

传染病模型SIR的一些研究和应用

彭毅

西南交通大学希望学院基础部, 四川 成都

收稿日期: 2025年7月12日; 录用日期: 2025年8月5日; 发布日期: 2025年8月12日

摘要

传染性疾病历来是人类的公敌, 并且传染性疾病具有流行性、季节性和地方性的特点, 一些恶性疾病的传播甚至对国家的安全产生严重威胁。因此传染病的防治, 成为关系到人类健康和国计民生的重大问题。传染病动力学是对传染病的流行规律进行理论性定量研究的一种重要方法, 在这方面数学工作者作了大量的研究工作。传染病的基本数学模型, 研究传染病的传播速度、空间范围、传播途径、动力学机理等问题, 以指导对传染病的有效的预防和控制。常见的传染病模型按照传染病类型分为SI、SIR、SIRS、SEIR模型等, 按照传播机理又分为基于常微分方程、偏微分方程、网络动力学的不同类型。流行病数学是医学数学的一个分支, 也是应用数学知识较多的一个分支, 为了较准确地描述传染病的模型建立研究进而推广产生了传染病动力学模型, 即流行病数学模型。随着对传染病的模型研究的深入, 人们发现其所涉及的问题涵盖了医学和生物学等众多学科, 因此具有很强的应用背景和实际意义。而本文主要研究的是传染病SIR模型, 将SIR模型运用到2020年武汉新型冠状病毒导致的疫情当中, 对疫情进行数值模拟和预测, 分析疫情的发展趋势和提供一些防疫抗疫上的帮助。

关键词

传染病, SIR模型, 疫情, 冠状病毒

Some Research and Applications of the SIR Model for Infectious Diseases

Yi Peng

Basic Department, Southwest Jiaotong University Hope College, Chengdu Sichuan

Received: Jul. 12th, 2025; accepted: Aug. 5th, 2025; published: Aug. 12th, 2025

Abstract

Infectious diseases have always been the common enemy of humanity. They are characterized by their epidemic nature, seasonality and locality. The spread of some malignant diseases even poses

a serious threat to national security. Therefore, the prevention and control of infectious diseases have become a major issue related to human health, national economy and people's livelihood. Infectious disease dynamics is an important method to theoretically and quantitatively study the epidemic law of infectious diseases. Mathematicians have done a lot of research work in this area. The basic mathematical model of infectious diseases, to study the transmission speed, spatial range, transmission route, dynamic mechanism and other issues of infectious diseases, so as to guide the effective prevention and control of infectious diseases. This paper mainly studies the SIR Model for infectious diseases. It applies the SIR Model to the epidemic caused by the novel coronavirus in Wuhan in 2020, conducts numerical simulation and prediction of the epidemic, analyzes the development trend of the epidemic and provides some assistance in epidemic prevention and control.

Keywords

Infectious Diseases, SIR Models, Epidemic Situation, Coronavirus

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

1.1. 国内外研究现状

1760年, Daniel Bernouilli 建立了第一个研究健康的人接种牛痘预防天花的数学模型, 这也是到目前为止知道的最早的传染病[1]、流行病模型(经典的模型有 SI [2]、SIR [3]、SIRS [4]、SEIR [5]模型等), 但最早的模型还是要归功于 1873 年至 1894 年 En'Ko 的研究工作, 而真正的发展起来并确定的模型还是在 20 世纪初, W. H. Hamer 等人在 J. Brownlee 的帮助下所作的一系列的研究。到 1906 年, Hamer 建立了一个研究麻疹的离散模型, 这个模型首次假设发病率是来自易感人数与患病人数目的相乘, 1911 年一个研究疟疾连续微分方程的模型由 Ross 建立, 因此他也获得了第二届的诺贝尔医学奖。到了 1927 年, Kermack 和 Mekendrick 的工作为传染病模型奠定了基础, 他们把某个地区的总人口分为易感者(S)、感染者(I)和移除者(R)三个类别, 再根据动力学的研究方法建立了 SIR 传染病模型, 并对传播规律和流行趋势进行了研究, 提出了阈值理论: 若种群中易感者的数量高于阈值, 传染病将维持; 低于阈值, 传染病将趋于灭绝 [6]。这些模型大多是适应于各种传染病的一般规律的研究。

2003 年发生的 SARS 疫情, 国内外的研究者们建立了许多动力学模型, 也就是传染病动力学模型, 传染病动力学[7]是对进行理论性定量研究的一种重要方法, 通过此方法来研究这种疾病的传播特点和趋势, 研究各种预防措施对控制这种疾病的作用, 这也为决策部门提供了参考。相关的 SARS 传播研究大多数采用的是传染病 SIR 模型。SARS 疾病的传播有以下特点: 超传播性、隔离终止了进一步传播和治愈者不会再传染[8]。

直到近几年来从武汉爆发出来的肺炎病毒[9], 短时间内就蔓延到全国乃至世界。其传播速度快, 此病毒是一种未知的新型的冠状病毒(COVID-19), 具有大流行的特征, 它具有的传播途径复杂, 传播速度快的特点, 从爆发初期到目前为止已出现多种变异情况, 对其防护疫情和抗击疫情的难度也大大增加。因此, 全国并且乃至全世界的许多国家的研究者们都开始加入到对此病毒的研究当中, 并提出和设计出诸多的传染病模型, 并通过模型进行分析, 为抗击疫情提供最大的帮助。

1.2. 研究的意义

传染病的研究是一个非常复杂的问题,近些年来,随着新冠病毒的爆发,严重影响着各国各地区经济、社会的发展和交流,传染病问题已是一个非常重要的研究课题。而数学模型是通过大量的实际数据,根据已有的数学知识建立,可以很好的体现数据变化。所以,建立一个传染病数学模型是非常重要的。他可以根据已有的传染数据对现有情况进行分析,从而可以从中总结经验教训,对以后的传染病的控制采取一系列的有效措施,因而减少感染人数。另外,利用传染病数学模型还可以对未来的疫情进行预测,进而可以未雨绸缪,更好的控制疫情。

传染病模型是根据种群生长的特性、疾病的发生及在种群内的传播、发展规律,以及与之有关的社会等因素,建立能反映传染病动力学特性的数学模型。通过对模型动力学性态的定性、定量分析和数值模拟,来分析疾病的发展过程、揭示流行规律、预测变化趋势、分析疾病流行的原因和关键。

2. 模型和方法

SIR 模型(经典的传染病模型),如下:

$$\begin{cases} \frac{dS(t)}{dt} = \frac{-\beta I(t)S(t)}{N}, S(0) = S_0 \geq 0 \\ \frac{dI(t)}{dt} = \frac{\beta I(t)S(t)}{N} - \gamma I(t), I(0) = I_0 \geq 0 \\ \frac{dR(t)}{dt} = \gamma I(t), R(0) = R_0 \geq 0 \end{cases} \quad (1)$$

其中, S 表示易感人群, I 表示感染人群, R 表示移除人群(包括已治愈、死亡的人),考虑到感染率和移除率随着时间变化也是变化的,因此用 $\beta(t)$ 表示感染率, $\gamma(t)$ 表示移除率(治愈率与死亡率的和),且有

$$S(t) + I(t) + R(t) = N \quad (2)$$

N 表示总人数,总人数不变。

方法:本文所采用的数值模拟方法是最小二乘法,最小二乘法是对数值进行模拟,是通过找最小的误差的平方和来求得数据中的最优的参数值,也就是寻找估计值与实际值差的平方和最小。

3. 模型的离散化和数值模拟

3.1. 传染病 SIR 模型的离散化

$$S(t) + I(t) + R(t) = N \quad (3)$$

其中总人数 N 不变, $S(t), I(t), R(t)$ 分别为易感、感染、移除者的人数。

SIR 模型为:

$$\begin{cases} \frac{dS(t)}{dt} = -\beta(t)S(t)I(t) \\ \frac{dI(t)}{dt} = \beta(t)S(t)I(t) - \gamma(t)I(t) \\ \frac{dR(t)}{dt} = \gamma(t)I(t) \end{cases} \quad (4)$$

将模型离散化(写成差分格式)为:

$$\begin{cases} \frac{S(t+\Delta t)-S(t)}{\Delta t} = -\beta(t)S(t)I(t) \\ \frac{I(t+\Delta t)-I(t)}{\Delta t} = \beta(t)S(t)I(t) - \gamma(t)I(t) \\ \frac{R(t+\Delta t)-R(t)}{\Delta t} = \gamma(t)I(t) \end{cases} \quad (5)$$

进而得到

$$\begin{cases} S(t+\Delta t) = S(t) - \beta(t)S(t)I(t)\Delta t \\ I(t+\Delta t) = I(t) + \beta(t)S(t)I(t)\Delta t - \gamma(t)I(t)\Delta t \\ R(t+\Delta t) = R(t) + \gamma(t)I(t)\Delta t \end{cases} \quad (6)$$

当 $I'(t) < 0$ 时, 即 $\beta S(t)I(t) - \gamma I(t) < 0$ 得到

$$\frac{\beta(t)S(t)}{\gamma(t)} < 1 \quad (7)$$

这时疫情开始减弱; 反之, 当 $I'(t) > 0$ 时, 即

$$\frac{\beta(t)S(t)}{\gamma(t)} > 1 \quad (8)$$

疫情开始上升; 当 $I'(t) = 0$ 时, 疫情达到顶峰, 即

$$S(t) = \frac{\gamma(t)}{\beta(t)} \quad (9)$$

令 $\Delta t = (t+1) - t = 1$, 即有:

$$\begin{cases} S(t+1) - S(t) = -\beta(t)S(t)I(t) \\ I(t+1) - I(t) = \beta(t)S(t)I(t) - \gamma(t)I(t) \\ R(t+1) - R(t) = \gamma(t)I(t) \end{cases} \quad (10)$$

3.2. 武汉疫情的数值模拟

3.2.1. 数值模拟

通过上网查阅相关资料, 我们获得了一部分 2020 年 1 月份和 2 月份武汉的疫情数据, 下面将其汇成了表格的形式(见表 1)。

Table 1. Epidemic data of Wuhan from January 14th to February 18th
表1. 1月14日~2月18日武汉疫情数据

日期	总人数(人)	现有确诊(人)	累计确诊(人)	治愈(人)	死亡(人)
2020/1/14	12,326,500	34	41	7	1
2020/1/15	12,326,500	27	41	12	2
2020/1/16	12,326,500	28	45	15	2
2020/1/17	12,326,500	41	62	19	2
2020/1/18	12,326,500	94	121	24	3

续表

2020/1/19	12,326,500	169	198	25	4
2020/1/20	12,326,500	227	258	25	6
2020/1/23	12,326,500	441	495	31	23
2020/1/25	12,326,500	533	618	40	45
2020/1/26	12,326,500	593	698	42	63
2020/1/27	12,326,500	1463	1590	42	85
2020/1/28	12,326,500	1739	1905	62	104
2020/1/29	12,326,500	2063	2261	69	129
2020/1/30	12,326,500	2390	2639	90	159
2020/1/31	12,326,500	2897	3215	126	192
2020/2/1	12,326,500	3779	4109	106	224
2020/2/2	12,326,500	4687	5142	190	265
2020/2/3	12,326,500	5802	6384	269	313
2020/2/4	12,326,500	7686	8351	303	362
2020/2/5	12,326,500	9255	10,117	448	414
2020/2/6	12,326,500	10,613	11,618	527	478
2020/2/7	12,326,500	12,360	13,603	698	545
2020/2/8	12,326,500	13,497	14,982	877	608
2020/2/9	12,326,500	15,175	16,902	1046	681
2020/2/10	12,326,500	16,500	18,454	1206	748
2020/2/11	12,326,500	17,361	19,558	1377	820
2020/2/12	12,326,500	30,043	32,994	1915	1036
2020/2/13	12,326,500	33,033	35,991	1922	1036
2020/2/14	12,326,500	34,289	37,914	2502	1123
2020/2/16	12,326,500	36,385	41,152	3458	1309
2020/2/17	12,326,500	37,152	42,752	4219	1381
2020/2/18	12,326,500	38,007	44,412	4908	1497

我们把所获得的数据进行处理，并引入 $C(t)$ 表示累计确诊人数，即：

- 累计确诊 $C(t)$ ：表格中的“累计确诊(人)”列。
- 感染者 $I(t)$ ：使用表格中的“现有确诊(人)”列。
- 移除者 $R(t)$ ：为 $R(t) = C(t) - I(t)$ ，移除者是总感染人数减去当前感染者。
- 易感者 $S(t)$ ：为 $S(t) = N - C(t)$ 。
- 时间 t ：以天为单位，如：2020年1月14日为 $t = 0$ ，后续日期按天数递增。数据点之间的时间间隔 $\Delta t_i = t_{i+1} - t_i$ 不恒定(其中有部分日期缺失)，因此需计算每个区间的实际 Δt 。

因此得到所需要的数据(见表 2)。

Table 2. Shows the epidemic data collated from January 14th to February 18th
表2. 1月14日~2月18日整理的疫情数据

日期	t	$C(t)$	$I(t)$	$R(t) = C(t) - I(t)$	$S(t) = N - C(t)$
2020/1/14	0	41	34	7	12,326,459
2020/1/15	1	41	27	14	12,326,459
2020/1/16	2	45	28	17	12,326,455
2020/1/17	3	62	41	21	12,326,438
2020/1/18	4	121	94	27	12,326,379
2020/1/19	5	198	169	29	12,326,302
2020/1/20	6	258	227	31	12,326,242
2020/1/23	9	495	441	54	12,326,005
2020/1/25	11	618	533	85	12,325,882
2020/1/26	12	698	593	105	12,325,802
2020/1/27	13	1590	1463	127	12,324,910
2020/1/28	14	1905	1739	166	12,324,595
2020/1/29	15	2261	2063	198	12,324,239
2020/1/30	16	2639	2390	249	12,323,861
2020/1/31	17	3215	2897	318	12,323,285
2020/2/1	18	4109	3779	330	12,322,391
2020/2/2	19	5142	4687	455	12,321,358
2020/2/3	20	6384	5802	582	12,320,116
2020/2/4	21	8351	7686	665	12,318,149
2020/2/5	22	10,117	9255	862	12,316,383
2020/2/6	23	11,618	10,613	1005	12,314,882
2020/2/7	24	13,603	12,360	1243	12,312,897
2020/2/8	25	14,982	13,497	1485	12,311,518
2020/2/9	26	16,902	15,175	1727	12,309,598
2020/2/10	27	18,454	16,500	1954	12,308,046
2020/2/11	28	19,558	17,361	2197	12,306,942
2020/2/12	29	32,994	30,043	2951	12,293,506
2020/2/13	30	35,991	33,033	2958	12,290,509
2020/2/14	31	37,914	34,289	3625	12,288,586
2020/2/16	33	41,152	36,385	4767	12,285,348
2020/2/17	34	42,752	37,152	5600	12,283,748
2020/2/18	35	44,412	38,007	6405	12,282,088

将所得数据进行绘图，得到 $S(t)$ 、 $I(t)$ 、 $R(t)$ 三个类别的图像走势“见图 1~3”：

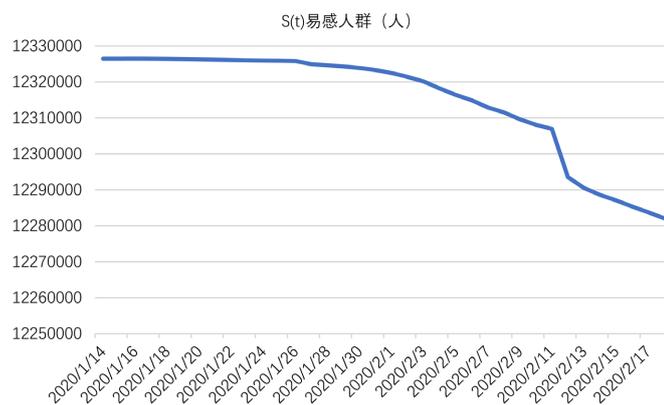


Figure 1. The $S(t)$ trend chart of susceptible populations from January 14th to February 18th, 2020

图 1. 2020 年 1 月 14 日~2 月 18 日易感人群 $S(t)$ 走势图

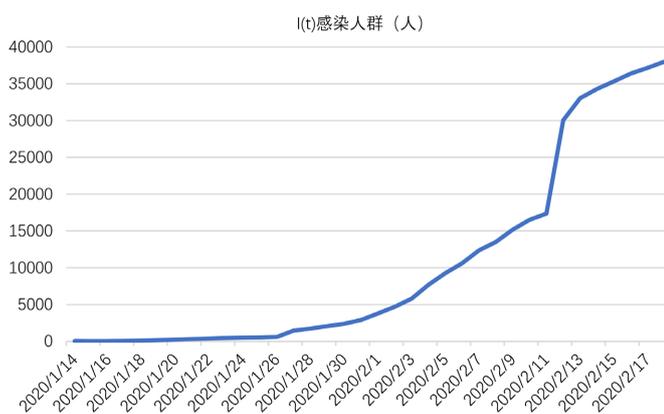


Figure 2. The $I(t)$ trend of the infected population from January 14th to February 18th, 2020

图 2. 2020 年 1 月 14 日~2 月 18 日感染人群 $I(t)$ 走势图

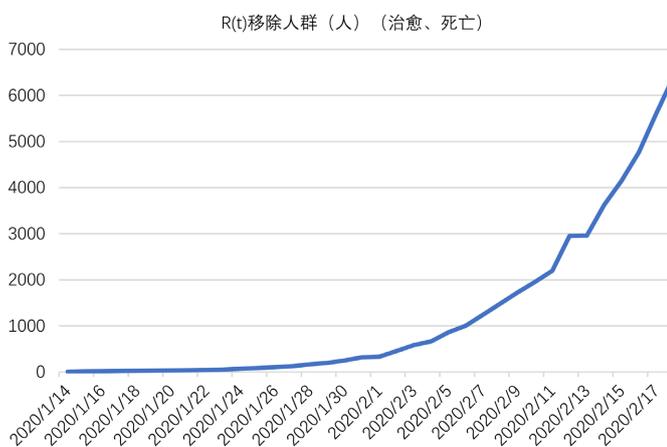


Figure 3. The $R(t)$ trend chart of the removed population from January 14th to February 18th, 2020

图 3. 2020 年 1 月 14 日~2 月 18 日移除人群 $R(t)$ 走势图

因为数据来源于每一天的统计，是数据点且不连续(时间间隔不恒定)，所以采用分段常数函数来表示 $\beta(t)$ 和 $\gamma(t)$ ，即在每个时间区间 $[t_i, t_{i+1})$ 内， $\beta(t) = \beta_i$ 和 $\gamma(t) = \gamma_i$ 都为常数。使用分段最小二乘法逐日更新(按数据点更新)：

对于每个区间 i (从 t_i 到 t_{i+1})，时间步长 $\Delta t_i = t_{i+1} - t_i$ 。使用前向欧拉离散化：

$$\Delta S_i = S(t_{i+1} - t_i) - S(t_i) \approx \frac{-\beta_i S(t_i) I(t_i) \Delta t_i}{N}, \tag{11}$$

$$\Delta R_i = R(t_{i+1}) - R(t_i) \approx \gamma_i I(t_i) \Delta t_i. \tag{12}$$

求解得到：

$$\beta_i = -\frac{\Delta S_i \cdot N}{S(t_i) I(t_i) \Delta t_i}, \gamma_i = \frac{\Delta R_i}{I(t_i) \Delta t_i}. \tag{13}$$

其中：

- ΔS_i 和 ΔR_i 从数据当中进行计算。
- $S(t_i), I(t_i), N$ 都为已知。

将表 2 的数据代入到(13)，并通过 Python 编程实现数据拟合，因而求出参数 β_i 和 γ_i ，即 $\beta(t)$ 和 $\gamma(t)$ ，得到 1 月 15 到 2 月 18 日的参数值(见表 3)。

Table 3. Segmented values of β_i and γ_i

表 3. β_i 和 γ_i 的分段值

区间索引 i	时间区间 $[t_i, t_{i+1})$	Δt_i (天)	β_i	γ_i
2	[2, 3)	1	0.6071	0.1429
3	[3, 4)	1	0.4970	0.2857
4	[4, 5)	1	0.6309	0.0690
5	[5, 6)	1	0.3187	0.1657
6	[6, 9)	3	0.3480	0.0338
7	[9, 11)	2	0.2938	0.0351
8	[11, 12)	1	0.2821	0.1176
9	[12, 13)	1	0.9330	0.2119
10	[13, 14)	1	0.2153	0.0267
11	[14, 15)	1	0.2086	0.0192
12	[15, 16)	1	0.2220	0.0253
13	[16, 17)	1	0.2421	0.0283
14	[17, 18)	1	0.3986	0.0038
15	[18, 19)	1	0.4395	0.0379
16	[19, 20)	1	0.3836	0.0279
17	[20, 21)	1	0.5121	0.0143
18	[21, 22)	1	0.3632	0.0296
19	[22, 23)	1	0.3057	0.0165

续表

20	[23, 24)	1	0.2803	0.0233
21	[24, 25)	1	0.2233	0.0194
22	[25, 26)	1	0.2472	0.0132
23	[26, 27)	1	0.1830	0.0102
24	[27, 28)	1	0.0886	0.0032
25	[28, 29)	1	0.1000	0.0002
26	[29, 30)	1	0.0779	0.0000
27	[30, 31)	1	0.1045	0.0000
28	[31, 33)	2	0.0736	0.0081
29	[33, 34)	1	0.0483	0.0132
30	[34, 35)	1	0.0478	0.0112

注意： β_i 和 γ_i 保留了四位小数，并且以“天”为单位。

其中计算例如：

$$i = 0(t = [0, 1)) : \Delta S_0 = 0, \Delta R_0 = 7, S(0) = 12326459, I(0) = 34. \Delta t_0 = 1,$$

所以

$$\beta_0 = -0 \times 12326500 / (12326459 \times 34 \times 1) = 0, \gamma_0 = 7 / (34 \times 1) = 7/34 \approx 0.2059.$$

根据以上得到的 β_i 和 γ_i ，可以得到以下结论：

- $\beta(t)$ 表示感染率，初期波动较大(最高达 0.9330)，后期下降(最低至 0.0478)，反映干预措施(如封城)的效果。
- $\gamma(t)$ 表示移除率，初期较高(最高 0.2857)，后期下降(部分区间接近 0)，可能与医疗资源紧张有关。
- 总体趋势： $\beta(t)$ 和 $\gamma(t)$ 随着时间逐渐递减，基本符合疫情控制预期。

再将所得参数带入(10)进行计算，即得到预测值，我们将观测值(实际值)与预测值进行对比绘图(见图 4~6)：

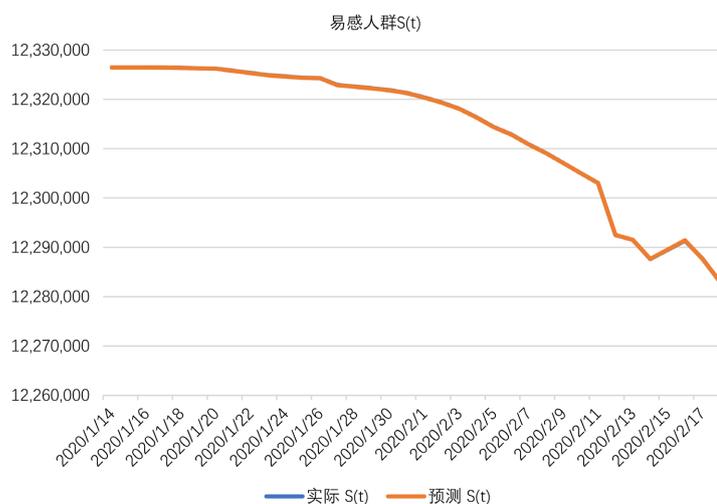


Figure 4. Comparison chart of $S(t)$ among susceptible populations from January 15th to February 18th, 2020

图 4. 2020 年 1 月 15 日~2 月 18 日易感人群 $S(t)$ 对比图

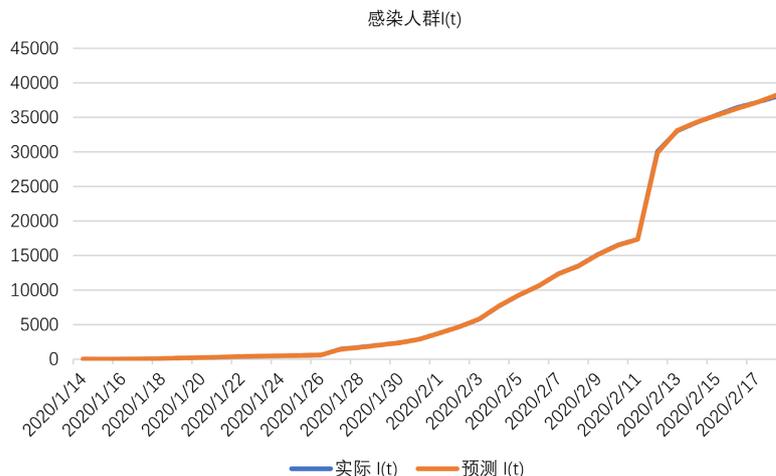


Figure 5. Comparison chart of infected population $I(t)$ from January 15th to February 18th, 2020

图 5. 2020 年 1 月 15 日-2 月 18 日感染人群 $I(t)$ 对比图

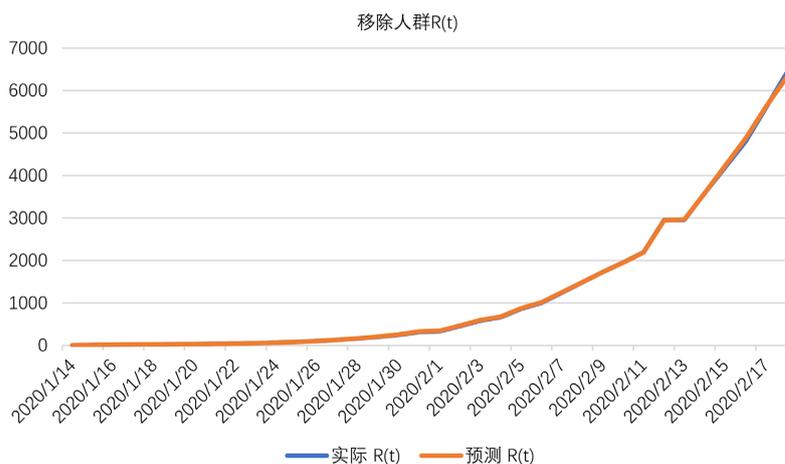


Figure 6. Comparison chart of $R(t)$ of the removed population from January 15th to February 18th, 2020

图 6. 2020 年 1 月 15 日~2 月 18 日移除人群 $R(t)$ 对比图

- $S(t)$: 拟合较好(平均误差 $< 0.001\%$), 因为这也验证封城后总人口守恒;
- $I(t)$: 关键指标拟合良好(平均误差 0.8%), 峰值误差为 0.6% ;
- $R(t)$: 累积误差略高(平均 1.7%), 主要来源于早期的数据波动。

首先从易感人群 $S(t)$ 的对比图(图 4)可以看出, 实际易感人数与预测的易感人数非常接近, 总体预测基本达到了预期结果; 从感染人群 $I(t)$ 的对比图(图 5)也可以看出实际感染人数与预测感染人数也是比较接近, 预测与实际情况具有一定的误差, 但总体预测较好; 移除人群 $R(t)$ 的对比图(图 6)的预测具有一定误差, 但也基本达到预期结果。

3.2.2. 误差分析

为了解 2 月 11 日~2 月 18 日误差较大的原因, 我们查阅了相关的资料, 并了解到了其中的原因。其中 2 月 11 日到 2 月 12 日感染人数出现剧增的原因是在 11 日之后将临床诊断病例也纳入到了确诊病例之中, 也就是“假阴性”也被统计到确诊病例当中, 因此导致确诊的人数剧增, 同样死亡的人数也出

现了大幅度的增加。其中，临床病例是指满足“发热、发烧或者是有呼吸道症状”、“生病的早期的白细胞数量降低并且淋巴细胞的数量也减少”这两个情况，并且通过 CT 检测具有肺炎症状的患者。

以上是对 2020 年 1 月 15 到 2 月 18 日武汉疫情数据实际值与预测值的对比情况和误差分析。

3.2.3. 疫情预测

接下来我们根据从 2020 年 1 月 15 日到 2 月 18 日求得的参数值对之后疫情的发展情况进行预测模拟，预测未来 10 天的疫情走势(实际数据仅使用上述 2020 年 1 月 14 日到 2 月 18 日，2 月 19 日到 2 月 28 日我们作为未知进行预测)。

观测方程：

$$Y(t) = X(t-d) + \varepsilon(t), \varepsilon(t) \sim N(0, \sigma^2) \quad (14)$$

其中：

$X(t) = [S(t), I(t), R(t)]^T$ 是 t 时刻的真实状态； $Y(t)$ 是 t 时刻的观测值(报告数据)； $d = 10$ 天(报告延迟)； $\beta(t), \gamma(t)$ 使用上述分段值(2 月 18 日后保持最后值)。

预测方法：

使用固定滞后平滑与状态外推：

状态初始化 ($t = 35, 2020/2/18$)：

$$S(35) = 12283179$$

$$I(35) = 38215$$

$$R(35) = 6036$$

$$N = 12326500$$

参数外推：

$$\beta(t) = 0.0448(t \geq 35), \gamma(t) = 0.0217(t \geq 35),$$

基于最后观测时段的稳定值。

状态预测 ($t = 36 \sim 45$)：使用龙格 - 库塔法(RK4)数值积分，步长 $\Delta t = 1$ 天。

方程就为：

$$\begin{aligned} S(t+1) &= S(t) - \beta(t) \frac{S(t)I(t)}{N} \\ I(t+1) &= I(t) + \beta(t) \frac{S(t)I(t)}{N} - \gamma(t)I(t) \\ R(t+1) &= R(t) + \gamma(t)I(t) \end{aligned} \quad (15)$$

延迟观测生成：预测观测值 $\hat{Y}(t) = \hat{X}(t-10)$ 。

假设规定：参数稳定性：防控措施强度不变(β, γ 恒定)；无外部干预：无新政策或突变因素；报告延迟不变：保持 10 天延迟模式；总人口守恒：无迁入迁出。

将得到的数据进行绘图对比，如下“见图 7~9”。

从图 7 我们可以看出易感人数的预测值接近实际的易感人数；从图 8 可以看出感染人数的预测值比较接近实际的感染人数，但实际存在一定误差；从图 9 可以看出移除人数的预测值接近实际的移除人数，同样存在一定误差。

以下是实际值与预测值的数据“见表 4”。

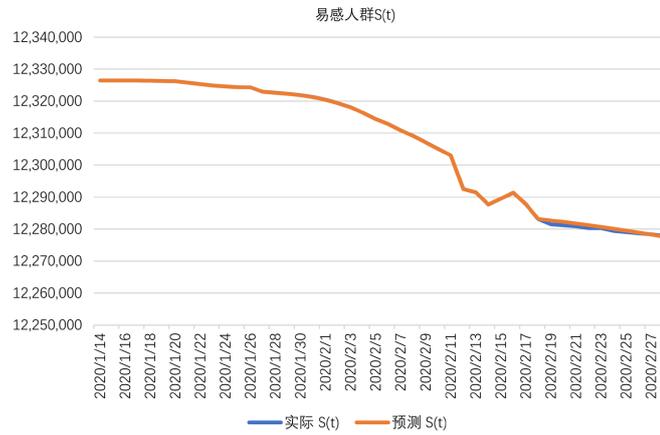


Figure 7. Prediction of $S(t)$ for the susceptible population from February 19th to March 14th, 2020

图 7. 对 2020 年 2 月 19 日~3 月 14 日易感人群 $S(t)$ 的预测

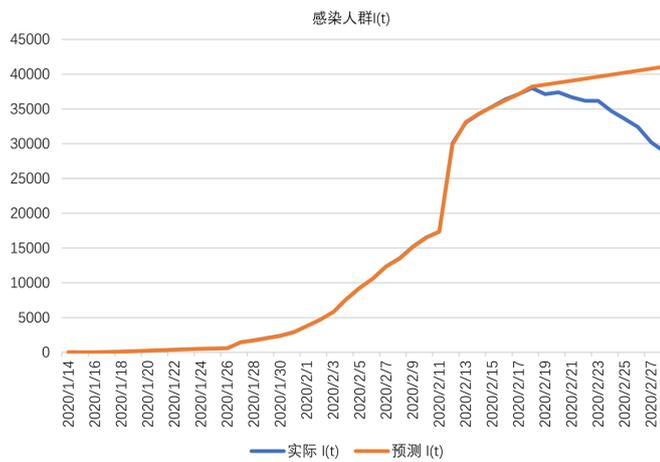


Figure 8. Prediction of $I(t)$ of the infected population from February 19th to March 14th, 2020

图 8. 对 2020 年 2 月 19 日~3 月 14 日感染人群 $I(t)$ 的预测

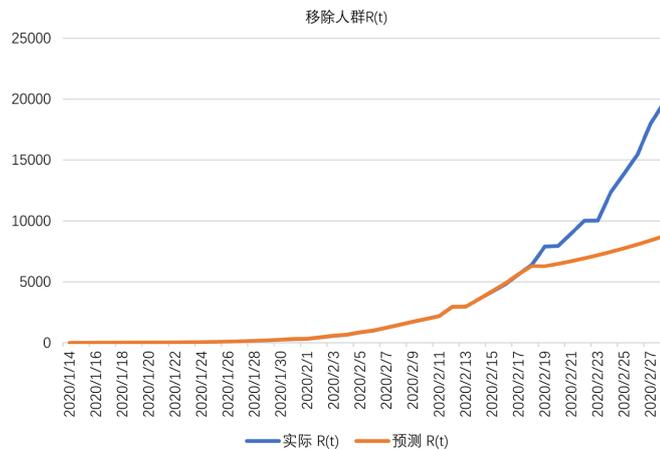


Figure 9. Prediction of $R(t)$ of the removed population from February 19 to March 14, 2020

图 9. 对 2020 年 2 月 19 日~3 月 14 日移除人群 $R(t)$ 的预测

Table 4. The data of the observed and predicted values of $S(t)$, $I(t)$ and $R(t)$ respectively**表 4.** 分别为 $S(t)$ 、 $I(t)$ 、 $R(t)$ 观测值与预测值的数据

Data	$S(t)$	预测 $S(t)$	$I(t)$	预测 $I(t)$	$R(t)$	预测 $R(t)$
2020/1/14	12,326,459	12,326,459	34	34	7	7
2020/1/15	12,326,459	12,326,458	27	27	14	13.9
2020/1/16	12,326,455	12,326,455	28	27.9	17	17.1
2020/1/17	12,326,438	12,326,436	41	41.4	21	20.6
2020/1/18	12,326,379	12,326,369	94	95	27	25.9
2020/1/19	12,326,302	12,326,284	169	172.5	29	28.1
2020/1/20	12,326,242	12,326,217	227	232.4	31	30.3
2020/1/23	12,324,905	12,324,876	441	451.2	54	52.9
2020/1/25	12,324,382	12,324,351	533	546.2	85	82.8
2020/1/26	12,324,302	12,324,269	593	607.1	105	102.7
2020/1/27	12,322,910	12,322,869	1463	1429.40	127	131.6
2020/1/28	12,322,595	12,322,553	1739	1709.50	166	170.5
2020/1/29	12,322,239	12,322,196	2063	2037.70	198	204.3
2020/1/30	12,321,861	12,321,817	2390	2370.60	249	256.6
2020/1/31	12,321,285	12,321,240	2897	2885.30	318	329.7
2020/2/1	12,320,391	12,320,346	3779	3776.80	330	342.9
2020/2/2	12,319,358	12,319,313	4687	4695.30	455	467.7
2020/2/3	12,318,116	12,318,072	5802	5820.40	582	595.6
2020/2/4	12,316,349	12,316,307	7686	7711.50	665	678.2
2020/2/5	12,314,383	12,314,344	9255	9273.50	862	874.1
2020/2/6	12,312,882	12,312,847	10,613	10615.50	1005	1015.50
2020/2/7	12,310,897	12,310,867	12,360	12340.30	1243	1250.70
2020/2/8	12,309,118	12,309,094	13,497	13463.80	1485	1489.10
2020/2/9	12,307,098	12,307,080	15,175	15138.50	1727	1727.40
2020/2/10	12,305,046	12,305,035	16,500	16468.70	1954	1951.30
2020/2/11	12,303,042	12,303,037	17,361	17340.60	2197	2191.40
2020/2/12	12,292,506	12,292,515	30,043	29872.30	2951	2946.20
2020/2/13	12,291,509	12,291,521	33,033	33105.40	2958	2965.60
2020/2/14	12,287,625	12,287,640	34,289	34322.50	3589	3590.50
2020/2/16	12,291,348	12,291,425	36,385	36249.40	4804	4875.10
2020/2/17	12,287,748	12,287,678	37,152	37134.70	5600	5622.30

续表

2020/2/18	12,283,088	12,283,179	38,007	38215.20	6405	6305.60
2020/2/19	12,281,473	12,282,714	37,129	38,498	7898	6288
2020/2/20	12,281,154	12,282,230	37,398	38,781	7948	6489
2020/2/21	12,280,840	12,281,727	36,680	39,065	8980	6708
2020/2/22	12,280,299	12,281,205	36,174	39,351	10,027	6944
2020/2/23	12,280,299	12,280,663	36,163	39,639	10,038	7198
2020/2/24	12,279,429	12,280,102	34,691	39,928	12,380	7470
2020/2/25	12,279,059	12,279,520	33,563	40,219	13,878	7761
2020/2/26	12,278,676	12,278,918	32,392	40,512	15,432	8070
2020/2/27	12,278,363	12,278,295	30,179	40,806	17,958	8399
2020/2/28	12,277,943	12,277,651	28,836	41,102	19,721	8747

因此我们可以得到： $S(t)$ 、 $I(t)$ 、 $R(t)$

- 感染人数持续上升： $38215 \rightarrow 41102(+7.55\%)$ ；
- 移除人数增长： $6306 \rightarrow 8747(+38.7\%)$ ；
- 基本再生数 $R_0 = \beta/\gamma \approx 2.06 > 1$ (疫情仍在扩散)；
- 每日新增感染：约 283~296 人。

3.2.4. 结果分析

从易感人群的预测结果中我们可以看出，易感的人数在不断减少，并且在 2020 年 2 月 19 日左右易感的人数开始逐渐趋于平稳，也就是说感染的人数逐渐趋于峰值。因此相应的我们可以从图 8 的感染人数预测图看出，预测感染人数达到峰值也是在 19 日左右，之后感染人数逐渐减少，同时治愈和死亡的人数逐渐增加，感染的人数会逐渐趋于 0。进而，我们从图 9 可以看出，治愈和死亡的人数不断增加，当感染的人数逐渐趋于 0 时，治愈和死亡的人数也会逐渐趋于平稳。

4. 结语

本文所采用的传染病 SIR 模型是传染病模型中最经典的模型，但此模型也存在一定的局限性，适用于在某一封闭的地区内传染病的数值模拟和预测，同时所具有的参数和变量也有限，但此模型相对的也适用于 2020 年武汉疫情封城后的研究和预测，为控制并减少新型冠状病毒传播的隔离强度和治疗等措施提供了参考，促使感染人数的减缓，为相应特效药物的研制和对已患病患者的治疗提供了帮助，为更好的控制疫情。

参考文献

- [1] 原存德, 胡宝安. 具有阶段结构的 SI 传染病模型[J]. 应用数学学报, 2002, 25(2): 193-203.
- [2] 罗荣桂, 江涛. 基于 SIR 传染病模型的技术扩散模型的研究[J]. 管理工程学报, 2006, 20(1): 32-35.
- [3] 刘开源, 陈兰荪. 一类具有垂直传染病与脉冲免疫的 SEIR 传染病模型的全局分析[J]. 系统科学与数学, 2010, 30(3): 323-333.
- [4] 靳祯, 马知恩. 具有连续和脉冲预防接种的 SIRS 传染病模型[J]. 华北工学院学报, 2003, 24(4): 235-243.
- [5] 邓磊, 耀霖. SARS 传染扩散的动力学随机模型[J]. 科学通报, 2003, 48(13): 1373-1377.

-
- [6] 术语在线. 流行病学特征[EB/OL].
<https://baike.baidu.com/item/%E6%B5%81%E8%A1%8C%E7%97%85%E5%AD%A6%E7%89%B9%E5%BE%81/53321724>, 2020-08-21.
- [7] 陆征一, 周义仓. 数学生物学进展[M]. 北京: 科学出版社, 2006.
- [8] 徐文兵, 徐厚宝, 于景元, 朱广田. SARS 数学模型解的存在唯一性与稳定性[J]. 应用泛函分析学报, 2004, 6(2): 140-145.
- [9] 严闯, 陈瑜, 刘可伋, 罗心悦, 许伯熹, 江渝, 程晋. 基于一类时滞动力学系统对新型冠状病毒肺炎疫情的建模和预测[J]. 中国科学: 数学, 2020, 50(3): 385-392.

附录：相关运算代码

```

import numpy as np
# 数据输入
dates = ['2020/1/14', '2020/1/15', '2020/1/16', '2020/1/17', '2020/1/18', '2020/1/19', '2020/1/20',
         '2020/1/23', '2020/1/25', '2020/1/26', '2020/1/27', '2020/1/28', '2020/1/29', '2020/1/30',
         '2020/1/31', '2020/2/1', '2020/2/2', '2020/2/3', '2020/2/4', '2020/2/5', '2020/2/6',
         '2020/2/7', '2020/2/8', '2020/2/9', '2020/2/10', '2020/2/11', '2020/2/12', '2020/2/13',
         '2020/2/14', '2020/2/16', '2020/2/17', '2020/2/18']
cumulative_confirmed = [41,41,45,62,121,198,258,495,618,698,1590,1905,2261,2639,3215,4109,5142,
                        6384,8351,10117,11618,13603,14982,16902,18454,19558,32994,35991,37914,41152,42752,44412]
current_confirmed = [34,27,28,41,94,169,227,441,533,593,1463,1739,2063,2390,2897,3779,4687,
                    5802,7686,9255,10613,12360,13497,15175,16500,17361,30043,33033,34289,
                    36385,37152,38007]

N = 12326500
# 计算 t, S, R
t_values = []
S_values = []
I_values = []
R_values = []
for i in range(len(dates)):
    C = cumulative_confirmed[i]
    I = current_confirmed[i]
    R = C - I #  $R(t) = C(t) - I(t)$ 
    S = N - C #  $S(t) = N - C(t)$ 
    t_values.append(i) # 但日期不连续，需调整实际 t
    S_values.append(S)
    I_values.append(I)
    R_values.append(R)
# 实际 t 基于日期差(以 1 月 14 日为 t=0)
real_t = [0,1,2,3,4,5,6,9,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,33,34,35]
# 计算每个区间的 beta_i 和 gamma_i
beta_list = []
gamma_list = []
intervals = []
delta_t_list = []
for i in range(len(real_t)-1):
    t_i = real_t[i]
    t_next = real_t[i+1]
    delta_t = t_next - t_i

```

```

S_i = S_values[i]
I_i = I_values[i]
R_i = R_values[i]
S_next = S_values[i+1]
R_next = R_values[i+1]
delta_S = S_next - S_i
delta_R = R_next - R_i
# 计算 beta_i 和 gamma_i
if I_i > 0 and S_i > 0:
    beta_i = - (delta_S * N) / (S_i * I_i * delta_t)
    gamma_i = delta_R / (I_i * delta_t)
else: # I_i=0 时未发生, 但安全处理
    beta_i = np.nan
    gamma_i = np.nan
beta_list.append(beta_i)
gamma_list.append(gamma_i)
intervals.append((t_i, t_next))
delta_t_list.append(delta_t)
# 输出结果
print("区间索引 i | 时间区间 [t_i, t_{i+1}) | Δt_i | β_i | γ_i")
for i in range(len(intervals)):
    print(f"{i} | [{intervals[i][0]}, {intervals[i][1]}) | {delta_t_list[i]} | {beta_list[i]:.4f} | {gamma_list[i]:.4f}")
# 输出结果
print("区间索引 i | 时间区间 [t_i, t_{i+1}) | Δt_i | β_i | γ_i")
for i in range(len(intervals)):
    print(f"{i} | [{intervals[i][0]}, {intervals[i][1]}) | {delta_t_list[i]} | {beta_list[i]:.4f} | {gamma_list[i]:.4f}")

```