**实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**： **应用回归分析实验** | **学期： 2023年秋季学期** | **成绩**： |
| **指导教师**： 陈丹 | **学生姓名**： **枫叶** | **学生学号**： |
| **实验名称**： **有偏估计** | | |
| **实验编号**：6 | **实验日期**： | **实验学时**： |
| **学院： 数学与统计学院** | **专业： 21统计学** | **年级**： **2021级** |

**一、实验目的**

掌握有偏估计的常用方法

**二、使用环境**

R4.3.2

**三、实验内容**

掌握有偏估计的常用方法

1. **算法介绍及结果**

本实验选用R自带的Loneley数据集，该数据集常用于多重共线性处理方法的测试，由于该数据集为时间序列数据，且观测年份仅16年，故不随机划分训练集和测试集，而是以前15年数据为训练集，以第16年数据为测试集[[1]](#footnote-1)。

选定Employed为因变量，其他变量为自变量，首先画出相关系数热力图（图1，画×的表示显著性检验不通过），在0.05显著性水平下，可以认为自变量之间存在较强的相关关系，这势必导致严重的多重共线性，下面进行进一步分析。

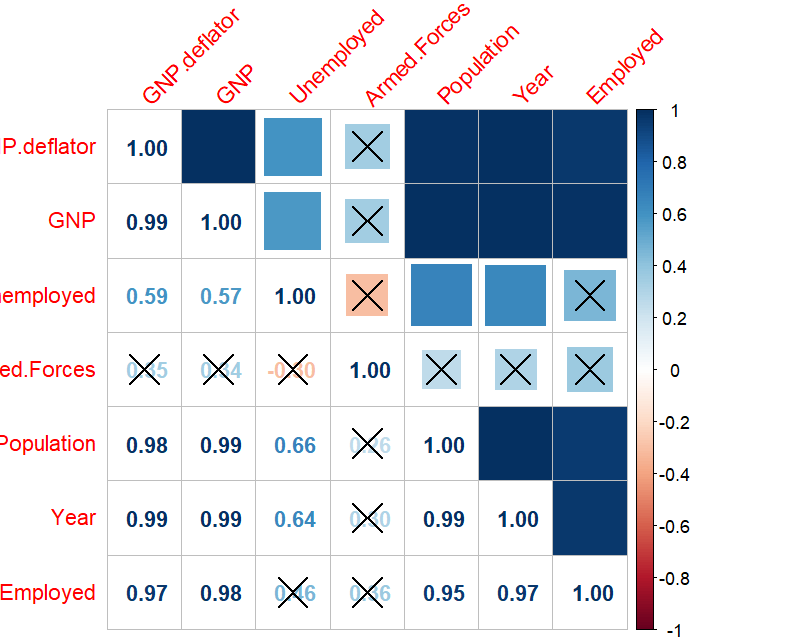


图 1

OLS回归结果如表1所示，值得注意的是军队扩招会使得就业人数下降，但事实上军队扩招往往是解决就业问题的一个重要手段，该项系数明显不合理。OLS回归的VIF值列于表2中，除Armed.Forces外，其余变量的VIF均远大于10，这说明存在严重的的多重共线性问题，下面分别尝试使用岭估计和主成分估计解决。

表 1 OLS回归结果

|  | OLS |
| --- | --- |
| (Intercept) | -3467.961\*\* |
|  | (871.584) |
| GNP.deflator | 0.035 |
|  | (0.085) |
| GNP | -0.034 |
|  | (0.033) |
| Unemployed | -0.020\*\* |
|  | (0.005) |
| Armed.Forces | -0.010\*\* |
|  | (0.002) |
| Population | -0.098 |
|  | (0.225) |
| Year | 1.823\*\* |
|  | (0.446) |
| R2 Adj. | 0.992 |
| + p < 0.1, \* p < 0.05, \*\* p < 0.01, \*\*\* p < 0.001 | |

表 2 VIF值

| 变量 | GNP.deflator | GNP | Unemployed | Armed.Forces | Population | Year |
| --- | --- | --- | --- | --- | --- | --- |
| VIF | 110.86 | 1487.70 | 32.03499 | 3.09 | 353.57 | 625.12 |

使用岭估计方法，画出岭迹图（图2），注意到随着的增加，Armed.Forces的系数逐渐收敛于零，其他变量的系数也都实现了收敛，且符号未发生改变，结合岭迹图与Hoerl-Kennard公式，选取值为0.005，并将岭估计的系数还原为原始变量系数，结果如表3所示

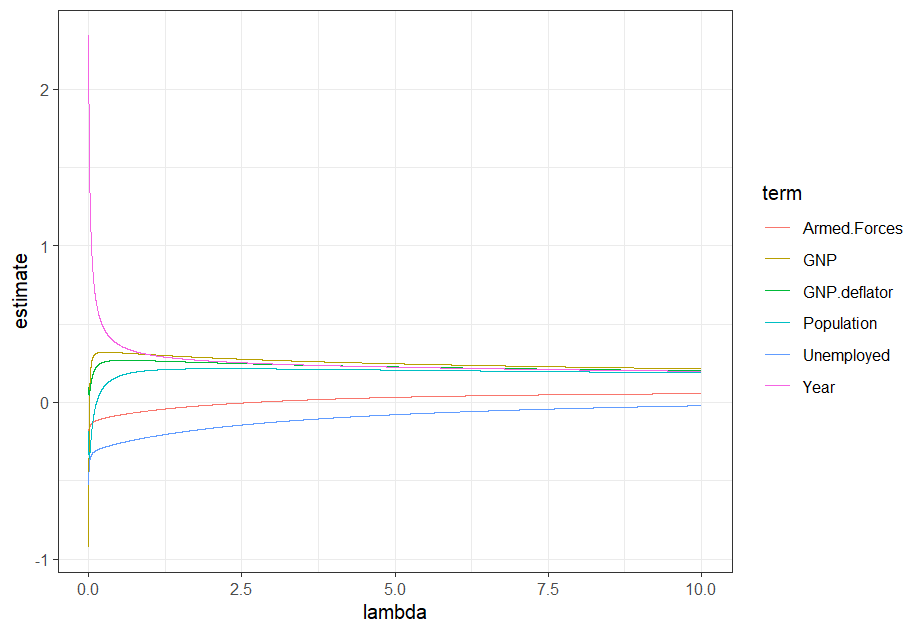


图 2 岭迹图

令人遗憾的是岭回归估计的系数中Armed.Forces仍为负，不过其系数相对更趋于零，此外，即使将岭参数选为10，该项系数依旧不会发生数量级上的下降，而选取更高的岭参数意味着更高的SSE，故岭估计对该样本的效果并不是很理想。

表 3 岭回归系数估计

|  | Ridge |
| --- | --- |
| (Intercept) | -2655.740 |
| GNP.deflator | 0.016 |
| GNP | -0.011 |
| Unemployed | -0.016 |
| Armed.Forces | -0.009 |
| Population | -0.170 |
| Year | 1.408 |

接下来使用主成分估计方法，各主成分的方差累积贡献率如图3所示，注意到第一个主成分的方差贡献率已经达到了90%，故只保留第一个主成分，对应的系数如表4所示，可以看到Armed.Forces系数已经为正，但此时出现了新的问题，失业人数的系数为负，这显然有违常理。

事实上，本文所选取变量具有一定缺陷，譬如关于就业问题有一个重要的公式：奥肯定律（公式(1)），其中为实际产出，为潜在产出，为实际失业率，为自然失业率，这意味着我们的模型中至少应该包含产出增长率和自然增长率等，若考虑更复杂的情况，譬如考虑消费者的跨期决策，则模型会更加复杂。此外，GNP平减指数应该用于修正GNP为实际GNP，同时应该引入与家庭决策更为相关的CPI作为变量，鉴于本文并非严谨地讨论就业问题，就不做更多修正了

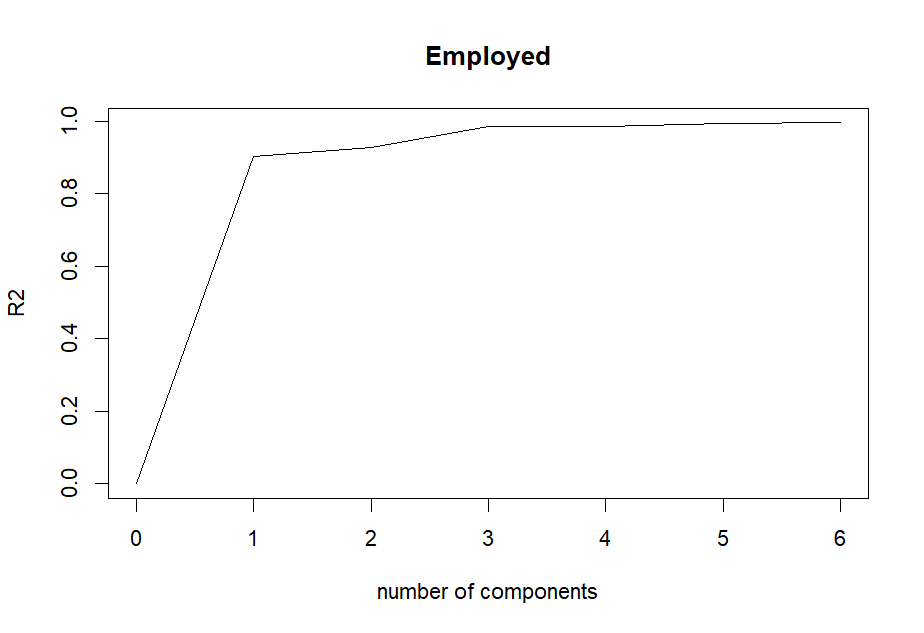


图 3 方差累积贡献率

表 4 PCR系数估计

|  | PCR |
| --- | --- |
| (Intercept) | -270.249 |
| GNP.deflator | 0.071 |
| GNP | 0.008 |
| Unemployed | 0.005 |
| Armed.Forces | 0.003 |
| Population | 0.106 |
| Year | 0.159 |

最后计算各模型的MSE，并用测试集进行预测，结果如表5所示，可以看到岭回归的预测值与实际值60.32最为接近，训练集上的MSE从小到大依次是OLS、岭回归和主成分估计，测试集上的MSE从小到大依次是岭回归、OLS和主成分估计

表 5 模型评估

| 模型 | 预测 | 内样本MSE | 外样本MSE |
| --- | --- | --- | --- |
| OLS | 59.86 | 0.7122 | 0.2158 |
| Ridge | 60.57 | 11.5145 | 0.0606 |
| PCR | 59.82 | 15.5450 | 0.2536 |

代码

|  |
| --- |
| library(corrplot)  library(MASS)  library(car)  library(flextable)  library(dplyr)  library(modelsummary)  library(ggplot2)  library(broom)  library(pls)  data(longley)  train <- longley[2:16,]  test <- longley[1,]  #相关系数  corr.p <-round(cor.mtest(as.matrix(train))$p,3)  corrplot(cor(train),method = "square",type = "upper", tl.pos = 'tp',tl.srt = 45,p.mat = corr.p)  corrplot(cor(train),method = "number",add = T,type = "lower", tl.pos = 'n', cl.pos = 'n',p.mat = corr.p)  #OLS  model0 <- lm(data=train,Employed~.)  options(digits = 4)  modelsummary(model0,output = "flextable",stars = T,gof\_map = "adj.r.squared") %>%  align(align = "center",part = "all") %>%  save\_as\_docx(path = "D:/预删除文件夹/大三上/应用回归分析/临时.docx")  data.frame(t(data.frame(vif(model0)))) %>%  flextable() %>%  align(align = "center",part = "all") %>%  save\_as\_docx(path = "D:/预删除文件夹/大三上/应用回归分析/临时.docx")  #岭估计  train\_mean <- unlist(lapply(train,mean))  train\_sd <- unlist(lapply(train,sd))  train\_scale <- data.frame(scale(train))  ridge <- lm.ridge(data=train\_scale,Employed~.,lambda = seq(0,10,0.001))  ggplot(tidy(ridge),aes(x=lambda,y=estimate,color=term)) +  geom\_line() +  theme\_bw()  MASS::select(ridge)  ridge2 <- lm.ridge(data=train\_scale,Employed~.,lambda = 0.005)  ridge\_coef <- ridge2$coef\*train\_sd[7]/train\_sd[1:6]  ridge\_coef0 <- as.vector(train\_mean[7]-ridge\_coef%\*%train\_mean[1:6])  options(digits = 1)  data.frame("X"=c("(Intercept)",names(ridge\_coef)),  "Ridge"=c(ridge\_coef0,ridge\_coef)) %>%  flextable() %>%  align(align = "center",part = "all") %>%  save\_as\_docx(path = "D:/预删除文件夹/大三上/应用回归分析/临时.docx")  #主成分  PCR <- pcr(data=train\_scale,Employed~.)  plot(PCR,"val",val.type = "R2")  options(digits = 1)  pcr\_coef <- PCR$coefficients[,,1]\*train\_sd[7]/train\_sd[1:6]  pcr\_coef0 <- as.vector(train\_mean[7]-pcr\_coef%\*%train\_mean[1:6])  data.frame("X"=c("(Intercept)",names(pcr\_coef)),  "pcr"=c(pcr\_coef0,pcr\_coef)) %>%  flextable() %>%  align(align = "center",part = "all") %>%  save\_as\_docx(path = "D:/预删除文件夹/大三上/应用回归分析/临时.docx")  #预测和MSE  y1 <- predict(model0,newdata = test)  y2 <- apply(test[1:6],1,function(x) sum(x\*c(ridge\_coef)))+ridge\_coef0  y3 <- apply(test[1:6],1,function(x) sum(x\*c(pcr\_coef)))+pcr\_coef0  data.frame(list(模型=c("OLS","Ridge","PCR")),  list(预测=c(y1,y2,y3)),  list(内样本MSE=c(sum(model0$residuals^2),  sum((train[7]-apply(train[1:6],1,function(x) sum(x\*c(ridge\_coef)))-ridge\_coef0)^2),  sum((train[7]-apply(train[1:6],1,function(x) sum(x\*c(pcr\_coef)))-pcr\_coef0)^2))),  list(外样本MSE=c((y1-test$Employed)^2,  (y2-test$Employed)^2,  (y3-test$Employed)^2))) %>%  flextable() %>%  align(align = "center",part = "all") %>%  save\_as\_docx(path = "D:/预删除文件夹/大三上/应用回归分析/临时.docx") |

**六、参考文献**

**七、教师评语**

1. 计划是这样的，但是最后模型评估的时候发现误把第一年作为测试集了，由于改动太过麻烦所以没有修正 [↑](#footnote-ref-1)