微分方程模型

Ordinary Differential Equation Model,Ode Model

October 5, 2021

目录

微分方程模型介绍

微分方程

微分方程是含有微分的方程,由此需要解释两个问题：

1. 方程。方程是指**含有未知数的等式**。使等式成立的未知数的值称为“解”或“根”。
2. 微分。微分是对函数的**局部变化率**的一种线性描述。微分可以近似地描述当函数自变量的取值作足够小的改变时，函数的值是怎样改变的。

微分方程模型介绍

例如，牛顿第二定律方程就是一个最典型的微分方程

$$f = ma = m\frac{\mathrm{d}v}{\mathrm{d}t} \tag{1}$$

例如，这是一个一阶微分方程初值问题

$$\begin{cases} \dfrac{\mathrm{d}y}{\mathrm{d}x} = 2 \\ y(0) = 1 \end{cases} \tag{2}$$

大家应该可以非常快速地看出它的解为

$$y = 2x + 1 \tag{3}$$

微分方程模型介绍

微分方程理论博大精深，我们本节仅就几个最经典的微分方程模型进行讨论，希望能够帮助大家理解微分方程建模的基本思想。我们将通过三个模型案例展开微分方程建模这一课程。他们分别是：

- 人口增长模型
- 传染病模型
- 猎物-捕食者模型

人口增长模型

认识人口数量的变化规律，建立人口模型，作出较准确的预报，是有效控制人口增长的前提。下表给出了近两个世纪的美国人口统计数据，我们以此研究人口增长模型。

年份	1790	1800	1810	1820	1830	1840	1850	1860
人口	3.9	5.3	7.2	9.6	12.9	17.1	23.2	31.4

年份	1870	1880	1890	1900	1910	1920	1930	1940
人口	38.6	50.2	62.9	76	92	106.5	123.2	131.7

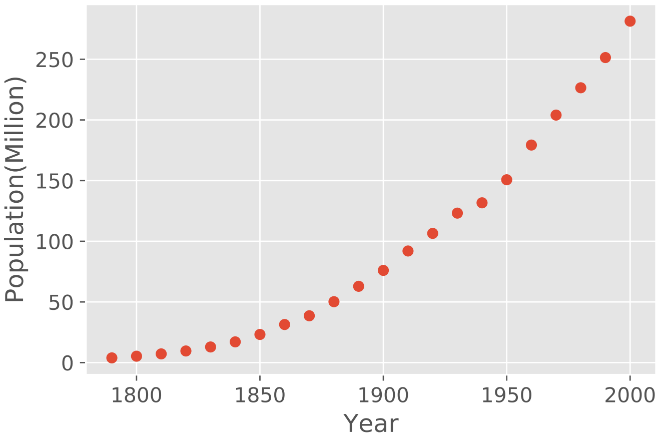
年份	1950	1960	1970	1980	1990	2000
人口	150.7	179.3	204	226.5	251.4	281.4

人口增长模型

先简单通过散点图看一下人口的增长模式。

```python
import matplotlib
import matplotlib.pyplot as plt
plt.style.use('ggplot') # 设置绘图风格为ggplot

year = [i for i in range(1790,2010,10)] # 时间列表
population =
    [3.9,5.3,7.2,9.6,12.9,17.1,23.2,31.4,38.6,
50.2,62.9,76,92,106.5,123.2,131.7,150.7,179.3,204,
226.5,251.4,281.4]
plt.scatter(year,population) # 绘制散点图
plt.xlabel("Year") # 加横坐标label
plt.ylabel("Population(Million)") # 加纵坐标label
plt.show()
```

人口增长模型



Population(Million)
250
200
150
100
50
0
1800
1850
1900
1950
2000
Year

指数增长模型

200多年前英国人口学家T. Malthus (1766-1834)调查了英国100多年的人口统计资料，得出了人口增长率不变的假设，并据此建立了著名的人口指数增长模型。

假设

指数增长模型假设人口增长率r不变

记时刻t的人口为$x(t)$,当考察一个国家或一个较大地区的人口时,$x(t)$是一个很大的整数。为了利用微积分这一数学工具，将$x(t)$视为连续、可微函数，记初始时刻$(t=0)$的人口为x_0,假设人口增长率为常数r,即单位时间内$x(t)$的增长量为r乘以$x(t)$,于是得到$x(t)$满足微分方程

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = rx \\ x(0) = x_0 \end{cases} \tag{4}$$

指数增长模型

由这个方程,我们通过分离变量法很容易解出其解析解:

分离变量法求解微分方程

先分离变量

$$\frac{\mathrm{d}x}{x} = r\mathrm{d}t \tag{5}$$

两边同时积分

$$\int \frac{\mathrm{d}x}{x} = \int r\mathrm{d}t \tag{6}$$

解得:

$$\begin{aligned} x &= e^{rt+C_1} \\ &= Ce^{rt} \end{aligned}$$

当$x = 0$时,

$$x(0) = C = x_0 \tag{7}$$

于是有

$\ln x = \ln x_0 + rt$

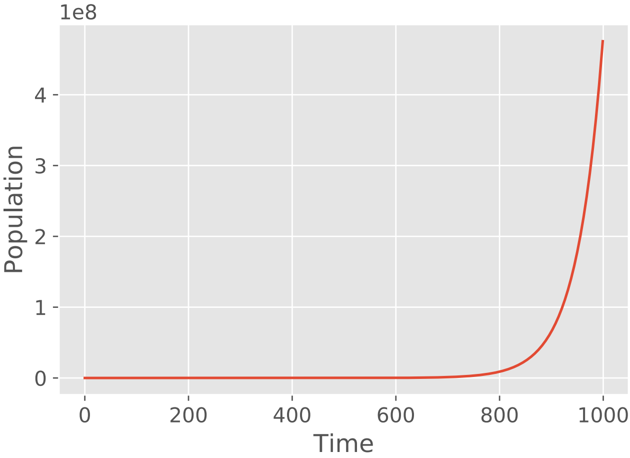
$$x(t) = x_0 e^{rt} \tag{8}$$

指数增长模型

当$r > 0$时，上式表示人口将按指数规律随时间无限增长，于是称为指数增长模型。

```python
# 指数增长模型的绘图
import numpy as np
x0 = 1  # 初始人口数
r = 0.02   # 增长率
t = [i for i in range(1000)] # 时间列表
x_t = [x0 * np.exp(r * time) for time in t] # 人口增长记录
plt.plot(t, x_t) # 绘图
plt.xlabel("Time") # 加横坐标label
plt.ylabel("Population") # 加纵坐标label
plt.show()
```

指数增长模型



我们来尝试用这个模型，带入19世纪的数据，来预测20世纪的人口增长规律。
为了估计指数增长模型中的参数r和x_0，需将原式取对数，得

$$y = rt + a \tag{9}$$

其中，

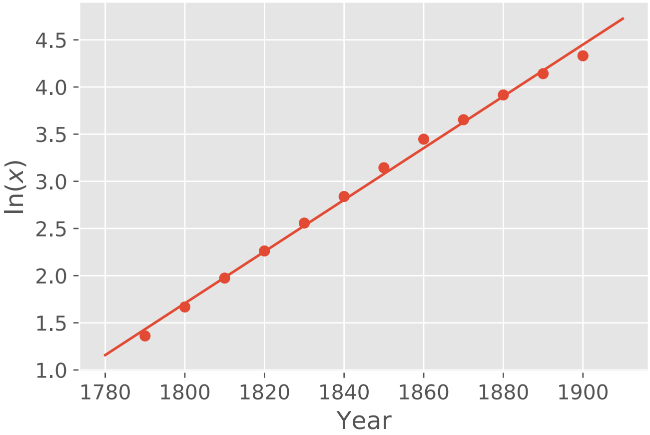
$$y = \ln x, a = \ln x_0 \tag{10}$$

$\ln x = rt + \ln x_0$

模型的参数估计、检验和预报

```
from sklearn.linear_model import LinearRegression
ln_population = np.log(population) # 对人口取对数，得
    到y
plt.scatter(year[0:12],ln_population[0:12])

## 使用LinearRegression 进行线性回归
lrModel = LinearRegression()
lrModel.fit(np.array(year[0:12]).reshape(-1,1),
    ln_population[0:12])   截距                      斜率
ln_population_fit = [ lrModel.intercept_ + lrModel.
    coef_ *i for i in range(1780,1920,10) ]
plt.plot(range(1780,1920,10),ln_population_fit) # 将
    线性回归以后的直线绘制在散点图上
plt.xlabel("Year") # 加横坐标labal
plt.ylabel(r'$\ln(x)$') # 加纵坐标label
plt.show()
```

模型的参数估计、检验和预报



模型的参数估计、检验和预报

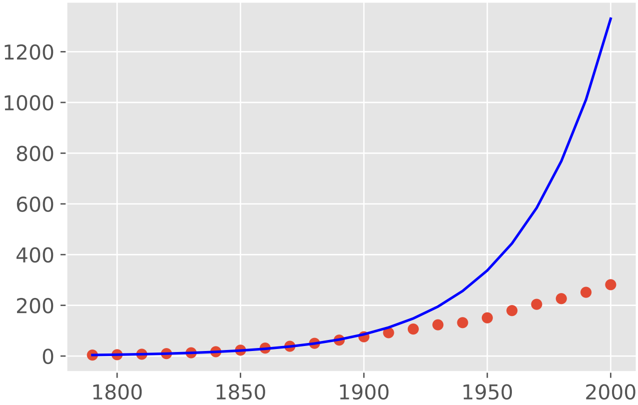
我们把以上的代码合起来，放在一起

```
## 对比一个世纪后的预测值和真实值
#输入原始数据
year = [i for i in range(1790,2010,10)] # 时间列表
population =
    [3.9,5.3,7.2,9.6,12.9,17.1,23.2,31.4,38.6,
50.2,62.9,76,92,106.5,123.2,131.7,150.7,179.3,204,
    226.5,251.4,281.4]
plt.scatter(year,population,label = 'Real Data')
from sklearn.linear_model import LinearRegression
ln_population = np.log(population) # 对人口取对数
lrModel = LinearRegression()
lrModel.fit(np.array(year[0:12]).reshape(-1,1),
    ln_population[0:12])
ln_population_fit = [ lrModel.intercept_ + lrModel.
    coef_ *i for i in range(1780,1920,10) ] # 给出线
    性回归直线表达式
```

模型的参数估计、检验和预报

```python
#将预测值和真实值进行对比
r = lrModel.coef_ # 预测直线增长率
x0 = np.exp(lrModel.intercept_) # 取x0
pop_predicted = [x0 * np.exp(lrModel.coef_ * time) for time in year] # 预测值
plt.plot(year,pop_predicted,'b',label = 'Predicted') #将预测的曲线绘制在散点图上
plt.show()
```

模型的参数估计、检验和预报



显然，使用1790 - 1900的数据得到的指数模型预测效果不好，不能够准确地预测20世纪的人口增长情况。

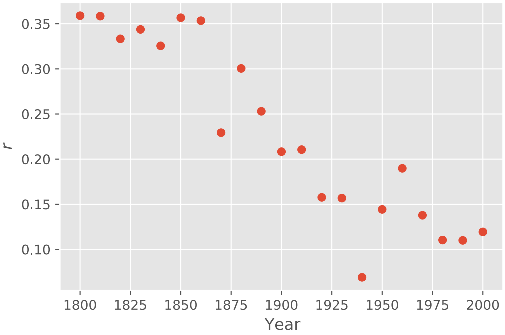
- 历史上，指数增长模型与19世纪以前欧洲一些地区人口统计数据可以很好地吻合，迁往加拿大的欧洲移民后代人口也大致符合这个模型。另外，用它作短期人口预测可以得到较好的结果。**这是因为在这些情况下，人口增长率是常数这个基本假设大致成立。**
- 但是长期来看，任何地区的人口都不可能无限增长，即指数模型不能描述、也不能预测较长时期的人口演变过程。这是因为，人口增长率事实上是在不断地变化着排除灾难、战争等特殊时期，一般说来，当人口较少时，增长较快，即增长率较大；人口增加到一定数量以后，增长就会慢下来，即增长率变小。

模型的参数估计、检验和预报

如果根据上面给出的数据计算一下美国人口的年增长率。

```python
## 增长率可视化
year = [i for i in range(1790,2010,10)] #时间列表
population =
    [3.9,5.3,7.2,9.6,12.9,17.1,23.2,31.4,38.6,
50.2,62.9,76,92,106.5,123.2,131.7,150.7,179.3,204,
226.5,251.4,281.4]
## 计算增长率
rate = []
for i in range(len(population)-1):
    rate.append((population[i+1] - population[i])/
        population[i])
## 可视化
plt.scatter(year[1:],rate)
plt.xlabel("Year") # 加横坐标label
plt.ylabel("r") # 加纵坐标label
plt.show()
```

模型的参数估计、检验和预报



可以看到增长率从19世纪开始就基本上在缓慢下降。如果用一个平均的年增长率作为r，用指数增长模型描述美国人口的变化，会发现结果与实际数据相差很大。看来，为了使人口预报特别是长期预报更好地符合实际情况，必须修改指数增长模型关于人口增长率是常数这个基本假设。

阻滞增长模型—logistic模型

通过分析人口增长到一定数量后增长率下降的主要原因，人们注意到，自然资源、环境条件等因素对人口的增长起着阻滞作用，**并且随着人口的增加，阻滞作用越来越大**。所谓阻滞增长模型就是考虑到这个因素，对指数增长模型的基本假设进行修改后得到的。

阻滞作用体现在对人口增长率r的影响上，使得r随着人口x的增加而下降。若将r表示为x的函数$r(x)$，则它应该是减函数。于是

$$\frac{\mathrm{d}x}{\mathrm{d}t} = r(x)x, x(0) = x_0 \tag{11}$$

阻滞增长模型—logistic模型

对$r(x)$一个最简单的假定是，设$r(x)$为x的线性函数，即

$$r(x) = r - sx (r, s > 0) \tag{12}$$

这里r称固有增长率，表示人口很少时(理论上是$x = 0$的增长率)。为了确定系数s的意义，引入自然资源和环境条件所能容纳的最大人口数量x_m，称人口容量，当$x = x_m$时人口不再增长，即增长率$r(x_m) = 0$. 于是

$$r(x) = r(1 - \frac{x}{x_m}) = r - \frac{r}{x_m} x \tag{13}$$

带回原方程，得到

$$\frac{\mathrm{d}x}{\mathrm{d}t} = rx(1 - \frac{x}{x_m}), x(0) = x_0 \tag{14}$$

方程右端的因子rx体现人口自身的增长趋势，因子$(1 - \frac{x}{x_m})$则体现了环境和资源对人口增长的阻滞作用。
显然，x越大，前一因子越大，后一因子越小，人口增长是两个因子共同作用的结果，上式称为阻滞增长模型。同样地，我们使用分离变量法来求解这个方程。

阻滞增长模型—logistic模型

分离变量法求解微分方程

将原方程分离变量后有 $\dfrac{\mathrm{d}x}{x(1-\frac{x}{x_m})} = r\mathrm{d}t$,等式左边变形$(\frac{1}{x} + \frac{1}{x_m - x})\mathrm{d}x = r\mathrm{d}t$.两边同时积分

$$\int (\frac{1}{x} + \frac{1}{x_m - x})\mathrm{d}x = \int r\mathrm{d}t \tag{15}$$

整理有

$$\ln x - \ln(x_m - x) = \ln(\frac{x}{x_m - x}) = rt + C \tag{16}$$

两边同时取指数

$$\frac{x}{x_m - x} = e^{rt+C} = e^{rt} \cdot e^{C} = C_1 \cdot e^{rt} \tag{17}$$

得到

$$x = \frac{x_m}{1 + C_1 e^{-rt}} \tag{18}$$

阻滞增长模型—logistic模型

$$\frac{\frac{dx}{dt}}{x} = r(1-\frac{x}{x_m}) = r - \frac{r}{x_m}x$$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = rx(1-\frac{x}{x_m}), x(0) = x_0$$

分离变量法求解微分方程

将初始条件$x(0) = \frac{x_m}{1+C_1}$代入，得到$C_1 = \frac{x_m}{x_0} - 1$，最终得到

$$x(t) = \frac{x_m}{1+(\frac{x_m}{x_0}-1)e^{-rt}} \tag{19}$$

In [10]:
```
## 对比一个世纪后的预测值和真实值
#输入原始数据
year = [i for i in range(1790, 2010, 10)] # 时间列表
population =[3.9, 5.3, 7.2, 9.6, 12.9, 17.1, 23.2, 31.4, 38.6, 50.2, 62.9, 76, 92, 106.5, 123.2, 131.7, 150.7, 179.3, 204, 226.5, 251.4, 281.4]
plt.scatter(year, population, label = 'Real Data')
ln_population = np.log(population) # 对人口取对数
lrModel = LinearRegression()
lrModel.fit(np.array(year[0:12]).reshape(-1, 1), ln_population[0:12])
ln_population_fit = [ lrModel.intercept_ + lrModel.coef_ *i for i in range(1780, 1920, 10) ] # 给出线性回归直线表达式

#将预测值和真实值进行对比
r = lrModel.coef_ # 预测直线增长率
x0 = np.exp(lrModel.intercept_) # 取x0
pop_predicted = [x0 * np.exp(lrModel.coef_ * time)for time in year] # 预测值
plt.plot(year, pop_predicted,'b', label = 'Predicted')#将预测的曲线绘制在散点图上
plt.legend()
plt.show()
```

阻滞增长模型—logistic模型

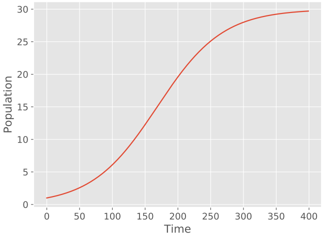
我们可以看一下 logistic模型的增长曲线。

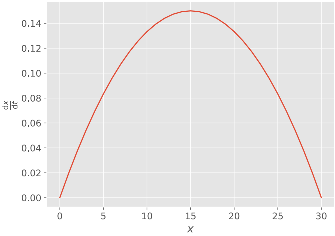
```
## 阻滞增长模型的绘图
## 输入初始参数
x0 = 1
xm = 30
r = 0.02

# 获得x 和 f(x)
t = [i for i in range(400)] # 取合适的值t
x_t = [xm * (1 + (xm/x0 -1)*np.exp(- r * time))
    **(-1) for time in t]

# 绘图增长曲线1:
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
plt.plot(t,x_t) # 绘图
plt.xlabel("Time") # 加横坐标label
plt.ylabel("Population") # 加纵坐标label
```

阻滞增长模型—logistic模型

```python
# 绘图增长速度2:
plt.subplot(1,2,2)
x = [xm*i/30 for i in range(31) ] # 对取值x
deri_x = [r * xx * (1 - xx/xm) for xx in x] # 表达式deri_x
plt.plot(x,deri_x) # 绘图
plt.xlabel(r'$x$') # 加横坐标label
plt.ylabel(r'$\frac{\mathrm{d}x}{\mathrm{d}t}$')
# 加纵坐标label
plt.show()
```

阻滞增长模型—logistic模型





上面的阻滞增长模型，是荷兰生物数学家Verhulst 19世纪中叶提出的，它不仅能够大体上描述人口及许多物种数量（如森林中的树木、鱼塘中的鱼群等）的变化规律，而且在社会经济领域也有广泛的应用，例如**耐用消费品的销售**就可以用它来描述。基于这个模型能够描述一些事物符合逻辑的客观规律，人们常称它为logistic模型。

用阻滞增长模型进行人口预报，先要作**参数估计**。除了初始人口x_0外，还要估计r和x_m。它们可以用人口统计数据拟合得到，也可以辅之以专家对于参数的估计。
原方程可以写为

$$\frac{\frac{\mathrm{d}x}{\mathrm{d}t}}{x} = r - sx, s = \frac{r}{x_m} \tag{20}$$

上式左端可以从实际人口数据用数值微分算出，右端对参数r, s是线性的，r, s可借助线性拟合获得。

模型的参数估计、检验和预报

```python
## 对比一个世纪后的预测值和真实值
import seaborn as sns
from sklearn.linear_model import LinearRegression #

# 输入初始数据
year = [i for i in range(1790,2010,10)] # 时间列表
population =
    [3.9,5.3,7.2,9.6,12.9,17.1,23.2,31.4,38.6,
50.2,62.9,76,92,106.5,123.2,131.7,150.7,179.3,204,
226.5,251.4,281.4]
y = []
for i in range(len(population)-1):
    y.append( (population[i+1] - population[i])/10 /
        population[i] )
plt.figure(figsize = (15,5))
```

$\frac{dx}{dt}/x$

模型的参数估计、检验和预报

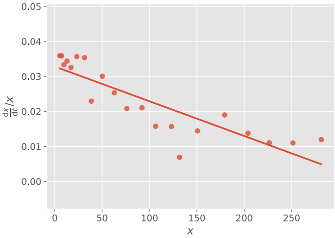
```
# 可视化线性回归结果
plt.subplot(1,2,1)
sns.regplot(population[1:],y,ci = 0)
plt.ylabel(r'$\frac{\mathrm{d}x}{\mathrm{d}t}/{x}$')
    # 加纵坐
    标label
plt.xlabel(r'$x$') # 加横坐标label
lrModel = LinearRegression()
lrModel.fit(np.array(population[7:-1]).reshape(-1,1)
    ,y[6:-1])
# 输入数据
r = lrModel.intercept_
xm = r/(- lrModel.coef_)
```

模型的参数估计、检验和预报

$$x(t) = \frac{x_m}{1 + (\frac{x_m}{x_0} - 1)e^{-rt}}$$

```python
# 可视化拟合结果
plt.subplot(1,2,2)
x0 = population[0]
plt.plot(year,population,label = 'Real Data') # 绘图
pop_predicted = [xm * (1 + (xm/x0 -1)*np.exp(- r * (time - 1790)))**(-1) for time in year] # 预测曲线
plt.plot(year,pop_predicted,label = 'Predicted') # 绘图
plt.xlabel('Year') # 加横坐标label
plt.xlabel('Population') # 加横坐标label
plt.show()
```

模型的参数估计、检验和预报



可以看出，这个模型整体拟合效果不错。我们可以用模型计算2000年的人口，用预测数据与已知的实际数据比较，来检验模型是否合适。