摘要

本文通过**灰色马尔科夫模型**预测 2019 年全国感染该疾病的发病与死亡人数。再通过该预测模型预测各个人群与地区的发病率特征，并基于 **TOPSIS** 得到防控排名前 3 位的重点区域与人群。最后通过**支持向量机回归**建立该传染病与经济发展的数学模型。

针对问题一，在传统**灰色预测模型**的基础上，利用**马尔科夫链**校正预测结果误差，得到**灰色马尔科夫模型**并进行预测。本组首先建立传统 **GM(1,1)** 模型，预测分析 2004-2019 年该流行病的发病人数和死亡人数。再通过**马尔科夫模型**，计算预测残差的期望值。最后利用残差期望校正传统灰色预测的固有偏差。预测得到 2019 年患病人数 **740960** 人、死亡人数 **2211** 人。该模型预测误差低于 **7%**，NSE 值趋近于 1，说明模型可信度很高。

针对问题二，基于问题一模型预测各人群、各地区的患病情况特征，再通过 **TOPSIS** 法选出各人群、各地区排名前三的重点防控对象。首先将附件中各个省市与各个职业的发病、死亡人数带入问题一的**灰色马尔科夫模型**中。预测出 2019 年**发病、死亡人数，发病、死亡人数增长率**和**患病死亡率**。并将上述五个指标带入 TOPSIS 综合评价模型，分别赋予权重 **0.15**、**0.3**、**0.15**、**0.3** 和 **0.1**，计算各人群、各地区的综合评价得分，选取排序前三位的省市——**新疆**、**西藏**和**青海**和排序前三位的职业——**农民**、**家政家务**和**退休人员**分别作为重点防控区域和重点防控人群。

针对问题三，结合各省市经济发展数据，通过**灰色关联度分析**筛选合适指标，并基于**支持向量机回归**建立该传染病与经济发展的数学模型。首先收集 31 个省市各年度的经济发展数据。将经济指标中**灰色关联度**排名前三的**人均 GDP**、**居民消费水平**和**一般预算收入**作为回归指标。再通过 **SVR** 得出经济指标分别对于患病人数和死亡人数的回归分析模型。通过误差检验得到预测误差为 **8%**，NSE 值分别为 **0.87**、**0.94**，预测精度较高。

针对问题四，基于问题二中的传染病传播模型以及对各地区、各人群的综合评价排名，并结合问题三中传染病与经济发展关系的模型结果与分析，给卫生健康委员会相关部门写一封公开信。

本文中所提到的模型优点主要有两点：一、利用马尔可夫模型改进后的灰度预测值与实际值拟合度更高，波动性保持一致；二、针对支持向量回归指标选取，利用灰色关联度筛选合适指标，相较于主观选取指标具有客观性、严谨性。

关键词： **灰色马尔科夫模型　TOPSIS　灰色关联度分析　支持向量机回归**

contents

一、 Introduction

1.1 Background

随着全球化的进程，人类活动范围日益扩大，人群流动频繁，传染病可在大范围内迅速传播，是对人类社会存在威胁的公共卫生问题。在疾病控制实际工作中，疾病的发病与流行趋势分析是极其重要的一环，科学、准确的分析能对卫生行政部分制定疾病预防与控制策略产生重要的影响，传染病早期预警将大大降低传染病的社会经济危害。

为了提高某传染病疫情和突发公共卫生事件报告的质量和时效，加强对全国感染病人的诊断、治疗和督导管理，卫生部建立了全国监管机制, 及时通报相关病情和相关数据, 并通过对疫情数据的动态分析，建立该传染病防治工作督导检查、防治效果评价和制定防治对策和策略，控制并逐渐消灭该传染病。构建预测模型从早期探测到传染病的爆发并及时预警，采取应对措施，是目前传染病防控的重要手段，具有重要的实际意义。

1.2 Work

围绕相关附件和条件要求，研究海运装载行动输送兵力任务的合理安排，依次提出以下问题:

问题一：根据合适的指标建立模型，分析流行病在 2004-2016 年的变化趋势，并预测 2019 年全国感染该病的发病数和死亡数。

问题二：基于 2004-2016 年每隔三年的不同地区的和职业分类的数据，建立疾病传播模型，并预测 2019 年传染病重点防控前 3 名的区域和职业人群。

问题三：结合地区经济发展的相关数据，选择一个角度建立传染病与经济发展相关的模型，并分析结论。

问题四: 综合模型结果及分析，给卫生健康委员会相关部门写一封公开信，谈谈对传染病疫情防治的看法和建议。

1.3 Problem Analysis

Analysis of question one Make reasonable predictions of the aging trend of China and the medical needs of the residents according to the data of residents' income, age structure of the population and the economic development level etc. in the relevant statistical analysis data of the National Bureau of Statistic. 根据国家统计局中 2009 年至 2018 年的相关数据，本组首先选择合适的指标后建立灰色预测模型，预测分析 2009-2018 年该流行病的我国人口老龄化趋势和居民医疗需求，再通过马尔科夫模型，由 2009-2018 年的数据模拟残差在各个区间的分布，计算 2010-2019 年预测残差的期望值。最后将预测结果与残差期望做差，校正传统灰色预测的固有偏差，经过两种模型的结合达到科学预测我国人口老龄化的未来发展和居民医疗需求趋势的目的。其思维流程图如图 **??** 所示：

Analysis of question two

Analysis of question three

Analysis of question four

Analysis of question five

二、 Symbol and Assumptions

2.1 Symbol Description

符号	说明
$X^{(i)}$	人数时间序列
a	发展灰度
u	内生控制灰度

2.2 Fundamental assumptions

(1) 为保证预测结果精确性，假设题目所给出数据真实可信。
(2) 假设重点防控的区域和人群中，发病、死亡人数的增长率比其基数更加重要。
(3) 假设与经济发展无关的该传染病的其它影响因素可以忽略不计。
(4) 假设各省流动人口可以忽略不计，并且传播仅在省内传播。
(5) 忽略各诊断方法的差异对总发病人数与死亡人数的影响。

三、 Establishment and solution of the model

3.1 The model of Problem 1

3.1.1 灰度预测 GM(1,1)

设 2004-2016 年总发病人数为时间序列：

$$X^{(0)} = [x^{(0)}(1), x^{(0)}(2), \cdots, x^{(0)}(13)]$$

通过一次累加生成 1-AGO 序列:

$$X^{(1)} = [x^{(1)}(1), x^{(1)}(2), \cdots, x^{(1)}(13)]$$

式中：$x^{(1)}(k)=\sum_{i=1}^{k}x^{(1)}(i),k=1,2,\cdots,13$。

根据 1-AGO 序列建立微分方程为 [1]：

$$\frac{dX^{(1)}}{dt}+aX^{(1)}=u \tag{1}$$

式中：a 称为发展灰度，u 称为内生控制灰度。设 $\widehat{\alpha}$ 为待估参数向量，且 $\widehat{\alpha}=[a,u]^T$, 利用最小二乘法求出：

$$\widehat{\alpha}=(B^TB)^{-1}B^TY_n \tag{2}$$

求解方程 (1), 可得第 $k+1$ 年传染病发病数**初步预测模型**为：

$$\widehat{X}(k+1)=[X^{(0)}(1)-\frac{u}{a}]e^{-ak}+\frac{u}{a},k=1,2,\cdots,16 \tag{3}$$

同理将死亡数作为向量 $X^{(0)}=[x^{(0)}(1),x^{(0)}(2),\cdots,x^{(0)}(13)]$ 带入模型可求得 2017-2019 年死亡数灰度预测值。

3.1.2 马尔科夫模型校正

利用马尔科夫模型对 GM(1,1) 预测误差项的状态及状态概率进行预估，并利用预测状态的期望值对 GM(1,1) 预测值进行修正 [2]。用 2004-2016 年预测数据与真实数据残差进行状态划分，设残差序列为：

$$\varepsilon=[\varepsilon(1),\varepsilon(2),\cdots,\varepsilon(13)]$$

最大残差绝对值为 $\delta_{max}=\underset{1\leqslant i\leqslant 13}{max}|\varepsilon(i)|$，将预测误差化均分为三个状态。令 $\lambda=\frac{\delta_{max}}{6}$。状态分别为 $E_1:(-3\lambda,-\lambda)$、$E_2:(-\lambda,\lambda)$ 和 $E_1:(\lambda,3\lambda)$。其中初始状态概率向量计算公式为：

$$\begin{cases} p_{Ek}=\frac{n_{Ek}}{13} \\ t_0=[p_{E1},p_{E2},p_{E3}] \end{cases} \tag{4}$$

式中：n_{Ek} 是状态 E_k 在 2004-2016 年内出现的次数，以状态 E_k 出现的频率代替其出现的概率 p_{Ek}。且构建状态转移矩阵为：

$$P=\begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix}$$

式中：P_{ij} 是由状态 E_i 经过一个时期转移到 E_j 的转移概率。

即马尔科夫模型可表示为：

$$t_{k+1} = t_k \cdot p \tag{5}$$

设状态区间的中间值分别为 $\overline{E}_1$、$\overline{E}_2$ 和 $\overline{E}_3$，即第 k 年 GM(1,1) 的误差期望为：

$$\eta = \begin{bmatrix} p_{E1} & p_{E2} & p_{E3} \end{bmatrix} \cdot \begin{bmatrix} \overline{E}_1 \\ \overline{E}_2 \\ \overline{E}_3 \end{bmatrix} \tag{6}$$

当第 k 年的患病人数的 GM(1,1) 预测值为 $\widehat{x}(k)$ 时，**修正后的灰色马尔可夫组合预测模型** $\overline{x}(k)$ 可以记作：

$$\overline{x}(k) = \widehat{x}(k) - \eta \tag{7}$$

3.1.3 预测结果评价指标

均方根误差 (RMSE)、平均相位误差绝对值 (MAPE) 和纳什效率系数 (NSE) 三者是常用来衡量预测结果的指标。RMSE 能评价患病人数和死亡人数中高值的预测结果，其计算公式为：

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^*)^2}$$

均方根误差越小，表明模型可靠性越高，结果越准确。

MAPE 用来评价预测数据中平稳部分的预测结果，其计算公式为：

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i^*}{y_i}\right| \times 100\%$$

MAPE 所求值为绝对值，是一个相对指标，当两个 MAPE 值进行比较时，值越小的说明模型可靠性越高。

NSE 可以用来评价模型的预测能力，其计算公式如下：

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{n}(y_i - y_i^*)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

求得 NSE 值越接近 1，表示模型质量越好，模型可信度越高。接近 0，表示模拟结果接近观测值的平均水平，即总体结果可信，但模拟误差较大。远远小于 0，则模型是不可信的。

3 . 2 solution of the model 1

通过 GM(1,1) 计算 2004-2016 年发病人数预测值得到灰度预测解如下：

$$\widehat{X}(k+1) = -2527359e^{-0.037k} + 3497638, k = 1, 2, \cdots, 16 \tag{8}$$

其误差状态区间如表 1 所示：

表 **1** 发病人数状态区间划分

状态	E_1	E_2	E_3
残差区间	$[-66389, -22130]$	$(-22130, 22130]$	$(22130, 66389]$

根据误差区间范围，将 2004-2016 年发病人数预测值归类于误差区间如表 2 所示：

表 **2** 发病人数误差状态区间

年份	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
状态区间	E_2	E_2	E_1	E_2	E_3	E_2	E_1	E_1	E_2	E_2	E_2	E_2	E_2

由此求得初始状态概率向量 t_0, 转移矩阵 P 为:

$$t_0' = [3/13, 9/13, 1/13]$$

$$P' = \begin{pmatrix} 1/3 & 2/3 & 0 \\ 1/4 & 5/8 & 1/8 \\ 0 & 1 & 0 \end{pmatrix} \tag{9}$$

得到由灰色预测与马尔科夫校正后预测解如图 1 所示:

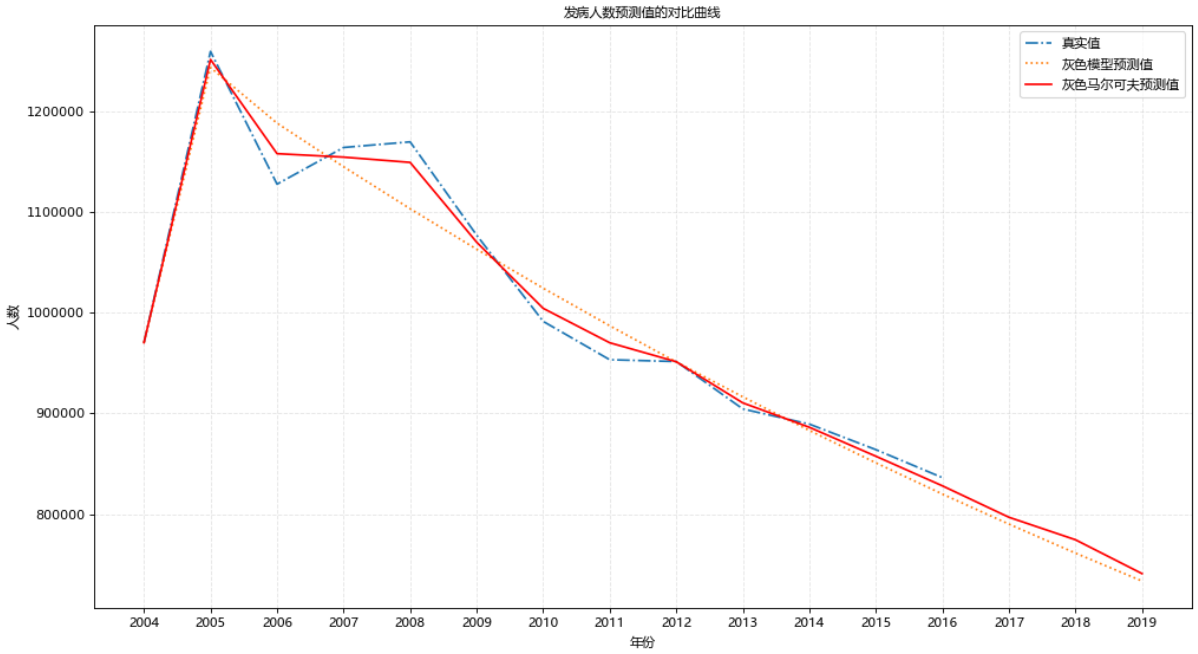

图 **1** 发病人数预测对比曲线图

同理，计算 2004-2016 年死亡人数预测值得到灰度预测解如下：

$$\widehat{X}(k+1) = -92315e^{-ak} + 93750, k = 1, 2, \cdots, 16 \tag{10}$$

其误差状态区间如表 3 所示：

表 **3** 死亡数状态区间划分

状态	E_1	E_2	E_3
残差区间	$[-684, -228]$	$(-228, 228]$	$(228, 684]$

将死亡人数预测值归类于误差区间如表 4 所示：

表 **4** 死亡人数误差状态区间

年份	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
状态区间	E_2	E_2	E_2	E_3	E_1	E_3	E_2	E_2	E_2	E_2	E_1	E_2	E_2

求得初始状态概率向量 t_0', 转移矩阵 P' 为:

$$t_0' = [2/13, 9/13, 2/13]$$

$$P' = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/8 & 3/4 & 1/8 \\ 1/2 & 1/2 & 0 \end{pmatrix} \tag{11}$$

得到由灰色预测与马尔科夫校正后预测解如图 2 所示：

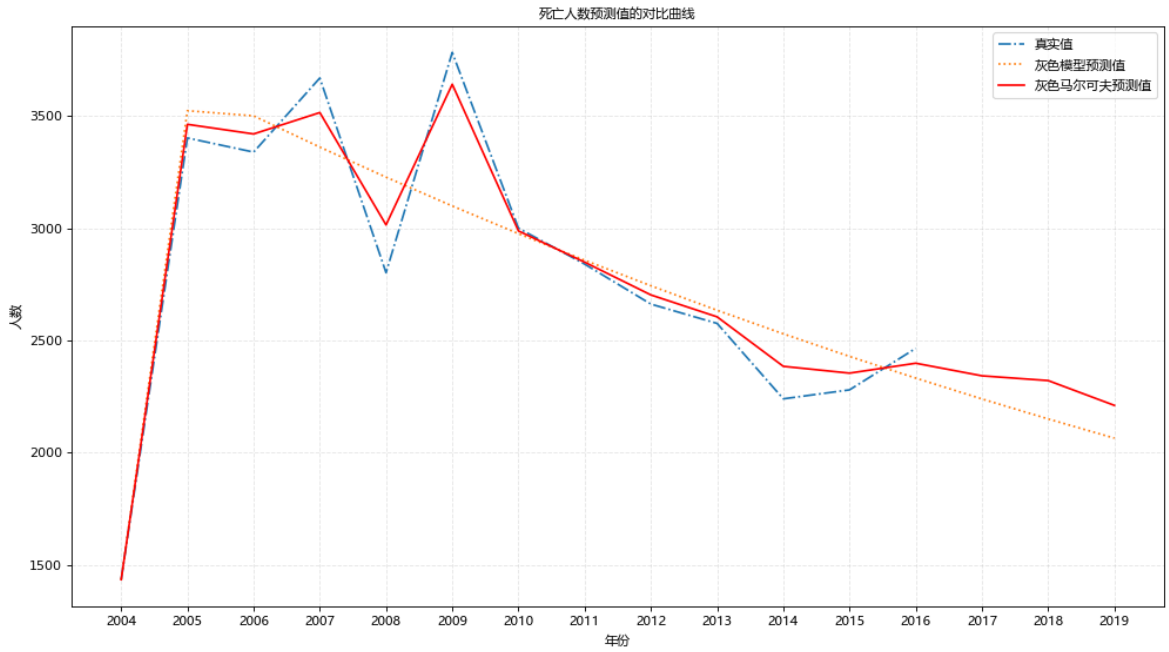

图 **2** 死亡人数预测对比曲线图

3.3 结果分析

根据灰色马尔可夫模型预估出 2019 年全国感染该疾病的发病人数为 7.4096×10^5，死亡人数为 2.211×10^3。由图 1 和图 2 中的预测解曲线直观对比可知，由马尔科夫模型校正后的预测值相较于传统灰色预测值的**拟合度更高**，波动性一致，且较于传统灰色模型预测值更能反应实际值的波动。两种模型预测指标如表 5 所示：

表 **5** 预测结果检验

检验参数	RMSE	MAPE	NSE
传统灰色预测数值 (患病数)	30040.04	0.0213	0.9455
灰色马尔科夫预测数值 (患病数)	12838.64	0.0095	0.9900
传统灰色预测数值 (死亡数)	265.88	0.0628	0.8178
灰色马尔科夫预测数值 (死亡数)	101.13	0.0273	0.9736

从上述的预测结果可以得出: 利用灰色马尔科夫模型修正后求出的患病人数与死亡人数的均方根误差值 RMSE 都小于传统灰色模型，表明**校正后结果可靠性更高**。且修正后模型 MPAE 值较于传统模型更接近 0，NSE 值更接近 1，说明改进后的灰色马尔科夫模型的**拟合程度更高，预测效果更好**，适用于传染病发病数和死亡数的短期预测。

3.4 The model of Problem 2

3.5 The model of Problem 3

3.6 The model of Problem 4

致卫生健康委员会相关部门的一封信:

四、 Sensitivity Analysis

五、 Strengths and Weakness

5.1 模型的优点

(1) 利用马尔可夫模型改进后的灰度预测值与实际值拟合度更高，波动性保持一致，预测的效果更好。
(2) 针对支持向量回归参数选取，利用灰色关联度筛选合适指标，相较于主观选取指标具有客观性、严谨性。

5.2 模型的缺点

问题一、二中的灰色预测模型只能做短期预测，并不适用于长期预测。

六、 模型改进

可以通过序列最小优化算法 (Sequential Minimal Optimization,SMO) 作为样本的训练算法，进而建立序列最小优化支持向量回归模型，从而减小算法复杂度，提高算法的求解速度。

参考文献

[1] Saad Ahmed Javed,Sifeng Liu. Correction to: Predicting the research output/growth of selected countries: application of Even GM (1, 1) and NDGM models[J]. Scientometrics,2019,120(3).

[2] 李立欣, 文海东, 许健开. 基于灰色马尔可夫模型的能源消耗预测 [J]. 中国科技信息,2018(15):74-75.

[3] Yawen Wang,Zhongzhou Shen,Yu Jiang. Analyzing maternal mortality rate in rural China by Grey-Markov model[J]. Medicine,2019,98(6).

[4] Saad Ahmed Javed,Sifeng Liu. Correction to: Predicting the research output/growth of selected countries: application of Even GM (1, 1) and NDGM models[J]. Scientometrics,2019,120(3).

[5] 成枢, 周龙飞, 高秀明. 基于灰色关联 GM(1,N)-Markov 修正模型的应用 [J]. 勘察科学技术,2019(03):43-48.

[6] 刘永阔, 谢春丽, 于竹君, 凌霜寒. 基于 GM(1,1) 模型与灰色马尔可夫 GM(1,1) 模型的核动力装置趋势预测方法研究 [J]. 原子能科学技术,2011,45(09):1075-1079.

[7] Kate Childs,Christopher Davis,Mary Cannon,Sarah Montague,Ana Filipe,Lily Tong,Peter Simmonds,Donald Smith,Emma C. Thomson,Geoff Dusheiko,Kosh Agarwal. Suboptimal SVR rates in African patients with atypical Genotype 1 subtypes: implications for global elimination of Hepatitis C[J]. Journal of Hepatology,2019.

[8] Yuyan Cao. Failure Prognosis for Electro-Mechanical Actuators Based on Improved SMO-SVR Method[A]. 中国航空学会制导、导航与控制分会、飞行器控制一体化技术重点实验室、IEEE 控制系统协会南京分会.Proceedings of 2016 IEEE Chinese Guidance, Navigation and Control Conference (IEEE CGNCC2016)[C].2016:6.

附录 A 代码

A.1 灰色马尔可夫模型–matlab 源代码

```
x = [970279 1259308 1127571 1163959 1169540 1076938 991350 953275 951508 904434 889381 864015
    836236];

%二次拟合预测GM(1,1)模型
sizexd2 = size(x,2);
%求数组长度

k=0;
for y1=x
k=k+1;
if k>1
x1(k)=x1(k-1)+x(k);
%累加生成
z1(k-1)=-0.5*(x1(k)+x1(k-1));
%z1维数减1，用于计算B
yn1(k-1)=x(k);
else
x1(k)=x(k);
end
end
%x1,z1,k,yn1

sizez1=size(z1,2);
%size(yn1);
z2 = z1';
z3 = ones(1,sizez1)';

YN = yn1'; %转置
%YN

B=[z2 z3];
au0=inv(B'*B)*B'*YN;
au = au0';
%B,au0,au

afor = au(1);
ufor = au(2);
ua = au(2)./au(1);
%afor,ufor,ua
%输出预测的 a u 和 u/a的值

constant1 = x(1)-ua;
```

```
afor1 = -afor;
x1t1 = 'x1(t+1)';
estr = 'exp';
tstr = 't';
leftbra = '(';
rightbra = ')';
%constant1,afor1,x1t1,estr,tstr,leftbra,rightbra

strcat(x1t1,'=',num2str(constant1),estr,leftbra,num2str(afor1),tstr,rightbra,'+',leftbra,num2str(ua),rightbra)
%输出时间响应方程

%*****************************************************
%二次拟合

k2 = 0;
for y2 = x1
k2 = k2 + 1;
if k2 > k
else
ze1(k2) = exp(-(k2-1)*afor);
end
end
%ze1

sizeze1 = size(ze1,2);
z4 = ones(1,sizeze1)';
G=[ze1' z4];
X1 = x1';
au20=inv(G'*G)*G'*X1;
au2 = au20';
%z4,X1,G,au20

Aval = au2(1);
Bval = au2(2);
%Aval,Bval
%输出预测的 A,B的值

strcat(x1t1,'=',num2str(Aval),estr,leftbra,num2str(afor1),tstr,rightbra,'+',leftbra,num2str(Bval),rightbra)
%输出时间响应方程

nfinal = sizexd2-1 + 3;
%决定预测的步骤数5 这个步骤可以通过函数传入

%nfinal = sizexd2 - 1 + 1;
%预测的步骤数 1

for k3=1:nfinal
```

```
x3fcast(k3) = constant1*exp(afor1*k3)+ua;
end
%x3fcast
%一次拟合累加值

for k31=nfinal:-1:0
if k31>1
x31fcast(k31+1) = x3fcast(k31)-x3fcast(k31-1);
else
if k31>0
x31fcast(k31+1) = x3fcast(k31)-x(1);
else
x31fcast(k31+1) = x(1);
end
end

end
x31fcast
%一次拟合预测值


for k4=1:nfinal
x4fcast(k4) = Aval*exp(afor1*k4)+Bval;
end
%x4fcast

for k41=nfinal:-1:0
if k41>1
x41fcast(k41+1) = x4fcast(k41)-x4fcast(k41-1);
else
if k41>0
x41fcast(k41+1) = x4fcast(k41)-x(1);
else
x41fcast(k41+1) = x(1);
end
end

end
x41fcast,x
%二次拟合预测值

%***精度检验p C***********////////////////////////////////
k5 = 0;
for y5 = x
k5 = k5 + 1;
if k5 > sizexd2
else
```

```
err1(k5) = x(k5) - x41fcast(k5);
end
end
%err1
%绝对误差


xavg = mean(x);
%xavg
%x平均值

err1avg = mean(err1);
%err1avg
%err1平均值

k5 = 0;
s1total = 0 ;
for y5 = x
k5 = k5 + 1;
if k5 > sizexd2
else
s1total = s1total + (x(k5) - xavg)^2;
end
end
s1suqare = s1total ./ sizexd2;
s1sqrt = sqrt(s1suqare);
%s1suqare,s1sqrt
%s1suqare 残差数列x的方差 s1sqrt 为x方差的平方根S1

k5 = 0;
s2total = 0 ;
for y5 = x
k5 = k5 + 1;
if k5 > sizexd2
else
s2total = s2total + (err1(k5) - err1avg)^2;
end
end
s2suqare = s2total ./ sizexd2;
%s2suqare 残差数列err1的方差S2

Cval = sqrt(s2suqare ./ s1suqare);
%nnn = 0.6745 * s1sqrt
%Cval C检验值

k5 = 0;
pnum = 0 ;
```

```
for y5 = x
k5 = k5 + 1;
if abs( err1(k5) - err1avg ) < 0.6745 * s1sqrt
pnum = pnum + 1;
%ppp = abs( err1(k5) - err1avg )
else
end
end
pval = pnum ./ sizexd2;

%p检验值

%arr1 = x41fcast(1:6)
```

A.2 TOPSIS 分析–matlab 源代码

```
function [ output_args ] = TOPSIS(A,W)

%A为决策矩阵，W为权值矩阵,M为正指标所在的列，N为负指标所在的列
[ma,na]=size(A);       %ma为A矩阵的行数，na为A矩阵的列数
for i=1:na
B(:,i)=A(:,i)*W(i); %按列循环得到[加权标准化矩阵]
end
V1=zeros(1,na);         %初始化正理想解和负理想解
V2=zeros(1,na);
BMAX=max(B);            %取加权标准化矩阵每列的最大值和最小值
BMIN=min(B);            %

for i=1:na
%if i<=size(M,2)     %循环得到理想解和负理想解，注意判断，不然会超个数
V1(i)=BMAX(i);
V2(i)=BMIN(i);
%end
%if i<=size(N,2)
%V1(N(i))=BMIN(N(i));
%V2(N(i))=BMAX(N(i));
%end
end

for i=1:ma              %按行循环求各方案的贴近度
C1=B(i,:)-V1;
S1(i)=norm(C1);     %S1,S2分别为离正理想点和负理想点的距离，用二阶范数

C2=B(i,:)-V2;
S2(i)=norm(C2);
```

```
T(i)=S2(i)/(S1(i)+S2(i)); %T为贴近度
end
C1
C2
output_args=T;
```

A.3 数据可视化–python 源代码

```
from example.commons import Faker
from pyecharts import options as opts
from pyecharts.charts import Bar
from pyecharts.globals import ThemeType
#
    新疆：0.8096，西藏：0.4148，青海：0.4021，广东：0.3517，四川：0.3420，贵州：0.3305.。。。。。。。。。陕西：0.2
#
    农民：0.6886，家政家务，0.4613，退休人员0.3995，公共人员：0.3869，商务人员0.2896，散居儿童：0.2840，工人：0.26
import pandas as pd
import numpy as np
import random
import matplotlib
from matplotlib import pyplot as plt
from matplotlib import font_manager
my_font = font_manager.FontProperties(fname="C:\Windows\Fonts\msyh.ttc")#微软雅黑字体位置
from sklearn.metrics import explained_variance_score,mean_squared_error
def bar_xyaxis_name() -> Bar:
c = (
Bar()
.add_xaxis(["新疆","西藏","青海","广东","四川","贵州","天津"])
.add_yaxis("", [0.6096,0.4148,0.4021,0.3517,0.3420,0.3305,0.3241])
# .add_yaxis("商家B", Faker.values())
.set_global_opts(
title_opts=opts.TitleOpts(title="不同区域TOPSIS打分情况"),
yaxis_opts=opts.AxisOpts(name="监测程度"),
toolbox_opts=opts.ToolboxOpts(),
xaxis_opts=opts.AxisOpts(name="重点监测区域"),
legend_opts=opts.LegendOpts(is_show=False)
)
)
return c

def bar_xyzaxis_name() -> Bar:
c = (
Bar()
.add_xaxis(["农民","家政家务","退休人员","公共人员","商务人员","散居儿童","工人"])
.add_yaxis("", [0.6886,0.4613,0.3995,0.3869,0.2896,0.2840,0.2661])
```

```
# .add_yaxis("商家B", Faker.values())
.set_global_opts(
title_opts=opts.TitleOpts(title="不同区域TOPSIS打分情况"),
# yaxis_opts=opts.AxisOpts(name="监测程度"),
toolbox_opts=opts.ToolboxOpts(),
# xaxis_opts=opts.AxisOpts(name="重点监测人权"),
legend_opts=opts.LegendOpts(is_show=False)
)
)
return c

# bar_xyzaxis_name().render()
# plt.figure(figsize=(15, 8), dpi=80)
x = [2004,2007,2010,2013,2016,2019]
# 两条曲线标注说明,
plt.plot(x,[58323.8, 69485.7 ,68906.70068, 67319.86143 ,59229.19782, 57590.64375
] , label='农民/x10人') # 虚线
plt.plot(x, [54916, 67490.08141 ,78063.99174 ,81181.69584, 106503.6705, 110400.3167
], label='家政家务/人') # 点划线
plt.plot(x, [47991 ,48952.06066 ,43076.23039 ,40811.81632 ,34969.5661, 33507.68682
], label='退休人员/人',color="red") # 点划线

# 设置刻度
_xtick_labels = ["{}年".format(int(i)) for i in x]
plt.xticks(x, _xtick_labels, fontproperties=my_font)
# plt.yticks(range(0, 9))

# 绘制网格
plt.grid(alpha=0.3, linestyle="--") # alpha为透明度 0-1
plt.title("三种重点检测职业分析图", fontproperties=my_font)
plt.xlabel("年份", fontproperties=my_font)
plt.ylabel("患病人数", fontproperties=my_font)
# 标注图例
plt.legend(prop=my_font, loc=0)
plt.show()
```

A.4 灰色关联度分析–python 源代码

```
import pandas as pd
import numpy as np
from numpy import *
import matplotlib.pyplot as plt

import seaborn as sns
```

```
def ShowGRAHeatMap(DataFrame):
colormap = plt.cm.RdBu
f, ax = plt.subplots(figsize=(14, 10.5))
ax.set_title('GRA HeatMap')
sns.heatmap(DataFrame.astype(float),
cmap=colormap,
ax=ax,
annot=True,
yticklabels=14,
xticklabels=10)
plt.show()
def GRA_ONE(gray, m=0):
# 读取为df格式
gray = (gray - gray.min()) / (gray.max() - gray.min())
# 标准化
std = gray.iloc[:, m] # 为标准要素
ce = gray.iloc[:, 0:] # 为比较要素
n, m = ce.shape[0], ce.shape[1] # 计算行列

# 与标准要素比较，相减
a = zeros([m, n])
for i in range(m):
for j in range(n):
a[i, j] = abs(ce.iloc[j, i] - std[j])

# 取出矩阵中最大值与最小值
c, d = amax(a), amin(a)

# 计算值
result = zeros([m, n])
for i in range(m):
for j in range(n):
result[i, j] = (d + 0.5 * c) / (a[i, j] + 0.5 * c)

# 求均值，得到灰色关联值,并返回
return pd.DataFrame([mean(result[i, :]) for i in range(m)])


def GRA(DataFrame):
list_columns = [
str(s) for s in range(len(DataFrame.columns)) if s not in [None]
]
df_local = pd.DataFrame(columns=list_columns)
for i in range(len(DataFrame.columns)):
df_local.iloc[:, i] = GRA_ONE(DataFrame, m=i)[0]
```

```
return df_local


# 从硬盘读取数据进入内存
wine = pd.read_excel("data3.xlsx",index_col="时域")

print(wine.head())

data_wine_gra = GRA(wine)
# data_wine_gra.to_csv(path+"GRA.csv") 存储结果到硬盘
print(data_wine_gra)

# 灰色关联结果矩阵可视化

ShowGRAHeatMap(data_wine_gra)


# import pandas as pd
# x=pd.read_excel('data.xlsx')
x=wine.T

# 1、数据均值化处理
x_mean=x.mean(axis=1)
for i in range(x.index.size):
x.iloc[i,:] = x.iloc[i,:]/x_mean[i]

# 2、提取参考队列和比较队列
ck=x.iloc[0,:]
cp=x.iloc[1:,:]

# 比较队列与参考队列相减
t=pd.DataFrame()
for j in range(cp.index.size):
temp=pd.Series(cp.iloc[j,:]-ck)
t=t.append(temp,ignore_index=True)

#求最大差和最小差
mmax=t.abs().max().max()
mmin=t.abs().min().min()
rho=0.5
#3、求关联系数
ksi=((mmin+rho*mmax)/(abs(t)+rho*mmax))


#4、求关联度
r=ksi.sum(axis=1)/ksi.columns.size
```

```
#5、关联度排序，得到结果r3>r2>r1
result=r.sort_values(ascending=False)

print(result)
```

A.5 SVR–python 源代码

```
\subsection{SMO-SVR--python源代码}
\begin{lstlisting}[language=python]
from __future__ import division
import time
import numpy as np
import pandas as pd
import random
from sklearn.svm import SVR
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import learning_curve
import matplotlib.pyplot as plt
from sklearn.cross_validation import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

wine = pd.read_excel("data3.xlsx",index_col="时域")

print(wine.head())
###############################################################################
# 生成随机数据
x = pd.DataFrame(wine,columns = ['人均国内总产值','居民消费水平','一般预算收入'])
y = wine['死亡数'].values.T

p = wine["预测值"]

print(y)
y_pe = []
for i in range(len(y)):
y_pe.append(random.gauss(0,40)+y[i])
print(y_pe)
# X_plot = np.linspace(0, 5, 100000)[:, None]
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.15, random_state=33)

svr_rbf = SVR(kernel='linear', gamma=0.1, C=100)
# svr_p= SVR(kernel='poly', gamma=0.2, C=100)

svr_rbf.fit(x_train, y_train)
```

```
# svr_p.fit(x_train, y_train)
y_rbf = svr_rbf.predict(x_test)



# y_p = svr_p.predict(x_test)
plt.plot(range(len(y)), y, linewidth=1,label ="True")
plt.plot(range(len(y_pe)), y_pe, linewidth=1,label = "Pre")
# # plt.plot(range(len(y_p)), y_p, linewidth=2,lable = "SMOSVR")
plt.grid(True)
plt.legend()
plt.show()
```