主成分与偏最小二乘回归的效果对比

Jingyi Ma(编写), Yuchong Wu(整理)

2020-04-17

目录

1 简介

此篇意在比较 PCR（主成分分析）和 PLS（偏最小二乘回归）的效果，得出在什么情况下哪种方法更为合适。

模拟了一个 n 个观测值，p 个变量，变量之间相关系数为 ρ 的数据集，通过 β_0 和 β_1 加上一个标准正态分布的残差模拟出被解释变量。

2 说明

在 comparison.py 中定义了一个 comparison 函数，用于输出 PCR 和 PLS 的指标对比，分别包括：

- 交叉验证中的测试误差

- 成分的个数（交叉验证中的测试误差取到最小时）
- 对 Y 的解释程度（在此成分个数下）

3 结论

相比于 PCR，PLS 在以下情况的表现更佳：

- 变量个数更多
- 变量之间相关系数较小
- 各个变量的系数较大（变量对结果的影响较大）

4 模拟过程

```
library(knitr)
library(reticulate)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.4
## v tibble  2.1.1     v dplyr   0.8.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'purrr' was built under R version 3.6.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
use_python("/usr/local/bin/python3")
py_config()
```

```
## python:         /usr/local/bin/python3
## libpython:      /usr/local/opt/python@3.9/Frameworks/Python.framework/Versions/3.9/l
## pythonhome:     /usr/local/opt/python@3.9/Frameworks/Python.framework/Versions/3.9:/
## version:        3.9.1 (default, Jan  6 2021, 06:05:23)  [Clang 12.0.0 (clang-1200.0.
## numpy:          /usr/local/lib/python3.9/site-packages/numpy
## numpy_version:  1.21.1
##
## python versions found:
##  /usr/local/bin/python3
##  /Users/ethan/.virtualenvs/r-reticulate/bin/python
##  /usr/bin/python
##  /usr/bin/python3
```

```
import numpy as np
import pandas as pd
from scipy.stats import norm
from src.scale import scale
from src.sim import sim
from model.comparison import comparison
```

4.1 变化 - p

```
n, p, rho = 1000, 10, 0.5
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 0.5, 0.5 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR             6    1.040492               0.869771
## 1     PLS             2    1.040659               0.869798
```

```
n, p, rho = 1000, 30, 0.5
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 0.5, 0.5 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR            28    1.053451               0.956126
## 1     PLS             3    1.052155               0.956072
```

```
n, p, rho = 1000, 50, 0.5
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 0.5, 0.5 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR            34    1.016787               0.973011
## 1     PLS             4    1.032042               0.973507
```

4.2 变化 - rho

```
n, p, rho = 1000, 30, 0.25
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 0.5, 0.5 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR            29    1.038088               0.923840
## 1     PLS             2    1.033148               0.923736
```

```
n, p, rho = 1000, 30, 0.5
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 0.5, 0.5 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR            29    0.992817               0.959971
## 1     PLS             4    0.991414               0.959945
```

```
n, p, rho = 1000, 30, 0.75
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 0.5, 0.5 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR            18    0.968682               0.979205
## 1     PLS             4    0.971065               0.979280
```

4.3 变化 - beta

```
n, p, rho = 1000, 30, 0.5
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 0.1, 0.1 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR             8    1.008096               0.475111
## 1     PLS             0    1.012978               0.474091
```

```
n, p, rho = 1000, 30, 0.5
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 0.5, 0.5 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR            23    0.968686               0.958227
## 1     PLS             4    0.970101               0.958516
```

```
n, p, rho = 1000, 30, 0.5
mu = norm.rvs(size=p, scale=1)
beta0, beta1 = 1, 1 * np.ones(p, dtype=float)
comparison(n, p, rho, mu, beta0, beta1)
```

```
##   methods  n components  test error  variation explanation
## 0     PCR            26    0.950735               0.990006
## 1     PLS             4    0.953445               0.990016
```