分的最优化可能无法起作用。因此, 需要将变换后参数的绝对值限制在例如 20 左右以下。

第 11 章

1. (1) 在 $y_{n+k} = y_{n+k-1} + v_{n+k} = \cdots = y_n + v_{n+1} + \cdots + v_{n+k}$ 中, 由 $E v_{n+i} = 0$ $E v_{n+i}^2 = \sigma^2$ 可得 $y_{n+k|n} = y_n$, $E \varepsilon_{n+k|n}^2 = k\sigma^2$
- (2) 由 $k\sigma^2 = 4 \times 40,000 = 160,000$ 得 $\sqrt{k}\sigma = 400$ 。正态分布中大于 1σ 的概率为 0.16。
- (3) 即使作为随机游走, 其方差 σ^2 也会随时间变动。因此, 可以考虑考虑波动率变化的模型

$$y_n = y_{n-1} + \sigma_n w_n, \\ \log \sigma_n^2 = a + b \log \sigma_{n-1}^2 + v_n \quad (\text{或 } \log \sigma_n^2 = \log \sigma_{n-1}^2 + v_n)$$

2. (示例解答) 可以考虑以三角函数模型为趋势的模型。
3. 省略

第 12 章

1. AR 模型的特征根接近 1 时, 与趋势分量的区分变得困难。因此, 在 AR 系数的估计中, 最好加以限制使特征根的绝对值不接近 1。
2. 令 $S_n = \sum_{j=1}^6 (a_j \cos \frac{\pi j n}{6} + b_j \sin \frac{\pi j n}{6})$, 即可得到用三角函数表示的季节分量。
3. (1) 令 $\beta_{n1} = \beta_{n7}$, $\beta_{n2} = \cdots = \beta_{n6}$, 可得到具有两个参数的星期效应分量。此时

$$t d_n = \beta_{n1}(d'_{n1} + d'_{n7}) + \beta_{n2}(d'_{n2} + \cdots + d'_{n6})。$$
 (2) 假设 $\beta_{n2} = \cdots = \beta_{n6}$, 同样可得到具有三个参数的模型。
4. 黄金周等假期因年份不同而休假模式有所变化, 需要考虑其影响。此外, 闰年的影响仅靠星期调整可能无法充分修正。
5. 省略

第 13 章

1. (示例解答) 股价数据
2. 在时变系数回归模型 $y_n = \beta_{0n} + \beta_{1n}x_{n1} + \cdots + \beta_{mn}x_{nm} + \varepsilon_n$ 中, 对系数 $\beta_{0n}, \beta_{1n}, \cdots, \beta_{mn}$, 例如考虑随机游走模型 $\beta_{jn} = \beta_{j,n-1} + v_n$ 即可。该模型可以用状态空间模型表示, 因此可以像时变系数 AR 模型一样用卡尔曼滤波进行估计。

第 14 章

1. 与正态分布相比, 分布集中在中心的同时尾部较重。因此适合于表达偶尔发生的结构变化和异常值。
2. 当 $y \sim N(0,1)$ 时令 $z = y^2$, $x = \log z$ 。此时 z 小于 t 的概率为

$$F(t) = \text{Prob}(y^2 < t) = \frac{2}{\sqrt{2\pi}} \int_0^{\sqrt{t}} e^{-\frac{y^2}{2}} dy$$

对 t 求导得

$$f(t) = \frac{dF}{dt} = \frac{1}{\sqrt{2\pi}} t^{-\frac{1}{2}} e^{-\frac{t}{2}}$$

(自由度 1 的 χ^2 分布)。进一步利用 (4.51) 的逆求对数变换 $x = h(z)$ 的分布, 可得如下 $g(x)$ 。

$$g(x) = \left| \frac{dh^{-1}}{dx} \right| f(e^x) = e^x \frac{1}{\sqrt{2\pi}} (e^x)^{-\frac{1}{2}} e^{-\frac{e^x}{2}} = \frac{1}{\sqrt{2\pi}} e^{\frac{x}{2} - \frac{e^x}{2}}.$$

3. 在趋势模型中将系统噪声的分布设为

$$p_v(x) = \begin{cases} \delta(0) & \text{以概率 } 1 - \alpha \\ N(0, \tau^2) & \text{以概率 } \alpha \end{cases}$$

即可。其中 $\delta(0)$ 是集中在原点的分布函数。

第 15 章

1. 设 y_n 为时间序列, t_n 为趋势, 则

$$\begin{aligned} t_n &= t_{n-1} + v_n, \\ \log \sigma_n^2 &= \log \sigma_{n-1}^2 + u_n, \\ y_n &= t_n + \sigma_n w_n, \end{aligned}$$

假设 $u_n \sim N(0, \tau_1^2)$, $v_n \sim N(0, \tau_2^2)$, $w_n \sim N(0, 1)$ 的模型。该模型的观测模型是非线性的。

2. 一般来说, 在蒙特卡罗法中, 精度提高一个数量级需要 100 倍的粒子数。
3. 如果完全不进行重采样, 许多粒子的权重将接近 0, 精度变差。
4. 当粒子数为 m , 数据数为 N , 状态维度为 k , 滞后为 L 时, 固定区间平滑需要保存 $m \times k \times N$ 个, 固定滞后平滑需要保存 $m \times k \times L$ 个左右的数据。注意一般 $N \gg L$ 。

第 16 章

1. 设 $f(x)$ 为均匀分布的密度函数, $f(x) = 1$ ($0 \leq x \leq 1$), $= 0$ (其他)。此时

$$g(x) = \int_{-\infty}^{\infty} f(y)f(x-y)dy = \int_0^1 f(x-y)dy = \begin{cases} x & 0 \leq x \leq 1 \\ 2-x & 1 \leq x \leq 2 \\ 0 & \text{其他} \end{cases}$$

因此 W 的密度函数为三角形。

2. 令 $y_0 = 1, v_n \equiv 0$, 则 $y_n = a^n$ 。因此, 取 m 使得 $a^m < c$ (c 例如为 10^{-3}) 以上即可忽略初始值的影响。当 $a = 0.9, 0.5, 0.1$ 时, m 分别为 66、10、5。
3. 利用截至当前的观测值 y_1, \dots, y_n , 构造状态向量 x_n , 用随机数生成 x_{n+1}, x_{n+2}, \dots 即可。利用 (9.5) 式的状态进行 AR 模型的仿真时, 令 $x_n = (y_n, y_{n-1}, \dots, y_{n-m+1})^T$ 即可。

索引

- AIC, 68, 104, 121, 127, 156
 ARMA 模型, 86, 146
 ARMA 模型的仿真, 229
 AR 模型, 103
 AR 模型, 230
 autocorrelation function, 33

 BFGS 公式, 239
 BLSALLFOOD 数据, 19, 48, 172, 175
 Box-Cox 变换, 73
 Burg 算法, 109

 DFP 公式, 239

 FFT, 54

 GFSR 法, 224
 Görtzel 法, 53

 K-L 情报量, 62

 maximum temperature data, 27, 162
 Mersenne Twister, 224

 PARCOR, 89, 188
 PARCOR 法, 107

 sunspot number data, 151
 sunspot number data continued, 153

 twisted GFSR 法, 224

 WHARD 数据, 19, 48, 155, 157, 170, 178

 赤池信息量准则, 68
 异常值, 21
 异常值, 206
 相位谱, 98
 一变量时系列, 20
 一样分布, 59
 均匀随机数, 223
 1 期先予测, 132, 134, 194
 移动平均系数, 86
 移动平均模型, 87
 脉冲响应函数, 149
 脉冲响应函数, 87
 后向 AR 模型, 240
 熵最大化原理, 62
 回归模型, 77
 χ^2 分布, 59, 226
 高斯型时间序列, 20
 高斯分布, 59
 可逆性, 150
 Kullback-Leibler 信息量, 62
 卡尔曼滤波, 242
 观测噪声, 131
 观测模型, 131
 气温数据, 164
 季节分量, 167
 季节分量模型, 167
 季节调整模型, 169, 232
 局部平稳 AR 模型, 120
 互谱密度函数, 97
 欠测值, 21, 142
 结构变化, 123
 同余法, 223
 柯西分布, 59, 204, 206, 228
 固定空间平滑化, 135
 相干性, 98
 Cholesky 分解, 230
 混合分布, 206
 最高气温数据, 17, 47, 81, 156
 最小二乘法, 106, 115
 最小二乘法, 77
 最大对数似然, 80
 最大似然估计, 150
 最尤推定值, 104, 141
 最大似然法, 66
 残差, 77
 散点图, 30, 35
 时间差分算子, 159
 时间移位算子, 87
 时系列, 15
 自回归移动平均模型, 86
 自回归系数, 86
 自回归模型, 86
 自协方差函数, 32, 88, 149
 自相关函数, 32
 地震波数据, 19, 48, 124, 183, 187, 202
 阶数, 77, 86
 阶数选择, 80
 指数分布, 59
 指数随机数, 226
 系统噪声, 131
 系统模型, 131
 实现值, 60
 自动划分, 121
 季度数据, 167
 时变 AR 系数, 184
 时变系数 AR 模型, 184
 时变谱, 188
 时变方差模型, 181, 201
 仿真, 223, 230
 仿真, 233
 弱平稳, 31
 频率响应函数, 91, 190
 拟牛顿法, 238
 状态, 131
 状态空间表示, 146
 状态空间模型, 131, 230
 初始状态, 149
 初始值, 152
 幅度谱, 98
 数值的实现, 195
 数值最优化, 67
 谱, 42
 谱窗, 52
 正规分布, 59

- 正态随机数, 225
 線形時系列, 21
 船舶数据, 17, 36, 47, 55, 114
 互协方差函数, 37
 互相关函数, 37
 相对功率贡献率, 98
 对数似然, 65
 对数尤度, 140
 对数似然関数, 66
 太阳黑子数数据, 17, 111, 152
 多项式回归模型, 155
 多元 AR 模型, 96, 112, 115
 多元时间序列, 35
 多变量時系列, 20
 多元自回归模型, 96
 長期予測, 135
 长期预测, 172
 月度数据, 167
 平稳 AR 分量, 172, 174
 定常時系列, 20
 平稳序列, 32
 平稳性, 150
 t 分布, 235
 数据的追加, 83
 数据的变换, 72
 特性方程, 93
 折中参数, 164
 趋势, 155
 趋势分量模型, 159
 趋势模型, 162
 二項分布, 207
 2 重指数分布, 59, 204
 双指数分布, 181, 235
 双指数随机数, 226
 Householder 变换, 79, 126
 豪斯霍尔德变换, 122
 Householder 法, 78, 107, 115
 白噪声, 32, 43
 参数估计, 139
 功率谱密度函数, 42
 功率谱, 90, 92
 可逆, 93
 Pearson 分布族, 59, 200
 非高斯型时间序列, 20
 非高斯型状态空间模型, 194
 非高斯型状态空间模型, 233
 非高斯型趋势模型, 198
 非高斯型模型, 193
 直方图, 29
 非线性最优化, 238
 非線形時系列, 21
 非定常時系列, 20
 非平稳二项过程模型, 207
 非平稳 Poisson 过程模型, 207
 非定常離散系列, 207
 标准季节调整模型, 169
 样本自协方差函数, 33
 样本自相关函数, 33
 样本谱, 46
 样本互协方差函数, 38
 样本互相关函数, 38
 样本平均, 33
 样本平均向量, 38
 周期图, 46
 滤波, 133, 134, 195
 傅里叶变换, 42, 54
 子集回归模型, 84
 分布函数, 228
 分布関数, 58
 平滑, 242
 平滑化, 133, 134
 平滑化公式, 195
 平均对数似然, 65
 平均值函数, 31
 平均向量, 37
 变化时点, 126
 偏自相关系数, 89
 变量选择, 84
 Poisson 分布, 207
 補間, 142
 白噪声, 32
 密度関数, 58
 Jacobian, 73
 似然, 66
 尤度, 140
 Yule-Walker 估计值, 105
 Yule-Walker 法, 112
 Yule-Walker 方程, 89, 97, 241
 星期效应, 175
 星期调整, 175
 予測, 133, 136
 预测误差, 240
 滞后, 32
 随机数, 223
 随机游走模型, 160, 235
 離散時間時系列, 20
 Levinson 算法, 104, 240