

目录

0 引言.....	1
1 构建 ARIMAX 模型.....	4
1.1 Lasso 变量选择.....	4
1.1.1 Lasso 方法理论介绍.....	4
1.1.2 数据来源与变量选择.....	4
1.2 ARIMAX 模型.....	6
1.2.1 ARIMAX 模型理论介绍.....	6
1.2.2 ARIMAX 模型构建.....	6
1.2.3 ARIMAX 模型拟合与预测评估.....	11
2 构建 BP 神经网络.....	13
2.1 BP 神经网络理论介绍.....	13
2.2 BP 神经网络模型构建.....	13
2.3 BP 神经网络模型拟合与预测评估.....	15
3 构建 ARIMAX-BP 组合模型.....	17
3.1 ARIMAX-BP 组合模型理论介绍.....	17
3.2 ARIMAX-BP 组合模型构建及预测效果评估.....	17
4 总结与展望.....	19
参考文献.....	20
附录.....	21
一、Lasso 变量选择的相关代码.....	21
二、构建 ARIMAX 模型的相关代码.....	22
三、构建 BP 神经网络的相关代码.....	26

福建省 GDP 预测方法研究

摘要：GDP 的合理预测对于政府政策的制定与调控以及人民生活水平的保障与提升具有重要指导意义。本文分别构建 ARIMAX 模型、BP 神经网络和 ARIMAX-BP 组合模型，对 2019-2021 年福建省 GDP 进行预测。组合模型的平均相对预测误差达到 0.36%，具有较好的预测效果，结果表明该模型在 GDP 预测方面具有一定参考意义。

关键词：Lasso 变量选择；ARIMAX 模型；BP 神经网络；组合模型

0 引言

国内生产总值（Gross Domestic Product, 简称 GDP），是指在一定时期内（一般情况下指一年内），一个国家（或地区）的所有常住单位所生产出的全部最终产品价值以及提供的劳动价值的总和，是国民经济核算的核心指标，衡量国民总产出以及收入总规模的重要指标，也是衡量一个国家或地区经济状况和发展水平的重要指标。

20 世纪 80 年代初，中国开始研究联合国国民经济体系的国内生产总值指标；1985 年，中国开始建立 GDP 核算制度；1993 年，中国正式取消国民收入核算，GDP 成为国民经济核算的核心指标；2010 年，中国 GDP 达到 58786 亿美元，成为世界第二大经济体。随着中国经济的飞速发展，中国的 GDP 也在稳步增长。近年来，世纪疫情与百年变局交织叠加，我国经济发展环境的复杂性、严峻性、不确定性上升。在此阶段，合理预测 GDP 取值能够有效促进经济的稳步发展，对于政策制定与人民生活保障具有重要意义。

伴随全球经济的高速发展，学术界对于 GDP 的预测也投入了极大的热情，在理论与实践方面都收获颇丰。总的来说，关于 GDP 的预测研究方向可以分成两大类，一是变量选择，二是模型选择^[1]。

在变量选择方面，也有两种主要思路。一是选取若干指标构建回归模型进而实现对 GDP 的合理预测，如肖争艳以 IS-LM 模型为基础构建了基准指标体系，并基于中国国情特点进一步构建宏观经济指标体系，研究发现该扩展指标能够有效提升 BVAR 模型和 LSTM 模型的 GDP 预测精度^[2]。二是将 GDP 作为自回归变量，构

建 GDP 时间序列模型以实现 GDP 的预测,如张梓运用时间序列分析方法对贵州省 GDP 建立 ARIMA(0, 1, 1)模型,相对误差控制在 7%以内,取得了较好的模型预测效果^[3]。

在模型选择方面,常用的 GDP 预测模型可分为以下几种:传统概率统计模型、灰色预测模型、机器学习模型及其组合模型^[4]。

传统概率统计模型的发展时间相对较长,体系也相对完善,包括自回归移动平均模型、回归模型、指数平滑模型等。王艳梅通过自回归移动平均模型 ARMA、普通最小二乘回归 OLS 以及分位数回归模型对山西省人均 GDP 进行分析研究,揭示山西省人均 GDP 的增长变化规律,为宏观调控政策的制定提供科学依据^[5]。

灰色预测模型是通过少量的、不完全的信息,建立数学模型进而作出预测的研究方法。陈冠霖基于我国东北三省 GDP 及第一、二、三产业数据为样本构建灰色预测及灰色关联度模型,对 GDP 增长情况进行预测,并进一步针对东北三省的经济的发展提出相关经济政策建议^[6]。

机器学习模型是通过算法的不断训练,从大型数据集中寻找模式和相关性,并根据数据分析结果做出最佳决策和预测。王森使用 2012-2018 年夜间灯光影像作为数据源,使用基于贝叶斯优化改进的随机森林算法预测 GDP,预测精度达到 97%,具有较高的准确性和鲁棒性^[7]。

组合模型是考虑到单一模型大多存在一定缺陷,仅仅依靠其本身改进的效果往往不尽人意,但是通过多个模型之间相互取长补短,却可以使得模型通用性和预测精度都得到一定程度的提升,使模型焕发出新的生命力。徐凤基 1978-2018 年进出口贸易量和 GDP 数据,使用 NARX 动态神经网络和 ARIMA 模型作为单项预测,建立组合预测模型,发现组合预测模型效果要优于单项预测^[8]。

考虑到 GDP 数据为时间序列数据,受到多种因素的影响,影响机制也相对复杂。同时注意到金玮对 1978-2004 年中国 GDP 数据进行了随机游走的假设检验,证明我国 GDP 数据可以看作是二阶差分后带随机性趋势的随机游走过程,难以用简单线性模型高度拟合^[9]。

基于上述分析,本文考虑构建 ARIMAX 与 BP 神经网络的组合模型,分别对 GDP 数据中的线性部分信息和非线性部分信息进行提取,进而实现对我国 GDP 数

据的合理预测，并将该组合模型与 ARIMAX 模型和 BP 神经网络的单一模型相比较，进一步体会机器学习模型与传统模型相结合对于 GDP 数据预测效果的提升。

1 构建 ARIMAX 模型

1.1 Lasso 变量选择

1.1.1 Lasso 方法理论介绍

Lasso 方法是一种常用的变量选择方法，本质上是使用模型系数的绝对值函数作为惩罚函数，以实现模型变量系数的压缩，产生稀疏解，通过将一些不是很重要的变量的系数压缩为零，实现降维的目的。

设样本自变量为 $X \in R^p$ ，因变量为 Y ，选择最小二乘函数作为损失函数，则 Lasso 方法的目标函数为：

$$\min_{\beta} Q(\beta_0, \beta) = \min_{\beta} \left[\frac{(Y - X\beta - \beta_0)^T (Y - X\beta - \beta_0)}{n} + \lambda \sum_{i=1}^p |\beta_i| \right] \quad (1)$$

上述 (1) 式等价于 (2) 式所示的带约束条件的优化问题：

$$\begin{aligned} \hat{\beta} &= \arg \min_{\beta} \left\{ \frac{(Y - X\beta - \beta_0)^T (Y - X\beta - \beta_0)}{n} \right\} \\ s.t. \quad &\sum_{i=1}^p |\beta_i| \leq t \end{aligned} \quad (2)$$

其中， λ 为非负正则化参数， $\lambda \sum_{i=1}^p |\beta_i|$ 为惩罚函数，参数 λ 与参数 t 存在一一对应关系。随着 λ 增大，惩罚项变大，保留的变量就越少。

1.1.2 数据来源与变量选择

由于 GDP 的影响因素众多，影响机制也相对复杂，本文在综合大量前人研究基础之上，初步选取与 GDP 高度相关的 12 个指标作为解释变量，分别是第一产业增加值 (First_ind)、第二产业增加值 (Second_ind)、第三产业增加值 (Third_ind)、农林渔业增加值 (af_add)、工业增加值 (ind_add)、建筑业增加值 (con_add)、批发和零售业增加值 (retail_ind)、交通运输仓储和邮政业增加值 (traf_add)、住宿业和餐饮业增加值 (ac_ind)、金融业增加值 (fin_add)、房地产业增加值 (rei_add)、其他行业增加值 (other_add)。本文采用国家统计局所公布数据为数据源，选取 1992-2021 年福建省 GDP 数据以及相关指标数据进行进一步分析研究，其中 1992-2018 年的相关数据为训练集，2019-2021 年的相关数据为测试集。

在正式分析开始之前，首先对变量之间相关性进行检验，得到变量相关图见图 1。其中，左下角存放变量的相关系数，右上角存放变量相关性检验的结果。由图 1 观察可知，这 12 个指标之间存在高度相关性。考虑到解释变量间的高度相关性会导致多重共线性问题，同时过多的解释变量也会导致后续模型过于复杂，在此本文使用 Lasso 方法对这 12 个变量进行进一步筛选，得到这 12 个变量回归系数见表 1。

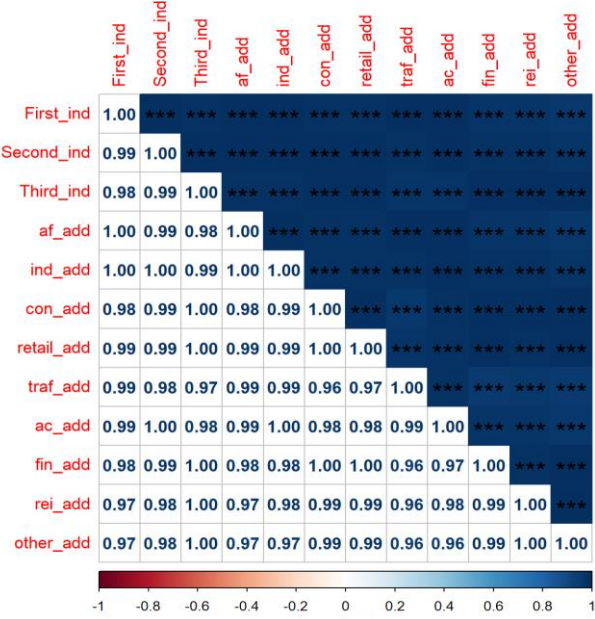


图 1 变量相关图

由表 1 数据可知，第二产业增加值 (Second_ind)、第三产业增加值 (Third_ind) 是与我国 GDP 数据最为相关的。故，本文初步考虑选取上述两个变量作为 ARIMAX 模型的输入变量，并进一步构建有关 GDP 的 ARIMAX 模型。

表 1 Lasso 回归变量系数表

变量名称	回归系数	变量名称	回归系数
First_ind	0.05469244722	retail_add	0.00002234745
Second_ind	0.48415940228	traf_add	0
Third_ind	0.46360389996	ac_add	0.00001298883
af_add	0.00008918140	fin_add	0.00003064628
ind_add	0.00003757063	rei_add	0
con_add	0	other_add	0.00007541960

1.2 ARIMAX 模型

1.2.1 ARIMAX 模型理论介绍

1970 年, Box 和 Jenkins 采用带输入变量的 ARIMA 模型, 为平稳多元时间序列建模。该模型的构建思路如下: 假设响应序列 $\{y_t\}$ 和输入序列 $\{x_{it}\}$ ($i = 1, 2, \dots, k$) 均平稳, 则首先构建响应序列和输入序列的回归模型:

$$y_t = \mu + \sum_{i=1}^k \frac{\theta_i(B)}{\phi_i(B)} B^{l_i} x_{it} + \varepsilon_t \quad (3)$$

其中, $\phi_i(B)$ 为第 i 个输入变量的自回归系数多项式; $\theta_i(B)$ 为第 i 个输入变量的移动平均系数多项式; l_i 为第 i 个输入变量的延迟阶数; $\{\varepsilon_t\}$ 为回归残差序列。

进一步, 使用 ARMA 模型提取残差序列 $\{\varepsilon_t\}$ 中的相关信息:

$$\varepsilon_t = \frac{\theta(B)}{\phi(B)} a_t \quad (4)$$

其中, $\phi(B)$ 为残差序列自回归系数多项式; $\theta(B)$ 为残差序列移动平均系数多项式; a_t 为零均值白噪声序列。

综合式 (3) (4), 最终得到带输入序列的 ARIMAX 模型如下:

$$\begin{cases} y_t = \mu + \sum_{i=1}^k \frac{\theta_i(B)}{\phi_i(B)} B^{l_i} x_{it} + \varepsilon_t \\ \varepsilon_t = \frac{\theta(B)}{\phi(B)} a_t \end{cases} \quad (5)$$

其中, $\phi_i(B)$ 为第 i 个输入变量的自回归系数多项式; $\theta_i(B)$ 为第 i 个输入变量的移动平均系数多项式; l_i 为第 i 个输入变量的延迟阶数; $\{\varepsilon_t\}$ 为回归残差序列; $\phi(B)$ 为残差序列自回归系数多项式; $\theta(B)$ 为残差序列移动平均系数多项式; a_t 为零均值白噪声序列。

1.2.2 ARIMAX 模型构建

由 1.1 中分析可知, 本文选取第二产业增加值 (Second_ind)、第三产业增加值 (Third_ind) 作为 ARIMAX 模型的输入变量, 并进一步构建有关 GDP 的 ARIMAX 模型。

在正式构建 ARIMAX 模型之前，需要首先对输入序列 $\{Second_ind_t\}$, $\{Third_ind_t\}$ 以及输出序列 $\{GDP_t\}$ 进行平稳性检验。为此，绘制 $\{Second_ind_t\}$, $\{Third_ind_t\}$, $\{GDP_t\}$ ，及其对数变换序列，对数变换序列的一阶差分序列的时序图如图 2 所示。

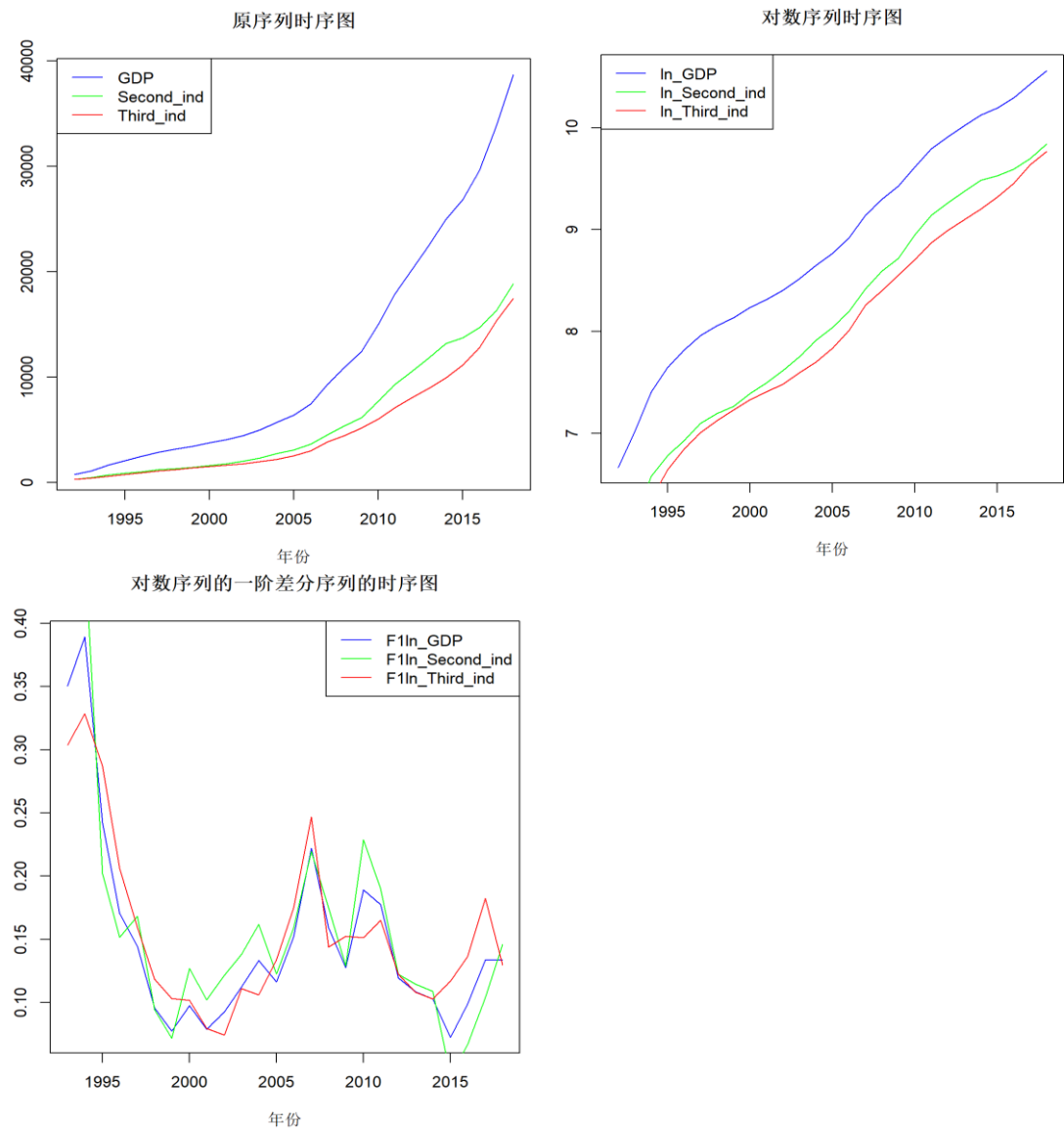


图 2 时间序列时序图

由图 2 观察可知，原序列和对数变换序列显然是不平稳的，但是对数变换序列的一阶差分序列似乎是平稳的。为了准确判断，在此本文对其进行 DF 检验，检验结果如表 2 所示，即对数变换序列的一阶差分序列是平稳的。进一步的，本文对对数变换序列的一阶差分序列进行纯随机性检验，检验结果见表 3，即对数变换序列的一阶差分序列是非纯随机的。

综合上述分析, 本文使用 $\{F1\ln_Second_ind_t\}$, $\{F1\ln_Third_ind_t\}$ 作为输入序列, $\{F1\ln_GDP_t\}$ 作为响应序列构建 ARIMAX 模型。

表 2 对数变换序列的一阶差分序列的 DF 检验结果

序列名称	Dickey-Fuller	P-value
$\{F1\ln_GDP_t\}$	-2.8271	≤ 0.01
$\{F1\ln_Second_ind_t\}$	-3.0445	≤ 0.01
$\{F1\ln_Third_ind_t\}$	-2.0963	0.0382

表 3 对数变换序列的一阶差分序列的纯随机性检验结果

序列名称	延迟阶数	P-value
$\{F1\ln_GDP_t\}$	6	0.00575
$\{F1\ln_GDP_t\}$	12	0.02327
$\{F1\ln_Second_ind_t\}$	6	0.04803
$\{F1\ln_Second_ind_t\}$	12	0.02985
$\{F1\ln_Third_ind_t\}$	6	0.00084
$\{F1\ln_Third_ind_t\}$	12	0.00116

接下来, 对序列进行预白噪声处理。对序列 $\{F1\ln_Second_ind_t\}$ 构建合适的 ARMA 模型。为此, 作序列 $\{F1\ln_Second_ind_t\}$ 的自相关图和偏自相关图如图 3 所示, 即该序列自相关系数拖尾, 偏自相关系数 4 阶截尾。故选择拟合 AR(4) 模型。对该模型残差序列 $\{ln_Second_ind.fit$residuals\}$ 进行白噪声检验, 得到结果见表 4, 即残差序列纯随机, 模型有效。

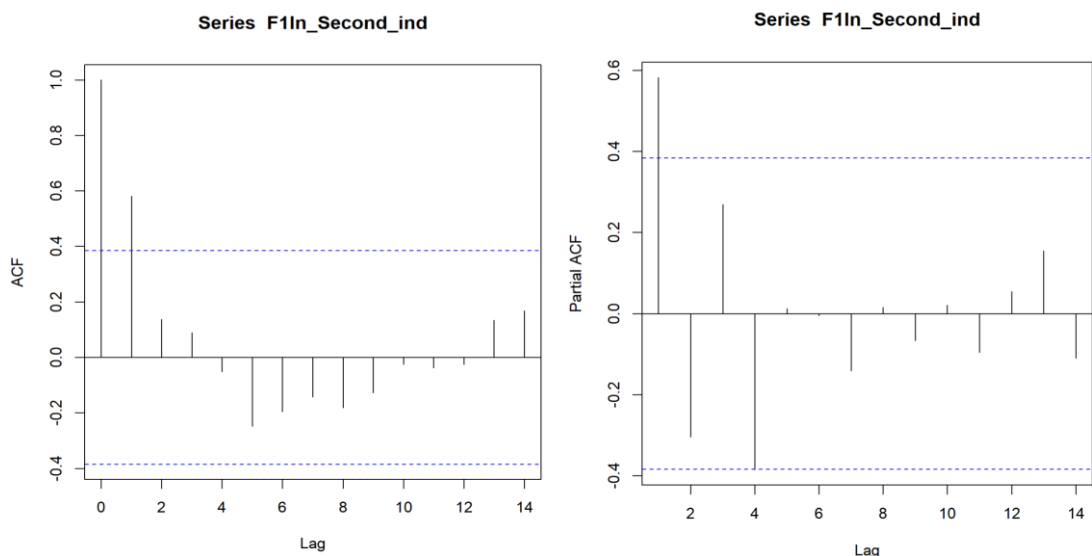


图 3 序列 $\{F1ln_Second_ind_t\}$ 的自相关图和偏自相关图

对序列 $\{F1ln_GDP_t\}$ 作相同变换, 得到残差序列 $\{ln_GDP.fit2$residuals\}$, 并绘制 $\{ln_GDP.fit2$residuals\}$ 与 $\{ln_Second_ind.fit$residuals\}$ 的互相关图, 见图 5。由图 5 可知, 该残差序列之间不存在滞后效应, 可以同期建模。

类似的, 对序列 $\{F1ln_Third_ind_t\}$ 进行相同操作, 绘制其自相关图和偏自相关图如图 4 所示, 即该序列自相关系数拖尾, 偏自相关系数 1 阶截尾。故选择拟合 AR(1)模型。对该模型残差序列 $\{ln_Second_ind.fit$residuals\}$ 进行白噪声检验, 得到结果见表 4, 即残差序列纯随机, 模型有效。

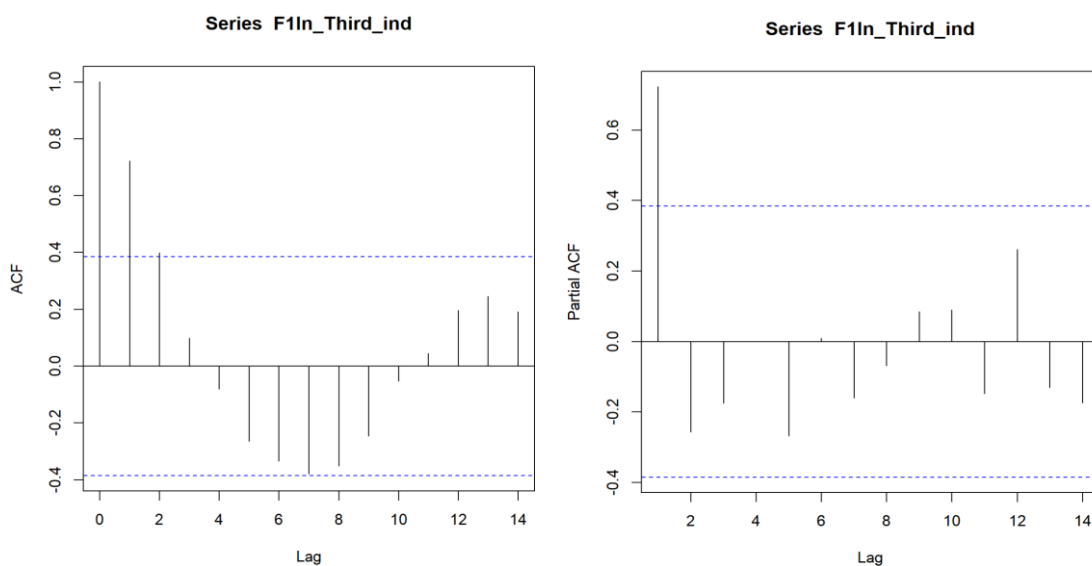


图 4 序列 $\{F1ln_Third_ind_t\}$ 的自相关图和偏自相关图

对序列 $\{F1\ln_GDP_t\}$ 作相同变换, 得到残差序列 $\{\ln_GDP.fit3$residuals\}$, 并绘制 $\{\ln_GDP.fit3$residuals\}$ 与 $\{\ln_Third_ind.fit$residuals\}$ 的互相关图, 见图 5。由图 5 可知, 该残差序列之间不存在滞后效应, 可以同期建模。

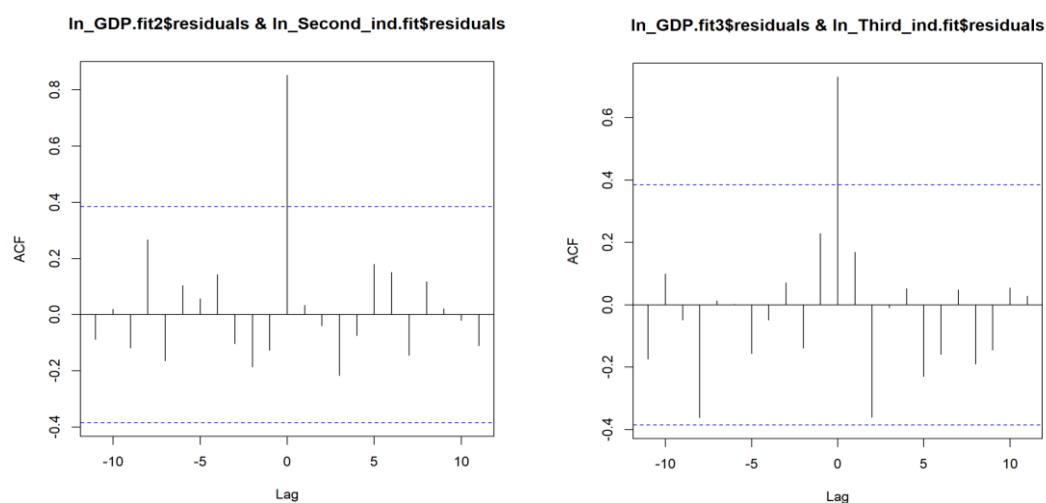


图 5 残差互相关图

表 4 残差序列的纯随机性检验结果

序列名称	延迟阶数	P-value
$\{\ln_Second_ind.fit$residuals\}$	6	0.5697
$\{\ln_Second_ind.fit$residuals\}$	12	0.5902
$\{\ln_Third_ind.fit$residuals\}$	6	0.8243
$\{\ln_Third_ind.fit$residuals\}$	12	0.8359
$\{F1\ln_GDP.fit$residuals\}$	6	0.8728
$\{F1\ln_GDP.fit$residuals\}$	12	0.8543

综上, 将输入序列 $\{F1\ln_Second_ind_t\}$, $\{F1\ln_Third_ind_t\}$ 与响应序列进行同期建模, 通过反复调试, 最终得到 ARIMAX 模型如下:

$$\nabla \ln_GDP = 0.4907\nabla \ln_Second_ind + 0.4944\nabla \ln_Third_ind - 0.0058 \quad (6)$$

或写作:

$$\ln_GDP = 0.4907\ln_Second_ind + 0.4944\ln_Third_ind - \frac{0.0058}{1-B} \quad (7)$$

并对上述 ARIMAX 模型的残差序列{F1ln_GDP.fit\$residuals}进行 ADF 检验和纯随机性检验，得到结果分别如表 5、表 4 所示，即该残差序列为白噪声序列，相关信息已经全部提取。

表 5 残差序列{F1ln_GDP.fit\$residuals}的 DF 检验结果

序列名称	延迟阶数	P-value
{F1ln_GDP.fit\$residuals}	1	0.01
{F1ln_GDP.fit\$residuals}	2	0.04283

1.2.3 ARIMAX 模型拟合与预测评估

使用(6)式所述模型，对 1992-2021 年福建省 GDP 数据进行拟合，得到拟合效果图见图 6，并进一步使用该模型对 2019-2021 年的福建省 GDP 数据计算得到其误差数据见表 6。

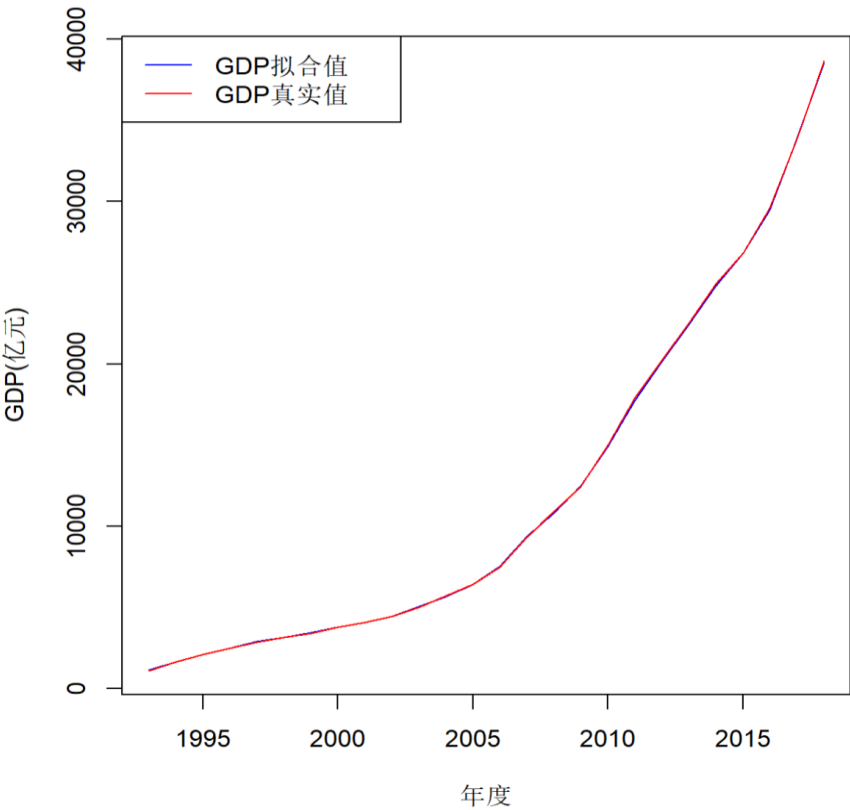


图 6 ARIMAX 模型拟合效果图

表 6 ARIMAX 模型预测结果误差表（单位：亿元）

年份	真实值	预测值	绝对误差	相对误差
2019	42326.6	42067.24	259.3644	0.006127694
2020	43608.6	43281.65	326.9516	0.007497412
2021	48810.4	48616.47	193.9258	0.003973043
平均相对误差：0.005866049				

由表 6 数据分析可知，ARIMAX 模型对于 2019-2021 年福建省 GDP 预测的平均相对误差为0.59%，整体模型预测效果较好。但是仍存在不足，如整体模型的预测结果普遍比真实结果更小，可能存在一定的有偏性。为此，本文在第 3 部分，引入 BP 神经网络对此进行修正，以进一步提高模型预测的准确性。

2 构建 BP 神经网络

2.1 BP 神经网络理论介绍

BP 神经网络，是基于误差逆向传播算法训练的多层前馈神经网络。典型的 BP 神经网络由输入层、隐藏层、输出层三部分组成，见图 7。

其主要思想分为两个部分，一是信息的前向传播——在前向传递中，先将样本数据导入到输入层，中间隐藏层对其进行学习训练，最终在输出层输出预测值；二是误差的反向传播——将输出层得到的预测值与期望值相比较，若二者相差甚远，则再由输出层将信息向前传播，对各层连接权值进行修整，以缩小误差。通过不断重复上述两个过程，对神经网络中的参数进行修正，在误差满足精度要求或迭代次数达到上限时终止迭代，至此训练好的模型可以用于后续的识别与预测。

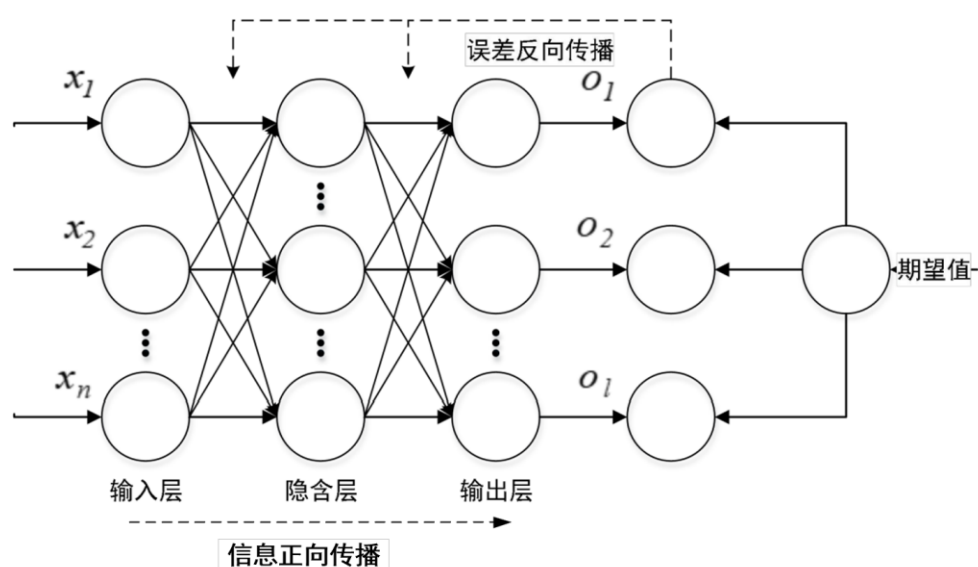


图 7 BP 神经网络结构流程图

2.2 BP 神经网络模型构建

BP 神经网络预测，本质上就是通过对历史数据的不断训练挖掘其中的非线性关系，不断更新修正神经网络中的权值和偏置，进而模拟其发展趋势，对未来发展进行预测。

本文使用基于时间序列的 BP 神经网络的滚动预测方法对于福建省 GDP 进行分析预测。同时，为保证模型效果的可比性，在本章研究中使用 1992-2018 年的福建省 GDP 数据为训练集，2019-2021 年的福建省 GDP 数据为测试集。

在训练样本构建方面，将 GDP 单一指标组为时间序列进行处理分析。具体而言，就是将 GDP 序列的前 M 个序列值和后 N 个序列值之间建立联系，以实现 BP 神经网络的搭建。在此，本文选定输入层的节点数为 5，以减缓运行速度压力，输出层节点数设定为 1，即按照每 5 年的时间间隔周期对 GDP 数据集进行分类，将 $t-1, t-2, t-3, t-4, t-5$ 期数据作为输入值，第 t 期数据作为输出值。按照上述方式，对 1992-2018 年的福建省 GDP 数据进行处理，滚动生成 22 个训练样本（见表 7），用以训练 BP 神经网络。

表 7 GDP 训练样本

滚动次数	x_1	x_2	x_3	x_4	x_5	y
1	784.7	1114.2	1644.4	2094.9	2484.3	2870.9
2	1114.2	1644.4	2094.9	2484.3	2870.9	3159.9
3	1644.4	2094.9	2484.3	2870.9	3159.9	3414.2
.....
21	20190.7	22503.8	24942.1	26819.5	29609.4	33842.4
22	22503.8	24942.1	26819.5	29609.4	33842.4	38687.8

在 BP 神经网络结构方面，考虑到 GDP 数据的自身特点，即其数据结构相对简单，在此仅构建单隐含层的 BP 神经网络模型。同时，考虑到隐藏层节点数的选取一直是 BP 神经网络结构的难题，目前为止并没有相关理论给出有关隐藏层节点数确定的具体方法。在此，仅使用经验公式进行初步判断： $n = \sqrt{N + M} + a$ ， $a \in [1, 10]$ ，其中 n 是隐藏层节点数， N 是输出层节点数， M 是输入层节点数。即本章所构建的 BP 神经网络的隐藏层节点数大致在 3-13 个。通过多次尝试验证，最终取隐藏层节点数为 9 个，隐藏层激活函数为 tan-sigmoid 函数，输出层激活函数为 purelin 函数。

至此，BP 神经网络的基本结构已经明确。在正式搭建 BP 神经网络之前，需要对 GDP 数据进行归一化处理，以消除数据数量级差别的影响，改善神经网络收敛速度、训练时间及预测误差。具体的归一化和反归一化公式如式 (8)、(9) 所示：

$$z_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, i = 1, 2, \dots, n \quad (8)$$

$$x_i = z_i * (x_{max} - x_{min}) + x_{min}, i = 1, 2, \dots, n \quad (9)$$

预备工作完成后，使用 R 语言 AMORE 软件包中的 newff 函数实现 BP 神经网络的搭建。在该宏包中，对于模型性能进行训练预测的函数为 train 和 sim，基础代码如下所示：

```
library(AMORE)
# 创建网络结构
net=newff(n.neurons = c(5,9,1),
          hidden.layer = 'tansig',output.layer = 'purelin',
          learning.rate.global = 1e-3,momentum.global = 0.5,
          error.criterium = 'LMS')
# 训练模型
result=train(net,X_train,Y_train,
             error.criterium = 'LMS', report=TRUE, show.step=1000, n.shows=5)
result
# 计算拟合值
fore.fit=sim(result$net,X_train)
fore.fit
# 计算预测值
fore=sim(result$net,X_test)
fore
```

2.3 BP 神经网络模型拟合与预测评估

基于 newff 函数所构建的 BP 神经网络，对 1997–2021 年福建省 GDP 数据进行拟合，得到拟合效果图见图 7，并进一步使用该模型对 2019–2021 年的福建省 GDP 数据计算得到其误差数据见表 8。

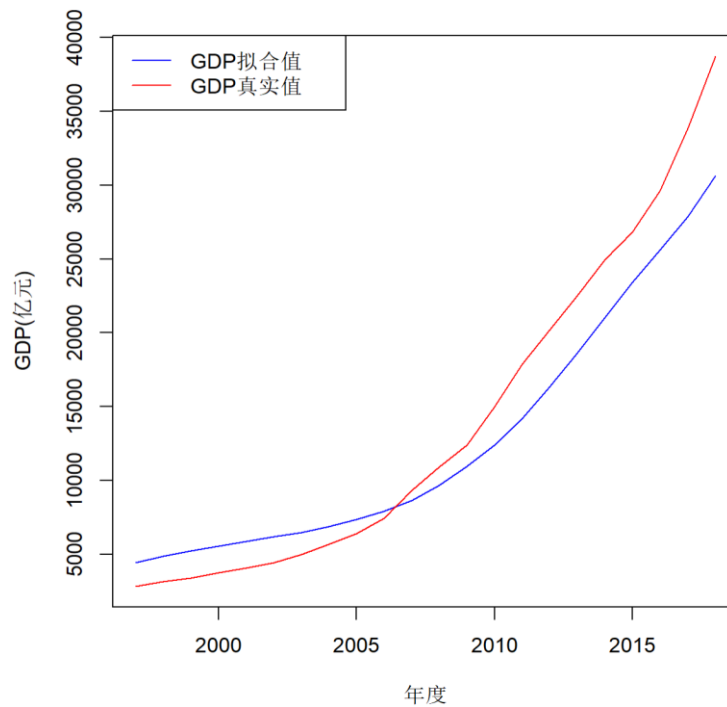


图 7 BP 神经网络拟合效果图

表 8 BP 神经网络预测结果误差表（单位：亿元）

年份	真实值	预测值	绝对误差	相对误差
2019	42326.6	48000.94	-5674.3443	0.134060951
2020	43608.6	48656.34	-5047.7366	0.115750945
2021	48810.4	49221.60	-411.2037	0.008424511
平均相对误差：0.0860788				

由表 8 数据分析可知，BP 神经网络对于 2019-2021 年福建省 GDP 预测的平均相对误差为8.6%，整体模型预测效果较好，具有一定参考价值。但是仍存在不足，如整体模型的预测结果普遍比真实结果更大，且该 BP 神经网络模型的预测效果相比于 ARIMAX 模型而言要逊色不少。综合上述分析，本文在第 3 部分构建 ARIMAX-BP 组合模型对福建省 GDP 数据进行预测，以进一步提高模型预测的准确性。

3 构建 ARIMAX-BP 组合模型

3.1 ARIMAX-BP 组合模型理论介绍

基于前述分析，本章考虑结合 ARIMAX 模型与 BP 神经网络的优点，建立 ARIMAX-BP 组合模型对福建省 GDP 进行预测。基本思路如下所述：

在本文第一部分，已采用 Lasso 方法对与 GDP 相关的多个指标变量进行筛选，得到第二产业增加值(Second_ind)、第三产业增加值(Third_ind)，并以此为输入变量，GDP 序列为输出变量，构建 ARIMAX 模型，实现对福建省 GDP 数据的初步预测。同时注意到该 ARIMAX 模型的残差序列是白噪声序列，无法使用传统模型对其中的相关信息进行进一步提取。

1989 年，Robert Hecht-Nielsen 证明 BP 神经网络可以逼近任意非线性函数，且具有良好的拟合效果。故而，本文考虑使用 BP 神经网络对 ARIMAX 模型的残差序列中的非线性信息进行进一步提取。即，将该残差序列视为时间序列，基于 BP 神经网络的滚动预测方法对其进行拟合预测，得到残差预测值。

最好，将 ARIMAX 模型预测得到的线性部分预测值与 BP 神经网络预测得到的非线性残差预测值相加，所得结果即为组合模型的最终预测值。

3.2 ARIMAX-BP 组合模型构建及预测效果评估

按照上述思路，使用 R 语言进行编程，构建 ARIMAX-BP 组合模型对 1997-2021 年福建省 GDP 数据进行拟合，得到拟合效果图见图 8，并进一步使用该模型对 2019-2021 年的福建省 GDP 数据计算得到其误差数据见表 9。

表 9 ARIMAX-BP 组合模型预测结果误差表（单位：亿元）

年份	真实值	预测值	绝对误差	相对误差
2019	42326.6	42164.27	162.33001	0.003835177
2020	43608.6	43385.22	223.37994	0.005122383
2021	48810.4	48720.85	89.54512	0.001834550
平均相对误差：0.00359737				

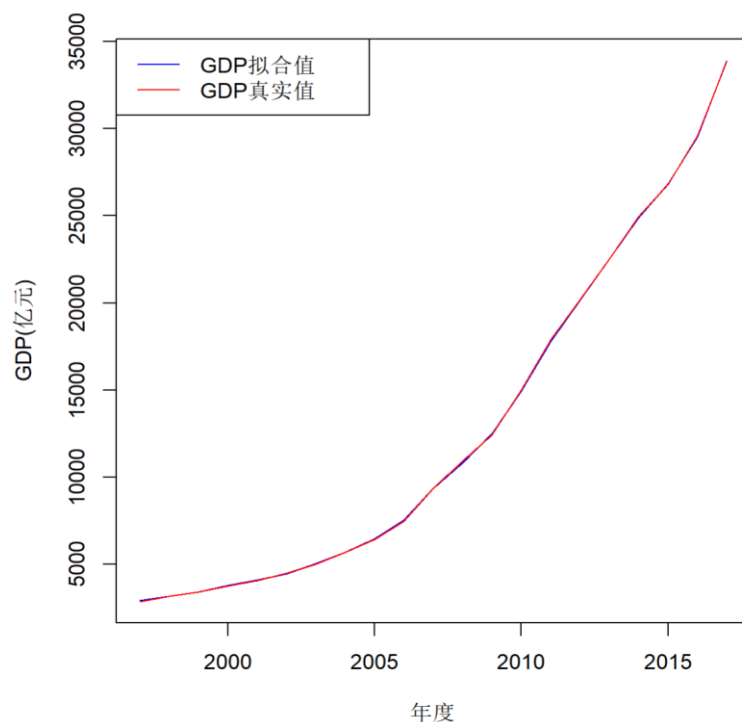


图 8 ARIMAX-BP 组合模型拟合效果图

由表 9 数据分析可知，ARIMAX-BP 组合模型对于 2019-2021 年福建省 GDP 预测的平均相对误差为0.36%。该组合模型相比于单一的 ARIMAX 模型和 BP 神经网络模型，其预测效果都有所提升，整体模型预测效果较好，具有较高的参考价值。

4 总结与展望

本文分别构建了 ARIMAX 模型、BP 神经网络以及 ARIMAX-BP 组合模型，对福建省 GDP 进行了拟合预测。通过前述研究，本文得出以下几点结论：

（1）并非所有的机器学习模型效果都优于传统模型。如本文在 2019-2021 年福建省 GDP 预测中，ARIMAX 模型的平均相对预测误差为**0.59%**，BP 神经网络的平均相对预测误差为**8.6%**，前者要显著地优于后者。具体在何种情况下，使用何种模型，还是要从实际数据结构和数据自身特点出发，不可一概而论。

（2）在单一模型的效果欠佳时，不妨考虑采用适当的模型加以组合，这或许会有较好的结果。如本文 2019-2021 年福建省 GDP 预测中，BP-ARIMAX 组合模型的平均相对预测误差为**0.36%**，要优于单一的 ARIMAX 模型和 BP 神经网络。

（3）ARIMAX-BP 组合模型在 GDP 预测方面效果较好，预测结果具有较高的参考价值。可以进一步地使用该模型对 GDP 数据作进一步预测，为后续的经济政策的制定和调整提供科学依据和指导方向。

参考文献

- [1] 贾万龙,王万雄. 基于 BP 神经网络的合肥都市圈 GDP 预测[J]. 西安电子科技大学学报(社会科学版), 2022, 32(1):37-44.
- [2] 肖争艳,刘玲君,赵廷蓉,等. 深度学习神经网络能改进 GDP 的预测能力吗?[J]. 经济与管理研究, 2020, 41(7):3-17.
- [3] 张梓. 基于 ARIMA 模型的贵州省 GDP 分析与预测[J]. 国土与自然资源研究, 2022(5):39-41.
- [4] 朱佳俊. 关于我国 GDP 的预测方法研究[D]. 山东:山东大学, 2019.
- [5] 王艳梅,陈希镇,董乃铭. 山西省人均 GDP 的发展态势——基于时间序列、最小二乘回归、分位数回归方法[J]. 科学技术与工程, 2012, 12(18):4575-4578.
- [6] 陈冠霖. 基于灰色模型的东北三省 GDP 预测分析[J]. 中国商论, 2022(3):10-13.
- [7] 王森,蔡小莉,鲍云飞,等. 基于 NPP/VIIRS 夜间灯光图像的 GDP 智能预测模型研究[J]. 科学技术与工程, 2022, 22(8):3191-3201.
- [8] 徐凤,常裕琦,朱家明,等. 基于组合预测模型对中国经济外贸依存度的预测[J]. 安徽大学学报(自然科学版), 2021, 45(5):37-44.
- [9] 金玮. 对 1978 年以来中国 GDP 事实特征的认识——基于实际商业周期理论[J]. 西安电子科技大学学报(社会科学版), 2007, 17(6):67-73.

附录

一、Lasso 变量选择的相关代码

```
# 移除所有对象
rm(list = ls())
# 使后续数据直观表示，而不使用科学计数法
options(scipen=200)

# 读取数据
rawdata <- read.table(file.choose(), header = T, sep = ",")
# 显示数据概况
colnames(rawdata)=c('year','First_ind','Second_ind','Third_ind',
                    'af_add','ind_add','con_add','retail_add',
                    'traf_add','ac_add','fin_add','rei_add',
                    'other_add','GDP')

skimr::skim(rawdata)
# 数据整理
data=as.matrix(rawdata)
x=data[,2:13]      # 自变量
y=data[,14]        # 因变量

##### 变量相关性分析 #####
# 方法一
library(ggcorrplot)
corr=cor(x,method='pearson')
ggcorrplot(corr,method = "circle",lab=T)
# 判断结果：
# 发现变量中存在高度相关性，为此使用 Lasso 进行变量选择

# 方法二
library(corrplot)
corr=cor(x,method='pearson')
# 变量相关性显著性检验结果
testRes=cor.mtest(x,conf.level=0.95)
# 下三角放相关系数
corrplot(corr,method='number',type='lower',tl.pos='lt')
# 上三角放色块叠加显著性星号
corrplot(corr,method='color',type='upper',add=T,
          tl.pos='n',cl.pos='n',diag=F,p.mat=testRes$p,
          sig.level=c(0.001,0.01,0.05),pch.cex=1.5,insig='label_sig')
# 判断结果：
# 发现变量中存在高度相关性，为此使用 Lasso 进行变量选择
```

```
##### 变量选择 #####
# 数据标准化
data1=as.data.frame(rawdata[,2:14])
zdata=as.data.frame(scale(data1))
data=as.matrix(zdata)
x=data[,1:12]      # 自变量
y=data[,13]        # 因变量

# LASSO 回归作变量选择
library(lars)
larx<-lars(x,y,type = "lasso") # 进行 Lasso 回归，结果存入 larx
larx                      # 查看 Lasso 回归的结果
plot(larx)                # 用图表形式查看 Lasso 回归结果
summary(larx)             # 输出数据的细节，查看 Cp、Df、Step
larx$Cp[which.min(larx$Cp)] # 直接选取最小的 Cp 值
larx$beta                 # 得到每一步对应的自变量对应的系数
coef=coef.lars(larx,mode="step",s=14) # 获取指定自变量系数，s=step+1
coef[coef!=0]             # 获取系数值不为零的自变量对应的系数值
# 判断结果：
# 选择 Second_ind, Third_ind 作为后续模型构建的所使用的输入变量
```

二、构建 ARIMAX 模型的相关代码

```
##### 数据导入 #####
# 移除所有对象
rm(list = ls())
# 使后续数据直观表示，而不使用科学计数法
options(scipen=200)
# 读取数据
rawdata <- read.table(file.choose(), header = T, sep = ",")
# 显示数据概况
colnames(rawdata)=c('year','First_ind','Second_ind','Third_ind',
                    'af_add','ind_add','con_add','retail_add',
                    'traf_add','ac_add','fin_add','rei_add',
                    'other_add','GDP')
skimr::skim(rawdata)

##### 数据预处理 #####
# 前言：以 1992-2018 年的数据为训练集，2019-2021 年的数据为测试集
# 选择 Second_ind, Third_ind 作为后续模型构建的所使用的输入变量；GDP 为响应变量
GDP=ts(rawdata[1:27,14],start=1992)
Second_ind=ts(rawdata[1:27,3],start=1992)
Third_ind=ts(rawdata[1:27,4],start=1992)
```

```

# 作时序图
plot(GDP,type='l,col='blue',xlab='年份',ylab=' ',main='原序列时序图')
lines(Second_ind,type='l,col='green')
lines(Third_ind,type='l,col='red')
legend('topleft',lty=c(1,1,1),col=c("blue",'green','red'),
      legend=c('GDP','Second_ind','Third_ind'))    # 添加图例

# 对原始数据作对数变换
ln_GDP=log(GDP)
ln_Second_ind=log(Second_ind)
ln_Third_ind=log(Third_ind)
# 作时序图
plot(ln_GDP,type='l,col='blue',xlab='年份',ylab=' ',main='对数序列时序图')
lines(ln_Second_ind,type='l,col='green')
lines(ln_Third_ind,type='l,col='red')
legend('topleft',lty=c(1,1,1),col=c("blue",'green','red'),
      legend=c('ln_GDP','ln_Second_ind','ln_Third_ind'))

# 一阶差分，并检验一阶差分序列的平稳性
F1ln_GDP=diff(ln_GDP,1,1)
F1ln_Second_ind=diff(ln_Second_ind,1,1)
F1ln_Third_ind=diff(ln_Third_ind,1,1)
# 作时序图
plot(F1ln_GDP,type='l,col='blue',xlab='年份',ylab=' ',main='对数序列的一阶差分序列的时序图')
lines(F1ln_Second_ind,type='l,col='green')
lines(F1ln_Third_ind,type='l,col='red')
legend('topright',lty=c(1,1,1),col=c("blue",'green','red'),
      legend=c('F1ln_GDP','F1ln_Second_ind','F1ln_Third_ind'))
# 作平稳性检验
library(fUnitRoots)
print(adfTest(F1ln_GDP,lag=1,type='nc'))
print(adfTest(F1ln_Second_ind,lag=1,type='nc'))
print(adfTest(F1ln_Third_ind,lag=1,type='nc'))
# 判断结果：
# 拒绝原假设：一阶差分序列平稳

# 作白噪声检验
for (i in 1:2)
  print(Box.test(F1ln_GDP,lag=6*i))
for (i in 1:2)
  print(Box.test(F1ln_Second_ind,lag=6*i))
for (i in 1:2)
  print(Box.test(F1ln_Third_ind,lag=6*i))

```


判断结果：拒绝原假设，即对数一阶差分序列非纯随机，可以用于拟合 ARIMAX 模型

构建 ARIMAX 模型

step1:对输入序列 F1ln_Second_ind, F1ln_Third_ind 分别拟合 ARMA 模型

对于 ln_Second_ind

acf(F1ln_Second_ind)

pacf(F1ln_Second_ind)

判断结果:ACF 拖尾，PACF 拖尾

模型拟合

ln_Second_ind.fit=arima(F1ln_Second_ind,order=c(4,0,0))

ln_Second_ind.fit

残差白噪声检验

for (i in 1:2)

print(Box.test(ln_Second_ind.fit\$residuals,lag=6*i))

判断结果：不拒绝原假设，即残差序列纯随机，模型有效

对于 ln_Third_ind

acf(F1ln_Third_ind)

pacf(F1ln_Third_ind)

判断结果:ACF 拖尾，PACF1 阶拖尾

模型拟合

ln_Third_ind.fit=arima(F1ln_Third_ind,order=c(1,0,0))

ln_Third_ind.fit

残差白噪声检验

for (i in 1:2)

print(Box.test(ln_Third_ind.fit\$residuals,lag=6*i))

判断结果：不拒绝原假设，即残差序列纯随机，模型有效

step2:仿照输入序列建模结果对输出序列 F2ln_GDP 拟合 ARMA 模型，并绘制残差互相关图，分析滞后效应

对于 ln_Second_ind

ln_GDP.fit2=arima(F1ln_GDP,order=c(4,0,0))

ccf(ln_GDP.fit2\$residuals,ln_Second_ind.fit\$residuals)

对于 ln_Third_ind

ln_GDP.fit3=arima(F1ln_GDP,order=c(1,0,0))

ccf(ln_GDP.fit3\$residuals,ln_Third_ind.fit\$residuals)

判断结果：无滞后效应，可以同期建模

step3: 拟合 ARIMAX 模型

模型拟合

library(forecast)

data=as.matrix(cbind(F1ln_Second_ind,F1ln_Third_ind))

colnames(data)=c('F1ln_Second_ind','F1ln_Third_ind')

```

auto.arima(y=F1ln_GDP,xreg=data)
F1ln_GDP.model=Arima(y=F1ln_GDP,xreg=data,order=c(0,0,0))
F1ln_GDP.model
# data=as.matrix(cbind(ln_Second_ind,ln_Third_ind))
# colnames(data)=c('ln_Second_ind','ln_Third_ind')
# auto.arima(y=ln_GDP,xreg=data)
# GDP.fit=arima(x=ln_GDP,xreg=data,order=c(1,0,0))
# GDP.fit

# 残差序列平稳性检验
# 对残差序列进行平稳性检验
library(fUnitRoots)
for (i in 1:2)
  {print(adfTest(F1ln_GDP.fit$residuals,lag=i,type='nc'))}
# 判断结果：拒绝原假设，即认为残差序列平稳
# 对残差序列进行纯随机性检验
for (i in 1:2)
  print(Box.test(F1ln_GDP.fit$residuals,lag=6*i))
# 判断结果：不拒绝原假设，即认为残差序列纯随机

##### 模型拟合 #####
F1ln_GDP.model$fitted      # 对数差分序列的拟合值
GDP.fit=exp(ln_GDP[1:26]+F1ln_GDP.model$fitted)  # 变换得到原序列的拟合值
# 绘制拟合效果图
GDP.fit=ts(GDP.fit,start=1993)
GDP.true=ts(GDP[2:27],start=1993)
ts.plot(GDP.fit,GDP.true,gpars=list(col=c("blue","red")),
        xlab='年度',ylab='GDP(亿元)')
legend('topleft',lty=c(1,1),col=c("blue","red"),
       legend=c('GDP 拟合值','GDP 真实值'))

##### 模型预测及效果评估 #####
library(forecast)
# 测试集数据导入
GDP=ts(rawdata[27:30,14],end=2021)
Second_ind=ts(rawdata[27:30,3],end=2021)
Third_ind=ts(rawdata[27:30,4],end=2021)
# 对原始数据作对数变换
ln_GDP=log(GDP)
ln_Second_ind=log(Second_ind)
ln_Third_ind=log(Third_ind)
# 作一阶差分

```

```

F1ln_GDP=diff(ln_GDP,1,1)
F1ln_Second_ind=diff(ln_Second_ind,1,1)
F1ln_Third_ind=diff(ln_Third_ind,1,1)
# 模型预测
data=as.matrix(cbind(F1ln_Second_ind,F1ln_Third_ind))
F1ln_GDP.fore=forecast(F1ln_GDP.model,h=3,xreg=data)
F1ln_GDP.fore          # 对数差分序列的预测
F1ln_GDP.forecast=as.vector(c(0.08374488,0.02231299,0.10870807))
GDP.fore=exp(ln_GDP[1:3]+F1ln_GDP.forecast)  # 变换得到原序列的拟合值
# 计算预测误差
GDP.true.test=GDP[2:4]
AE=GDP.true.test-GDP.fore  # 绝对误差
AE
RE=abs(AE)/GDP.true.test  # 相对误差
RE
mre=mean(abs(RE))          # 平均相对误差
mre

```

三、构建 BP 神经网络的相关代码

```

##### 准备工作 #####
# 移除所有对象
rm(list = ls())
# 使后续数据直观表示，而不使用科学计数法
options(scipen=200)

# 归一化函数
pre_uni <- function(x)
{
  a=min(x)
  b=max(x)
  x=(x-a)/(b-a)
  return(x)
}

# 反归一化函数
anti_uni <- function(x,x_pre)
{
  a=min(x_pre)
  b=max(x_pre)
  x=(b-a)*x+a
  return(x)
}

```

```

##### 数据导入 #####
# 读取数据
rawdata <- read.table(file.choose(), header = T, sep = ",")
# 显示数据概况
colnames(rawdata)=c('year','First_ind','Second_ind','Third_ind',
                    'af_add','ind_add','con_add','retail_add',
                    'traf_add','ac_add','fin_add','rei_add',
                    'other_add','GDP')
skimr::skim(rawdata)

##### 数据预处理 #####
# 前言：选择历年 GDP 作为时间序列建立网络结构模型
# 输入节点取 5 个，输出节点取 1 个
# 数据归一化处理：最小-最大规范化
GDP=rawdata[,14]
data=pre_uni(GDP)

# 数据拆分:以 1992-2018 年的数据为训练集，2019-2021 年的数据为测试集
# data_train=data[1:27]
# data_test=data[28:30]

# 训练集赋值
# 生成训练样本：以滑动窗口的方式生成输入序列和输出序列
# 训练样本个数=27-（5+1）+1=22
# 预先声明矩阵，用于储存输入序列
X_train=matrix(rep(0,times=22*5),nrow=22,ncol=5)
# 滑动窗口，生成输入序列
for (i in 1:22)
{
  for (j in 1:5)
  {
    X_train[i,j]=data[i+j-1]
  }
}
# 输出序列
Y_train=as.vector(data[6:27])

# 测试集赋值
# 输入序列
X_test=matrix(rep(0,times=3*5),nrow=3,ncol=5)
# 滑动窗口，生成输入序列
for (i in 1:3)

```

```

{
  for (j in 1:5)
  {
    X_test[i,j]=data[22+i+j-1]
  }
}
# 输出序列
Y_test=as.vector(data[28:30])

##### 构建 BP 神经网络 #####
# 创建网络：单隐藏层的 3 层神经网络
# 输入节点 5 个；输出节点 1 个
# 隐藏层节点数：3~13 个
# 激活函数：隐藏层 tan-sigmod,输出层 purelin
library(AMORE)
# 创建网络结构
net=newff(n.neurons = c(5,9,1),hidden.layer = 'tansig',output.layer = 'purelin',
          learning.rate.global = 1e-3,momentum.global = 0.5,error.criterium = 'LMS')
# 训练模型
result=train(net,X_train,Y_train,
             error.criterium = 'LMS', report=TRUE, show.step=1000, n.shows=5)
result

##### 模型评估 #####
# step1: 绘制拟合曲线与真实曲线
# 计算拟合值
fore.fit=sim(result$net,X_train)
fore.fit
# 反归一化，得到拟合值
GDP.true.train=rawdata[6:27,14]
GDP.fit=anti_uni(fore.fit,GDP.true.train)
GDP.fit
fit=ts(GDP.fit,start=1997)
true=ts(GDP.true.train,start=1997)
ts.plot(fit,true,gpars=list(col=c("blue","red")),
        xlab='年度',ylab='GDP(亿元)')
legend('topleft',lty=c(1,1),col=c("blue","red"),
       legend=c('GDP 拟合值','GDP 真实值'))

# step2: 计算预测误差
# 计算预测值
fore=sim(result$net,X_test)
fore

```

```

# 反归一化，得到预测值
GDP.true.test=rawdata[28:30,14]
GDP.fore=anti_uni(fore,GDP.true.test)
GDP.fore
# 计算预测误差
AE=GDP.true.test-GDP.fore    # 绝对误差
AE
RE=abs(AE)/GDP.true.test    # 相对误差
RE
mre=mean(abs(RE))          # 平均相对误差
mre

```

四、构建 ARIMAX-BP 组合模型的相关代码

```

##### 数据导入 #####
# 移除所有对象
rm(list = ls())
# 使后续数据直观表示，而不使用科学计数法
options(scipen=200)
# 读取数据
rawdata <- read.table(file.choose(), header = T, sep = ",")
# 显示数据概况
colnames(rawdata)=c('year','First_ind','Second_ind','Third_ind',
                    'af_add','ind_add','con_add','retail_add',
                    'traf_add','ac_add','fin_add','rei_add',
                    'other_add','GDP')
skimr::skim(rawdata)

##### 构建 ARIMAX 模型 #####
# 前言：以 1992-2018 年的数据为训练集，2019-2021 年的数据为测试集
# 选择 Second_ind, Third_ind 作为后续模型构建的所使用的输入变量；GDP 为响应变量
# 测试集数据导入
GDP=ts(rawdata[1:27,14],start=1992)
Second_ind=ts(rawdata[1:27,3],start=1992)
Third_ind=ts(rawdata[1:27,4],start=1992)
# 对数变换
ln_GDP=log(GDP)
ln_Second_ind=log(Second_ind)
ln_Third_ind=log(Third_ind)
# 一阶差分
F1ln_GDP=diff(ln_GDP,1,1)
F1ln_Second_ind=diff(ln_Second_ind,1,1)
F1ln_Third_ind=diff(ln_Third_ind,1,1)
# 构建 ARIMAX 模型

```

```

library(forecast)
data=as.matrix(cbind(F1ln_Second_ind,F1ln_Third_ind))
colnames(data)=c('F1ln_Second_ind','F1ln_Third_ind')
F1ln_GDP.model=Arima(y=F1ln_GDP,xreg=data,order=c(0,0,0))
F1ln_GDP.model

##### 残差序列构建 BP 神经网络 #####
#### 准备工作 ####
# 归一化函数
pre_uni <- function(x)
{
  a=min(x)
  b=max(x)
  x=(x-a)/(b-a)
  return(x)
}
# 反归一化函数
anti_uni <- function(x,x_pre)
{
  a=min(x_pre)
  b=max(x_pre)
  x=(b-a)*x+a
  return(x)
}

##### 数据预处理 #####
# 前言：选择 ARIMAX 模型的残差序列 F1ln_GDP.model$residuals 作为时间序列，建立网
络结构模型
# 输入节点取 5 个，输出节点取 1 个
# 数据归一化处理：最小-最大规范化
F1ln_GDP.model$residuals
F1ln_GDP.resid=F1ln_GDP.model$residuals
data=pre_uni(F1ln_GDP.resid)

# 训练集赋值
# 生成训练样本：以滑动窗口的方式生成输入序列和输出序列
# 训练样本个数=26-（5+1）+1=21
# 预先声明矩阵，用于储存输入序列
X_train=matrix(rep(0,times=21*5),nrow=21,ncol=5)
# 滑动窗口，生成输入序列
for (i in 1:21)
{
  for (j in 1:5)
  {

```

```

        X_train[i,j]=data[i+j-1]
    }
}
# 输出序列
Y_train=as.vector(data[6:26])

# 测试集赋值
# 输入序列
X_test=X_train[19:21,]
# X_test=matrix(rep(0,times=3*5),nrow=3,ncol=5)
## 滑动窗口，生成输入序列
# for (i in 1:3)
# {
#   for (j in 1:5)
#   {
#     X_test[i,j]=data[21+i+j-1]
#   }
# }

##### 构建 BP 神经网络 #####
# 创建网络：单隐藏层的 3 层神经网络
# 输入节点 5 个；输出节点 1 个
# 隐藏层节点数：3~13 个
# 激活函数：隐藏层 tan-sigmod,输出层 purelin
library(AMORE)
# 创建网络结构
net=newff(n.neurons = c(5,9,1),hidden.layer = 'tansig',output.layer = 'purelin',
          learning.rate.global = 1e-3,momentum.global = 0.5,error.criterion = 'LMS')
# 训练模型
result=train(net,X_train,Y_train,
             error.criterion = 'LMS', report=TRUE, show.step=1000, n.shows=5)

result

##### 计算拟合值 #####
fore.fit=sim(result$net,X_train)
fore.fit
# 反归一化，得到拟合值
F1ln_GDP.residuals.train=F1ln_GDP.model$residuals
F1ln_GDP.residuals.fit=anti_uni(fore.fit,F1ln_GDP.residuals.train)
F1ln_GDP.residuals.fit

```



```
##### 组合模型拟合效果图 #####
# 组合模型的拟合值
# F1ln_GDP.model$fitted      # ARIMAX 模型中对数差分序列的拟合值
# F1ln_GDP.residuals.fit     # BP 神经网络模型中关于对数差分序列残差的拟合值
GDP.fit=exp(ln_GDP[5:25]+F1ln_GDP.model$fitted[5:25]+F1ln_GDP.residuals.fit)
# 绘制拟合效果图
GDP.fit=ts(GDP.fit,start=1997)
GDP.true=ts(GDP[6:26],start=1997)
ts.plot(GDP.fit,GDP.true,gpars=list(col=c("blue","red")),
        xlab='年度',ylab='GDP(亿元)')
legend('topleft',lty=c(1,1),col=c("blue","red"),
       legend=c('GDP 拟合值','GDP 真实值'))

##### 组合模型预测效果评估 #####
#### 计算 ARIMAX 模型的预测值 ####
library(forecast)
# 测试集数据导入
GDP=ts(rawdata[27:30,14],end=2021)
Second_ind=ts(rawdata[27:30,3],end=2021)
Third_ind=ts(rawdata[27:30,4],end=2021)
# 对原始数据作对数变换
ln_GDP=log(GDP)
ln_Second_ind=log(Second_ind)
ln_Third_ind=log(Third_ind)
# 作一阶差分
F1ln_GDP=diff(ln_GDP,1,1)
F1ln_Second_ind=diff(ln_Second_ind,1,1)
F1ln_Third_ind=diff(ln_Third_ind,1,1)
# ARIMAX 模型的预测值
data=as.matrix(cbind(F1ln_Second_ind,F1ln_Third_ind))
F1ln_GDP.fore=forecast(F1ln_GDP.model,h=3,xreg=data)
F1ln_GDP.fore      # 对数差分序列的预测
F1ln_GDP.forecast=as.vector(c(0.08374488,0.02231299,0.10870807))

#### 计算 BP 神经网络对于残差序列的预测值 ####
# 计算预测误差
F1ln_GDP.residuals.fore=sim(result$net,X_test)
F1ln_GDP.residuals.fore
# 反归一化, 得到预测值
F1ln_GDP.residuals.true=F1ln_GDP.model$residuals[24:26]
F1ln_GDP.residuals.fore=anti_uni(F1ln_GDP.residuals.fore,F1ln_GDP.residuals.true)
F1ln_GDP.residuals.fore
```

```

##### 计算 ARIMAX-BP 组合模型预测值及预测误差 #####
GDP.fore=exp(ln_GDP[1:3]+F1ln_GDP.forecast+F1ln_GDP.residuals.fore)
GDP.fore
# 计算预测误差
GDP.true.test=GDP[2:4]
# 计算预测误差
AE=GDP.true.test-GDP.fore      # 绝对误差
AE
RE=abs(AE)/GDP.true.test      # 相对误差
RE
mre=mean(abs(RE))             # 平均相对误差
mre

```

全文检测报告

基本信息

报告编号: 2022121102407785774A795811

文档名称: 曾淑慧+福建省GDP预测方法研究

过滤操作: 已过滤参考文献

文档作者: 曾淑慧

已过滤自引"曾淑慧"的相似影响

提交方式: 上传文档检测

提交时间: 2022年12月11日

正文字符数: 19193

检测范围: 大雅全文库

总体结论

文献相似度: 15.45%

文献原创度: 84.55%

去除参考文献相似度: 15.45%

单篇最大相似度: 1.5%

去除本人已发表论文相似度: 15.45%

单篇最大重复数: 302

重复字符数: 2966

最相似文献题名: 福州主城区大气环境污染时空特征分析及质量评价

过滤前的相似度: 16.99%

过滤前的重复字数: 2966

相似片段分布



典型相似文献

相似图书

序号	题名	作者	出处	相似度
1	统计预测与决策	魏艳华;王丙参;郝淑双	成都: 西南交通大学出版社, 2014.05	1.12%
2	地理信息技术在公共卫生与健康领域的应用	施迅;王法辉	北京: 高等教育出版社, 2016.02	0.98%
3	应用时间序列分析	王燕	北京: 中国人民大学出版社, 2005.07	0.84%
4	应用时间序列分析 第3版	王燕	北京: 中国人民大学出版社, 2012.12	0.84%
5	高级计量经济学	李宝仁;王琴英;祝金甫;辛士波	北京: 经济科学出版社, 2012.10	0.84%
6	应用时间序列分析	王燕	北京: 中国人民大学出版社, 2015.12	0.84%
7	基于R应用的统计学丛书 时间序列分析 基于R	王燕	北京: 中国人民大学出版社, 2015.09	0.82%
8	精算中常用的统计模型	黄向阳;金阳;肖宇谷	北京: 中国人民大学出版社, 2009.02	0.71%
9	时间序列分析实验	李正辉;李庭辉	北京: 中国统计出版社, 2010.08	0.61%
10	经济学基础	王国娜	北京: 中国财政经济出版社, 2017.09	0.51%
11	改革开放以来我国犯罪问题的宏观分析	张威	北京: 中国政法大学出版社, 2018.05	0.51%
12	利率市场化与货币政策传导机制	李庆华	北京: 中国石化出版社, 2015.06	0.51%