

部分线性模型

司继春

2026年10月

部分线性模型

- 回归方程:

$$y = \gamma \cdot w + \tilde{x}'\tilde{\beta} + u$$

其中 w 为我们关心的核心解释变量, 而 $\tilde{x}$ 为其他控制变量 (control variables) 或者协变量 (covariates) .

- 传统上, 假定

$$\mathbb{E}(y|\tilde{x}) = \tilde{x}'\delta, \mathbb{E}(w|\tilde{x}) = \tilde{x}'\eta$$

从而根据FWL定理, 使用线性回归估计即可.

- 然而: 假设了 y, w 对 $\tilde{x}$ 是线性函数
- 一个改进: 部分线性模型

$$y_i = \gamma \cdot w_i + g(\tilde{x}_i) + u_i$$

假定 $\mathbb{E}(u_i|\tilde{x}_i) = 0$.

半参数回归

- 针对以上半参数回归模型，两边对 $\tilde{x}$ 求条件期望，得到

$$\mathbb{E}(y_i|\tilde{x}_i) = \gamma \mathbb{E}(w_i|\tilde{x}_i) + g(\tilde{x}_i)$$

相减得到

$$y_i - \mathbb{E}(y_i|\tilde{x}_i) = \gamma \cdot [w_i - \mathbb{E}(w_i|\tilde{x}_i)] + u_i$$

或者

$$y_i^\perp = \gamma \times w_i^\perp + u_i$$

- Robinson (1988) 建议 $\mathbb{E}(y_i|\tilde{x}_i)$, $\mathbb{E}(w_i|\tilde{x}_i)$ 可以使用非参数的方法进行估计。
 - 缺点：维数的诅咒、收敛速度慢

高维与非线性

- 考虑如下部分线性模型：

$$y_i = \gamma \times w_i + g(f_i) + \epsilon_i$$

假设 $\mathbb{E}(\epsilon_i|w_i, f_i) = 0$ ，从而外生性假设成立。

- 此外，假设核心解释变量

$$w_i = m(f_i) + v_i$$

其中 $\mathbb{E}(v_i|f_i) = 0$ 。

- 由于 $g(\cdot)$ 和 $m(\cdot)$ 函数形式未知，我们可以使用一些控制变量 x_i 的线性组合来逼近两个函数，从而有

$$y_i = \gamma \times w_i + x_i' \beta + r_{gi} + \epsilon_i$$

$$w_i = x_i' \eta + r_{mi} + v_i$$

其中 x_i 可以是 f_i 的多项式逼近等，而 r_{gi}, r_{mi} 分别为用 $x_i' \beta$ 和 $x_i' \eta$ 逼近 $g(\cdot)$ 和 $m(\cdot)$ 的误差。

- 由于使用了逼近的方法， $x_i = P(f_i)$ 可以是比较高维的。

双重选择

- Belloni、Chernozhukov和Hansen (2014) 研究了这一模型, 发现如果简单的使用Lasso对 x_i 进行变量筛选, 其结果 $\hat{\gamma}$ 通常是有偏、非正态的.
 - 由于对 x_i 进行变量选择时具有不确定性, $\hat{\gamma}$ 的分布有时会出现多峰的情形, 这会造成统计推断上的困难.
- 为此, 他们提出了一种“双重选择”(double selection)的方法. 首先, 将上式中的第二个式子带入到第一个式子, 我们得到了

$$y_i = x_i' \zeta + r_i + \varepsilon_i$$

$$w_i = x_i' \eta + r_{mi} + v_i$$

其中 $\zeta = \gamma\eta + \beta$, $r_i = \gamma r_{mi} + r_{gi}$, $\varepsilon_i = \gamma v_i + \epsilon_i$.

- 分别对以上两个方程分别进行Lasso回归, 然后取

$$\mathcal{J} = \left\{ k : \hat{\zeta}_k^{\text{lasso}} \neq 0 \right\} \cup \left\{ k : \hat{\eta}_k^{\text{lasso}} \neq 0 \right\}$$

后双重Lasso估计量

- 使用 $\mathcal{J}$ 中的变量进行如下回归：

$$y_i = \gamma \times w_i + \sum_{k \in \mathcal{J}} \xi_i x_{ik} + u_i, k \in \mathcal{J}$$

以上估计量被称为“后双重Lasso估计量”（post-double-Lasso estimator）。

- Belloni、Chernozhukov和Hansen（2014）证明在一定的稀疏性假设下，以上估计量为渐近正态的：

$$\sqrt{N}(\hat{\gamma} - \gamma_0) \overset{a}{\sim} \mathcal{N}\left(0, (\mathbb{E}v_i^2)^{-1} \mathbb{E}(v_i^2 \epsilon_i^2) (\mathbb{E}v_i^2)^{-1}\right)$$

后双重Lasso估计量

NTV中的双重选择

- 在NTV的例子中，使用1999年的数据研究了观看到NTV的概率对于投票的影响. 为了保证外生性其加入了一些人口、工资等滞后变量的多项式进行控制. 我们可以使用双重选择的方法对这些多项式进行模型选择 (ntv_double_selec.do)
- 在Stata中，命令dsregress实现了以上算法. 此外还有dslogit、dspoisson等分别针对Logit回归和Poisson回归的拓展.
- 在以上代码中，选项controls为控制变量，其中括号中的为不使用Lasso选择的变量，而括号外的变量使用Lasso进行变量选择.

冗余参数

- 对于模型

$$y_i = \gamma \times w_i + x_i' \beta + r_{gi} + \epsilon_i$$

$$w_i = x_i' \eta + r_{mi} + v_i$$

真正关心的参数是 γ ，而 β, η 为冗余参数（nuisance parameters）。

- 如果使用机器学习的方法对这些参数进行估计，通常正则化会导致这些参数有偏，或者在Lasso中被错误地删除，这样就会导致核心参数的估计也被影响。
- 为此，一个比较常见的方法是通过Neyman正交化（Neyman orthogonalization, Neyman, 1959, 1979）的手段降低这种影响。

Neyman正交化

- 考虑模型

$$y_i = \gamma \times w_i + g(f_i) + \epsilon_i$$

- 该模型中，对于 γ 的识别基于如下矩条件：

$$\mathbb{E}(w_i \epsilon_i) = \mathbb{E}([y_i - \gamma \times w_i - g(f_i)] w_i) = 0$$

其中 $g(f_i)$ 是需要使用机器学习的方法（如Lasso等）进行估计和预测的部分，而我们对其系数并不感兴趣，从而是这里的冗余参数。

Neyman正交化

- 现在考虑如果我们没有很好的估计 $g(\cdot)$ ，而是有一个偏差，比如实际中带入的函数为

$$g(f_i) + t[\tilde{g}(f_i) - g(f_i)] \triangleq g(f_i) + th(f_i)$$

其中 $\tilde{g}(f_i) - g(f_i)$ 为偏离的方向，而 t 为偏离的程度

- 从而实际上我们使用的矩条件为

$$\mathbb{E}([y_i - \gamma \times w_i - g(f_i) - th(f_i)] w_i) = 0$$

然而如果我们考虑将以上矩条件对 t 求导数，会得到

$$\partial_t \mathbb{E}([y_i - \gamma \times w_i - g(f_i) - th(f_i)] \epsilon_i) |_{t=0} = -\mathbb{E}[w_i \times h(f_i)]$$

由于 w_i 与 f_i 通常是相关的，从而以上偏导数通常不为0。

- 以上计算的导数可以看做是矩条件对 $g(\cdot)$ 函数求导，也被称为Gâteaux导数，也可以记为 $\partial_g \mathbb{E}([y_i - \gamma \times w_i - g(f_i)] w_i)$ 。

Neyman正交化

- 现在，我们考虑一个“正交化”的版本. 对于原方程，两边分别对 f_i 求条件期望并相减，得到

$$y_i - \mathbb{E}(y_i|f_i) = \gamma \times [w_i - \mathbb{E}(w_i|f_i)] + \epsilon_i$$

如果记 $\mathbb{E}(y_i|f_i) = \psi(f_i)$ ，而 $\mathbb{E}(w_i|f_i) = m(f_i)$ ，那么此时估计 γ 的矩条件为\

$$\begin{aligned}\mathbb{E}(\epsilon_i [w_i - m(f_i)]) &= \mathbb{E}([(y_i - \psi(f_i)) - \gamma \times (w_i - m(f_i))] (w_i - m(f_i))) \\ &= \mathbb{E}[(y_i - \psi(f_i)) (w_i - m(f_i)) - \gamma (w_i - m(f_i))^2] \\ &= 0\end{aligned}$$

- $\psi(\cdot)$ 和 $m(\cdot)$ 两个函数作为冗余参数，分别使用机器学习的方法进行估计

Neyman正交化

- 同样实际使用的估计版本可能带有偏差：

$$\psi(f_i) + t [\tilde{\psi}(f_i) - \psi(f_i)] \triangleq \psi(f_i) + th_{\psi}(f_i)$$

$$m(f_i) + t [\tilde{m}(f_i) - m(f_i)] \triangleq m(f_i) + th_m(f_i)$$

- 同样计算

$$\begin{aligned} & \partial_t \mathbb{E} \left[(y_i - \psi(f_i) - th_{\psi}(f_i)) (w_i - m(f_i) - th_m(f_i)) - \gamma (w_i - m(f_i) - th_m(f_i))^2 \right] \\ &= \mathbb{E} [-h_{\psi}(f_i) (w_i - m(f_i)) - h_m(y_i - \psi(f_i)) + 2th_{\psi}(f_i) h_m(f_i)] \\ & \quad + \mathbb{E} [2\gamma h_m(f_i) (w_i - m(f_i) - th_m(f_i))] \end{aligned}$$

Neyman正交化

- 根据定义, 有

$$\mathbb{E}[h_{\psi}(f_i)(w_i - m(f_i))] = \mathbb{E}(\mathbb{E}[h_{\psi}(f_i)(w_i - m(f_i)) | f_i]) = \mathbb{E}(h_{\psi}(f_i) \mathbb{E}[(w_i - m(f_i))])$$

类似的有 $\mathbb{E}[h_m(y_i - \psi(f_i))] = 0$

- 从而以上导数可以化简为 $2t\mathbb{E}[h_{\psi}(f_i)h_m(f_i)] - 2\gamma t\mathbb{E}[h_m(f_i)]^2$
- 当 $t = 0$ 时, 有

$$\partial_t \mathbb{E} \left[(y_i - \psi(f_i) - th_{\psi}(f_i))(w_i - m(f_i) - th_m(f_i)) - \gamma (w_i - m(f_i) - th_m(f_i))^2 \right] \Big|_{t=0}$$

- 这也就意味着, 即使估计的冗余参数 $\psi(\cdot)$ 和 $m(\cdot)$ 存在偏差, 其对估计 γ 的矩条件的影响也是更高阶的影响, 从而影响较小.

Partialing-Out Regression

- 以上对于 γ 的估计步骤可以看做是Chernozhukov、Hansen和Spindler (2015a,b) 的特例，其中 $\psi(\cdot)$ 和 $m(\cdot)$ 两个函数可以使用Lasso、后Lasso等方法结合多项式逼近等方法获得，当然也可以使用其他机器学习方法，如决策树、随机森林等。
- 如果使用Lasso等方法，可以结合Belloni、Chernozhukov和Hansen (2014) 的双重选择方法，如此便诞生了“排除回归” (partialing-out regression)，其计算步骤如下：
 - ① 使用 w_i 对 x_i (f_i 的多项式等逼近) 进行Lasso回归，并进行后Lasso估计，得到残差 $w^\perp$ ；
 - ② 使用 y_i 对 x_i 进行Lasso回归，并进行后Lasso估计，得到残差 $y^\perp$ ；
 - ③ 使用 $y^\perp$ 对 $w^\perp$ 做回归得到 γ 的估计。

后双重Lasso估计量

NTV中的排除回归

- 在NTV的例子中，简单的将命令改为poregress即可

```
1 poregress Votes_SPS_ Watch_probit if year==1999, controls((i.  
   region) `E_controls') cluster(region) selec(cv)
```

POReg的偏差

- 以上正交化的方法虽然可以使得估计量更加可靠，但是仍然会存在一些偏差。
- 为了说明这一点，我们仍然考虑部分线性模型。使用正交化的方法，估计量可以写为

$$\hat{\gamma} = \left(\frac{1}{N} \sum_{i=1}^N \hat{v}_i^2 \right)^{-1} \frac{1}{N} \sum_{i=1}^N \hat{v}_i (y_i - \hat{g}(f_i))$$

其中 $\hat{v}_i = w_i - \hat{m}(f_i)$, $\hat{m}(\cdot), \hat{g}(\cdot)$ 可以使用机器学习的方法得到

- 这里我们需要收敛速度比 $N^{-1/4}$ 更快，通常在Lasso等回归中该条件可以通过稀疏性保证。

POReg的偏差

- 以上的估计量中, $\frac{1}{N} \sum_{i=1}^N \hat{v}_i^2 \xrightarrow{P} \mathbb{E}(v_i^2)$, 而收敛速度主要取决于第二项.
- 如果我们将 $\hat{v}_i = w_i - \hat{m}(f_i)$ 以及

$$y_i - \hat{g}(f_i) = \gamma(m(f_i) + v_i) + g(f_i) - \hat{g}(f_i) + \epsilon_i$$

带入到第二项, 我们会发现可以将

$$\frac{1}{\sqrt{N}} \sum_{i=1}^N \hat{v}_i (y_i - \hat{g}(f_i))$$

分解为如下几种成分:

POReg的偏差

- 两个误差项的乘积，即

$$\frac{1}{\sqrt{N}} \sum_{i=1}^N v_i \epsilon_i \stackrel{a}{\sim} \mathcal{N}(0, \sigma_{v\epsilon}^2)$$

这一项是常见的一项；

- 估计误差的乘积，或者正则化的偏差乘积项，即

$$\frac{1}{\sqrt{N}} \sum_{i=1}^N (\hat{g}(f_i) - g(f_i)) (\hat{m}(f_i) - m(f_i))$$

这一项是两个估计误差的乘积，即使两个成分的收敛速度都比较慢，其乘积也会更快，如果收敛速度比 $N^{-1/4}$ 更快，也会随着 $N \rightarrow \infty$ 变为无穷小项；

- 误差项与估计误差的乘积项，即诸如

$$\frac{1}{\sqrt{N}} \sum_{i=1}^N v_i (\hat{g}(f_i) - g(f_i)), \frac{1}{\sqrt{N}} \sum_{i=1}^N \epsilon_i (\hat{m}(f_i) - m(f_i))$$

收敛速度问题

- 为了说明第三项，如果假设 $\hat{g}(f_i) = g(f_i) + \frac{\epsilon_i}{N^{1/2-\delta}}$ ，从而 $\hat{g}(f_i) - g(f_i) = O_p(N^{-1/2+\delta})$ ，这是一个近乎于参数模型的相对比较快的收敛速度。
- 然而如果带入到上面，我们会发现

$$\frac{1}{\sqrt{N}} \sum_{i=1}^N v_i (\hat{g}(f_i) - g(f_i)) = \sqrt{N} O_p(N^{-1/2+\delta}) = O_p(N^\delta) \rightarrow \infty$$

从而这一项会带来一定的偏差。

- 注意这里一个重要的阻碍在于，由于估计 $\hat{g}(\cdot)$ 所使用的样本中包含了可能与 v_i 相关的信息，或者样本 i 即参与了估计 $\hat{g}(\cdot)$ ，又参与了对 γ 的估计，从而 v_i 与 $\hat{g}(f_i) - g(f_i)$ 可能是相关的。

交叉拟合

- Chernozhukov等（2018）提出可以使用“交叉拟合”（cross-fitting）的办法消除这一项。
- 所谓交叉拟合，即估计冗余参数 $\hat{g}, \hat{m}$ 所用的样本与估计 α 所用的样本为不同样本，如此，我们就消除了 v_i 与 $\hat{g}(f_i) - g(f_i)$ 之间的相关性。
- 比如，如果估计冗余参数 $\hat{g}, \hat{m}$ 所用的样本与估计 α 所用的样本是独立的，且注意到 $\mathbb{E}(v_i|f_i) = 0$ ，那么有

$$\mathbb{V}(v_i(\hat{g}(f_i) - g(f_i))) = \mathbb{E}[v_i^2(\hat{g}(f_i) - g(f_i))^2] - (\mathbb{E}[v_i(\hat{g}(f_i) - g(f_i))])^2 = \mathbb{E}[(\hat{g}(f_i) - g(f_i))^2 v_i^2]$$

由于 $\hat{g}(f_i) - g(f_i) = o_p(1)$ ，从而

$$\mathbb{V}\left(\frac{1}{\sqrt{N}} \sum_{i=1}^N v_i(\hat{g}(f_i) - g(f_i))\right) = \frac{\sum_{i=1}^N \mathbb{E}[(\hat{g}(f_i) - g(f_i))^2 \mathbb{E}(v_i^2|f_i)]}{N} \xrightarrow{p} 0$$

从而以上第三项也会是一个 $o_p(1)$ 。

交叉拟合

- 为了实现交叉拟合，一种简单的方法是将样本一分为二： I 和 I^c .
- 其中，样本 I 用于估计冗余参数 $\hat{g}, \hat{m}$ ；然后将 I^c 中的样本带入到 $\hat{g}, \hat{m}$ 中，得到样本外残差，再使用这些样本外残差估计 γ .
- 不过，这种做法会使得我们失去大约一半的样本. 解决这一问题的方法很简单，就是将 I 和 I^c 的角色互换：再使用 I^c 估计冗余参数，使用 I 估计 γ 即可.

交叉拟合

- 或者，我们可以将样本随机分为 S 份： $I_1, I_2, \dots, I_S$ ，然后使用如下步骤估计 γ ：
 - ① 将 $I_s^c = I/I_s$ ，即第 s 份样本的补集用于估计冗余参数（如 g, m ）；
 - ② 使用第1步估计的模型，在 I_s 上计算残差；
 - ③ 重复以上步骤 S 次，计算所有样本的“样本外残差”；
 - ④ 使用样本外残差带入正交化的矩条件估计 γ 。

双重机器学习

- 双重机器学习可以看做是以上Neyman正交化方法的拓展，即使用交叉拟合的方式进一步减少偏误.
- 因而，对于以上Neyman正交化的方法，我们都可以使用双重机器学习的方法进行估计.
- 比如，以上介绍的正交化的回归方法和工具变量方法都可以扩展到双重机器学习方法.

双重机器学习

NTV中的DML

- 在NTV的例子中，我们可以使用交叉拟合版本的poregress命令；xporegress进行交叉拟合：

```
1 | xporegress Votes_SPS_ Watch_probit if year==1999, controls((i.  
   | region) `E_controls') cluster(region) selec(cv) xfold(15)
```

其中xfold选项即S的值.

双重机器学习

NTV中的排除回归

- 我们也可以使用其他机器学习方法，如随机森林、梯度提升（gradient boosting）等机器学习方法。为此我们可以直接在Stata中调用Python，或者也可以使用ddml命令配合pystacked命令等方式进行。
- 使用方法可以参考<https://statalasso.github.io/docs/ddml/models/>
- ntv_ddml.do
 - 其中ddml init用于声明模型，partial代表部分线性模型；kfolds为样本划分的折数；E[Y|X]用于设定预测y的模型，这里使用了随机森林；E[D|X]用于设定预测w的模型
 - ddml crossfit命令就会进行交叉拟合，并将预测的样本外残差放入到Y1_pystacked_1 D1_pystacked_1两个变量中；最后可以使用ddml estimate命令进行估计。

双重机器学习

DDML与文本

- Dube等（2020）使用双重机器学习估计了在线劳动市场（Mturk网站）的劳动供给弹性问题，其基本回归方程为

$$\ln(\text{duration}) = -\eta \ln(\text{reward}) + g(x) + \epsilon$$

$$\ln(\text{reward}) = m(x) + v$$

其中duration为任务空缺的持续期，而reward为任务的奖励。

- 除了雇主、发布时间、任务时间要求等固定效应外，他们还将任务的标题、描述等使用Doc2Vec嵌入方法结合词袋模型等扩展了文本维度，从而 x 的维度很高。
- 为此，他们使用随机森林对 $\ln(\text{duration})$ 和 $\ln(\text{reward})$ 进行了交叉拟合，再将交叉拟合的样本外残差进行回归，从而进行了双重机器学习的估计。