一文打通普通标准误、稳健标准误和聚类标准误

陈普
(华东交通大学经管学院)

2022 年 2 月 5 日

1 普通标准误

对一个回归模型，可以书写如下，

$$y_i = x_i'\beta + \epsilon_i$$

其中，x_i 是第 i 个个体 k 个变量的 $k \times 1$ 的观测值列向量，y_i, ϵ_i 都是标量。该模型的 OLS 解可以写为，

$$\hat{\beta} = (X'X)^{-1}(X'y) = (X'X)^{-1}X'(X\beta + \epsilon) = \beta + (X'X)^{-1}X'\epsilon$$

其中，X 是 $n \times k$ 的数据矩阵，ϵ 是 $n \times 1$ 的误差项。那么 $\hat{\beta}$ 的方差协方差矩阵为，

$$E(\hat{\beta}\hat{\beta}') = E[(X'X)^{-1}X'\epsilon\epsilon'X(X'X)^{-1}] = (X'X)^{-1}X'\Omega X(X'X)^{-1} \tag{1}$$

其中,$\Omega = E(\epsilon\epsilon')$ 是 ϵ 的方差协方差矩阵。如果,我们认为误差项是同方差且无自相关,那么 $\Omega = \sigma I_{n\times n}$，此时，基于(1)式，可知 $\hat{\beta}$ 的方差协方差矩阵为，

$$E(\hat{\beta}\hat{\beta}') = \sigma^2(X'X)^{-1}$$

上式就是 R，Stata 等软件报告的对 $\hat{\beta}$ 普通标准误的计算方式。

2 怀特标准误——异方差稳健标准误

但是如果误差是异方差，同时还有自相关，就不能把 Ω 写成 $\sigma^2 I$，从而(1)式就不能获得上式的简化，而只能写成，

$$E(\hat{\beta}\hat{\beta}') = (X'X)^{-1}X'\epsilon\epsilon'X(X'X)^{-1}$$

为了方便说明，我们要把 $X'\epsilon$ 写成 $\sum_i x_i\epsilon_i$，这样上式可以写成，

$$E(\hat{\beta}\hat{\beta}') = (X'X)^{-1}\left[\sum_i \sigma_i^2 x_i x_i'\right](X'X)^{-1} \tag{2}$$

其中 σ_i^2 是 ϵ_i 的方差。因为有这个方差在，那么可以做如下的渐进估计，

$$E(\hat{\beta}\hat{\beta}') = n(X'X)^{-1}\left[p\lim \frac{1}{n}\sum_i \sigma_i^2 x_i x_i'\right](X'X)^{-1}$$

然后，怀特的异方差稳健估计量就是把中间中括号里面的东西用如下矩阵替代，

$$W = \frac{1}{n}\sum_i \hat{\epsilon}_i^2 x_i x_i'$$

上面阐述有点抽象，但我们发现无论哪种情况，主要区别在于 Ω 不同。因此，可以按照 Ω 的形状归类，印象更为深刻。在普通标准误下，

$$\Omega = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix} = \sigma^2 I_{n\times n}$$

其中 σ^2 可以用 $\sum_{i=1}^{n} \hat{\epsilon_i}^2/(n-k)$ 来估计。在 White 异方差稳健标准误下，

$$\Omega = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

其中 σ_i^2 直接用 $\hat{\epsilon}_i^2$ 进行估计。

3 聚类标准误

所谓聚类标准误，指的是误差在某个组别，比如在同一个省的，该误差项具有同方差结构。但是不在同一个省，误差项就异方差了。即 Ω 是一个块对角结构，若数据有 m 个省份，那么该结构如下，

$$\Omega = \begin{bmatrix} \underbrace{\begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_1^2 \end{pmatrix}}_{Province_1} & \cdots & \cdots & 0 \\ 0 & \underbrace{\begin{pmatrix} \sigma_2^2 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_2^2 \end{pmatrix}}_{Province_2} & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \underbrace{\begin{pmatrix} \sigma_m^2 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_m^2 \end{pmatrix}}_{Province_m} \end{bmatrix} \tag{3}$$

如果在组内还允许自相关也是可以的，这样 Ω 可以写成 (这里假设了每组有 T 个观测值)，

$$
\Omega = \begin{bmatrix}
\underbrace{\begin{pmatrix} \sigma^2_{1,11} & \cdots & \cdots & \sigma^2_{1,1T} \\ \sigma^2_{1,21} & \cdots & \ddots & \sigma^2_{1,2T} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^2_{1,n1} & \sigma^2_{1,12} & \cdots & \sigma^2_{1,TT} \end{pmatrix}}_{Province_1} & \cdots & \cdots & 0 \\
0 & \underbrace{\begin{pmatrix} \sigma^2_{2,11} & \cdots & \cdots & \sigma^2_{2,1T} \\ \sigma^2_{2,21} & \cdots & \ddots & \sigma^2_{2,2T} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^2_{2,n1} & \sigma^2_{2,12} & \cdots & \sigma^2_{2,TT} \end{pmatrix}}_{Province_2} & \cdots & 0 \\
0 & 0 & \ddots & \vdots \\
0 & 0 & \cdots & \underbrace{\begin{pmatrix} \sigma^2_{m,11} & \cdots & \cdots & \sigma^2_{m,1T} \\ \sigma^2_{m,21} & \cdots & \ddots & \sigma^2_{m,2T} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^2_{m,n1} & \sigma^2_{m,12} & \cdots & \sigma^2_{m,TT} \end{pmatrix}}_{Province_m}
\end{bmatrix} \tag{4}
$$

4 代码实现

```
> rm(list = ls())
> library(pacman)
> p_load(wooldridge,lmtest,sandwich,tidyverse)
> data("airfare")
> lmmod <- lm(dist ~ fare, data = airfare)
> coeftest(lmmod) # 正常标准误计算


t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) 78.780558  18.267761  4.3125 1.648e-05 ***
fare         5.094971   0.094241 54.0632 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> # 手动计算
> bigX <- data.frame(cnst = rep(1,nrow(airfare)), fare = airfare$fare) %>%
   as.matrix()
> hat_eps <- residuals(lmmod) %>% matrix(ncol = 1)
> hat_sig2 <- sum(hat_eps*hat_eps)/(nrow(airfare)-2)
> sqrt(diag(hat_sig2 * solve(t(bigX) %*% bigX)))
```

```
      cnst        fare
18.26776055  0.09424098
```

从上可以看到，我们手动计算的和代码计算的是一致的。接下来再看看异方差稳健标准误。

```
> coeftest(lmmod, vcov. = vcovHC, type = 'HC') # White 标准误计算


t test of coefficients:

             Estimate Std. Error t value  Pr(>|t|)
(Intercept) 78.780558  13.268128  5.9376 3.106e-09 ***
fare         5.094971   0.086834 58.6747 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> # 手动计算
> W_diag <- hat_eps %*% t(hat_eps) %>% diag()
> W <- matrix(0, ncol = nrow(airfare), nrow = nrow(airfare))
> diag(W) <- W_diag
> solve(t(bigX) %*% bigX) %*% t(bigX) %*% W %*% bigX %*% solve(t(bigX) %*%
   bigX) %>%
  diag() %>% sqrt()
```

```
      cnst        fare
13.26812780  0.08683426
```

聚类标准误必须要指定组别，我们可以利用面板数据估计包plm来方便的估计聚类标准误。比如对于普通的回归，如果我们要计算(3)式的聚类标准误，可以操作如下，

```
> library(plm)
> data("Hedonic")
> rlt <- plm(mv~zn, data = Hedonic, index = 'townid', model = 'pooling')
> se_cluster <- coeftest(rlt, vcov. = vcovHC, method = 'white2') # 在townid
   上的聚类标准误
> se_white <- lm(mv~zn, data = Hedonic) %>% coeftest(type = 'HC') # White标
   准误
> ans <- cbind(se_cluster[,2],se_white[,2])
> colnames(ans) <- c('cluster_se','White_se')
> ans
```

```
             cluster_se     White_se
(Intercept) 0.019720793 0.0188544848
zn          0.000579765 0.0007273307
```

我们也可以计算在(4)式下的聚类标准误，

```
> coeftest(rlt, vcov. = vcovHC, method = 'arellano')


t test of coefficients:
```

```
             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 9.8699033  0.0578704 170.5519 < 2.2e-16 ***
zn          0.0063681  0.0011543   5.5168 5.527e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

之所以要在面板设置下计算聚类标准误，是因为plm包对vcovHC函数添加了功能，这个函数本来是在sandwich中，它有一个参数type，可以指定误差的方差结构，HC就是 White 的算法。而在plm中，它给这个函数添加了一个参数method，该参数包含三个选项'white1','white2','arellano'，第一个选项对应最普通的怀特标准误，第二个选项对应(3)式的聚类标准误，第三个选项对应(4)式的聚类标准误。