武漢理工大學

数学建模暑期培训论文

第 4 题

基于贝叶斯优化和灰度预测的传染病动力学模型

第 26 组

姓名	方向
许鸢飞（组长）	编程
尹可汗	编程建模
李想	写作建模

2020 年 8 月 20 日

摘要

本文通过对新型冠状病毒传播方式进行机理分析，建立了基于贝叶斯优化的传染病动力学模型，通过贝叶斯方法优化参数，对安徽省和英国海峡群岛两地区的疫情状况进行了合理的预测，同时利用灰色预测，AR 时间序列，量化消费者心理，较准确预测了社会消费品零售总额的受疫情影响的变化.

针对问题一，本文通过分析病毒传播方式建立了解决医疗挤兑的**传染病动力学模型**，将地区人口依据是否感染病毒，是否接受隔离或治疗进行分组，将感染率，治愈率，开始隔离政策的天数作为微分方程模型的参数，利用**贝叶斯优化**方法，得到了**安徽省和英国海峡群岛**两个地区参数设定范围的最优值. 创新性地提出分段治愈率参数，模拟医疗资源扩充和医疗技术提高解决疫情初期的医疗挤兑从而提高了治愈率. 我们发现对于**安徽省，每提前执行隔离措施 5 天，新增确诊和死亡率的峰值都下降到 50% 左右；对于海峡群岛，每提前执行隔离措施 5 天，新增确诊和死亡率的峰值都下降到 25% 左右.**

针对问题二，我们从**销售额确定性增长趋势，周期性随机波动以及疫情下消费者悲观指数**三个方面建立了**社会消费品零售总额预测模型**，该模型以国家统计局搜集的 2015-2020 年销售额数据为基础，利用**灰色预测**方法对消费额的长期增长趋势进行预测，并以预测模型与原始数据的差值波动建立 **AR 时间序列**来模型模拟消费额的**季节性变化因素**，并引入**消费者信心，居民消费价格指数**等经济学指标，计算疫情期间 CPI 数据与基准数据的差值比例来描述消费者的**悲观程度，结果显示社会消费品零售总额将在 7 月增长 0.4%, 经济也会开启复苏阶段. 模型预测 2020 年下半年我国社会消费品零售总额分别为 33525.9，33205，34028，34626，38225，38908 亿元.**

本文根据上述新冠病毒传播模型的预测结果和疫情突发事件下的销售额变动分析，从**患者人群，疫情防控措施，突发事件的经济影响以及消费者心理**等多个角度向**国家卫生部门**写了一封信, 给出了一些相关的建议.

本文的优点:1. 根据对病毒传播过程做了严谨的分析建立动力学模型，可解释性强. 创新性地提出分段治愈率参数，模拟医疗挤兑的解决过程. 2. 通过贝叶斯方法，参考文献提供的参数范围，依据不同地区的病例数据对模型进行参数搜索，使得模型预测不同地区疫情时可推广性强. 3. 量化了消费者信心, 对消费品零售额变化的原因给出了经济学解释.

关键词： 动力学模型 贝叶斯优化 灰度预测 时间序列 消费者信心量化

目录

一、 问题重述

1.1 问题背景

新冠冠状肺炎 (WHO 命名为 2019-nCOV）于 2019 年末袭击湖北省武汉市，截止 2020 年 8 月 18 日已经造成全球范围内超 2 千万人感染，超 77 万死亡. 新型冠状病毒大流行是全球百年一遇的特大卫生危机，目前全球范围内新冠肺炎的致死率约为 3%，疫情在东亚和西欧的肆虐已经基本接近尾声，而美洲和南亚的情况则不容乐观，目前已被证实的是新冠肺炎的传播速度和致死率远超 SARS 病毒，各国政府已经大多已对居民采取隔离治疗措施，但除少数国家外，大部分国家的疫情发展趋势还在恶化. 图1是截止 8.18 日的全球新冠肺炎病例确诊地图.

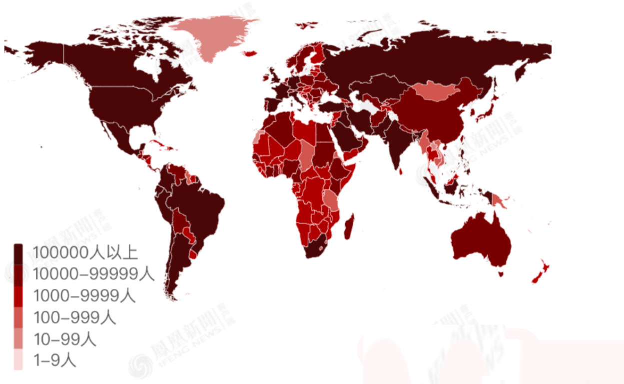

图 **1**　全球确诊地图

1.2 待解决的问题

问题一: 通过建立的模型预测至少两个不同国家或者地区的确诊病例和死亡病例的变化，分析各个国家或地区卫生部门采取的举措对降低该国家或地区疫情流行程度的影响并做对此做出评论.

问题二: 大流行对国民经济发展产生了巨大的影响，收集某一方面经济发展的数据，建立相应的数学模型以预测未来该方面经济发展的趋势.

问题三: 基于一二问的模型和预测结果, 向相关国家或者地区的卫生部门写一封信，给出对该国或地区的卫生预防建议.

二、 模型假设

1. 不考虑疫情期间国家或地区人口因迁入或迁出，自然出生或死亡的等因素的影响.
2. 治愈后的患者由于抗体或者防护措施增强等因素，不会再被感染
3. 隔离后的患者或者易感染者不会传播或感染病毒.

三、 符号说明

符号	含义
N	国家或地区的总人口
S	易感染人群人数
E	潜伏期易感染者
I	已发病但未隔离患者
Q	已发病且已隔离但未被治疗患者
J	已发病已隔离且正被治疗者
R	治愈者
k	潜伏期患者与已发病患者传染强度的比率
β	病毒传播速率
ε	潜伏期感染者转入未隔离感染者的比例
λ	潜伏期感染者转入已隔离感染者的比例
θ	未隔离感人者转入正在治疗的人群的比例
σ	已隔离感染者转入正在治疗的人群的比例
γ	正在治疗的已感染者治愈的比例
δ	未隔离的已感染者的死亡率
η	接受治疗的病人的死亡率
h	自我隔离的易感染者的比例

注：表中未说明的符号以首次出现处为准

四、 问题一模型的建立与求解

4.1 问题分析

问题一要求我们对不同国家或者地区的确诊病例和死亡病例的变化进行预测，我们会对病毒的传播原理做出机理分析，将本地区的人口按照是否感染病毒，是否接受隔离，

是否接受治疗等方面进行划分，病毒在不同人群中间的传播规律是不同的，我们将建立微分方程模型用以描述病毒在人口间的传播. 我们画出了问题一的思维导图. 如图2.

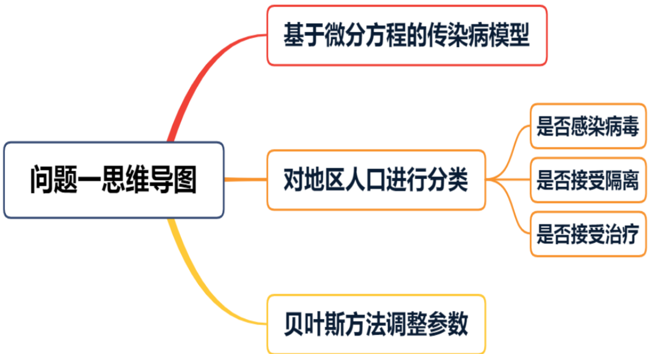

图 **2**　问题一思维导图

4.2 解决医疗挤兑的传染病动力学微分方程模型

为了理清新冠病毒在人群中的传播规律，我们有必要对一些概念作出解释. 我们将该地区人口按照感染病毒与否分为易感染者和患者，在本次新冠病毒大流行中，几乎所有年龄阶段的人都有几率感染病毒，老年人感染病毒的几率较大而年轻人较小，但在此我们不对易感染者做出基于年龄，性别等方面的分类. 我们的模型中将人口分为以下几个组，人群之中转变的关系如下图所示.

易感染者分组: 我们假设接受隔离的易感染者与易感染者总数之间存在一个比例关系 h, 易感染者被分为**未自我隔离**的易感染者 $(1-h)S$ 和**自我隔离**的易感染者 hS，S 为易感染者，即尚未感染的总人口. 我们假定只有未隔离的易感者才有感染病毒的风险，这部分易感者可能被潜伏期患者 E，或已发病且已经隔离但尚未治疗的感染者 Q 感染.E 人群的传染性比 Q 弱，比例系数 k 处在 0-1 之间.

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -(\beta(kE+I)(1-h)S. \tag{1}$$

未隔离易感者比例 h 会随着政府的建议或者强制措施而变化. 我们假设无隔离政策是所有人气都是未隔离状态，居家令颁布后 $(t \geq QS)$ 自我隔离的易感者比例为 0.9，即：

$$h = \begin{cases} 0, & t < QS\ . \\ 0.9, & t \geq QS. \end{cases} \tag{2}$$

易感染者朝感染者的转变: 据相关研究发现新冠病毒在人体内有一段时间约在的潜伏期，所以易感染者往往不会立即表现出发病状况，而是要在一段时间的潜伏期后才体现出症状，但这时的感染者已经有了轻度传染性. 我们用 E 表示已经感染病毒但还在潜

伏期的人群，此人群中一部分会被隔离起来，一部分则由于医疗条件的限制或者自身意愿不愿被隔离.

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \beta(kE + I)(1 - h)S - (\varepsilon + \lambda)E. \tag{3}$$

其中 I 表示已发病但尚未隔离的感染者，他们具有强传染性，可以将病毒传染给易感染者使其变为潜伏期患者,Q 表示已发病且已经隔离但尚未治疗的感染者,β 被用来表示病毒的传播速率.ε 和 λ 分别表示潜伏期患者转入 I 和 Q 的比例关系.

感染者内部: 在感染者内部，经历过一定的潜伏期后，潜伏期患者会出现感染症状，此时他们可能被送入隔离，也可能因为医疗条件或其他原因而无法被隔离. 对于没有隔离的已出现症状的感染者，可能会面临死亡风险也可能在后期被送入隔离治疗，隔离治疗的病人组用 J 来表示.

$$\frac{\mathrm{d}I}{\mathrm{d}t} = \varepsilon E - (\theta + \delta)I, \tag{4}$$

其中 θ 表示患者被隔离治疗的比例，而 δ 表示该组患者的死亡率.

对于被送入隔离的感染者 Q，它们可能会在一段时间后接受治疗.

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = \lambda E - \sigma Q, \tag{5}$$

σ 表示该组已感染者转入治疗的比例.

对于被送入隔离治疗环境的患者组 J，他们可能会在治疗后痊愈并不再感染该病毒，或者面临死亡.

$$\frac{\mathrm{d}J}{\mathrm{d}t} = \theta I + \sigma Q - (\gamma + \eta)J. \tag{6}$$

其中 γ 表示患者的治愈率，而 η 表示该组患者的死亡率.

治愈: 对于患者来说，一旦感染病毒就意味着最终只有两个结果，治愈或者死亡. 我们假设治愈着主要来着在医院就诊的患者，参数 γ 表示医院的接诊能力或医疗水平，直接影响治愈率：

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \gamma J, \tag{7}$$

在我们的**医疗挤兑模型**中，疾病爆发初期，由于接诊能力，治疗技术和医用物资等因素的限制，医院尚未做好迎接大规模疫情的准备，所以前期一段时间内 $0 \le t < MUS$ 保持较低的治愈率 γ_0，出现医疗资源挤兑的情况. 然后由于相应的医院扩容，物资补给政策，我们假设医院治疗能力在一段时间 $MUS \le t < MUE$ 内线性增加，直到保持在一个较高的水平 γ_1.

$$\gamma = \begin{cases} \gamma_0, & 0 \le t < MUS\ . \\ \gamma_0 + \frac{(\gamma_1 - \gamma_0)(MUE - t)}{(MUE - MUS)}, & MUS \le t < MUE\ . \\ \gamma_1, & MUE \le t. \end{cases} \tag{8}$$

死亡：发病但未隔离的患者 I 和治疗中的患者 J 每天都有一定比率死亡:

$$\frac{\mathrm{d}D}{\mathrm{d}t} = \delta I + \eta J. \tag{9}$$

显然上述微分方程组满足如下约束，该地区的总人口（用各个人群之和进行表示）保持不变.

$$N = S + E + I + Q + R + D. \tag{10}$$

$$\frac{\mathrm{d}N}{\mathrm{d}t} = 0. \tag{11}$$

经过前面的推导，我们已经为系统内各人群的数量变化建立**常微分方程**模型.

模型综合

$$\begin{cases} \frac{\mathrm{d}S}{\mathrm{d}t} = -(\beta(kE+I)(1-h)S, \\ \frac{\mathrm{d}E}{\mathrm{d}t} = \beta(kE+I)(1-h)S - (\varepsilon+\lambda)E, \\ \frac{\mathrm{d}I}{\mathrm{d}t} = \varepsilon E - (\theta+\delta)I, \\ \frac{\mathrm{d}Q}{\mathrm{d}t} = \lambda E - \sigma Q, \\ \frac{\mathrm{d}J}{\mathrm{d}t} = \theta I + \sigma Q - (\gamma+\eta)J, \\ \frac{\mathrm{d}R}{\mathrm{d}t} = \gamma J, \\ \frac{\mathrm{d}D}{\mathrm{d}t} = \delta I + \eta J. \end{cases} \tag{12}$$

4.3 模型求解

对于各个地区，微分方程模型的参数各不相同. 我们根据实际统计的确诊，死亡，治愈人数，与输出的解比较，定义出损失函数. 然后依据损失函数对模型的参数进行调优.

4.3.1 数据预处理

我们使用附件 1，来着约翰 • 霍普金斯大学系统科学与工程中心（CSSE）的 COVID-19 数据存储库作为待研究地区确诊，死亡，治愈人数的参考. 附件 1 给出了各地区按天统计的累计确诊，死亡，治愈病例数. 我们差分计算出每日新增的病例数，并计算每周 7 日的滑动平均值. 以安徽省数据为例，数据处理的结果见图3.

$$\Delta D_i = D_{i+1} - D_i, \Delta C_i = C_{i+1} - R_i, \Delta R_i = R_{i+1} - R_i. \tag{13}$$

$$\begin{aligned} DMA_i &= \frac{\sum_{p=i}^{i+w} D_p}{w}, \\ CMA_i &= \frac{\sum_{p=i}^{i+w} C_p}{w}, \\ RMA_i &= \frac{\sum_{p=i}^{i+w} R_p}{w}. \end{aligned} \tag{14}$$

$$
\begin{aligned}
\Delta DMA_i &= \frac{\sum_{p=i}^{i+w} \Delta D_p}{w}, \\
\Delta CMA_i &= \frac{\sum_{p=i}^{i+w} \Delta C_p}{w}, \\
\Delta RMA_i &= \frac{\sum_{p=i}^{i+w} \Delta R_p}{w}.
\end{aligned} \tag{15}
$$

其中 $\Delta D_i, \Delta C_i, \Delta R_i$ 为第 i 天新增确诊，死亡，治愈的人数，$\Delta DMA_i, \Delta CMA_i, \Delta RMA_i$ 为 i 到 $i+w$ 天新增确诊，死亡，治愈的人数均值，w 为滑动平均的窗口大小.

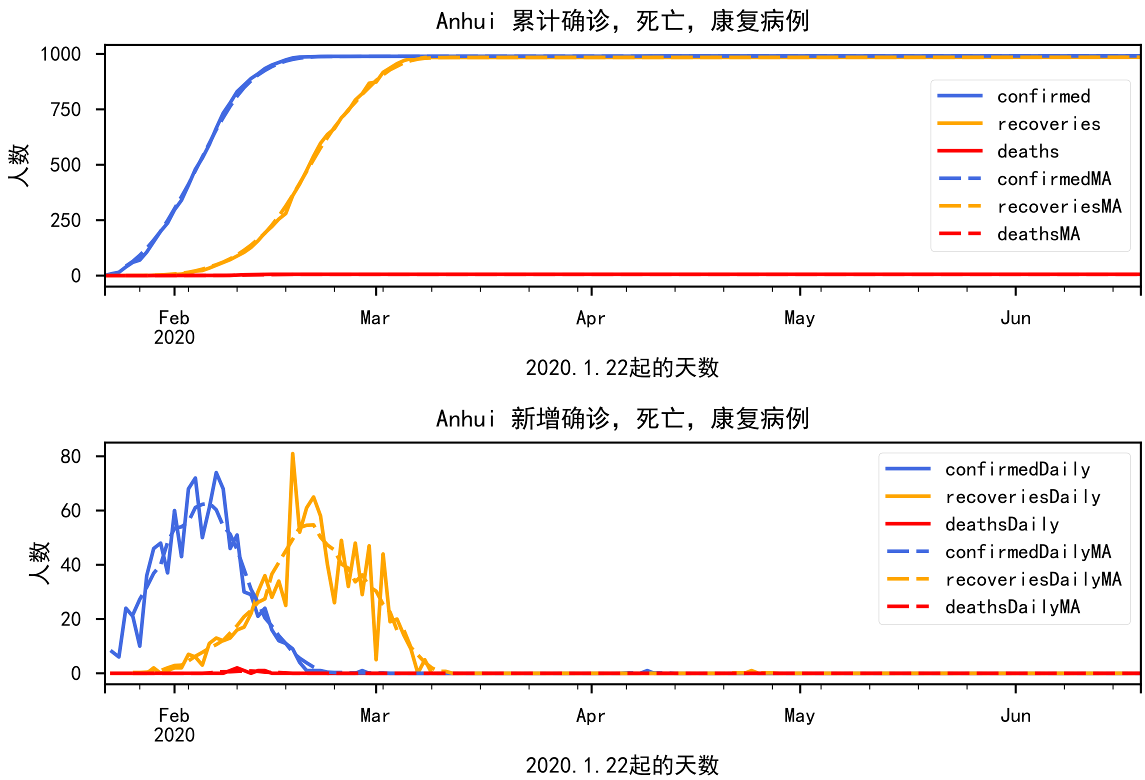

图 **3**　安徽省确诊，死亡，治愈病例数据

4.3.2 贝叶斯参数调优

我们将微分方程所有可调节的参数设为 $\mathbf{p}$，对于每一个确定的 $\mathbf{p}$，我们都能求出微分方程的数值解，即 $\Delta D_i', \Delta C_i', \Delta R_i'$. 这三个数列和真实数据 $\Delta D_i, \Delta C_i, \Delta R_i$ 符合的越好，说明模型的预测能力越强. 为了对 $\mathbf{p}$ 进行调优，我们需要定义损失函数为模型预测的每日新增病例与真实数据的绝对值误差：

$$
L(\mathbf{p}) = \frac{1}{n}\left(\sum_i |\Delta D_i' - \Delta D_i| + \sum_i |\Delta D_i' - \Delta D_i| + \sum_i |\Delta D_i' - \Delta D_i|\right). \tag{16}
$$

我们仅能对函数 $L(\mathbf{p})$ 在点 $\mathbf{x}$ 进行抽样求值. 每次求值都需要对微分方程模型在进行模拟然后计算损失函数，这涉及较大的计算量. 这类优化是贝叶斯优化技术最有用的领域.

贝叶斯优化用较少的采样数目逼近全局最优值. 贝叶斯优化包含了关于 $L(\mathbf{p})$ 的先验信念，并用从 $L(\mathbf{p})$ 中抽取的样本更新先验，从而得到一个更好地接近 $L(\mathbf{p})$ 的后验. 用于近似目标函数的模型称为“代理模型”. 贝叶斯优化还使用了一个“获取函数”，将采样导向可能优于当前最佳观测的区域.

代理模型 (Surrogate model) 我们使用的代理模型是高斯过程 Gaussian Process (GP). 高斯过程是一个随机过程，其中任意点 $\mathbf{x} \in \mathbb{R}^d$ 被赋予一个随机值 $f(\mathbf{x})$，其中是这些变量的有限数量的联合高斯分布：

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K}). \tag{17}$$

其中 $\mathbf{f} = (f(\mathbf{x}_1), ..., f(\mathbf{x}_N)), \boldsymbol{\mu} = (m(\mathbf{x}_1), ..., m(\mathbf{x}_N)), K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$，$m$ 是均值函数. 高斯过程的函数的分布“平滑度”由 K 定义.

一个 GP 的先验 $p(\mathbf{f}|\mathbf{X})$ 可以在观察到一些数据 y 后转化为 GP 的后验 $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$，根据：

$$p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{y}) = \int p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{f}) p(\mathbf{f}|\mathbf{X}, \mathbf{y})\, d\mathbf{f} = \mathcal{N}(\mathbf{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*). \tag{18}$$

GPs 定义了我们对目标函数的先验知识，我们可以使用它们来描述关于目标函数的先验信念，例如“平滑性”.GP 后验计算成本低，用于在搜索空间中评估哪些采样点可能产生改进.

获取函数 (Acquisition functions) 在搜索空间中提出采样点是通过获取函数来实现的. 获取函数 $u(\mathbf{x}|\mathcal{D}_{1:t})$ 综合权衡利用 (exploitation) 和勘探 (exploration). 利用是指在替代模型预测的目标高的地方进行采样，勘探是指在预测不确定性高的地方进行采样. 两者会倾向于提高采集函数值，使得获取函数最大化的采样点为下一步的试验点.

$$\mathbf{x}_t = \mathrm{argmax}_{\mathbf{x}} u(\mathbf{x}|\mathcal{D}_{1:t-1}). \tag{19}$$

优化算法 (Optimization algorithm) 贝叶斯优化算法过程如下.

For $t = 1, 2, ...$ 重复执行:

- 通过代理模型 (GP) 求得获取函数 (AF)
- 通过获取函 (AF) 数找到下一个采样点 $\mathbf{x}_t = \mathrm{argmax}_{\mathbf{x}} u(\mathbf{x}|\mathcal{D}_{1:t-1})$
- 对目标函数进行求值 $y_t = f(\mathbf{x}_t) + \epsilon_t$.
- 将样本 $(\mathbf{x}_t, y_t)$ 加入先验样本 $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (\mathbf{x}_t, y_t)\}$
- 更新代理模型 (GP)

预期提升 (Expected improvement) 定义为：

$$\mathrm{EI}(\mathbf{x}) = \mathbb{E} \max(f(\mathbf{x}) - f(\mathbf{x}^+), 0). \tag{20}$$

其中 $f(\mathbf{x}^+)$ 是目前最优的目标函数值 $\mathbf{x}^+$ 是最优解 $\mathbf{x}^+ = \mathrm{argmax}_{\mathbf{x}_i \in \mathbf{x}_{1:t}} f(\mathbf{x}_i)$.

预期提升可以在 GP 模型下求得解析解：

$$\mathrm{EI}(\mathbf{x})=\begin{cases}(\mu(\mathbf{x})-f(\mathbf{x}^{+})-\xi)\Phi(Z)+\sigma(\mathbf{x})\phi(Z), & \text{if } \sigma(\mathbf{x})>0.\\ 0, & \text{if } \sigma(\mathbf{x})=0.\end{cases} \tag{21}$$

其中：

$$Z=\begin{cases}\frac{\mu(\mathbf{x})-f(\mathbf{x}^{+})-\xi}{\sigma(\mathbf{x})}, & \text{if } \sigma(\mathbf{x})>0.\\ 0, & \text{if } \sigma(\mathbf{x})=0.\end{cases} \tag{22}$$

式22中的 ξ 确定优化期间的探究量，较高的 ξ 值将导致更强的探索.ξ 的推荐值是 0.01.

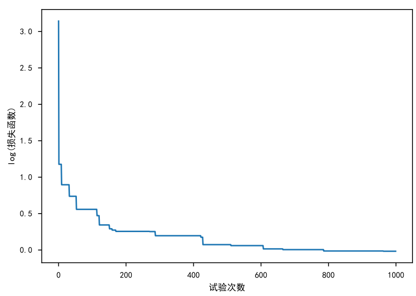

图 **4**　安徽省微分方程参数调优损失函数

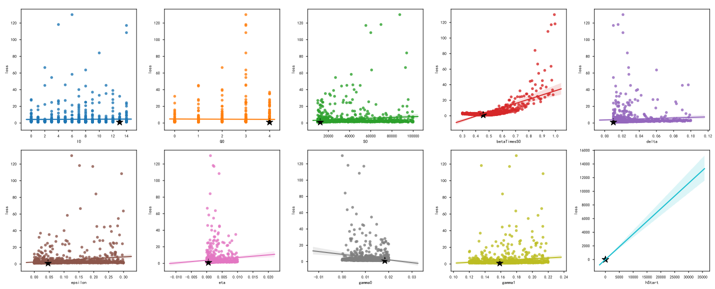

图 **5**　安徽省微分方程参数调优试验

4.4 结果分析

4.4.1 最佳参数

贝叶斯参数优化器在参数空间中进行 1000 轮参数搜索试验后，为安徽省和海峡群岛的疫情传播情况找到的最优参数见表1. 其中参数搜索范围参考文献[2].

表 **1**　微分方程模型参数

	海峡群岛 (英国)	安徽省	搜索范围
初始易感人数 S_0	26380.72	12338.98	$10^4 \sim 10^5$
βS_0	0.734773	0.456662	$0.3 \sim 1.0$
δ	0.010046	0.010046	$0.01 \sim 0.1$
ϵ	7.84^{-05}	0.046334	$0 \sim 0.3$
η	0.003384	0.000546	$0 \sim 0.01$
γ_0	0.016848	0.018348	$0 \sim 0.02$
$\gamma 1$	0.166284	0.158608	$0.12 \sim 0.22$
开始隔离政策天数	58	12	$0 \sim 150$
k	0.692089	0.696911	$0.6 \sim 0.7$
λ	0.17209	0.201207	$0.15 \sim 0.21$
开始医疗升级天数	87	34	$0 \sim 150$
结束医疗升级天数	84	19	$0 \sim 150$
σ	0.077206	0.293288	$0 \sim 0.35$
θ	0.122481	0.162654	$0.1 \sim 0.17$

4.4.2 结果展示

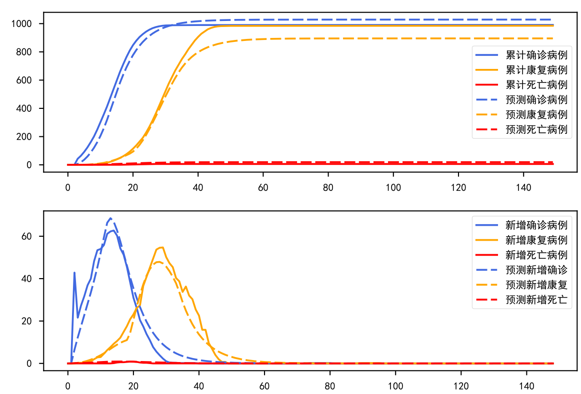

图 **6**　安徽省真实数据及模型预测

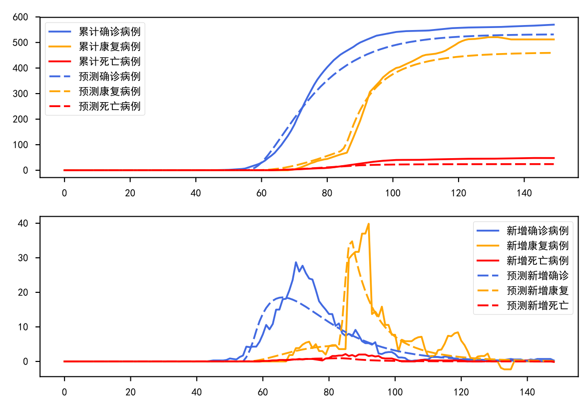

图 **7**　海峡群岛 (英国) 真实数据及模型预测

可以看到安徽省（图6）在 19 天左右，海峡群岛（图7）在 84 天左右，新增治愈人数陡增，这符合解决医疗挤兑的传染病动力学模型假设，即：疾病爆发初期，治愈率 γ_0 较低，出现医疗资源挤兑的情况. 然后由于相应的医院扩容，物资补给政策，我们假设医院治疗能力增加，直到保持在一个较高的水平 γ_1.

4.4.3 灵敏度分析

除了医疗资源的变化，政府的隔离政策也对疫情的传播有重大影响，当疫情传播开始之后，何时执行居家令（stay-at-home）和就地避难令（shelter-in-place）对确诊人数，死亡人数也有重大影响. 我们对安徽省（图8）和海峡群岛（图9）关于隔离开始时间进行灵敏度分析，分别在模型搜索到最的近似实际考试时间的基础上，分别提前和延迟 5 天，然后观察新增确诊人数和死亡人数的曲线. 我们发现对于安徽省，每提前执行隔离措施 5 天，新增确诊和死亡率的峰值都下降到 50% 左右；对于海峡群岛，每提前执行隔离措施 5 天，新增确诊和死亡率的峰值都下降到 25% 左右.

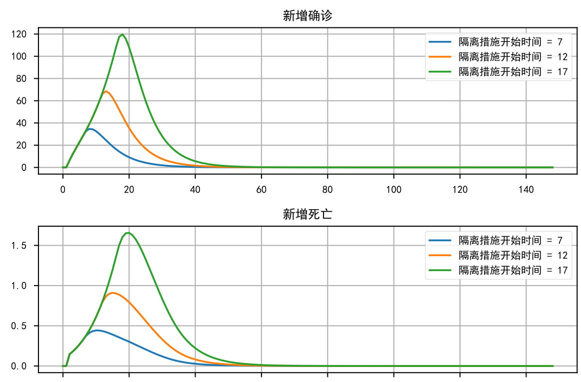

图 **8**　安徽省隔离开始时间灵敏度分析

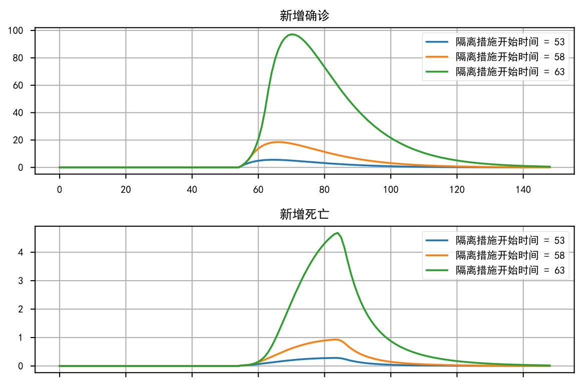

图 **9**　海峡群岛 (英国) 隔离开始时间灵敏度分析

五、 问题二模型的建立与求解

5.1 问题分析

新冠病毒大流行对世界经济造成了巨大的影响，人们因隔离而无法工作，国家的援助也不能保证市场按原有计划运行，国家经济会在消费，投资，进出口等方面遭遇大幅衰退. 为此，我们选择对反映经济体消费能力的社会消费品零售总额进行分析，通过构建灰色预测模型和时间序列模型预测社会消费品零售总额的正常增长，引入消费者信心这一概念，研究在疫情背景下，消费品零售总额与消费者信心的函数关系，消费者信心用居民消费价格指数的变化来表示. 问题二的思维导图如图10.

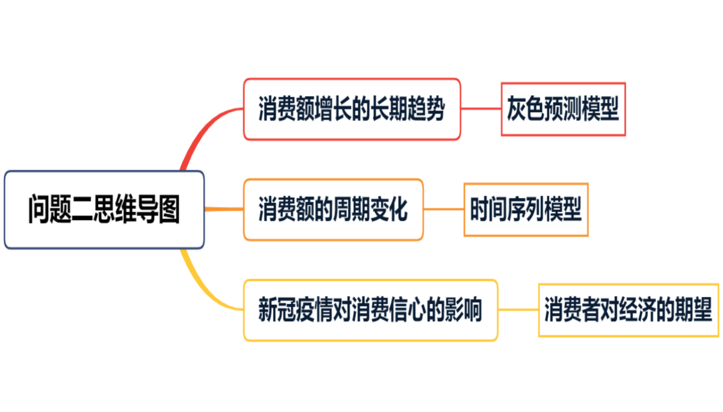

图 **10**　问题二思维导图

5.2 社会消费品零售总额预测模型

自 2020 年 1 月底新冠病毒大流行爆发以来，由于一系列防控措施所导致的经济封锁，已经在消费，投资与进出口等方方面面影响了社会的经济运转，消费是现代经济社会的支柱，贡献了中国近 4 成的 GDP，消费端因疫情封锁受到的打击会直接影响家庭消费与企业运转，更重要的是会降低民众对经济体的信心，为此我们收集了自 2012 年以来社会消费品销售总额的数据，对该数据做无疫情影响下的预测分析，然后与受新冠病毒大流行影响的真实数据做对比，以判断消费受到疫情影响的程度.

在中国经济长期增长的背景下，我们认为社会销售品零售额已经由了长期增长的趋势，但作为一般销售品，有其消费增长的周期性规律，这主要体现在不同月份人们消费的需求与喜好的不同. 我们认为消费品销售额度 w 由两方面组成，一方面体现在消费品长期增长的趋势 Y，另一方面体现在消费品受季节影响而导致的波动 T, 我们按照年度及月份从小到大排列，从 2015 年一月开始编号,$k=1,2,....66$.

$$W(k)=Y(k)+T(k). \tag{23}$$

其中 $Y(k)$ 反映了销售额长期增长的确定性趋势，$T(K)$ 反映了销售额的季节性随机变化趋势.

5.2.1 销售额长期增长确定性趋势的灰色预测

从长远来看，我们的销售额增长趋势整体上是一种稳步上升的趋势，所以本文对于整个销售额的这种确定性增长趋势做灰度预测，本文建立了预测商品销售额长期稳定趋势的一阶线性微分方程 GM(1,1). 我们用 $Y(k)$ 反映了销售额长期增长的确定性趋势.

累加以弱化波动性和随机性:

$$Y^{(1)}=(Y^{(1)}(1),Y^{(1)}(2),...,Y^{(1)}(n)). \tag{24}$$

其中，n 为数据个数.

$$Y^{(1)}(t)=\sum_{k=1}^{t}Y^{0}(k),t=1,2,....n. \tag{25}$$

生成均值序列:

$$z^{(1)}(k)=\alpha Y^{(1)}(k)+(1-\alpha)Y^{(1)}(k-1) \qquad k=2,3,...n. \tag{26}$$

白化微分方程:

$$\frac{\mathrm{d}Y^{(1)}}{\mathrm{d}t}+aY^{(1)}=b. \tag{27}$$

其中 a 为发展系数，b 为灰色作用量，a 的有效区间是 (-2,2), 记 a,b 构成的矩阵为:

$$\beta=\begin{pmatrix} a \\ b \\ . \end{pmatrix} \tag{28}$$

最小二乘法求解参数:

$$\hat{\beta}=(a,b)^{T}=\left(B^{T}B\right)^{-1}B^{T}Y_{n}. \tag{29}$$

GM(1,1) 离散解:

$$\hat{Y}^{(1)}(k)=[Y^{(0)}(1)-\frac{b}{a}]e^{-a(k-1)}+\frac{b}{a}, \qquad k=2,3,4...n. \tag{30}$$

原始序列预测模型:

$$\hat{Y}^{(0)}(k)=Y^{(1)}(k)-Y^{(1)}(k-1), \qquad k=2,3,4...n. \tag{31}$$

$$\hat{Y}^{(0)}(k)=[Y^{(0)}(1)-\frac{b}{a}]e^{-a(k-1)}(1-e^{a}), \qquad k=2,3,4...n. \tag{32}$$

5.2.2 销售额的季节性随机变化趋势的时间序列 AR 模型分析

$T(k)$ 表示数据序列的随机变化趋势因素，通过对 2015 到 2020 年的数据进行观察我们可以看出来，销售额增长的过程总是以 12 个月为周期出现相同的变化趋势，于是我们将 $Z(k)=X(k)-Y(k)$ 的误差变化作为我们销售额的季节变化波动影响.

提取周期项

$$L(k)=\begin{cases}0, & k<13. \\ Z(k)-Z(k-12), & 13<k<60.\end{cases} \tag{33}$$

单位根检验

我们在使用时间序列之前需要对时间序列的平稳性进行检验，这里采用的 ADF 检验方法，能够有效判断该序列是否平稳.

$$\Delta Z_{t}=\sigma Z_{t}-1+\sum_{i=1}^{12}\beta_{i}\Delta Z_{t-i}+\epsilon_{t}. \tag{34}$$

零假设 H0：σ=0，原序列存在单位根，为非平稳序列.
备选假设 H1：σ<0，原序列不存在单位根，为平稳序列.

样本自相关系数

样本自相关系数的确定首要条件是该序列为平稳性序列，然后利用协方差的相关性的计算公式，并根据数据代入进行运算.$\hat{Z}_t$ 表示的是 s 周期内的序列 Z_{t-s} 到 Z_t 的均值

$$\hat{\rho} = \frac{cov(Z_t, Z_{t-12})}{\sqrt{Var(Z_t)}\sqrt{Var(Z_t - 12)}} = \frac{\sum_{t=s+1}^{T}(Z_t - \overline{Z})(Z_{t-s} - \overline{Z})}{\sum_{t=1}^{T}(Z_t - \overline{Z})^2}. \tag{35}$$

样本偏相关系数

样本偏自相关系数的确定首要条件也是该序列为平稳性序列，然后利用协方差的相关性的计算公式根据数据代入进行运算. 最后得到一个关于 $\hat{\phi}$ 的序列.

$$P_k = \frac{cov[(Z_t, \hat{Z}_t), (Z_{t+k} - \hat{Z}_{t+k})]}{\sqrt{Var(Z_t - \hat{Z}_t)}\sqrt{Var(Z_{t+k}, \hat{Z}_{t+k})}}, \tag{36}$$

$$\hat{\phi}_{k+1,k+1} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^{k}\hat{\phi}_{kj}\hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^{k}\hat{\phi}_{kj}\hat{\rho}_{k+1-j}}. \tag{37}$$

随机波动趋势预测 AR 序列模型

根据自相关系数和偏自相关系数的图像，我们可以看到该模型的 ACF 自相关系数是拖尾形状，而偏自相关系数是截尾情况，从而确定该模型的阶数 $\rho = 2$ 从而得出该模型的一种表达式情况. 其中 $\epsilon \sim N(0, \sigma^2)$

$$Z_t = \sum_{i=1}^{\rho}\alpha_i Z_{t-i} + \epsilon. \tag{38}$$

5.2.3 消费者缩减支出幅度的分析

新冠疫情造成社会经济形势在各方面的恶化，极大打击了消费者的信心，消费者对经济形势保持悲观的情况下，会减少支出以为将来可能的花销做准备. 要分析新冠疫情对经济消费端的打击，我们就必须对消费者信心做出预测.

在一个长期增长且稳定的经济体中，消费者价格指数常常保持稳定，因为市场对经济增长有统一却明确的信心，在遭到新冠疫情袭击后，消费者为了抢购物资而推高的消费价格指数，事实上导致了消费品消费额的总体下降，因为售出货品的数量减少的速率大于价格提高的速率. 需要指出的是，一次经济危机常导致消费者价格指数先高于基准预期，后低于基准预期，最后走向稳定. 消费者对经济恶化预期到达某一阈值 F 后，才会做出减少消费的决定，且由于必需品支出无法压缩，消费者压缩支出的幅度 P 不会超出必需品占消费的百分比, 此时消费者悲观程度以达到峰值. 消费者对于压缩支出的欲望是边际递减的:

$$P = \begin{cases} 0, & D \le F. \\ k_1 lnD + b_1, & F < D <= D_{max}. \end{cases} \tag{39}$$

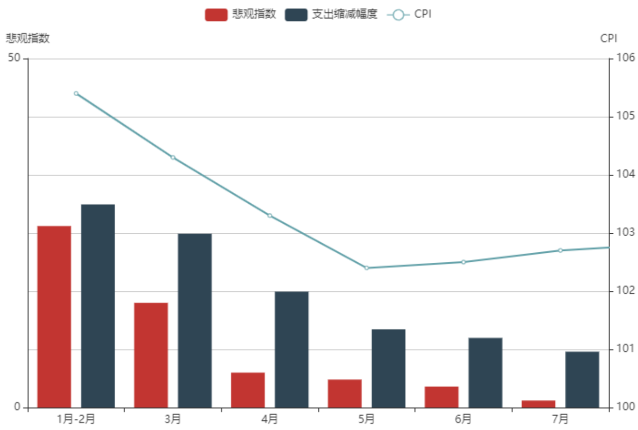

图 **11　2020** 年以来悲观指数**,** 缩减支出幅度和 **CPI** 变化情况

D 表示消费者对经济预期的悲观程度，是消费者信心的反映.D_{max} 表示悲观程度的峰值,k_1,b_1 是常数系数, 在经济危机期间，消费者信心是趋于悲观的，并以当期 CPI 值 C_i 与 CPI 基准值 C_{ave} 之差的绝对值表示.

$$D_i = |C_i - C_{ave}|. \tag{40}$$

5.3 模型求解

5.3.1 确定性销售额增长 GM(1,1) 求解

step1 我们将 2015-2019 年的每月份销售额原始数据矩阵记为 $A_{5\times12}$, 代表这 5 年的 60 个数据. 从而计算出他们的平均值 $Y_i^{(0)} = \frac{1}{12}\sum_{j=1}^{12} a_i j, i = 1, 2..5$. 在生成均值序列是，我们的参数 $\alpha = 0.5$

step2 在白化微分方程的迭代下，通过最小二乘法的计算求得 $\beta = (a, b)^T$, 从而求得发展系数 $a = -0.0061095, b = 24666.8866$ 从而得到该灰色模型的离散解

$$\hat{Y}^{(1)}(k) = 4059060.67e^{0.0061095(k-1)} - 4037464.05, \qquad k = 2, 3, 4...n. \tag{41}$$

然后取 k=6,从而估算出在灰度预测下的确定性增长幅度的均值预测 $\hat{Y}^{(0)}(6) = 37586.9425$

step3 在求出原始数列的预测模型后，我们利用前 5 年的各月销售额数据，计算出每月销售额所占的比例 $(r_1, r_2, ...r_{12})$

$$r_j = \frac{\sum_{i=1}^{5} a_{ij}}{\sum_{i=1}^{5}\sum_{j=1}^{1} 2a_{ij}}. \tag{42}$$

从而得到各月销售额确定性的分配比例 $r = (0.0807, 0.0814, 0.0818, 0.0821, 0.0825, 0.0830, 0.0834, 0.0839, 0.0843, 0.0850, 0.0856, 0.0864)$ 最终得到我们的确定性增长部分的增长数据 Y(k)

表 **2** 销售额确定性增长部分的销售额灰度预测结果

月份	销售额预测	月份	销售额预测	月份	销售额预测
1	36390.14264	5	37233.4705	9	38027.79963
2	36696.2505	6	37431.51006	10	38332.88239
3	36875.2686	7	37622.22469	11	38629.19532
4	37029.70192	8	37822.66108	12	38952.20202

5.3.2 周期性随机波动的时间序列求解

step1 平稳性检验：将灰度预测的模型对其原始数据进行残差波动计算，得到 T(t) 通过对原始序列的趋势观察和 ADF 检验我们得出了，利用 matlab 的 adftest 函数可以得出**该模型 adf=1, 属于平稳性时间序列.**

step2 定阶：在该模型为平稳性序列的基础上，使用 autocorr 和 parcorr 函数，进一步得出了该模型的自相关系数和偏自相关系数的图像，根据图像信息可知该模型的自相关系数呈拖尾状，而偏自相关系数呈截尾状，则**该模型确定为时间序列的 AR 模型，且他的阶数** $\rho = 2$

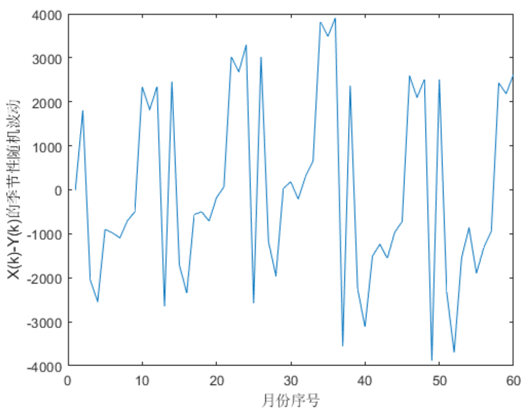

图 **12** **2015-2019** 年的随机波动

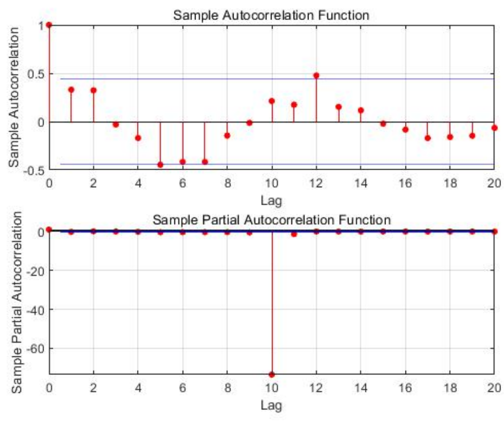

图 **13** 自相关系数和偏自相关系数图像

step3 参数求解：在确定好我们的时间序列模型为 AR 模型且 $\rho = 2$ 后，我们可以对该时间序列进行拟合和求解，从而计算得出 $\alpha_1 = -0.6069\alpha_2 = -0.38018$ 从而得到未来个月内的波动情况.

最终得到我们的周期性随机波动的增长数据 T(k)

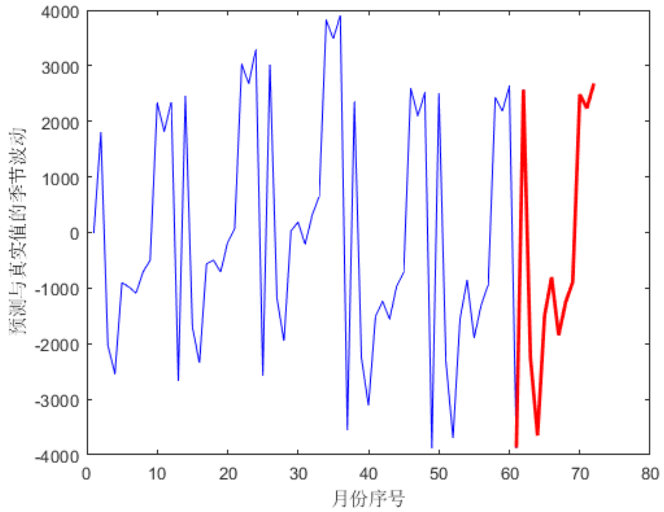

图 **14** **2020** 年的波动预测

表 **3** 销售额确定性增长部分的销售额灰度预测结果

月份	销售额波动预测	月份	销售额波动预测	月份	销售额波动预测
1	-3871.939676	5	-1497.852934	9	-887.4429241
2	2567.104346	6	-802.965961	10	2484.516924
3	-2259.204392	7	-1845.352286	11	2237.164318
4	-3652.111398	8	-1252.244212	12	2681.111115

5.3.3 悲观缩减支出与悲观程度的拟合

我们以 2015 年至 2019 年 CPI 的基准值 102.8 为判断经济复苏的依据, 数据来源于国家统计局, 东方财富网等相关网站，在大流行期间，根据时间序列模型拟合的消费者预期消费增长率与现实增长率的差值 P_i，以及每个月消费者的悲观程度 D_i, 我们依据 2020 年 1 月到 5 月的数据进行拟合，发现 P 与 D 满足以下函数关系:

$$P = 9.0627ln(D) + 21.102. \tag{43}$$

用 2020 年 6 月份数据进行检验，2020 年 6 月消费者对支出缩减的预估是 9.97%，根据时间序列模型求出的该月 8.1% 的增长，我们得到 2020 年 6 月预计增长率为-1.87%, 与实际增长率-1.8% 十分接近，证明我们的预测效果较好，根据 2020 年 7 月消费者悲观程度计算，我们得到消费者的实际支出会增加 0.4%, 这说明经济已经步入复苏阶段，2020 年 7 月社会消费品零售总额预估为 33205 亿元.

5.4 结果分析

5.4.1 结果展示

经过上述 3 个方面得模型预测影响，最终我们得到了我国在当前疫情影响下得结果，经过对疫情和时间序列得模拟我们可以得出，当前疫情形势下，我国销售额变化呈现出了先迅速降低，然后随着疫情的慢慢褪去以及消费者信心指数的提升，我们可以看出来当前疫情形势下在 7 月时达到了同年的相同增长情况，借此我们可以认为从 2020 年 7 月开始，我国的销售额恢复达到了 2019 年的水平，且各项指标都将回复成时间序列下的常规状态，借此我们可以利用波动规律对之后的 2019 年数据进行有效预测, 得到下半年各月之间的销售额分别为 33525.9,33205,34028,34626.6,38225.68,38908.4 亿元.

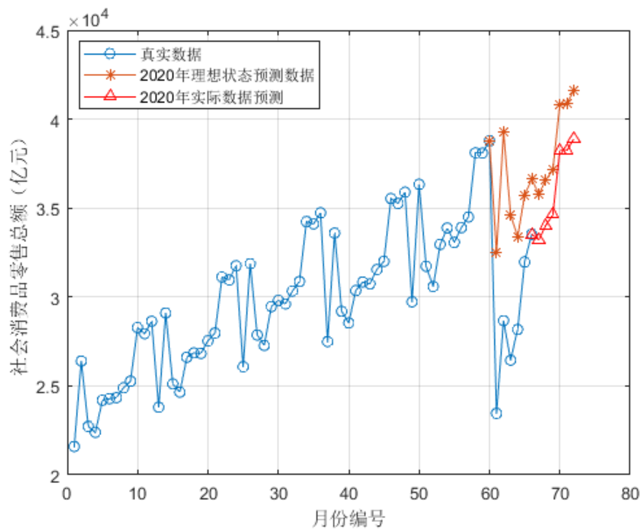

图 **15　2020** 年的销售额实际预测与理想预测的对比

表 **4　2020** 年的下半年销售额预测情况

月份	预测销售额	月份	预测销售额
7	33525.9	10	34626.6
8	33205	11	38225.68
9	34028	12	38908.4

5.4.2 灰度预测模型检验

我们的灰度预测拟合度具有准指数的性质，光滑比小于 0.5 的数据占比为 98.3051% 除去前两个时期外，光滑比小于 0.5 的数据占比为 100%, 平均相对残差为 0.061574, 平

均级比偏差为 0.066514 表明其拟合效果很好，因此在确定性趋势下的

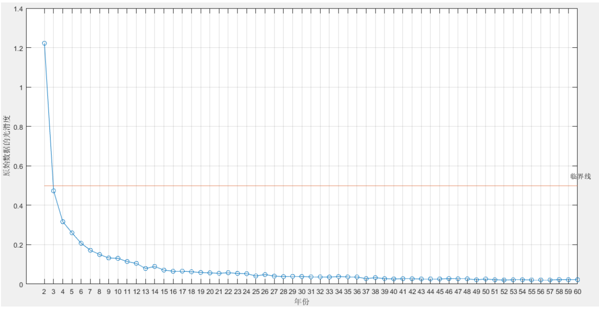

图 **16**　灰度预测各个平滑度

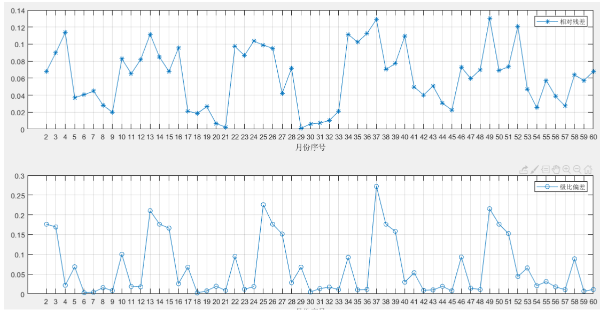

图 **17**　灰度预测相对残差和级比残差

5.4.3 自回归残差检验

自回归标准残差，基本在 0 的附近，而他的 QQ 图残差基本完全落在 45° 线上，表明确实是符合其正态分布的假设. 且 AR(2) 的定阶对应的 BIC 值是最小的.

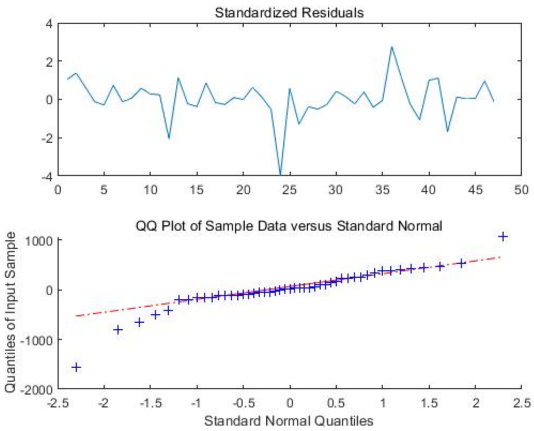

图 **18**　标准化残差与 **QQ** 图正态检验

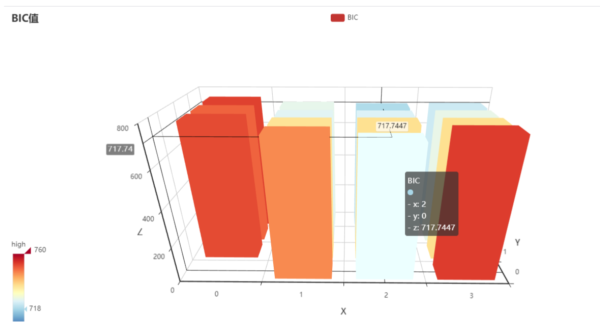

图 **19**　定阶对应的 **BIC** 值

六、 模型评价

6.1 模型的优点

1. 在建立改进的 SEIR 模型是，考虑了愿意自我隔离的人的比例随疫情发展情况的变化，通过对病毒传播的机理分析，我们依据隔离与治疗与否，将感染者细分成三类，考虑到了患者送医时间，隔离比例对病毒传播的影响，对传播过程模拟的仿真度高.

2. 问题二模型考虑了消费增长的趋势性与周期性因素，研究疫情对经济的短期影响时，引入了消费者悲观程度，并作出了经济学解释, 模型与真实经济发展状况契合程度较高.

6.2 模型的缺点

1. 由于缺少数据，无法对感染者进行进一步分类，感染者按照病情发展程度，其治愈率，死亡率，传播病毒能力都是不同的，如果有更详细的数据，我们可以将模型进一步完善.

2. 由于缺少资料，未就疫情后经济复苏的潜力进行量化.

6.3 改进与展望

已有不少学者应有对传统 SEIR 模型提出改进，魏等[2] 提出了对已发病感染者按照轻症，中度和重症进行分类，使用马尔科夫链蒙特卡洛算法结合 Gibbs 抽样的方法得到了更准确的参数，模型拟合效果较理想. 在模型建立过程中，没有考虑到人口迁入迁出的影响，晏等[7] 对新冠肺炎感染人群做了人口学分析，可以参考他们的结论进一步完善模型,.

七、 致国家卫生部门的一封信

尊敬的卫生部门领导:

您好.

新冠病毒大流行已经造成了全球范围内严重的公共卫生危机, 各国已经为战胜危机付出了巨大努力，各国应对新冠病毒的措施包括但不限于发布居家令, 建立发热门诊，建设专门应对病毒的临时医院，强制佩戴口罩，研发特效药与疫苗. 目前，全球范围内的病毒传播还处于扩散阶段，这主要是因为部分国家防控意识薄弱，医疗水平落后，且国际援助水平有限. 为了更快且彻底的摆脱新冠病毒危机，我们对新冠病毒的传播过程进行建模并分析，根据所得结果向贵部门提出相应建议:

1. 立刻出来相应法规隔离养老院等老年人集聚地并提供帮助，全球新冠肺炎患者中，60 岁及以上年龄患者占一半以上, 加强对老年人的保护能够有效切断病毒在人际间

的传播.

2. 加强医疗卫生投入，对已发病或者潜伏期但已检测为阳性的患者应收尽收，我们发现提高收治效率能够显著降低新增感染人数.

3. 尽快发布居家隔离令，强制居民在家隔离, 可以有效降低前期感染人数，尽早压平感染曲线.

4. 展开大规模的病毒筛查，建立分级诊断机制，将人群中的潜伏期患者找出，避免造成易感人群的规模性的群体感染. 只有找出所有潜伏期患者，新冠病毒传播的源头才能被阻隔.

5. 传统的防控手段并不能使全部人口免于病毒感染，应加快对该病毒特效药和疫苗的研发，易感染人群越早完成疫苗的接种，病毒就能更早的被全面控制.

希望我们做的调查和预测能为您提供一些有效信息，为抗击疫情献上我们的绵薄之力，同时我们衷心祝愿我国能够早日战胜新冠疫情，恢复正常生活，加快全面小康的进程.

参考文献

[1] 陈柯宇. 基于 SEIR 模型的新型冠状病毒疫情实证分析 [J]. 统计与管理,2020,35(06):31-38.

[2] 魏永越, 卢珍珍, 杜志成, 张志杰, 赵杨, 沈思鹏, 王波, 郝元涛, 陈峰. 基于改进的 SEIR+CAQ 传染病动力学模型进行新型冠状病毒肺炎疫情趋势分析 [J]. 中华流行病学杂志, 2020, 41(4): 470-475.

[3] 徐致靖. 复杂社会系统中的传染病动力学建模与案例研究 [D]. 中国人民解放军军事医学科学院,2015:34-25.

[4] 颉俊慧. 两类传染病动力学建模与分析 [D]. 山西大学,2017:7-11.

[5] Arapović Jurica,Skočibušić Siniša. The first two months of the COVID-19 pandemic in Bosnia and Herzegovina: Single-center experience.[J]. Bosnian journal of basic medical sciences,2020,20(3).

[6] Carlos Victor Chaves de Lima,Estelita Lima Cândido,José Arinelson da Silva,Letícia Viana Albuquerque,Lívia de Menezes Soares,Mayara Maciel do Nascimento,Sara Alves de Oliveira,Modesto Leite Rolim Neto. Effects of quarantine on mental health of populations affected by Covid-19[J]. Journal of Affective Disorders,2020,275.

[7] 晏月平, 李忠骥. 新冠肺炎感染人群的人口学分析——以我国 10 省市区为例 [J]. 人口与发展,2020,26(03):73-85.

附录 A 代码

A.1 python 源程序

```python
import paramater_tuning
import solve_scipy
import read_data
import plot
from common import *

def get_bestParameters():
    # anhui = paramater_tuning.MedicalUpgrade("Anhui", 1000, 2, 12, 19, 34)
    # anhui.get_best_parameters_for_medical_upgrade()
    CI = paramater_tuning.MedicalUpgrade("Channel Islands", 1000, 55, 58, 84, 87)
    CI.get_best_parameters_for_medical_upgrade()


if __name__ == '__main__':
    get_bestParameters()
```

```python

import pandas as pd
import numpy as np
import matplotlib
matplotlib.rcParams["font.sans-serif"] = ["SimHei"]
matplotlib.rcParams['axes.unicode_minus'] = False
import matplotlib.pyplot as plt
from matplotlib import style
style.use('seaborn-paper')

WINDOW = 7
DAYS = 150

bestParametersDict = {
    "Channel Islands": {'I0': 2, 'Q0': 3, 'S0': 26380.717689599554, 'betaTimesS0':
        0.7347729066021397, 'delta': 0.010045853476576706, 'epsilon': 7.840097082522992e-05,
        'eta': 0.003383864399097301, 'gamma0': 0.016848020203597488, 'gamma1':
        0.16628438395216183, 'hStart': 58, 'k': 0.6920892202047502, 'lamb':
        0.17208981925301345, 'mEnd': 87, 'mStart': 84, 'sigma': 0.07720585767550427, 'start':
        55, 'theta': 0.12248114734032356},
    "Anhui":         {'I0': 13, 'Q0': 4, 'S0': 12338.977128653125, 'betaTimesS0':
        0.45666217056059893, 'delta': 0.010045504375222879, 'epsilon': 0.04633446975602694,
        'eta': 0.000545652721259959, 'gamma0': 0.018348160210604064, 'gamma1':
        0.15860752405127965, 'hStart': 12, 'k': 0.6969110600952818, 'lamb': 0.2012066716037268,
        'mEnd': 34, 'mStart': 19, 'sigma': 0.2932882661742646, 'start': 2, 'theta':
        0.1626536380573757}
```

```python
}


def save_para_excel():
    df = pd.DataFrame(bestParametersDict["Channel Islands"], index=["海峡群岛(英国)"])
    df = df.append(pd.DataFrame(bestParametersDict["Anhui"], index=["安徽省"]))
    df.T.to_excel("../cache/para.xlsx")
    print(df.T)


if __name__ == '__main__':
    save_para_excel()
```

```python
from hyperopt import fmin, tpe, hp, Trials, STATUS_OK
import read_data
from solve_scipy import *
import plot
from common import *


def get_medical_upgrade_solution(p):
    # u0 = (p["S0"], 109, 10, 33, 5, 0, 0, 0)
    u0 = (p["S0"], 109, p["I0"], p["Q0"], 0, 0, 0, 0)
    parameters = (p["betaTimesS0"] / p["S0"], p["start"], p["hStart"], p["k"], p["epsilon"],
        p["lamb"],
              p["theta"], p["delta"], p["sigma"], p["gamma0"], p["gamma1"],
              p["mStart"], p["mEnd"], p["eta"])
    t = np.arange(0, DAYS, 1)
    seir_solution = odeint(f2, u0, t, args=parameters)
    return seir_solution[:DAYS, -3:]


class MedicalUpgrade():
    def __init__(self, province, max_evals, start=None, hStart=None, mStart=None, mEnd=None):
        self.DATA_JRD = np.array(read_data.read_cleaned_data(province)[
                         ["confirmedMA", "recoveriesMA", "deathsMA"]][:DAYS].fillna(0))

        self.DATA_JRD_DIFF = np.diff(self.DATA_JRD, axis=0)
        self.max_evals = max_evals
        self.province = province

        self.space = {
            "S0": hp.loguniform("S0", np.log(1e4), np.log(1e5)),
            "betaTimesS0": hp.uniform("betaTimesS0", 0.3, 1),
            "I0": hp.choice("I0", range(15)),
            "Q0": hp.choice("Q0", range(5)),
```

```python
        "start": start if start is not None else hp.choice("start", range(40, 60)), #
            疫情开始传播
        "hStart": hStart if hStart is not None else hp.choice("hStart", range(45, 60)), #
            开始实行隔离政策的天数
        "mStart": mStart if mStart is not None else hp.choice("mStart", range(80, 85)), #
            开始医疗条件升级的天数
        "mEnd": mEnd if mEnd is not None else hp.choice("mEnd", range(85, 90)), #
            开始医疗条件升级的天数
        "k": hp.uniform("k", 0.6, 0.7), # 潜伏期患者与已发病患者传染强度的比率
        "epsilon": hp.uniform("epsilon", 0, 0.3), # 潜伏期感染者转入未隔离感染者的比例
        "lamb": hp.uniform("lamb", 0.15, 0.21), # 潜伏期感染者转入已隔离感染者的比例

        "theta": hp.uniform("theta", 0.1, 0.17), # 未隔离感人者转入正在治疗的人群的比例
        "sigma": hp.uniform("sigma", 0., 0.35), #
            医疗资源扩充前，已隔离感人者转入正在治疗的人群的比例
        "gamma0": hp.uniform("gamma0", 0, 0.02), # 医疗资源扩充后，正在治疗的已感染者治愈的比例
        "gamma1": hp.uniform("gamma1", 0.12, 0.22), # 正在治疗的已感染者治愈的比例
        "delta": hp.uniform("delta", 0.01, 0.1), # 未隔离的已感染者的死亡率
        "eta": hp.uniform("eta", 0, 0.01), # 接受治疗的病人的死亡率
    }




def opt_medical_upgrade(self, max_evals):
    """
    调用hyperopt库的API接口在搜索空间中确定最佳超参数优化总距离（成本）
    :param max_evals:
    :return:
    """
    def medical_upgrade_objective(p):
        """
        模型超参数搜索优化总距离（成本）
        :param p:
        :return:
        """
        seir_solution = get_medical_upgrade_solution(p)
        # loss = (np.abs(DATA_JRD - seir_solution)).mean()
        loss = np.abs(np.diff(seir_solution, axis=0) - self.DATA_JRD_DIFF).mean()
        # loss = ((np.diff(seir_solution, axis=0)-DATA_JRD_DIFF)**2).mean()
        result = {"total_loss": loss}
        return {'loss': result["total_loss"], "result": result, 'parameters': p, 'status':
            STATUS_OK}
    trials = Trials()
    best = fmin(fn=medical_upgrade_objective, space=self.space, algo=tpe.suggest,
                max_evals=max_evals, trials=trials)
```

```python
        return best, trials


    def get_best_parameters_for_medical_upgrade(self):
        """
        封装opt_journey解析参数搜索结果
        :param max_evals:
        :return:
        """
        best, trials = self.opt_medical_upgrade(self.max_evals)
        # Sort the trials with lowest loss first
        trials_list = sorted(trials.results, key=lambda x: x['loss'])
        bestParameters = trials_list[0]['parameters']
        result = trials_list[0]["result"]
        # print(trials_list[0])
        print("best parameter found after {} trials".format(len(trials_list)))
        print(bestParameters)
        print("lowest loss is")
        print(result)
        plot.plot_trials(trials_list, name=self.province)
        loss_step = [step["loss"] for step in trials.results]
        plot.plot_loss_time(loss_step, name=self.province)

        seir_solution = get_medical_upgrade_solution(bestParameters)
        plot.compare_result_with_data(seir_solution, self.DATA_JRD, province=self.province)
        test_SEIR(bestParameters, name=self.province)
        return bestParameters
```

```python
from common import *
import plot


def read_data(province='Hubei'):
    confirmed_df = pd.read_csv(
        '../input/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv')
    confirmed_df = confirmed_df[confirmed_df['Province/State'] == province]
    deaths_df = pd.read_csv(
        '../input/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv')
    deaths_df = deaths_df[deaths_df['Province/State'] == province]
    recoveries_df = pd.read_csv(
        '../input/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_recovered_global.csv')
    recoveries_df = recoveries_df[recoveries_df['Province/State'] == province]

    provinceDF = pd.DataFrame(
        index=confirmed_df.columns,
        columns=["confirmed", "deaths", "recoveries"])
```

```python
    provinceDF["confirmed"] = confirmed_df.T
    provinceDF["deaths"] = deaths_df.T
    provinceDF["recoveries"] = recoveries_df.T

    provinceDF = provinceDF[4:]
    #provinceDF.index = pd.date_range("20200122", "20200720")

    provinceDF["confirmedMA"] = provinceDF["confirmed"].rolling(window=WINDOW,
        center=True).mean()
    provinceDF["deathsMA"] = provinceDF["deaths"].rolling(window=WINDOW, center=True).mean()
    provinceDF["recoveriesMA"] = provinceDF["recoveries"].rolling(window=WINDOW,
        center=True).mean()

    provinceDF["confirmedDaily"] = provinceDF["confirmed"].diff()
    provinceDF["deathsDaily"] = provinceDF["deaths"].diff()
    provinceDF["recoveriesDaily"] = provinceDF["recoveries"].diff()

    provinceDF["confirmedDailyMA"] = provinceDF["confirmedDaily"].rolling(window=WINDOW,
        center=True).mean()
    provinceDF["deathsDailyMA"] = provinceDF["deathsDaily"].rolling(window=WINDOW,
        center=True).mean()
    provinceDF["recoveriesDailyMA"] = provinceDF["recoveriesDaily"].rolling(window=WINDOW,
        center=True).mean()
    plot.plot_provinceDF(provinceDF, province)
    return provinceDF.plot


def read_cleaned_data(province='Hubei'):
    try:
        return pd.read_excel("../cache/{}.xlsx".format(province))
    except FileNotFoundError:
        print("Preprocessing data..")
    df = pd.read_csv(
        '../input/covid_19_clean_complete.csv')
    provinceDF = df[df['Province/State'] == province]
    provinceDF.index = pd.date_range("20200122", "20200727", freq='1D')

    provinceDF["confirmedMA"] = provinceDF["confirmed"].rolling(window=WINDOW,
        center=True).mean()
    provinceDF["deathsMA"] = provinceDF["deaths"].rolling(window=WINDOW, center=True).mean()
    provinceDF["recoveriesMA"] = provinceDF["recoveries"].rolling(window=WINDOW,
        center=True).mean()

    provinceDF["confirmedDaily"] = provinceDF["confirmed"].diff()
    provinceDF["deathsDaily"] = provinceDF["deaths"].diff()
    provinceDF["recoveriesDaily"] = provinceDF["recoveries"].diff()
```

```
    provinceDF["confirmedDailyMA"] = provinceDF["confirmedDaily"].rolling(window=WINDOW,
        center=True).mean()
    provinceDF["deathsDailyMA"] = provinceDF["deathsDaily"].rolling(window=WINDOW,
        center=True).mean()
    provinceDF["recoveriesDailyMA"] = provinceDF["recoveriesDaily"].rolling(window=WINDOW,
        center=True).mean()
    plot.plot_provinceDF(provinceDF, province)
    provinceDF.to_excel("../cache/{}.xlsx".format(province))
    return provinceDF.plot


if __name__ == '__main__':
    read_cleaned_data(province="Channel Islands")
    read_cleaned_data(province="Anhui")
```

```
import read_data
import paramater_tuning
import plot
from common import *


def compare_hStart(province, change=5):
    # DATA_JRD = np.array(read_data.read_cleaned_data(province)[
    #                     ["confirmedMA", "recoveriesMA", "deathsMA"]][:DAYS].fillna(0))
    seir_solutions = []
    bestParameters = bestParametersDict[province]
    hStarts = (bestParameters["hStart"]-change, bestParameters["hStart"],
        bestParameters["hStart"]+5)
    for hStart in hStarts:
        bestParameters["hStart"] = hStart
        seir_solution = paramater_tuning.get_medical_upgrade_solution(bestParameters)
        seir_solution = np.diff(seir_solution, axis=0)
        seir_solutions.append(seir_solution)
    plot.compare_results(seir_solutions, "hStart", "隔离措施开始时间", hStarts,
        province=province)


if __name__ == '__main__':
    compare_hStart("Channel Islands")
    compare_hStart("Anhui")
```

```
from scipy.integrate import odeint
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.backends.backend_pdf import PdfPages
```

```python
from common import *

def lorenz(w, t, p, r, b):
    # 给出位置矢量w，和三个参数p, r, b计算出
    # dx/dt, dy/dt, dz/dt的值
    x, y, z = w
    # 直接与lorenz的计算公式对应
    return np.array([p*(y-x), x*(r-z)-y, x*y-b*z])

def test_lorentz():
    t = np.arange(0, 30, 0.01) # 创建时间点
    # 调用ode对lorenz进行求解，用两个不同的初始值
    track1 = odeint(lorenz, (0.0, 1.00, 0.0), t, args=(10.0, 28.0, 3.0))
    track2 = odeint(lorenz, (0.0, 1.01, 0.0), t, args=(10.0, 28.0, 3.0))

    fig = plt.figure()
    ax = Axes3D(fig)
    ax.plot(track1[:,0], track1[:,1], track1[:,2])
    ax.plot(track2[:,0], track2[:,1], track2[:,2])
    plt.show()

def f1(u, t,
      beta, start, hStart,
      k, epsilon, lamb, theta, delta, sigma, gamma, eta):
    S, E, I, Q, J, C, R, D = u

    h = 0.9*(t > hStart)
    # beta = (31+t)/(22+5*t)
    return np.array([
        -beta*(k*E+I)*(1-h)*S,
        beta*(k*E+I)*(1-h)*S - (epsilon+lamb)*E,
        epsilon*E - (theta+delta)*I,
        lamb*E - (sigma+delta)*Q,
        theta*I + sigma*Q - (gamma+delta)*J,
        theta * (I+Q) + sigma * Q,
        gamma*J,
        delta*I + eta*J
    ])*(t > start)

def f2(u, t,
      beta, start, hStart,
      k, epsilon, lamb, theta, delta, sigma,
      gamma0, gamma1, mStart, mEnd,
      eta):
    S, E, I, Q, J, C, R, D = u

    h = 0.9*(t > hStart)
```

```
    if t < mStart:
        gamma = gamma0
    elif t > mEnd:
        gamma = gamma1
    else:# 在时间介于mStart和mEnd之间时，gamma线性增加，表示医疗资源扩充或者引进医疗技术。
        gamma = gamma0 + (t-mStart)*(gamma1-gamma0)/(mEnd-mStart)

    dS = -beta*(k*E+I)*(1-h)*S
    dE = beta*(k*E+I)*(1-h)*S - (epsilon+lamb)*E
    dI = epsilon*E - (theta+delta)*I
    dQ = lamb*E - sigma*Q
    dJ = theta*I + sigma*Q - (gamma+delta)*J
    dC = theta * I + sigma * Q
    dR = gamma*J
    dD = delta*I + eta*J
    return np.array([dS, dE, dI, dQ, dJ, dC, dR, dD])*(t > start)


def test_SEIR(province, p=None, save=True):
    if p is None:
        p = bestParametersDict[province]

    u0 = (p["S0"], 109, p["I0"], p["Q0"], 0, 0, 0, 0)

    parameters = (p["betaTimesS0"]/p["S0"], p["start"], p["hStart"], p["k"], p["epsilon"],
        p["lamb"],
                p["theta"], p["delta"], p["sigma"], p["gamma0"], p["gamma1"],
                p["mStart"], p["mEnd"], p["eta"])
    t = np.arange(0, 200, 1) # 创建时间点
    # 调用ode对lorenz进行求解，用两个不同的初始值
    track1 = odeint(f2, u0, t, args=parameters)
    r, c = 2, 4
    fig, axs = plt.subplots(r, c, figsize=(3*c, 3*r))
    titles = ["S", "E", "I", "Q", "J", "C", "R", "D"]
    for i in range(r):
        for j in range(c):
            track = track1[:, c*i+j]
            axs[i, j].plot(t, track)
            axs[i, j].set(title="{}".format(titles[c*i+j]))
    if save:
        pdf = PdfPages("..//img//{}_SEIR.pdf".format(province))
        pdf.savefig()
        pdf.close()
    else:
        plt.show()
```

```python
if __name__ == '__main__':
    test_SEIR("Anhui")
    test_SEIR("Channel Islands")
```

A.2 matlab 源程序

```matlab
%% 输入原始数据并做出时间序列图
clear;clc
year =[1:1:60]';
x0=[21596.6250000000;26395.8750000000;22722.8000000000;22386.7000000000;24194.8000000000;24280.3000000000;24338
    23809.6350000000;29100.6650000000;25114.1000000000;24645.8000000000;26610.7000000000;26857.4000000000;26827.
    26081.8650000000;31877.8350000000;27863.7000000000;27278.5000000000;29459.2000000000;29807.6000000000;29609.
    27486.8100000000;33594.9900000000;29193.6000000000;28541.9000000000;30359.1000000000;30841.6000000000;30733.
    29728.8000000000;36335.2000000000;31725.7000000000;30586.1000000000;32955.7000000000;33878.1000000000;33073.

% year =[1995:1:2004]'; % 横坐标表示年份，写成列向量的形式（加'就表示转置）
% x0 = [174,179,183,189,207,234,220.5,256,270,285]';
    %原始数据序列，写成列向量的形式（加'就表示转置）
% % year = [2009:2015]; %
    其实本程序写成了行向量也可以，因为我怕你们真的这么写了，所以在后面会有判断。
% x0 = [730, 679, 632, 599, 589, 532, 511];
% year = [2010:2017]'; % 该数据很特殊，可以通过准指数规律检验，但是预测效果却很差
% x0 = [1.321,0.387,0.651,0.985,1.235,0.987,0.854,1.021]';
% year = [2014:2017]';
% x0 = [2.874,3.278,3.337,3.390]';

% 画出原始数据的时间序列图
figure(1); % 因为我们的图形不止一个，因此要设置编号
plot(year,x0,'o-'); grid on; % 原式数据的时间序列图
set(gca,'xtick',year(1:1:end)) % 设置x轴横坐标的间隔为1
xlabel('年份'); ylabel('社会消费品零售总额（亿元）'); % 给坐标轴加上标签


%% 因为我们要使用GM(1,1)模型，其适用于数据期数较短的非负时间序列
ERROR = 0; % 建立一个错误指标，一旦出错就指定为1
% 判断是否有负数元素
if sum(x0<0) > 0 %
    x0<0返回一个逻辑数组(0-1组成)，如果有数据小于0，则所在位置为1，如果原始数据均为非负数，那么这个逻辑数组中全为0，
    disp('亲，灰色预测的时间序列中不能有负数哦')
    ERROR = 1;
end

% 判断数据量是否太少
n = length(x0); % 计算原始数据的长度
disp(strcat('原始数据的长度为',num2str(n))) % strcat()是连接字符串的函数，第一讲学了，可别忘了哦
```

```
if n<=3
    disp('亲，数据量太小，我无能为力哦')
    ERROR = 1;
end

% 数据太多时提示可考虑使用其他方法（不报错）
if n>10
    disp('亲，这么多数据量，一定要考虑使用其他的方法哦，例如ARIMA，指数平滑等')
end

% 判断数据是否为列向量，如果输入的是行向量则转置为列向量
if size(x0,1) == 1
    x0 = x0';
end
if size(year,1) == 1
    year = year';
end


%%
     对一次累加后的数据进行准指数规律的检验(注意，这个检验有时候即使能通过，也不一定能保证预测结果非常好，例如上面的第
if ERROR == 0 % 如果上述错误均没有发生时，才能执行下面的操作步骤
    disp('------------------------------------------------------------')
    disp('准指数规律检验')
    x1 = cumsum(x0); % 生成1-AGO序列，cumsum是累加函数哦~ 注意：1.0e+03
        *0.1740的意思是科学计数法,即10^3*0.1740 = 174
    rho = x0(2:end) ./ x1(1:end-1) ; % 计算光滑度rho(k) = x0(k)/x1(k-1)

    % 画出光滑度的图形，并画上0.5的直线，表示临界值
    figure(2)
    plot(year(2:end),rho,'o-',[year(2),year(end)],[0.5,0.5],'-'); grid on;
    text(year(end-1)+0.2,0.55,'临界线') % 在坐标(year(end-1)+0.2,0.55)上添加文本
    set(gca,'xtick',year(2:1:end)) % 设置x轴横坐标的间隔为1
    xlabel('年份'); ylabel('原始数据的光滑度'); % 给坐标轴加上标签



    disp(strcat('指标1：光滑比小于0.5的数据占比为',num2str(100*sum(rho<0.5)/(n-1)),'%'))
    disp(strcat('指标2：除去前两个时期外，光滑比小于0.5的数据占比为',num2str(100*sum(rho(3:end)<0.5)/(n-3)),'%'))
    disp('参考标准：指标1一般要大于60%，指标2要大于90%，你认为本例数据可以通过检验吗？')

    Judge = input('你认为可以通过准指数规律的检验吗？可以通过请输入1，不能请输入0：');
    if Judge == 0
        disp('亲，灰色预测模型不适合你的数据哦~ 请考虑其他方法吧 例如ARIMA，指数平滑等')
        ERROR = 1;
    end
    disp('------------------------------------------------------------')
end
```

```
%%
    当数据量大于4时，我们利用试验组来选择使用传统的GM(1,1)模型、新信息GM(1,1)模型还是新陈代谢GM(1,1)模型；
    如果数据量等于4，那么我们直接对三种方法求一个平均来进行预测
if ERROR == 0 % 如果上述错误均没有发生时，才能执行下面的操作步骤
   if  n > 4 %
       数据量大于4时，将数据分为训练组和试验组(根据原数据量大小n来取，n为5-7个则取最后两年为试验组，n大于7则取最后三
      disp('因为原数据的期数大于4，所以我们可以将数据组分为训练组和试验组') %
          注意，如果试验组的个数只有1个，那么三种模型的结果完全相同，因此至少要取2个试验组
      if n > 7
         test_num = 3;
      else
         test_num = 2;
      end
      train_x0 = x0(1:end-test_num); % 训练数据
      disp('训练数据是：')
      disp(mat2str(train_x0')) % mat2str可以将矩阵或者向量转换为字符串显示，
          这里加一撇表示转置，把列向量变成行向量方便观看
      test_x0 = x0(end-test_num+1:end); % 试验数据
      disp('试验数据是：')
      disp(mat2str(test_x0')) % mat2str可以将矩阵或者向量转换为字符串显示
      disp('---------------------------------------------------------------')

      % 使用三种模型对训练数据进行训练，返回的result就是往后预测test_num期的数据
      disp(' ')
      disp('***下面是传统的GM(1,1)模型预测的详细过程***')
      result1 = gm11(train_x0, test_num);
          %使用传统的GM(1,1)模型对训练数据，并预测后test_num期的结果
      disp(' ')
      disp('***下面是进行新信息的GM(1,1)模型预测的详细过程***')
      result2 = new_gm11(train_x0, test_num);
          %使用新信息GM(1,1)模型对训练数据，并预测后test_num期的结果
      disp(' ')
      disp('***下面是进行新陈代谢的GM(1,1)模型预测的详细过程***')
      result3 = metabolism_gm11(train_x0, test_num);
          %使用新陈代谢GM(1,1)模型对训练数据，并预测后test_num期的结果

      % 现在比较三种模型对于试验数据的预测结果
      disp(' ')
      disp('---------------------------------------------------------------')
      % 绘制对试验数据进行预测的图形（对于部分数据，可能三条直线预测的结果非常接近）
      test_year = year(end-test_num+1:end); % 试验组对应的年份
      figure(3)
      plot(test_year,test_x0,'o-',test_year,result1,'*-',test_year,result2,'+-',test_year,result3,'x-');
          grid on;
      set(gca,'xtick',year(end-test_num+1): 1 :year(end)) % 设置x轴横坐标的间隔为1
      legend('试验组的真实数据','传统GM(1,1)预测结果','新信息GM(1,1)预测结果','新陈代谢GM(1,1)预测结果')
```

```
    % 注意：如果lengend挡着了图形中的直线，那么lengend的位置可以自己手动拖动
xlabel('年份'); ylabel('社会消费品零售总额（亿元）'); % 给坐标轴加上标签
% 计算误差平方和SSE
SSE1 = sum((test_x0-result1).^2);
SSE2 = sum((test_x0-result2).^2);
SSE3 = sum((test_x0-result3).^2);
disp(strcat('传统GM(1,1)对于试验组预测的误差平方和为',num2str(SSE1)))
disp(strcat('新信息GM(1,1)对于试验组预测的误差平方和为',num2str(SSE2)))
disp(strcat('新陈代谢GM(1,1)对于试验组预测的误差平方和为',num2str(SSE3)))
if SSE1<SSE2
    if SSE1<SSE3
        choose = 1; % SSE1最小，选择传统GM(1,1)模型
    else
        choose = 3; % SSE3最小，选择新陈代谢GM(1,1)模型
    end
elseif SSE2<SSE3
    choose = 2; % SSE2最小，选择新信息GM(1,1)模型
else
    choose = 3; % SSE3最小，选择新陈代谢GM(1,1)模型
end
Model = {'传统GM(1,1)模型','新信息GM(1,1)模型','新陈代谢GM(1,1)模型'};
disp(strcat('因为',Model(choose),'的误差平方和最小，所以我们应该选择其进行预测'))
disp('-------------------------------------------------------------')

%% 选用误差最小的那个模型进行预测
predict_num = input('请输入你要往后面预测的期数：  ');
% 计算使用传统GM模型的结果，用来得到另外的返回变量：x0_hat,
    相对残差relative_residuals和级比偏差eta
[result, x0_hat, relative_residuals, eta] = gm11(x0, predict_num); %
    先利用gm11函数得到对原数据拟合的详细结果

% % 判断我们选择的是哪个模型，如果是2或3，则更新刚刚由模型1计算出来的预测结果
if choose == 2
    result = new_gm11(x0, predict_num);
end
if choose == 3
    result = metabolism_gm11(x0, predict_num);
end

%% 输出使用最佳的模型预测出来的结果
disp('-------------------------------------------------------------')
disp('对原始数据的拟合结果：')
for i = 1:n
    disp(strcat(num2str(year(i)), ' ： ',num2str(x0_hat(i))))
end
disp(strcat('往后预测',num2str(predict_num),'期的得到的结果：'))
for i = 1:predict_num
```

```
        disp(strcat(num2str(year(end)+i), ' :  ',num2str(result(i))))
    end

    %%
        如果只有四期数据，那么我们就没必要选择何种模型进行预测，直接对三种模型预测的结果求一个平均值~
else
    disp('因为数据只有4期，因此我们直接将三种方法的结果求平均即可~')
    predict_num = input('请输入你要往后面预测的期数：  ');
    disp(' ')
    disp('***下面是传统的GM(1,1)模型预测的详细过程***')
    [result1, x0_hat, relative_residuals, eta] = gm11(x0, predict_num);
    disp(' ')
    disp('***下面是进行新信息的GM(1,1)模型预测的详细过程***')
    result2 = new_gm11(x0, predict_num);
    disp(' ')
    disp('***下面是进行新陈代谢的GM(1,1)模型预测的详细过程***')
    result3 = metabolism_gm11(x0, predict_num);
    result = (result1+result2+result3)/3;

    disp('对原始数据的拟合结果：')
    for i = 1:n
        disp(strcat(num2str(year(i)), ' :  ',num2str(x0_hat(i))))
    end
    disp(strcat('传统GM(1,1)往后预测',num2str(predict_num),'期的得到的结果：'))
    for i = 1:predict_num
        disp(strcat(num2str(year(end)+i), ' :  ',num2str(result1(i))))
    end
    disp(strcat('新信息GM(1,1)往后预测',num2str(predict_num),'期的得到的结果：'))
    for i = 1:predict_num
        disp(strcat(num2str(year(end)+i), ' :  ',num2str(result2(i))))
    end
    disp(strcat('新陈代谢GM(1,1)往后预测',num2str(predict_num),'期的得到的结果：'))
    for i = 1:predict_num
        disp(strcat(num2str(year(end)+i), ' :  ',num2str(result3(i))))
    end
    disp(strcat('三种方法求平均得到的往后预测',num2str(predict_num),'期的得到的结果：'))
    for i = 1:predict_num
        disp(strcat(num2str(year(end)+i), ' :  ',num2str(result(i))))
    end
end

%%
    绘制相对残差和级比偏差的图形(注意：因为是对原始数据的拟合效果评估，所以三个模型都是一样的哦~~~)
figure(4)
subplot(2,1,1) % 绘制子图（将图分块）
plot(year(2:end), relative_residuals,'*-'); grid on; % 原数据中的各时期和相对残差
legend('相对残差'); xlabel('月份序号');
```

```
    set(gca,'xtick',year(2:1:end)) % 设置x轴横坐标的间隔为1
    subplot(2,1,2)
    plot(year(2:end), eta,'o-'); grid on; % 原数据中的各时期和级比偏差
    legend('级比偏差'); xlabel('月份序号');
    set(gca,'xtick',year(2:1:end)) % 设置x轴横坐标的间隔为1
    disp(' ')
    disp('****下面将输出对原数据拟合的评价结果***')

    %% 残差检验
    average_relative_residuals = mean(relative_residuals); % 计算平均相对残差 mean函数用来均值
    disp(strcat('平均相对残差为',num2str(average_relative_residuals)))
    if average_relative_residuals<0.1
        disp('残差检验的结果表明：该模型对原数据的拟合程度非常不错')
    elseif average_relative_residuals<0.2
        disp('残差检验的结果表明：该模型对原数据的拟合程度达到一般要求')
    else
        disp('残差检验的结果表明：该模型对原数据的拟合程度不太好，建议使用其他模型预测')
    end

    %% 级比偏差检验
    average_eta = mean(eta); % 计算平均级比偏差
    disp(strcat('平均级比偏差为',num2str(average_eta)))
    if average_eta<0.1
        disp('级比偏差检验的结果表明：该模型对原数据的拟合程度非常不错')
    elseif average_eta<0.2
        disp('级比偏差检验的结果表明：该模型对原数据的拟合程度达到一般要求')
    else
        disp('级比偏差检验的结果表明：该模型对原数据的拟合程度不太好，建议使用其他模型预测')
    end
    disp(' ')
    disp('-------------------------------------------------------------')

    %% 绘制最终的预测效果图
    figure(5) % 下面绘图中的符号m:洋红色 b:蓝色
    plot(year,x0,'-o', year,x0_hat,'-*m', year(end)+1:year(end)+predict_num,result,'-*b' );
        grid on;
    hold on;
    plot([year(end),year(end)+1],[x0(end),result(1)],'-*b')
    legend('原始数据','拟合数据','预测数据') %
        注意：如果lengend挡着了图形中的直线，那么lengend的位置可以自己手动拖动
    set(gca,'xtick',[year(1):1:year(end)+predict_num]) % 设置x轴横坐标的间隔为1
    xlabel('年份'); ylabel('社会消费品零售总额（亿元）'); % 给坐标轴加上标签
end

% % 注意：代码文件仅供参考，一定不要直接用于自己的数模论文中
% % 国赛对于论文的查重要求非常严格，代码雷同也算作抄袭
% %
```

```
    视频中提到的附件可在售后群（购买后收到的那个有道云笔记中有加入方式）的群文件中下载。包括讲义、代码、优秀的作业、
%%
    关注我的微信公众号《数学建模学习交流》，后台发送“软件”两个字，可获得常见的建模软件下载方法；发送“数据”两个字
%%
    购买更多优质精选的数学建模资料，可关注我的微信公众号《数学建模学习交流》，在后台发送“买”这个字即可进入店铺(我的
%%
    视频价格不贵，但价值很高。单人购买观看只需要58元，三人购买人均仅需46元，视频本身也是下载到本地观看的，所以请大家
%% 如何修改代码避免查重的方法：https://www.bilibili.com/video/av59423231（必看）
```

```
function Px = gray(k)
X0=2.507755833;
a=-0.0681;
b=25521.4260044839;
res=[X0-b/a]*exp(0.0681*k)+b/a;
fprintf("%f\n",X0-b/a);
fprintf("%f\n",b/a);
Total=12*res;
r=[0.0718154817521152,0.0877744776970297,0.0762326278676047,0.0744577153842544,0.0801160196495294,0.08127970916
Px=Total*r;
end
```

```
function [result, x0_hat, relative_residuals, eta] = gm11(x0, predict_num)
    % 函数作用：使用传统的GM(1,1)模型对数据进行预测
    %     x0：要预测的原始数据
    %     predict_num：  向后预测的期数
    % 输出变量
        （注意，实际调用时该函数时不一定输出全部结果，就像corrcoef函数一样~，可以只输出相关系数矩阵，也可以附带输出p
    %     result：预测值
    %     x0_hat：对原始数据的拟合值
    %     relative_residuals：  对模型进行评价时计算得到的相对残差
    %     eta：  对模型进行评价时计算得到的级比偏差

    n = length(x0); % 数据的长度
    x1=cumsum(x0); % 计算一次累加值
    z1 = (x1(1:end-1) + x1(2:end)) / 2; % 计算紧邻均值生成数列（长度为n-1）
    % 将从第二项开始的x0当成y，z1当成x，来进行一元回归 y = kx +b
    y = x0(2:end); x = z1;
    % 下面的表达式就是第四讲拟合里面的哦~ 当是要注意，此时的样本数应该是n-1，少了一项哦
    k = ((n-1)*sum(x.*y)-sum(x)*sum(y))/((n-1)*sum(x.*x)-sum(x)*sum(x));
    b = (sum(x.*x)*sum(y)-sum(x)*sum(x.*y))/((n-1)*sum(x.*x)-sum(x)*sum(x));
    a = -k; %注意：k = -a哦
    % 注意：  -a就是发展系数，b就是灰作用量

    disp('现在进行GM(1,1)预测的原始数据是：')
    disp(mat2str(x0')) % mat2str可以将矩阵或者向量转换为字符串显示
    disp(strcat('最小二乘法拟合得到的发展系数为',num2str(-a),'，灰作用量是',num2str(b)))
```

```
    disp('***************分割线***************')
    x0_hat=zeros(n,1); x0_hat(1)=x0(1); % x0_hat向量用来存储对x0序列的拟合值，这里先进行初始化
    for m = 1: n-1
        x0_hat(m+1) = (1-exp(a))*(x0(1)-b/a)*exp(-a*m);
    end
    result = zeros(predict_num,1); % 初始化用来保存预测值的向量
    for i = 1: predict_num
        result(i) = (1-exp(a))*(x0(1)-b/a)*exp(-a*(n+i-1)); % 带入公式直接计算
    end

    % 计算绝对残差和相对残差
    absolute_residuals = x0(2:end) - x0_hat(2:end); % 从第二项开始计算绝对残差，因为第一项是相同的
    relative_residuals = abs(absolute_residuals) ./ x0(2:end); %
        计算相对残差，注意分子要加绝对值，而且要使用点除
    % 计算级比和级比偏差
    class_ratio = x0(2:end) ./ x0(1:end-1) ; % 计算级比 sigma(k) = x0(k)/x0(k-1)
    eta = abs(1-(1-0.5*a)/(1+0.5*a)*(1./class_ratio)); % 计算级比偏差
end


% % 注意：代码文件仅供参考，一定不要直接用于自己的数模论文中
% % 国赛对于论文的查重要求非常严格，代码雷同也算作抄袭
% %
    视频中提到的附件可在售后群（购买后收到的那个有道云笔记中有加入方式）的群文件中下载。包括讲义、代码、优秀的作业、
% %
    关注我的微信公众号《数学建模学习交流》，后台发送“软件”两个字，可获得常见的建模软件下载方法；发送“数据”两个字
% %
    购买更多优质精选的数学建模资料，可关注我的微信公众号《数学建模学习交流》，在后台发送“买”这个字即可进入店铺(我的
% %
    视频价格不贵，但价值很高。单人购买观看只需要58元，三人购买人均仅需46元，视频本身也是下载到本地观看的，所以请大家
% % 如何修改代码避免查重的方法：https://www.bilibili.com/video/av59423231（必看）
```

```
function [result] = metabolism_gm11(x0, predict_num)
% 函数作用：使用新陈代谢的GM(1,1)模型对数据进行预测
% 输入变量
%    x0：要预测的原始数据
%    predict_num： 向后预测的期数
% 输出变量
%    result：预测值
    result = zeros(predict_num,1); % 初始化用来保存预测值的向量
    for i = 1 : predict_num
        result(i) = gm11(x0, 1); % 将预测一期的结果保存到result中
        x0 = [x0(2:end); result(i)]; %
            更新x0向量，此时x0多了新的预测信息，并且删除了最开始的那个向量
    end
end
```

```
%% 注意：代码文件仅供参考，一定不要直接用于自己的数模论文中
%% 国赛对于论文的查重要求非常严格，代码雷同也算作抄袭
%%
    视频中提到的附件可在售后群（购买后收到的那个有道云笔记中有加入方式）的群文件中下载。包括讲义、代码、优秀的作业、
%%
    关注我的微信公众号《数学建模学习交流》，后台发送“软件”两个字，可获得常见的建模软件下载方法；发送“数据”两个字
%%
    购买更多优质精选的数学建模资料，可关注我的微信公众号《数学建模学习交流》，在后台发送“买”这个字即可进入店铺(我的
%%
    视频价格不贵，但价值很高。单人购买观看只需要58元，三人购买人均仅需46元，视频本身也是下载到本地观看的，所以请大家
%% 如何修改代码避免查重的方法：https://www.bilibili.com/video/av59423231（必看）
```

```
function [result] = new_gm11(x0, predict_num)
% 函数作用：使用新信息的GM(1,1)模型对数据进行预测
% 输入变量
%    x0：要预测的原始数据
%    predict_num： 向后预测的期数
% 输出变量
%    result：预测值
    result = zeros(predict_num,1); % 初始化用来保存预测值的向量
    for i = 1 : predict_num
        result(i) = gm11(x0, 1); % 将预测一期的结果保存到result中
        x0 = [x0; result(i)]; % 更新x0向量，此时x0多了新的预测信息
    end
end
```

```
[water,txt,raw]=xlsread('./data1.xlsx');
water=[0
1795.800065
-2041.074142
-2542.064004
-899.9517817
-981.5447866
-1091.250378
-705.9759629
-499.2289999
2337.483004
1823.152493
2346.571857
-2653.431559
2461.394534
-1702.547624
-2349.405845
-564.2529931
-498.4969846
```

```
-710.6457888
-181.807428
70.41002236
3027.398433
2679.64962
3289.855345
-2574.827684
3030.332184
-1175.883454
-1954.443058
31.60985684
184.0667167
-210.9811077
310.3576965
651.074384
3820.460152
3485.206138
3907.203421
-3545.346979
2356.205898
-2253.187048
-3114.274978
-1507.857115
-1237.542741
-1559.041202
-965.4619049
-718.8143196
2592.292021
2098.24752
2511.242517
-3875.732713
2506.912041
-2327.833075
-3694.177982
-1552.832667
-860.2071822
-1896.311647
-1306.156249
-941.9512431
2431.493047
2183.466231
2627.257845
];
% water=water(1:5,:);
% water=[9.40 8.81 8.65 10.01 11.07 11.54 12.73 12.43 11.64 11.39 11.1 10.85
% 10.71 10.24 8.48 9.88 10.31 10.53 9.55 6.51 7.75 7.8 5.96 5.21
% 6.39 6.38 6.51 7.14 7.26 8.49 9.39 9.71 9.65 9.26 8.84 8.29
```

```
% 7.21 6.93 7.21 7.82 8.57 9.59 8.77 8.61 8.94 8.4 8.35 7.95
% 7.66 7.68 7.85 8.53 9.38 10.09 10.59 10.83 10.49 9.21 8.66 8.39
% 8.27 8.14 8.71 10.43 11.47 11.73 11.61 11.93 11.55 11.35 11.11 10.49
% 10.16 9.96 10.47 11.70 10.1 10.37 12.47 11.91 10.83 10.64 10.29 10.34];
water=water';
x=water(:)';
%water(:)为将water中的数据转化为一列数据
r11=autocorr(x);
%计算自相关系数
r12=parcorr(x);
%计算偏相关函数
figure
subplot(211),autocorr(r11);
subplot(212),parcorr(r12);
s=12;
%地下水按照12个月的季节性变化
n=12;
%预报数据的个数
m1=length(x);
%原始数据的个数
for i=s+1:m1
    y(i-s)=x(i)-x(i-s);
    %周期差分：相邻两个年份同一个月分的地下水位的差
end
m2=length(y);
%周期差分后数据的个数
w=diff(y);
%消除趋势性的差分运算
r21=autocorr(w);
%计算自相关系数
r22=parcorr(w);
%计算偏相关函数
adf=adftest(w);
%若adf==1，则表明是平稳时间序列。
figure
subplot(211),autocorr(r21);
subplot(212),parcorr(r22);
m3=length(w);
%计算最终差分后数据的个数
k=0;
for i = 0:3
    for j = 0:3 %0:L,L的值不确定
        if i == 0 & j == 0
            continue
        elseif i == 0
            ToEstMd = arima('MALags',1:j,'Constant',0); %指定模型的结构
        elseif j == 0
```

```
            ToEstMd = arima('ARLags',1:i,'Constant',0); %指定模型的结构
        else
            ToEstMd = arima('ARLags',1:i,'MALags',1:j,'Constant',0); %指定模型的结构
        end
        k = k + 1;
        R(k) = i;
        M(k) = j;
        [EstMd,EstParamCov,LogL,info] = estimate(ToEstMd,w');
        %模型拟合,估计模型参数
        numParams = sum(any(EstParamCov));
        %计算拟合参数的个数
        [aic(k),bic(k)] = aicbic(LogL,numParams,m2);
    end
end
fprintf('R,M,AIC,BIC的对应值如下\n%f');%显示计算结果
check = [R',M',aic',bic']

%模型验证:
 res=infer(EstMd,w');
%求条件方差, 条件方差增加, 波动性增加
figure
subplot(2,1,1)
plot(res./sqrt(EstMd.Variance));
%画出标准化残差
title('Standardized Residuals');
subplot(2,1,2),qqplot(res);
%QQ图中残差基本完全落在45°线上即为符合正态性假设。否则模型可能出现错误.
% subplot(2,2,3),autocorr(res);
% subplot(2,2,4),parcorr(res);


%定阶:
%由差分后的由差分后的自相关图与偏自相关数据图可知:
%自相关系数在滞后1阶后就快速的减为0, 偏自相关系数同自相关系数
%所以, p=1,q=1

%模型预测:
p=input('输入阶数P=');
q=input('输入阶数q=');
ToEstMd=arima('ARLags',1:p,'MALags',1:q,'Constant',0);
%指定模型的结构
 [EstMd,EstParamCov,LogL,info] = estimate(ToEstMd,w');
 %模型拟合,估计模型参数
 dy_forest=forecast(EstMd,n,'Y0',w');
 %预测确定的模型输出,注意已知数据应为列向量, 所以用w'.
 yhat=y(m2)+cumsum(dy_forest);
 %求一阶差分的还原值
```

```
yhat=yhat';
for j=1:n
    x(m1+j)=yhat(j)+x(m1+j-s);
    %求x的预测值
end
what=x(m1+1:end);
%截取n个预报值
%画图:
figure
h4 = plot(x,'b');
%h5 = plot(x,'b');
%  h4 = plot(x,'b');
hold on
h5 = plot(length(x)-11:length(x),what,'r','LineWidth',2);
xlabel('月份序号'); ylabel('预测与真实值的季节波动');
hold off
```