2023年Stata中国用户大会

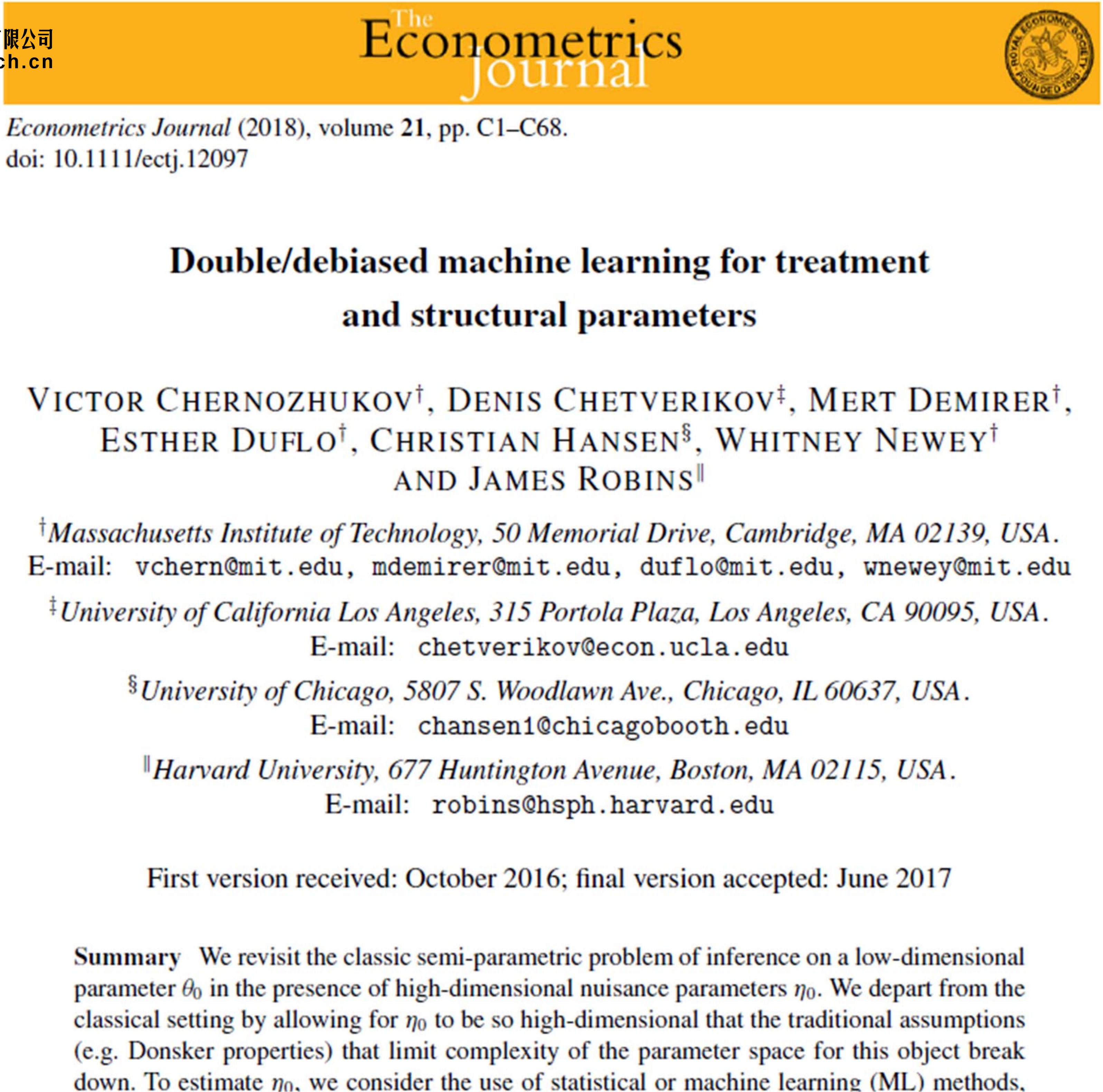
双重机器学习及Stata应用

陈强
山东大学经济学院
qiang2chen2@126.com
www.econometrics-stata.com
视频号：山东大学陈强教授
公众号：计量经济学及Stata应用

双重机器学习

- 正如2SLS就是做两个OLS，“双重机器学习” (Double Machine Learning) 就是做两次机器学习

- 若只做单次机器学习会有“正则化偏差” (regularization bias)，而双重机器学习具有纠偏功能，故也称为“双重纠偏机器学习” (Double-Debiased Machine Learning，DDML)



The Econometrics Journal

Econometrics Journal (2018), volume **21**, pp. C1–C68.
doi: 10.1111/ectj.12097

Double/debiased machine learning for treatment and structural parameters

Victor Chernozhukov[†], Denis Chetverikov[‡], Mert Demirer[†], Esther Duflo[†], Christian Hansen[§], Whitney Newey[†] and James Robins[‖]

[†] *Massachusetts Institute of Technology, 50 Memorial Drive, Cambridge, MA 02139, USA.*
E-mail: vchern@mit.edu, mdemirer@mit.edu, duflo@mit.edu, wnewey@mit.edu

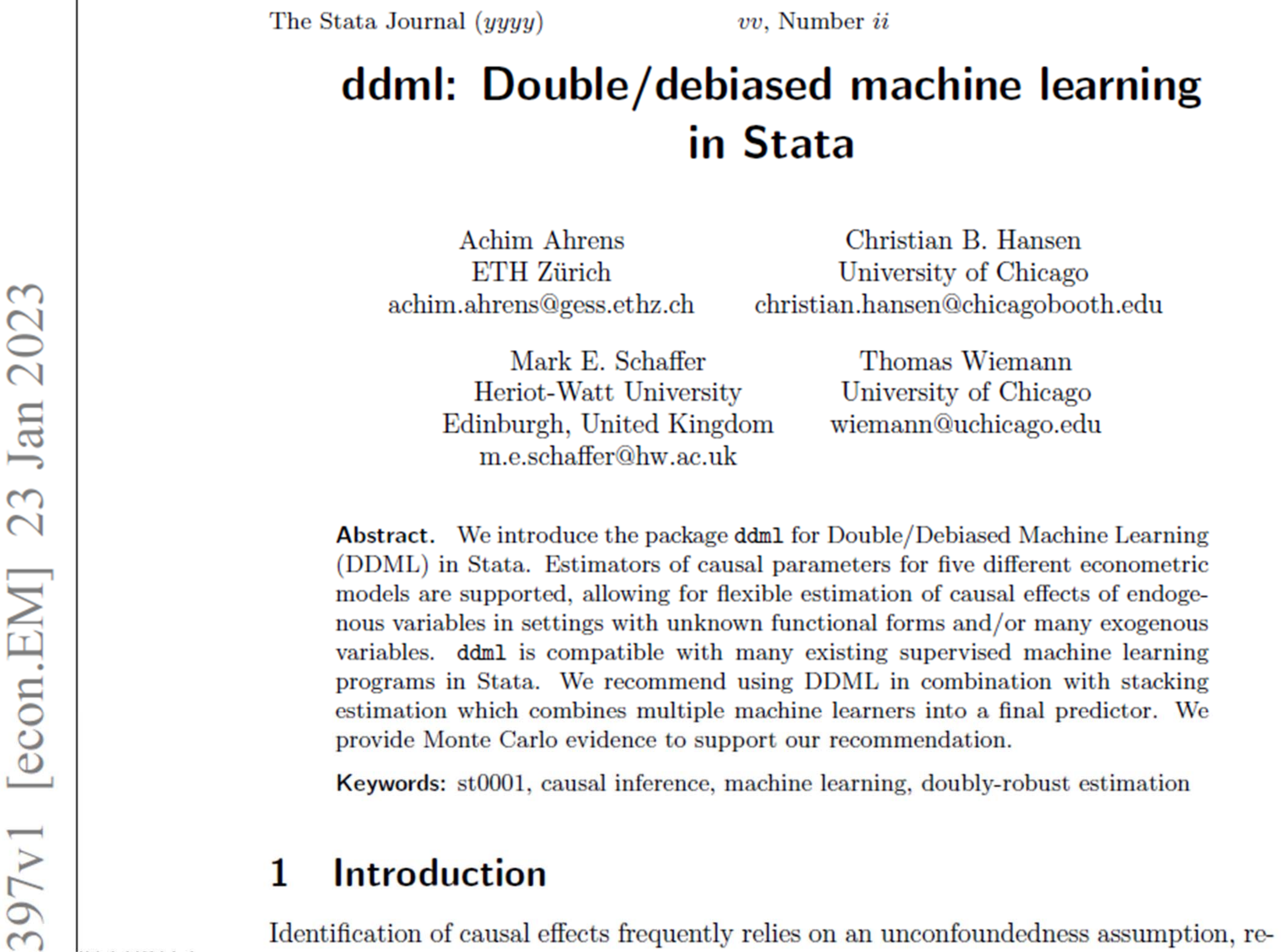
[‡] *University of California Los Angeles, 315 Portola Plaza, Los Angeles, CA 90095, USA.*
E-mail: chetverikov@econ.ucla.edu

[§] *University of Chicago, 5807 S. Woodlawn Ave., Chicago, IL 60637, USA.*
E-mail: chansen1@chicagobooth.edu

[‖] *Harvard University, 677 Huntington Avenue, Boston, MA 02115, USA.*
E-mail: robins@hsph.harvard.edu

First version received: October 2016; final version accepted: June 2017

Summary We revisit the classic semi-parametric problem of inference on a low-dimensional parameter θ_0 in the presence of high-dimensional nuisance parameters η_0. We depart from the classical setting by allowing for η_0 to be so high-dimensional that the traditional assumptions (e.g. Donsker properties) that limit complexity of the parameter space for this object break down. To estimate η_0, we consider the use of statistical or machine learning (ML) methods,



397v1 [econ.EM] 23 Jan 2023

The Stata Journal (*yyyy*) *vv*, Number *ii*

ddml: Double/debiased machine learning in Stata

Achim Ahrens
ETH Zürich
achim.ahrens@gess.ethz.ch

Christian B. Hansen
University of Chicago
christian.hansen@chicagobooth.edu

Mark E. Schaffer
Heriot-Watt University
Edinburgh, United Kingdom
m.e.schaffer@hw.ac.uk

Thomas Wiemann
University of Chicago
wiemann@uchicago.edu

Abstract. We introduce the package `ddml` for Double/Debiased Machine Learning (DDML) in Stata. Estimators of causal parameters for five different econometric models are supported, allowing for flexible estimation of causal effects of endogenous variables in settings with unknown functional forms and/or many exogenous variables. `ddml` is compatible with many existing supervised machine learning programs in Stata. We recommend using DDML in combination with stacking estimation which combines multiple machine learners into a final predictor. We provide Monte Carlo evidence to support our recommendation.

Keywords: st0001, causal inference, machine learning, doubly-robust estimation

1 Introduction

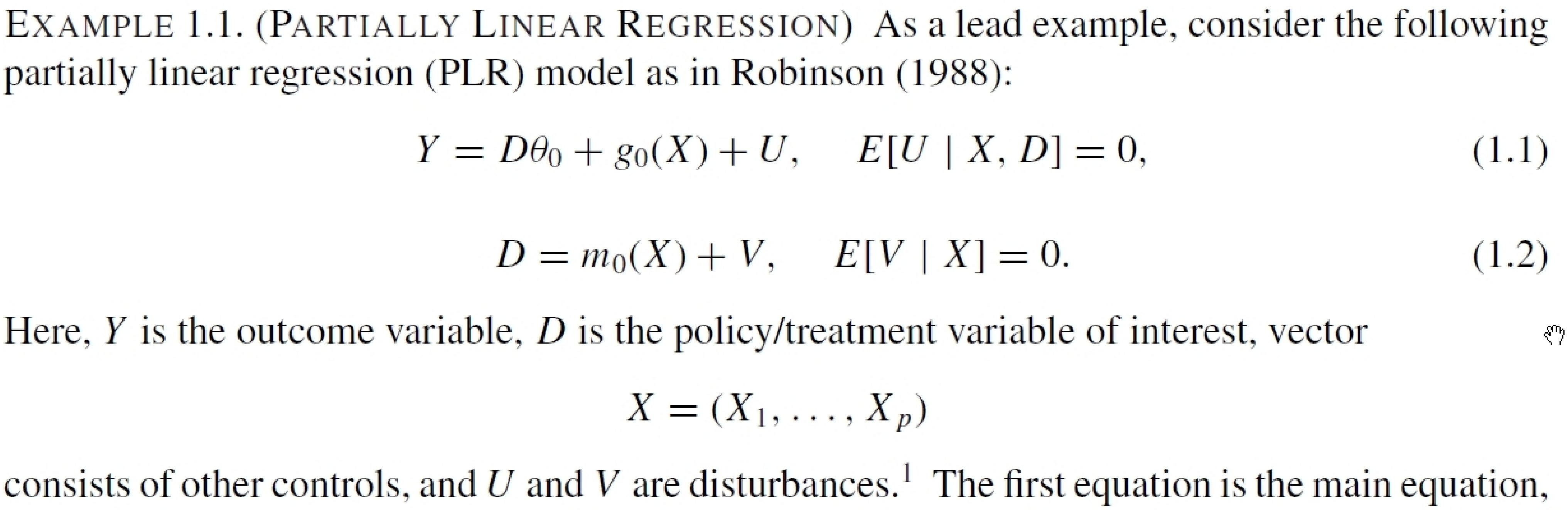
Identification of causal effects frequently relies on an unconfoundedness assumption, re-

DDML适用的数据类型

- 目前DDML主要适用于独立同分布(i.i.d.)的横截面数据，应用场景包括：

1. 部分线性模型(partially linear model)
2. 交互模型(interactive model)
3. 部分线性工具变量模型(partially linear IV model)
4. 灵活部分线性工具变量模型(flexible partially linear IV model)
5. 交互工具变量模型(interactive IV model)

1. 部分线性模型 (Robinson, 1988)



EXAMPLE 1.1. (PARTIALLY LINEAR REGRESSION) As a lead example, consider the following partially linear regression (PLR) model as in Robinson (1988):

$$Y = D\theta_0 + g_0(X) + U, \qquad E[U \mid X, D] = 0, \tag{1.1}$$

$$D = m_0(X) + V, \qquad E[V \mid X] = 0. \tag{1.2}$$

Here, Y is the outcome variable, D is the policy/treatment variable of interest, vector

$$X = (X_1, \dots, X_p)$$

consists of other controls, and U and V are disturbances.[1] The first equation is the main equation,

- 其中，Y 为结果变量，D 为处理变量(可以是连续变量)，θ_0 为感兴趣的参数(parameter of interest)，$\mathbf{X}$ 为控制变量(维度可能超过样本容量，即高维数据)，$g_0(\cdot)$ 与 $m_0(\cdot)$ 为未知函数(连函数形式也未知)

传统的参数回归

- 由于扰动项 U 均值独立于D与 $\mathbf{X}$，故部分线性模型并无内生性

- 传统的参数回归(parametric regression)假设 $g_0(\cdot)$ 的函数形式已知（比如线性函数，或加上平方项与交互项），然后直接对主方程(1.1)进行OLS估计。

- 但对 $g_0(\cdot)$ 的函数形式很可能误设(misspecified)，则会导致偏差，因为根据方程(1.2)，处理变量 D 也依赖于 $\mathbf{X}$

经典的半参数回归

- 为了避免函数形式误设，经典的半参数回归(Robinson, 1988)使用非参数回归(nonparametric regression)来估计 $\mathrm{E}(Y\mid\mathbf{X})$ 与 $\mathrm{E}(D\mid\mathbf{X})$

- 比如，“核回归”(kernel regression)或“局部线性回归”(local linear regression)。

- 但传统的非参数回归容易遇到“维度诅咒”(curse of dimensionality)，由于协变量 $\mathbf{X}$ 的维度通常较高。

引入机器学习

- 很多机器学习的方法在高维数据依然适用，比如拉索估计量(Lasso)、随机森林(random forest)、梯度提升法(gradient boosting)、神经网络(neural networks)等。

- 但机器学习方法通常有“正则化偏差”(regularization bias)，例如以Lasso为代表的惩罚回归(penalized regression)

- 若直接以机器学习方法估计 $g_0(\cdot)$，可能导致偏差

天真的“单重”机器学习：例1

- 直接使用Lasso估计主方程：

$$Y_i = D_i\theta_0 + g_0(\mathbf{X}_i) + U_i$$

- 但Lasso假设 $g_0(\mathbf{X})$ 为稀疏(sparse)的线性函数，函数形式可能误设

- 即使 $g_0(\mathbf{X})$ 确为稀疏的线性函数，在有限样本中，Lasso也存在正则化偏差

天真的“单重”机器学习：例2

- 设 θ_0 的初始值为0，以非参数的随机森林(random forest)估计方程 $Y = g_0(\mathbf{X}) + U$，得到 $\hat{g}_0(\mathbf{X})$

- OLS回归：$Y - \hat{g}_0(\mathbf{X}) \xrightarrow{OLS} D$，得到 $\hat{\theta}_0$
- 随机森林回归：$Y - D\hat{\theta}_0 \xrightarrow{random\ forest} \mathbf{X}$

- 迭代直至收敛

单重机器学习的偏差

- 考察迭代收敛后的最后一步OLS回归：

$$Y_i - \hat{g}_0(\mathbf{X}_i) = D_i\theta_0 + error_i$$

- 无常数项的OLS估计量为：

$$\hat{\theta}_0 = \left(\frac{1}{n}\sum_{i=1}^{n} D_i^2\right)^{-1} \frac{1}{n}\sum_{i=1}^{n} D_i\left(Y_i - \hat{g}_0(\mathbf{X}_i)\right)$$

- 但估计量 $\hat{\theta}_0$ 的收敛速度一般慢于 $\sqrt{n}$ ，导致 $\sqrt{n}(\hat{\theta}_0 - \theta_0)$ 的渐近分布出现偏差，并不以0为中心

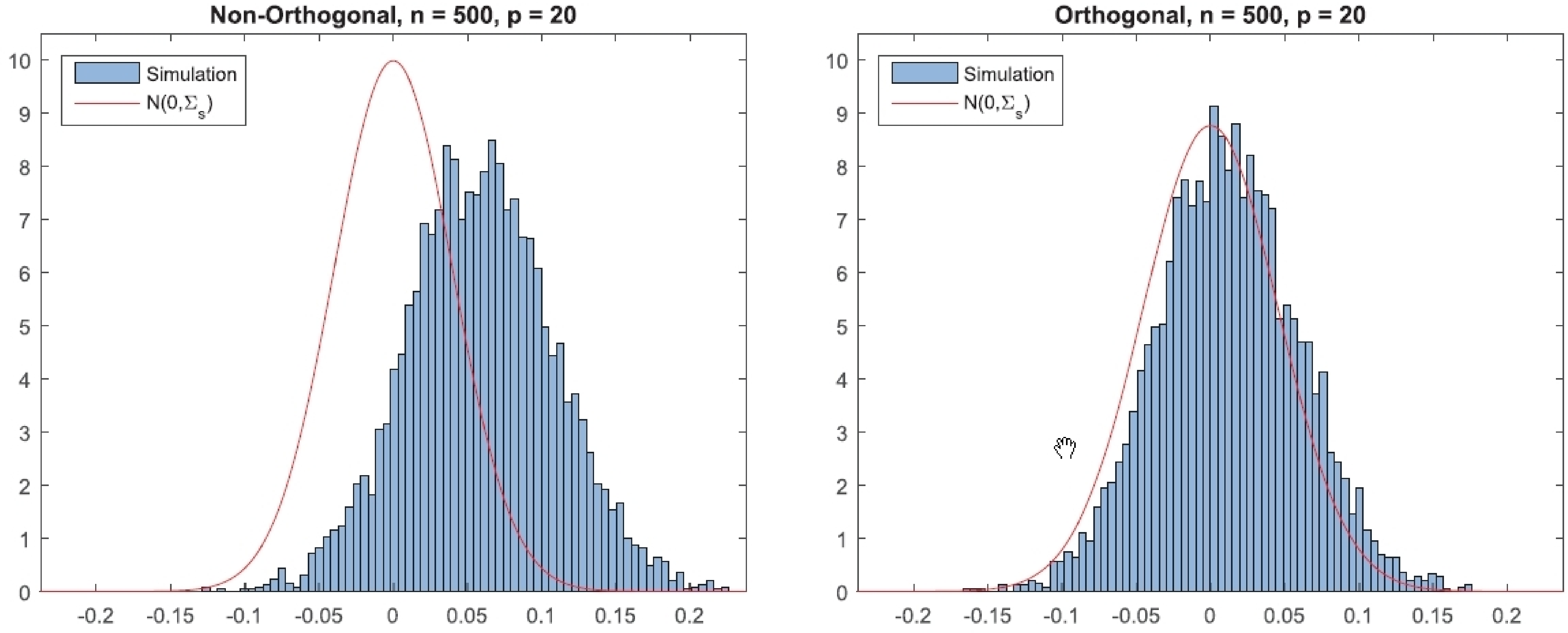


Figure 1. Comparison of the conventional and double ML estimators. [Colour figure can be viewed at wileyonlinelibrary.com]

单重机器学习的偏差来源

- 将 $Y_i = D_i\theta_0 + g_0(\mathbf{X}_i) + U_i$ 代入OLS表达式：

$$\hat{\theta}_0 = \left(\frac{1}{n}\sum_{i=1}^{n} D_i^2\right)^{-1} \frac{1}{n}\sum_{i=1}^{n} D_i\left(D_i\theta_0 + g_0(\mathbf{X}_i) + U_i - \hat{g}_0(\mathbf{X}_i)\right)$$

$$= \theta_0 + \left(\frac{1}{n}\sum_{i=1}^{n} D_i^2\right)^{-1} \frac{1}{n}\sum_{i=1}^{n} D_i U_i$$

$$+\left(\frac{1}{n}\sum_{i=1}^{n} D_i^2\right)^{-1} \frac{1}{n}\sum_{i=1}^{n} D_i\left(g_0(\mathbf{X}_i) - \hat{g}_0(\mathbf{X}_i)\right)$$

单重机器学习的偏差来源 (续)

$$\sqrt{n}\left(\hat{\theta}_0-\theta_0\right)=\left(\frac{1}{n}\sum_{i=1}^{n}D_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}D_iU_i$$

$$+\left(\frac{1}{n}\sum_{i=1}^{n}D_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}D_i\left(g_0(\mathbf{X}_i)-\hat{g}_0(\mathbf{X}_i)\right)$$

- 代入 $D_i=m_0(\mathbf{X}_i)+V_i$ ，上式第二项可写为

$$\left(\mathrm{E}\,D_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}m_0(\mathbf{X}_i)\left(g_0(\mathbf{X}_i)-\hat{g}_0(\mathbf{X}_i)\right)+\underbrace{o_p(1)}_{\xrightarrow{p}0}$$

单重机器学习的偏差来源 (续2)

$$\left(\mathrm{E}\,D_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n} m_0(\mathbf{X}_i)\left(g_0(\mathbf{X}_i)-\hat{g}_0(\mathbf{X}_i)\right)+o_p(1)$$

- 由于机器学习存在正则化偏差，故通常 $\left(g_0(\mathbf{X}_i)-\hat{g}_0(\mathbf{X}_i)\right)$收敛到0的速度慢于$1/\sqrt{n}$；记收敛速度为 $n^{-\varphi_g}$，且 $\varphi_g<\frac{1}{2}$
- 由于$m_0(\mathbf{X}_i)\neq 0$，上式数量级为$\sqrt{n}n^{-\varphi_g}\to\infty$
- 解决思路：将D_i 替换为$\left(D_i-\hat{m}_0(\mathbf{X}_i)\right)$

回到经典的半参数回归 (Robinson Difference Estimator)

- 给定 $\mathbf{X}_i$，对主方程两边同时求条件期望：

$$Y_i = D_i\theta_0 + g_0(\mathbf{X}_i) + U_i$$

$$\mathrm{E}(Y_i \mid \mathbf{X}_i) = \theta_0\,\mathrm{E}(D_i \mid \mathbf{X}_i) + g_0(X_i) + \underbrace{\mathrm{E}(U_i \mid \mathbf{X}_i)}_{=0}$$

- 两方程相减可得：

$$Y_i - \mathrm{E}(Y_i \mid \mathbf{X}_i) = \theta_0\left[D_i - \mathrm{E}(D_i \mid \mathbf{X}_i)\right] + U_i$$

- 代入非参估计量，进行OLS回归

$$Y_i - \widehat{\mathrm{E}(Y_i \mid \mathbf{X}_i)} = \theta_0\left[D_i - \widehat{\mathrm{E}(D_i \mid \mathbf{X}_i)}\right] + error_i$$

引入双重机器学习

- 分别以机器学习方法估计 $\widehat{\mathrm{E}(Y_i\,|\,\mathbf{X}_i)}$ 与 $\widehat{\mathrm{E}(D_i\,|\,\mathbf{X}_i)}$

- 将$\left[Y_i-\widehat{\mathrm{E}(Y_i\,|\,\mathbf{X}_i)}\right]$对$\left[D_i-\widehat{\mathrm{E}(D_i\,|\,\mathbf{X}_i)}\right]$进行OLS回归

- 这种类似离差的做法称为“partialing out”，使得估计方程满足“内曼正交性” (Neyman orthogonality) (Neyman, 1959, 1979)

从投影的角度理解

- $\widehat{\mathrm{E}(Y_i \mid \mathbf{X}_i)}$ 为 Y_i 对 $\mathbf{X}_i$ 的投影(并一定是线性投影)，而 $\left[Y_i - \widehat{\mathrm{E}(Y_i \mid \mathbf{X}_i)}\right]$ 为残差
- 类似地， $\widehat{\mathrm{E}(D_i \mid \mathbf{X}_i)}$ 为 D_i 对 $\mathbf{X}_i$ 的投影，而 $\left[D_i - \widehat{\mathrm{E}(D_i \mid \mathbf{X}_i)}\right]$ 为残差
- 将 $\left[Y_i - \widehat{\mathrm{E}(Y_i \mid \mathbf{X}_i)}\right]$ 对 $\left[D_i - \widehat{\mathrm{E}(D_i \mid \mathbf{X}_i)}\right]$ 进行回归则类似于“偏回归”(partial regression, Frisch-Waugh-Lovell Theorem)

两阶段估计法的误差

- 第一阶段(机器学习)的估计误差是否会在第二阶段(OLS)进一步放大?

- 若满足“内曼正交性”(Neyman orthogonality)，则第一阶段估计误差的微小变动对于最优化一阶条件(得分函数)的影响将很小(locally insensitive)，即得分函数对于第一阶段估计误差的一阶导数为0。

得分函数(score function)

- 计量经济学的估计量一般通过求极值(最小值或最大值)而得，称为“极值估计量”(extremum estimator)
- 最优化的一阶条件称为“得分函数”(score function)，比如MLE的一阶条件
- 得分函数的本质：在参数的真实值处，得分函数的期望为0。通过此矩条件，可识别参数

部分线性模型的得分函数

- 对于正交化之后的回归方程：

$$Y_i - \mathrm{E}(Y_i \mid X_i) = \theta_0 \left[D_i - \mathrm{E}(D_i \mid X_i) \right] + U_i$$

- 其识别条件为(扰动项与解释变量不相关)：

$$\mathrm{E}\left[U_i \left(D_i - \mathrm{E}(D_i \mid X_i) \right) \right] = 0$$

- 由此可得

$$\mathrm{E}\left[\left(Y_i - \mathrm{E}(Y_i \mid X_i) - \theta_0 \left(D_i - \mathrm{E}(D_i \mid X_i) \right) \right) \left(D_i - \mathrm{E}(D_i \mid X_i) \right) \right] = 0$$

部分线性模型的得分函数(续)

$$\mathrm{E}\left[\left(Y_i-\mathrm{E}(Y_i\mid X_i)-\theta_0\left(D_i-\mathrm{E}(D_i\mid X_i)\right)\right)\left(D_i-\mathrm{E}(D_i\mid X_i)\right)\right]=0$$

- 由此矩条件(moment condition)得到“得分函数”：

$$\psi\left(W;\theta,l,m\right)\equiv\left(Y-l(X)-\theta\left(D-m(X)\right)\right)\left(D-m(X)\right)$$

- 其中，$W\equiv(Y,X,D)$。记条件期望函数(Conditional Expectation Function, CEF) $l_0(X)\equiv\mathrm{E}(Y\mid X)$，$m_0(X)=\mathrm{E}(D\mid X)$，则得分函数满足矩条件：

$$\mathrm{E}\left[\psi\left(W;\theta_0,l_0,m_0\right)\right]=0$$

内曼正交性

- 根据“变分法”的思想，任意给定第一阶段估计误差 $\delta_l(X)$ 与 $\delta_m(X)$，以及扰动幅度 $r \geq 0$ ，则有 $l(X) = l_0(X) + r\delta_l(X)$ 与 $m(X) = m_0(X) + r\delta_m(X)$ 。代入得分函数：

$$\psi(W;\theta,l,m) \equiv \big(Y - l(X) - \theta\big(D - m(X)\big)\big)\big(D - m(X)\big)$$

$$= \big(Y - l(X)\big)\big(D - m(X)\big) - \theta\big(D - m(X)\big)^2$$

$$= \big(Y - l_0(X) - r\delta_l(X)\big)\big(D - m_0(X) - r\delta_m(X)\big)$$

$$-\theta\big(D - m_0(X) - r\delta_m(X)\big)^2$$

内曼正交性 (续)

- 得分函数的期望：

$$\mathrm{E}\psi\left(W;\theta,l,m\right)=\mathrm{E}\left(Y-l_0(X)-r\delta_l(X)\right)\left(D-m_0(X)-r\delta_m(X)\right)$$
$$-\theta\mathrm{E}\left(D-m_0(X)-r\delta_m(X)\right)^2$$

- 内曼正交性的定义：

$$\left.\frac{\partial\mathrm{E}\psi\left(W;\theta,l,m\right)}{\partial r}\right|_{r=0}=0$$

偏导数的第一项

$$\left.\frac{\partial \mathrm{E}\left(Y-l_0(X)-r\delta_l(X)\right)\left(D-m_0(X)-r\delta_m(X)\right)}{\partial r}\right|_{r=0}$$

$$=\mathrm{E}\left.\frac{\partial\left[\left(Y-l_0(X)-r\delta_l(X)\right)\left(D-m_0(X)-r\delta_m(X)\right)\right]}{\partial r}\right|_{r=0}$$

$$=\mathrm{E}\left[-\delta_l(X)\left(D-m_0(X)\right)\right]+\mathrm{E}\left[-\delta_m(X)\left(Y-l_0(X)\right)\right]$$

$$=-\mathrm{E}_X\,\mathrm{E}\left[\delta_l(X)\left(D-m_0(X)\right)\middle|X\right]-\mathrm{E}_X\,\mathrm{E}\left[\delta_m(X)\left(Y-l_0(X)\right)\middle|X\right]$$

$$=-\mathrm{E}_X\,\delta_l(X)\underbrace{\left[\mathrm{E}(D\mid X)-m_0(X)\right]}_{=0}-\mathrm{E}_X\,\delta_m(X)\underbrace{\left[\mathrm{E}(Y\mid X)-l_0(X)\right]}_{=0}$$

$$=0$$

偏导数的第二项

$$-\theta \left.\frac{\partial \mathrm{E}\left(D-m_0(X)-r\delta_m(X)\right)^2}{\partial r}\right|_{r=0}$$

$$=-\theta\,\mathrm{E} \left.\frac{\partial\left(D-m_0(X)-r\delta_m(X)\right)^2}{\partial r}\right|_{r=0}$$

$$=-2\theta\,\mathrm{E}\left[-\delta_m(X)\left(D-m_0(X)\right)\right]$$

$$=2\theta\,\mathrm{E}_X\,\mathrm{E}\left[\delta_m(X)\left(D-m_0(X)\right)\middle| X\right]$$

$$=2\theta\,\mathrm{E}_X\,\delta_m(X)\underbrace{\left[\mathrm{E}(D\,|\,X)-m_0(X)\right]}_{=0}$$

$$=0$$

DDML估计量

- OLS回归：

$$Y_i - \widehat{\mathrm{E}(Y_i \mid \mathbf{X}_i)} = \theta_0 \left[D_i - \widehat{\mathrm{E}(D_i \mid \mathbf{X}_i)} \right] + error_i$$

- 记 $\hat{l}_0(\mathbf{X}_i) \equiv \widehat{\mathrm{E}(Y_i \mid \mathbf{X}_i)}$，$\hat{m}_0(\mathbf{X}_i) \equiv \widehat{\mathrm{E}(D_i \mid \mathbf{X}_i)}$ ，

  $\hat{V}_i \equiv D_i - \widehat{\mathrm{E}(D_i \mid \mathbf{X}_i)}$， 则

$$\hat{\theta}_0 = \left(\frac{1}{n} \sum_{i=1}^{n} \hat{V}_i^2 \right)^{-1} \frac{1}{n} \sum_{i=1}^{n} \hat{V}_i \left(Y_i - \hat{l}_0(\mathbf{X}_i) \right)$$

- 使用OLS标准误即可(异方差稳健标准误、聚类稳健标准误等)

DDML估计量的性质

$$\hat{\theta}_0 = \left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i\left(Y_i - \hat{l}_0(\mathbf{X}_i)\right)$$

$$= \left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i\left(D_i\theta_0 + g_0(\mathbf{X}_i) + U_i - \hat{l}_0(\mathbf{X}_i)\right)$$

$$= \left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i\left((\hat{V}_i + \hat{m}_0(\mathbf{X}_i))\theta_0 + g_0(\mathbf{X}_i) + U_i - \hat{l}_0(\mathbf{X}_i)\right)$$

$$= \theta_0 + \left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i\left(\hat{m}_0(\mathbf{X}_i)\theta_0 + g_0(\mathbf{X}_i) + U_i - \hat{l}_0(\mathbf{X}_i)\right)$$

$$
\begin{aligned}
&\sqrt{n}\left(\hat{\theta}_0-\theta_0\right)\\
&=\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\hat{V}_i\left(\hat{m}_0(\mathbf{X}_i)\theta_0+g_0(\mathbf{X}_i)+U_i-\hat{l}_0(\mathbf{X}_i)\right)\\
&=\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\hat{V}_i\left(m_0(\mathbf{X}_i)\theta_0+\left(\hat{m}_0(\mathbf{X}_i)-m_0(\mathbf{X}_i)\right)\theta_0+g_0(\mathbf{X}_i)+U_i-\hat{l}_0(\mathbf{X}_i)\right)\\
&=\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left(m_0(\mathbf{X}_i)-\hat{m}_0(\mathbf{X}_i)+V_i\right)\left(U_i+\left(\hat{m}_0(\mathbf{X}_i)-m_0(\mathbf{X}_i)\right)\theta_0+l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)\\
&=\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left(m_0(\mathbf{X}_i)-\hat{m}_0(\mathbf{X}_i)\right)U_i+\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}V_iU_i\\
&-\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{\theta_0}{\sqrt{n}}\sum_{i=1}^{n}\left(\hat{m}_0(\mathbf{X}_i)-m_0(\mathbf{X}_i)\right)^2+\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{\theta_0}{\sqrt{n}}\sum_{i=1}^{n}V_i\left(\hat{m}_0(\mathbf{X}_i)-m_0(\mathbf{X}_i)\right)\\
&+\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left(m_0(\mathbf{X}_i)-\hat{m}_0(\mathbf{X}_i)\right)\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)\\
&+\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}V_i\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)
\end{aligned}
$$

内曼正交性带来的改进

- 如果 $(m_0(\mathbf{X}_i)-\hat{m}_0(\mathbf{X}_i))$ 与 $\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)$ 均以快于 $n^{-1/4}$ 的速度收敛于0，则
- $\left(m_0(\mathbf{X}_i)-\hat{m}_0(\mathbf{X}_i)\right)^2$ 与 $(m_0(\mathbf{X}_i)-\hat{m}_0(\mathbf{X}_i))\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)$ 都将以快于 $n^{-1/2}$ 收敛于0
- 故相应的项依概率收敛于0

新问题

- 最后一项：$\left(\frac{1}{n}\sum_{i=1}^{n}\hat{V}_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i=1}^{n}V_i\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)$

- 其中，V_i 与 $\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)$ 相关，因为 $V_i \to D_i \to Y_i \to \hat{l}_0(\mathbf{X}_i)$

- 若 $\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)$ 收敛于0的速度慢于 $n^{-1/2}$，则此项不收敛于0

解决方法

- 若用一半样本进行第一阶段的机器学习估计，而用另一半样本进行第二阶段的OLS估计，则V_i与$\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)$相互独立，故此项依均方收敛于0：

$$\mathrm{E}\left[\frac{1}{\sqrt{n}}\sum_{i=1}^{n}V_i\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)\right]=\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathrm{E}_{\mathbf{X}_i}\,\mathrm{E}\left[V_i\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)\middle|\mathbf{X}_i\right]$$

$$=\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mathrm{E}_{\mathbf{X}_i}\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)\underbrace{\mathrm{E}\left(V_i\middle|\mathbf{X}_i\right)}_{=0}=0$$

$$\mathrm{Var}\left[\frac{1}{\sqrt{n}}\sum_{i=1}^{n}V_i\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)\right]=\frac{1}{n}\sum_{i=1}^{n}\mathrm{E}\left[V_i^2\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)^2\right]$$

$$=\frac{1}{n}\sum_{i=1}^{n}\mathrm{E}(V_i^2)\,\mathrm{E}\left(l_0(\mathbf{X}_i)-\hat{l}_0(\mathbf{X}_i)\right)^2\rightarrow 0$$

*K*折交叉拟合

- 以上方法仅使用一半样本估计θ_0，导致效率损失。

- 更一般地，可进行*K*折交叉拟合(K-fold cross-fitting)，比如 *K* = 4或5

- 首先，将样本均分为*K*个子样本，记为 $I_1,\cdots,I_K$，而相应的补集为 $I_1^c,\cdots,I_K^c$

K折交叉拟合 (续)

- 其次，对于I_k中的每位个体i，均使用其补集I_k^c进行第一阶段的机器学习估计，得到$\hat{m}_{I_k^c}$与$\hat{l}_{I_k^c}$；然后进行样本外预测(out-of-sample predictions)，得到$\hat{m}_{I_k^c}(\mathbf{X}_i)$与$\hat{l}_{I_k^c}(\mathbf{X}_i)$
- 最后，进行OLS回归：

$$Y_i - \hat{l}_{I_{k_i}^c}(\mathbf{X}_i) = \theta_0\left[D_i - \hat{m}_{I_{k_i}^c}(\mathbf{X}_i)\right] + error_i$$

- 其中，k_i表示个体i所归属的子样本

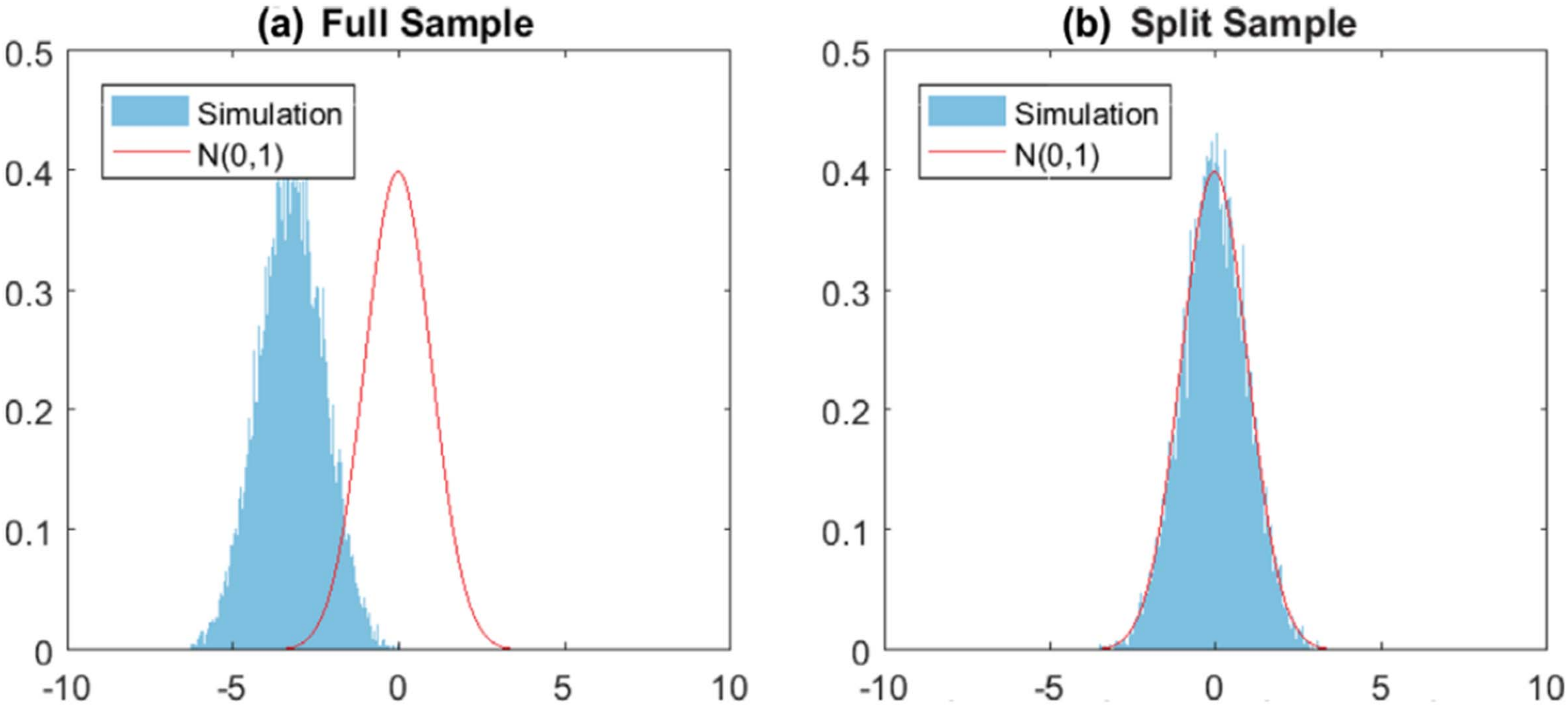
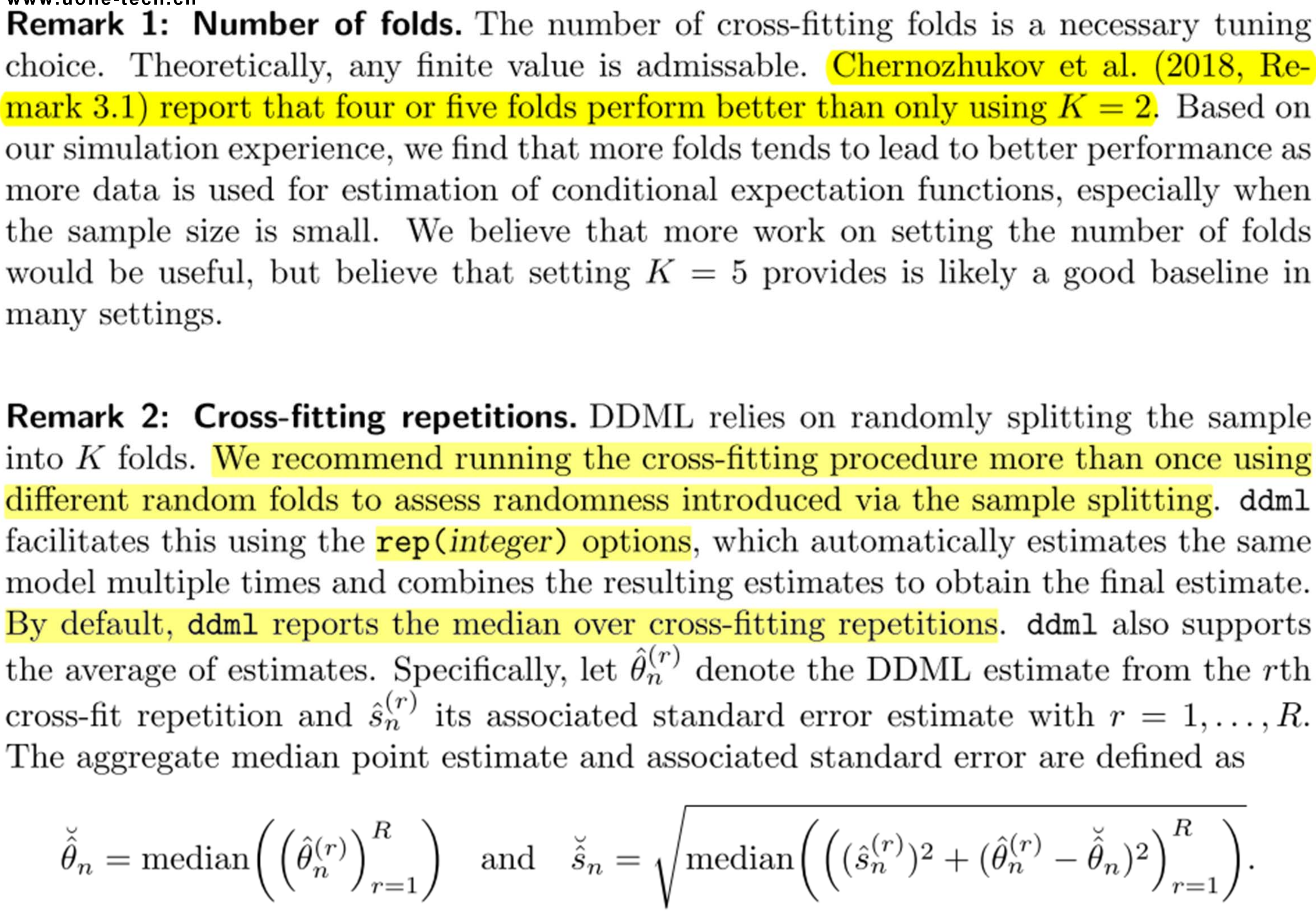


Figure 2. Comparison of full-sample and cross-fitting procedures. [Colour figure can be viewed at wileyonlinelibrary.com]

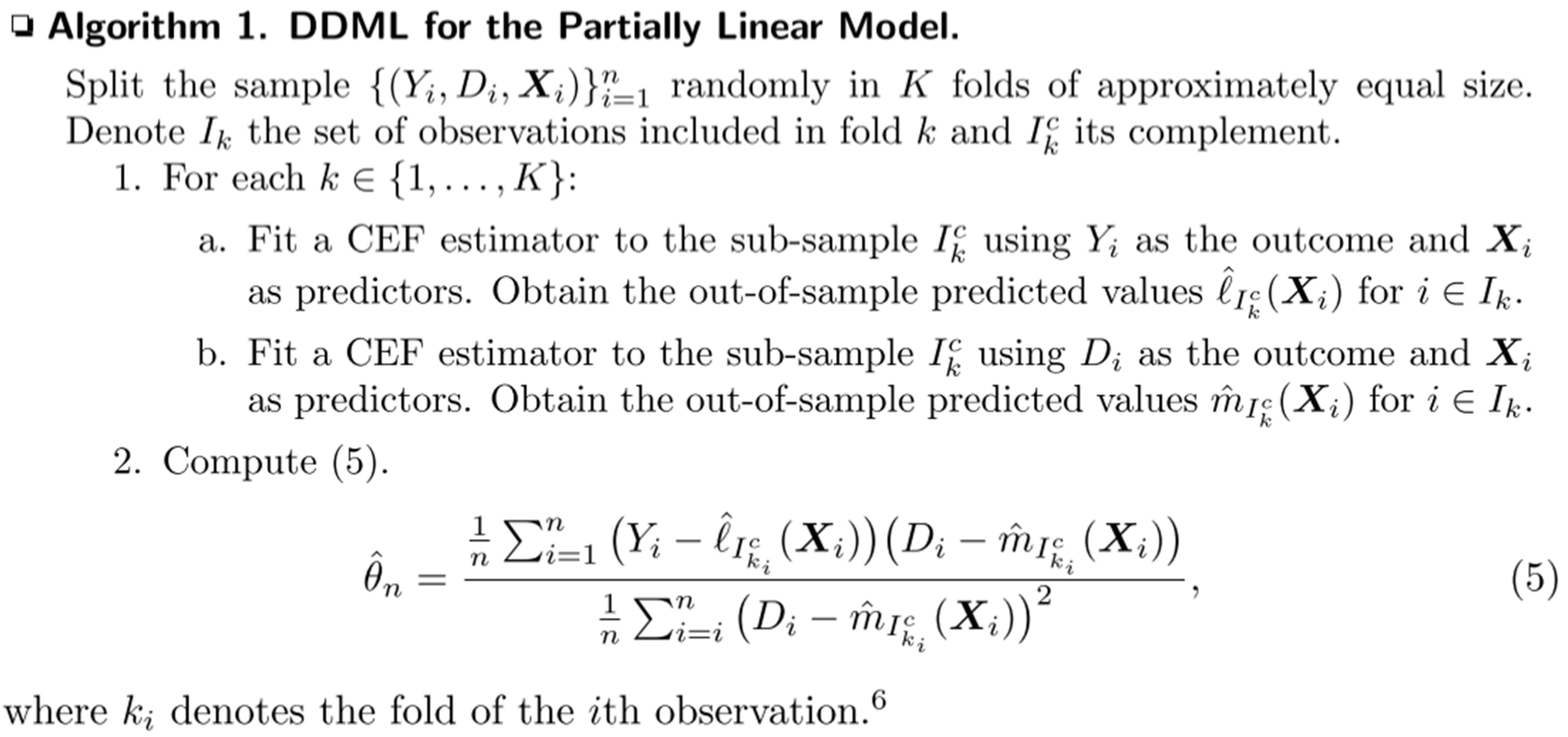


Remark 1: Number of folds. The number of cross-fitting folds is a necessary tuning choice. Theoretically, any finite value is admissable. Chernozhukov et al. (2018, Remark 3.1) report that four or five folds perform better than only using $K = 2$. Based on our simulation experience, we find that more folds tends to lead to better performance as more data is used for estimation of conditional expectation functions, especially when the sample size is small. We believe that more work on setting the number of folds would be useful, but believe that setting $K = 5$ provides is likely a good baseline in many settings.

Remark 2: Cross-fitting repetitions. DDML relies on randomly splitting the sample into K folds. We recommend running the cross-fitting procedure more than once using different random folds to assess randomness introduced via the sample splitting. `ddml` facilitates this using the `rep(`*integer*`)` options, which automatically estimates the same model multiple times and combines the resulting estimates to obtain the final estimate. By default, `ddml` reports the median over cross-fitting repetitions. `ddml` also supports the average of estimates. Specifically, let $\hat{\theta}_n^{(r)}$ denote the DDML estimate from the rth cross-fit repetition and $\hat{s}_n^{(r)}$ its associated standard error estimate with $r = 1, \ldots, R$. The aggregate median point estimate and associated standard error are defined as

$$\breve{\hat{\theta}}_n = \text{median}\left(\left(\hat{\theta}_n^{(r)}\right)_{r=1}^{R}\right) \quad \text{and} \quad \breve{\hat{s}}_n = \sqrt{\text{median}\left(\left((\hat{s}_n^{(r)})^2 + (\hat{\theta}_n^{(r)} - \breve{\hat{\theta}}_n)^2\right)_{r=1}^{R}\right)}.$$

DDML的算法



❑ **Algorithm 1. DDML for the Partially Linear Model.**

Split the sample $\{(Y_i, D_i, \boldsymbol{X}_i)\}_{i=1}^n$ randomly in K folds of approximately equal size. Denote I_k the set of observations included in fold k and I_k^c its complement.

1. For each $k \in \{1, \ldots, K\}$:

   a. Fit a CEF estimator to the sub-sample I_k^c using Y_i as the outcome and $\boldsymbol{X}_i$ as predictors. Obtain the out-of-sample predicted values $\hat{\ell}_{I_k^c}(\boldsymbol{X}_i)$ for $i \in I_k$.

   b. Fit a CEF estimator to the sub-sample I_k^c using D_i as the outcome and $\boldsymbol{X}_i$ as predictors. Obtain the out-of-sample predicted values $\hat{m}_{I_k^c}(\boldsymbol{X}_i)$ for $i \in I_k$.

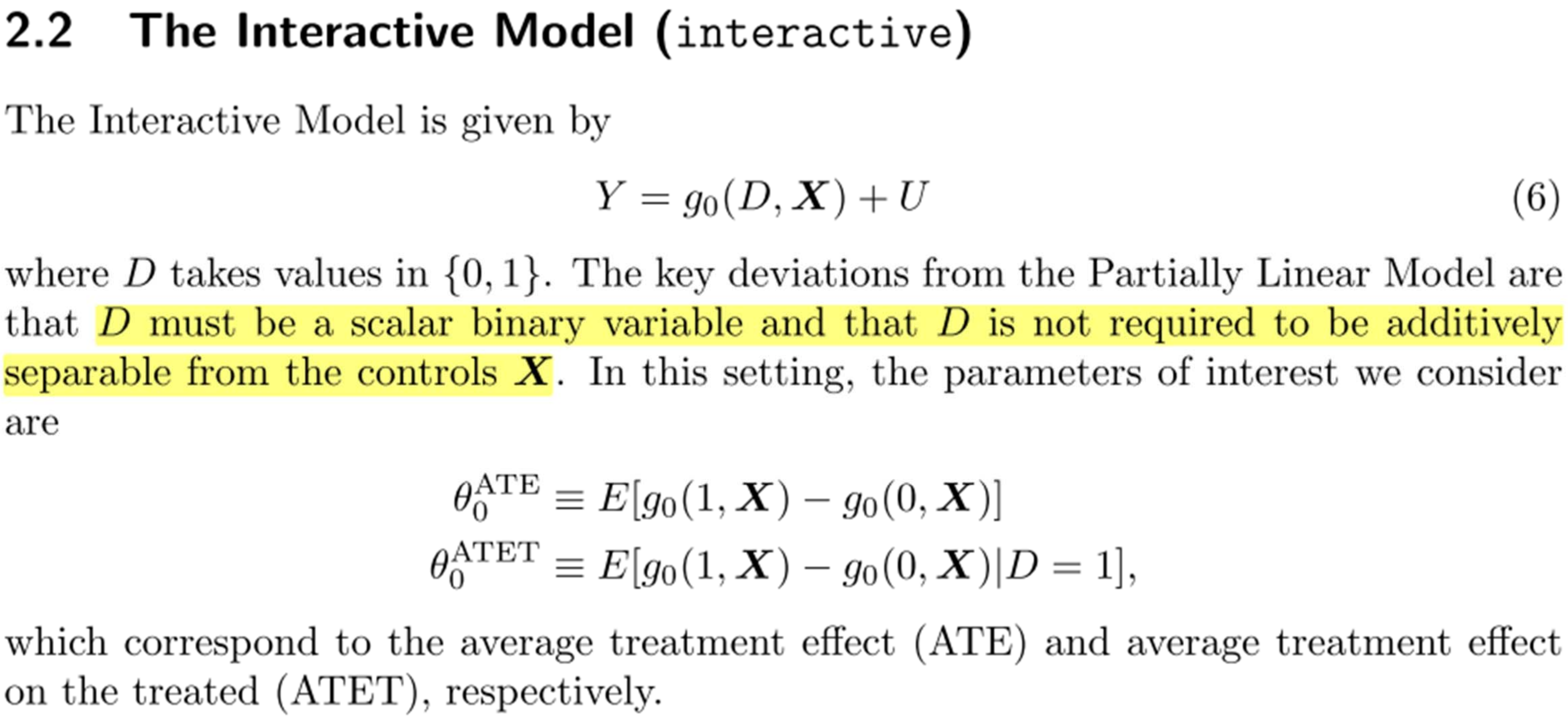
2. Compute (5).

$$\hat{\theta}_n = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{\ell}_{I_{k_i}^c}(\boldsymbol{X}_i)\right)\left(D_i - \hat{m}_{I_{k_i}^c}(\boldsymbol{X}_i)\right)}{\frac{1}{n}\sum_{i=i}^{n}\left(D_i - \hat{m}_{I_{k_i}^c}(\boldsymbol{X}_i)\right)^2}, \qquad (5)$$

where k_i denotes the fold of the ith observation.[6]

- Chernozhukov et al. (2018)证明，在一定正则条件下，DDML估计量为一致估计，且服从渐近正态分布。

2. 交互模型



2.2 The Interactive Model (interactive)

The Interactive Model is given by

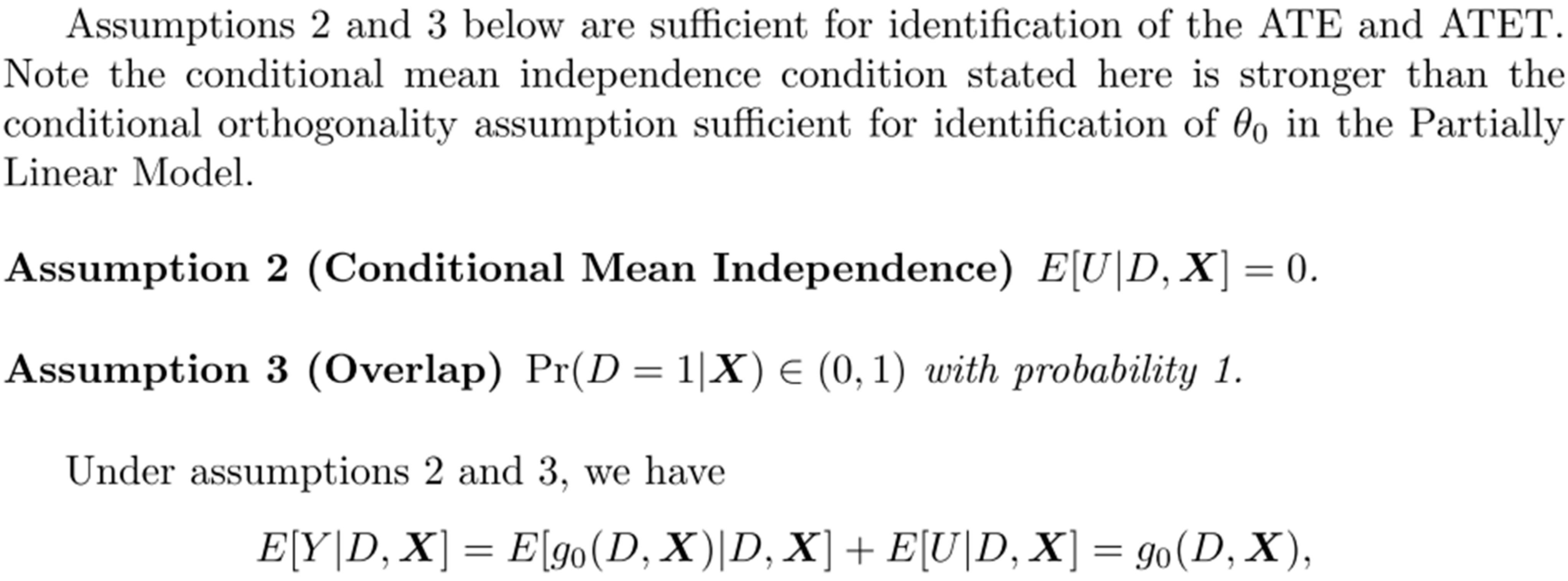
$$Y = g_0(D, \boldsymbol{X}) + U \tag{6}$$

where D takes values in $\{0, 1\}$. The key deviations from the Partially Linear Model are that D must be a scalar binary variable and that D is not required to be additively separable from the controls $\boldsymbol{X}$. In this setting, the parameters of interest we consider are

$$\theta_0^{\text{ATE}} \equiv E[g_0(1, \boldsymbol{X}) - g_0(0, \boldsymbol{X})]$$
$$\theta_0^{\text{ATET}} \equiv E[g_0(1, \boldsymbol{X}) - g_0(0, \boldsymbol{X}) | D = 1],$$

which correspond to the average treatment effect (ATE) and average treatment effect on the treated (ATET), respectively.

交互模型的假定与识别



Assumptions 2 and 3 below are sufficient for identification of the ATE and ATET. Note the conditional mean independence condition stated here is stronger than the conditional orthogonality assumption sufficient for identification of θ_0 in the Partially Linear Model.

Assumption 2 (Conditional Mean Independence) $E[U|D,\boldsymbol{X}] = 0$.

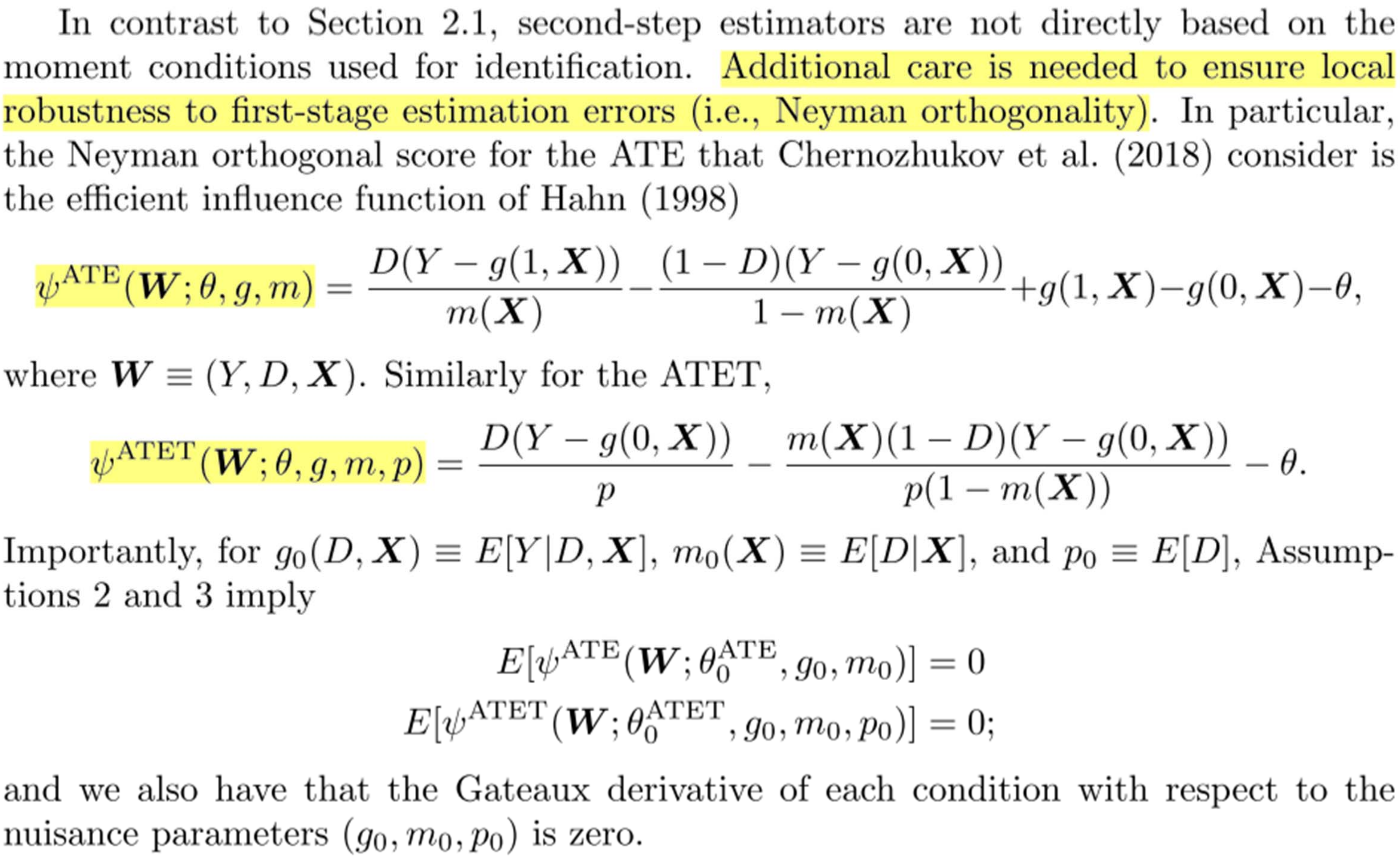
Assumption 3 (Overlap) $\Pr(D = 1|\boldsymbol{X}) \in (0,1)$ *with probability 1.*

Under assumptions 2 and 3, we have

$$E[Y|D,\boldsymbol{X}] = E[g_0(D,\boldsymbol{X})|D,\boldsymbol{X}] + E[U|D,\boldsymbol{X}] = g_0(D,\boldsymbol{X}),$$

- 故可直接使用“结果回归”(outcome regression)识别参数，但……

交互模型的得分函数与内曼条件



In contrast to Section 2.1, second-step estimators are not directly based on the moment conditions used for identification. Additional care is needed to ensure local robustness to first-stage estimation errors (i.e., Neyman orthogonality). In particular, the Neyman orthogonal score for the ATE that Chernozhukov et al. (2018) consider is the efficient influence function of Hahn (1998)

$$\psi^{\text{ATE}}(\boldsymbol{W};\theta,g,m)=\frac{D(Y-g(1,\boldsymbol{X}))}{m(\boldsymbol{X})}-\frac{(1-D)(Y-g(0,\boldsymbol{X}))}{1-m(\boldsymbol{X})}+g(1,\boldsymbol{X})-g(0,\boldsymbol{X})-\theta,$$

where $\boldsymbol{W}\equiv(Y,D,\boldsymbol{X})$. Similarly for the ATET,

$$\psi^{\text{ATET}}(\boldsymbol{W};\theta,g,m,p)=\frac{D(Y-g(0,\boldsymbol{X}))}{p}-\frac{m(\boldsymbol{X})(1-D)(Y-g(0,\boldsymbol{X}))}{p(1-m(\boldsymbol{X}))}-\theta.$$

Importantly, for $g_0(D,\boldsymbol{X})\equiv E[Y|D,\boldsymbol{X}]$, $m_0(\boldsymbol{X})\equiv E[D|\boldsymbol{X}]$, and $p_0\equiv E[D]$, Assumptions 2 and 3 imply

$$E[\psi^{\text{ATE}}(\boldsymbol{W};\theta_0^{\text{ATE}},g_0,m_0)]=0$$
$$E[\psi^{\text{ATET}}(\boldsymbol{W};\theta_0^{\text{ATET}},g_0,m_0,p_0)]=0;$$

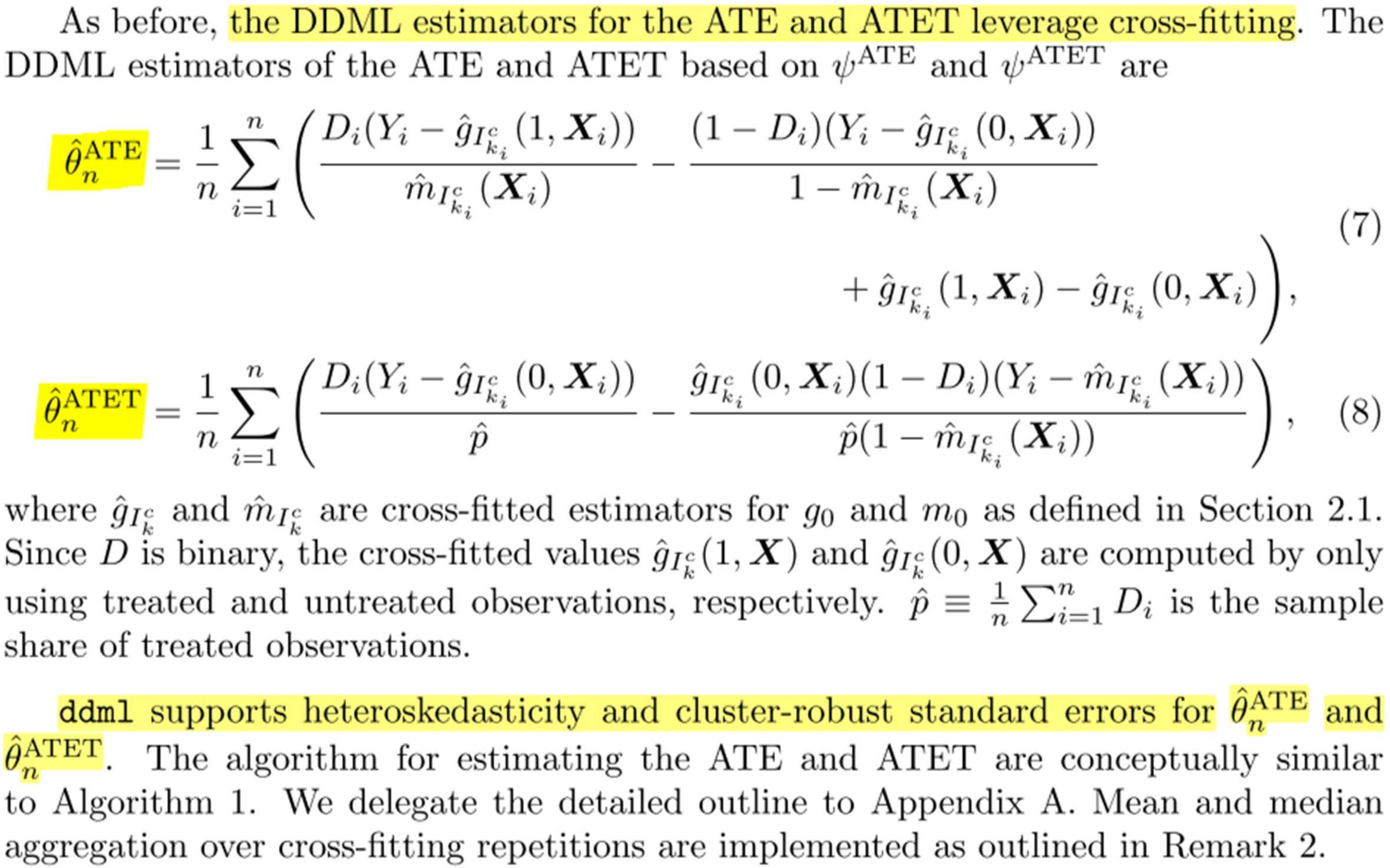
and we also have that the Gateaux derivative of each condition with respect to the nuisance parameters (g_0,m_0,p_0) is zero.

得分函数的解释

- 两个得分函数的前两项类似于“逆概加权法”(inverse probability weighting)

- 但将 Y 替换为其正交化的残差 $\left[Y - \mathrm{E}(Y \mid D, \mathbf{X})\right]$，以保证满足内曼正交条件

交互模型的算法



As before, the DDML estimators for the ATE and ATET leverage cross-fitting. The DDML estimators of the ATE and ATET based on ψ^{ATE} and ψ^{ATET} are

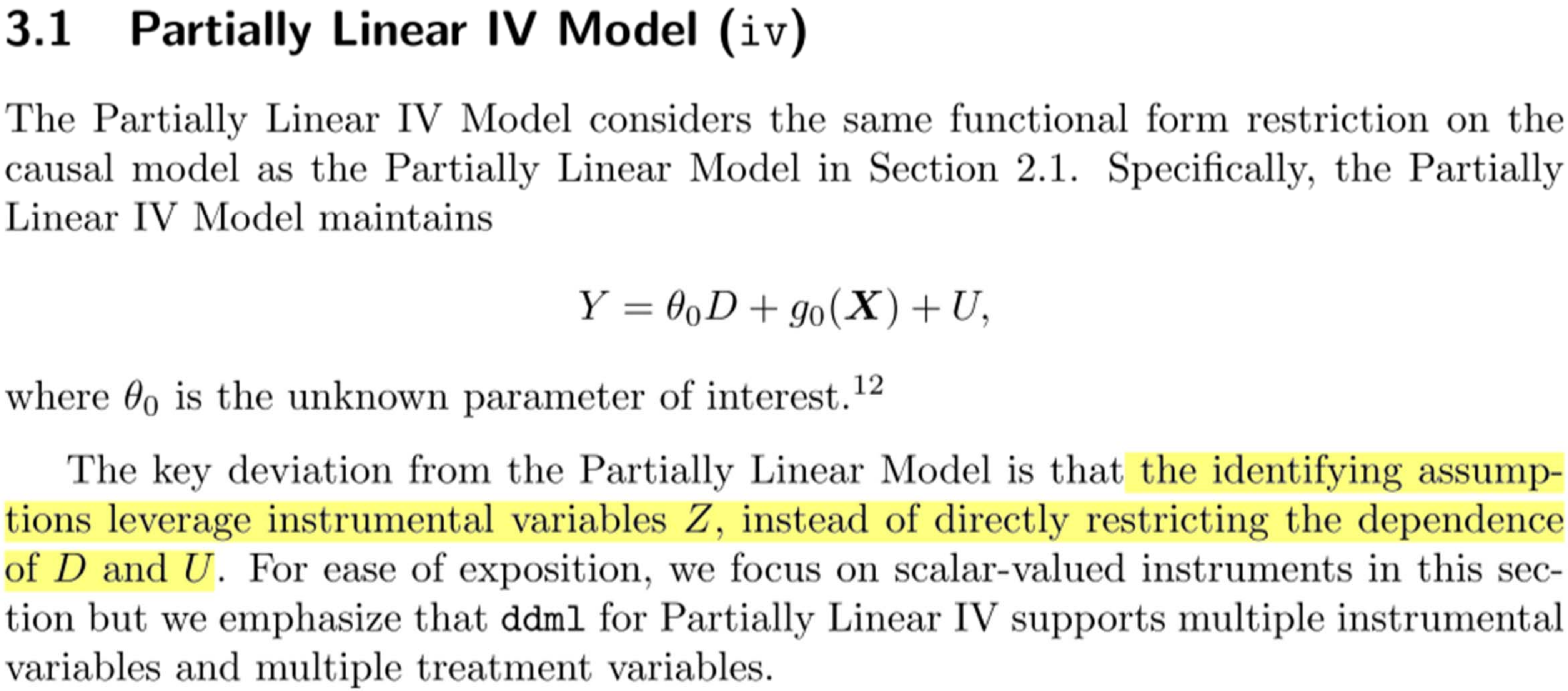
$$\hat{\theta}_n^{\text{ATE}} = \frac{1}{n}\sum_{i=1}^{n}\Bigg(\frac{D_i(Y_i - \hat{g}_{I^c_{k_i}}(1,\boldsymbol{X}_i))}{\hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i)} - \frac{(1-D_i)(Y_i - \hat{g}_{I^c_{k_i}}(0,\boldsymbol{X}_i))}{1-\hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i)} + \hat{g}_{I^c_{k_i}}(1,\boldsymbol{X}_i) - \hat{g}_{I^c_{k_i}}(0,\boldsymbol{X}_i)\Bigg), \tag{7}$$

$$\hat{\theta}_n^{\text{ATET}} = \frac{1}{n}\sum_{i=1}^{n}\Bigg(\frac{D_i(Y_i - \hat{g}_{I^c_{k_i}}(0,\boldsymbol{X}_i))}{\hat{p}} - \frac{\hat{g}_{I^c_{k_i}}(0,\boldsymbol{X}_i)(1-D_i)(Y_i - \hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i))}{\hat{p}(1-\hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i))}\Bigg), \tag{8}$$

where $\hat{g}_{I^c_k}$ and $\hat{m}_{I^c_k}$ are cross-fitted estimators for g_0 and m_0 as defined in Section 2.1. Since D is binary, the cross-fitted values $\hat{g}_{I^c_k}(1,\boldsymbol{X})$ and $\hat{g}_{I^c_k}(0,\boldsymbol{X})$ are computed by only using treated and untreated observations, respectively. $\hat{p} \equiv \frac{1}{n}\sum_{i=1}^{n} D_i$ is the sample share of treated observations.

`ddml` supports heteroskedasticity and cluster-robust standard errors for $\hat{\theta}_n^{\text{ATE}}$ and $\hat{\theta}_n^{\text{ATET}}$. The algorithm for estimating the ATE and ATET are conceptually similar to Algorithm 1. We delegate the detailed outline to Appendix A. Mean and median aggregation over cross-fitting repetitions are implemented as outlined in Remark 2.

3. 部分线性工具变量模型



3.1 Partially Linear IV Model (iv)

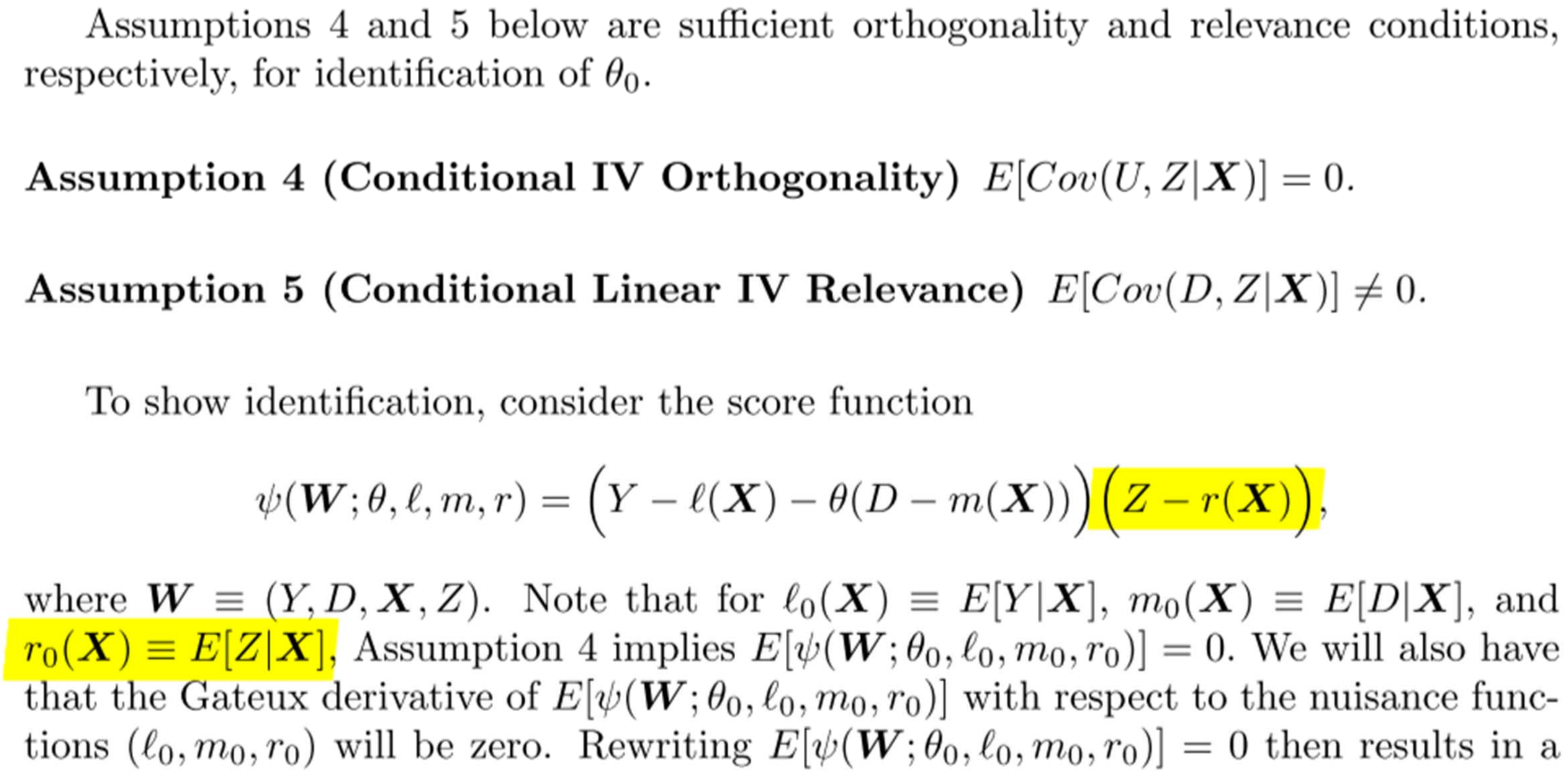
The Partially Linear IV Model considers the same functional form restriction on the causal model as the Partially Linear Model in Section 2.1. Specifically, the Partially Linear IV Model maintains

$$Y = \theta_0 D + g_0(\boldsymbol{X}) + U,$$

where θ_0 is the unknown parameter of interest.[12]

The key deviation from the Partially Linear Model is that the identifying assumptions leverage instrumental variables Z, instead of directly restricting the dependence of D and U. For ease of exposition, we focus on scalar-valued instruments in this section but we emphasize that ddml for Partially Linear IV supports multiple instrumental variables and multiple treatment variables.

部分线性工具变量模型的假定



Assumptions 4 and 5 below are sufficient orthogonality and relevance conditions, respectively, for identification of θ_0.

Assumption 4 (Conditional IV Orthogonality) $E[Cov(U, Z|\boldsymbol{X})] = 0$.

Assumption 5 (Conditional Linear IV Relevance) $E[Cov(D, Z|\boldsymbol{X})] \neq 0$.

To show identification, consider the score function

$$\psi(\boldsymbol{W}; \theta, \ell, m, r) = \Big(Y - \ell(\boldsymbol{X}) - \theta(D - m(\boldsymbol{X}))\Big)\Big(Z - r(\boldsymbol{X})\Big),$$

where $\boldsymbol{W} \equiv (Y, D, \boldsymbol{X}, Z)$. Note that for $\ell_0(\boldsymbol{X}) \equiv E[Y|\boldsymbol{X}]$, $m_0(\boldsymbol{X}) \equiv E[D|\boldsymbol{X}]$, and $r_0(\boldsymbol{X}) \equiv E[Z|\boldsymbol{X}]$, Assumption 4 implies $E[\psi(\boldsymbol{W}; \theta_0, \ell_0, m_0, r_0)] = 0$. We will also have that the Gateux derivative of $E[\psi(\boldsymbol{W}; \theta_0, \ell_0, m_0, r_0)]$ with respect to the nuisance functions (ℓ_0, m_0, r_0) will be zero. Rewriting $E[\psi(\boldsymbol{W}; \theta_0, \ell_0, m_0, r_0)] = 0$ then results in a

Wald expression given by

$$\theta_0 = \frac{E\left[\left(Y - \ell_0(\boldsymbol{X})\right)\left(Z - r_0(\boldsymbol{X})\right)\right]}{E\left[\left(D - m_0(\boldsymbol{X})\right)\left(Z - r_0(\boldsymbol{X})\right)\right]}, \tag{9}$$

where Assumption 5 is used to ensure a non-zero denominator.

The DDML estimator based on Equation (9) is given by

$$\hat{\theta}_n = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{\ell}_{I^c_{k_i}}(\boldsymbol{X}_i)\right)\left(Z_i - \hat{r}_{I^c_{k_i}}(\boldsymbol{X}_i)\right)}{\frac{1}{n}\sum_{i=i}^{n}\left(D_i - \hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i)\right)\left(Z_i - \hat{r}_{I^c_{k_i}}(\boldsymbol{X}_i)\right)}, \tag{10}$$

where $\hat{\ell}_{I^c_k}$, $\hat{m}_{I^c_k}$, and $\hat{r}_{I^c_k}$ are appropriate cross-fitted CEF estimators.

Standard errors corresponding to $\hat{\theta}_n$ are equivalent to the IV standard errors where $Y_i - \hat{\ell}_{I^c_{k_i}}(\boldsymbol{X}_i)$ is the outcome, $D_i - \hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i)$ is the endogenous variable, and $Z_i - \hat{r}_{I^c_{k_i}}(\boldsymbol{X}_i)$ is the instrument. `ddml` supports conventional standard errors available for linear instrumental variable regression in Stata, including heteroskedasticity and cluster-robust standard errors. Mean and median aggregation over cross-fitting repetitions are implemented as outlined in Remark 2. In the case where we have multiple instruments or endogenous regressors, we adjust the algorithm by residualizing each instrument and endogenous variable as above and applying two-stage least squares with the residualized outcome, endogenous variables, and instruments.

部分线性IV估计量的推导

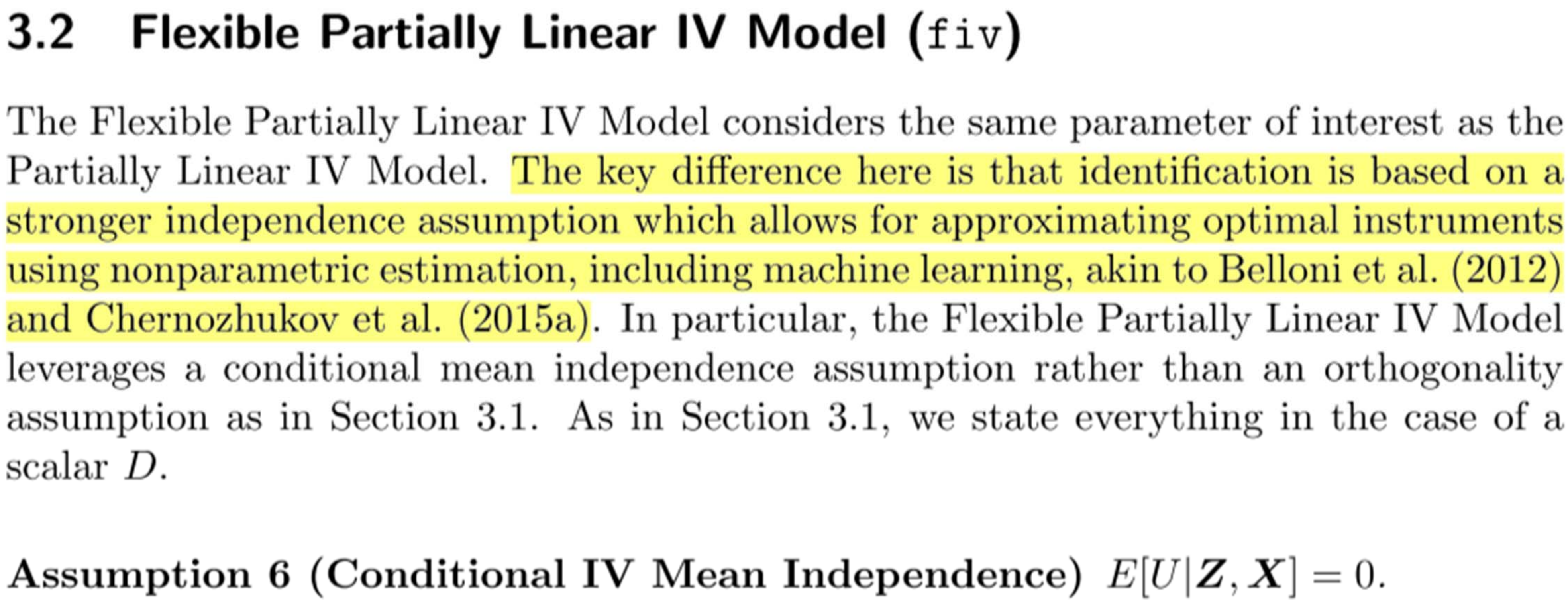
- 从Robinson difference模型出发：

$$Y_i - \mathrm{E}(Y_i \mid \mathbf{X}_i) = \theta_0 \left[D_i - \mathrm{E}(D_i \mid \mathbf{X}_i) \right] + U_i$$

- 方程两边同乘正交化的工具变量$\left[Z_i - \mathrm{E}(Z_i \mid \mathbf{X}_i) \right]$，并求期望：

$$\begin{aligned} &\mathrm{E}\left[Y_i - \mathrm{E}(Y_i \mid \mathbf{X}_i) \right]\left[Z_i - \mathrm{E}(Z_i \mid \mathbf{X}_i) \right] \\ &= \theta_0 \underbrace{\mathrm{E}\left[D_i - \mathrm{E}(D_i \mid \mathbf{X}_i) \right]\left[Z_i - \mathrm{E}(Z_i \mid \mathbf{X}_i) \right]}_{\neq 0} + \underbrace{\mathrm{E} U_i \left[Z_i - \mathrm{E}(Z_i \mid \mathbf{X}_i) \right]}_{=0} \end{aligned}$$

4. 灵活部分线性IV模型



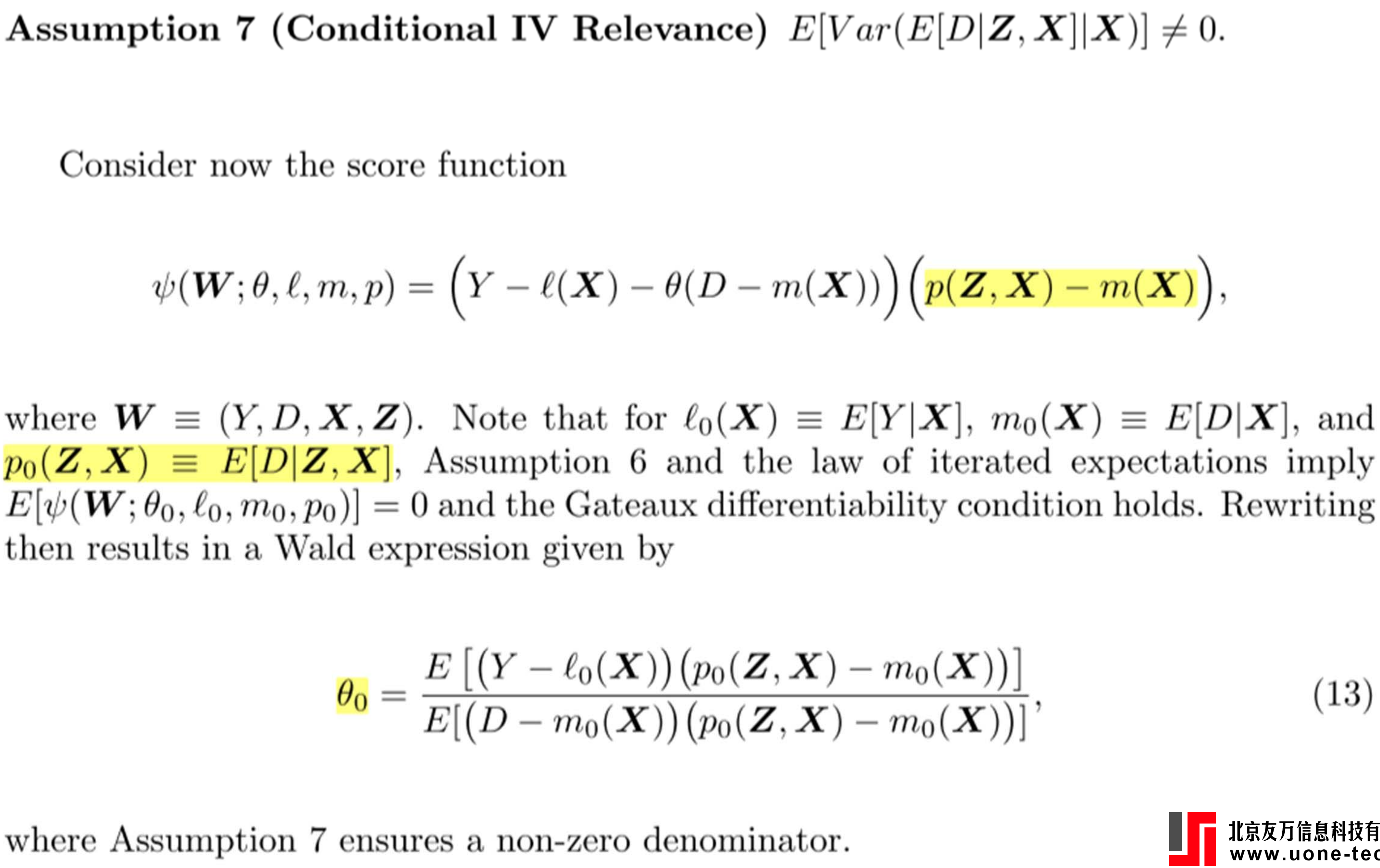
3.2 Flexible Partially Linear IV Model (`fiv`)

The Flexible Partially Linear IV Model considers the same parameter of interest as the Partially Linear IV Model. The key difference here is that identification is based on a stronger independence assumption which allows for approximating optimal instruments using nonparametric estimation, including machine learning, akin to Belloni et al. (2012) and Chernozhukov et al. (2015a). In particular, the Flexible Partially Linear IV Model leverages a conditional mean independence assumption rather than an orthogonality assumption as in Section 3.1. As in Section 3.1, we state everything in the case of a scalar D.

Assumption 6 (Conditional IV Mean Independence) $E[U|\boldsymbol{Z}, \boldsymbol{X}] = 0$.

- 也称为“Optimal IV model”。IV的高次项与非线性项仍是IV。可使用 $\mathrm{E}(D\,|\,\mathbf{Z},\mathbf{X})$ 作为最优IV。

灵活部分线性IV模型的得分函数



Assumption 7 (Conditional IV Relevance) $E[Var(E[D|\boldsymbol{Z},\boldsymbol{X}]|\boldsymbol{X})] \neq 0.$

Consider now the score function

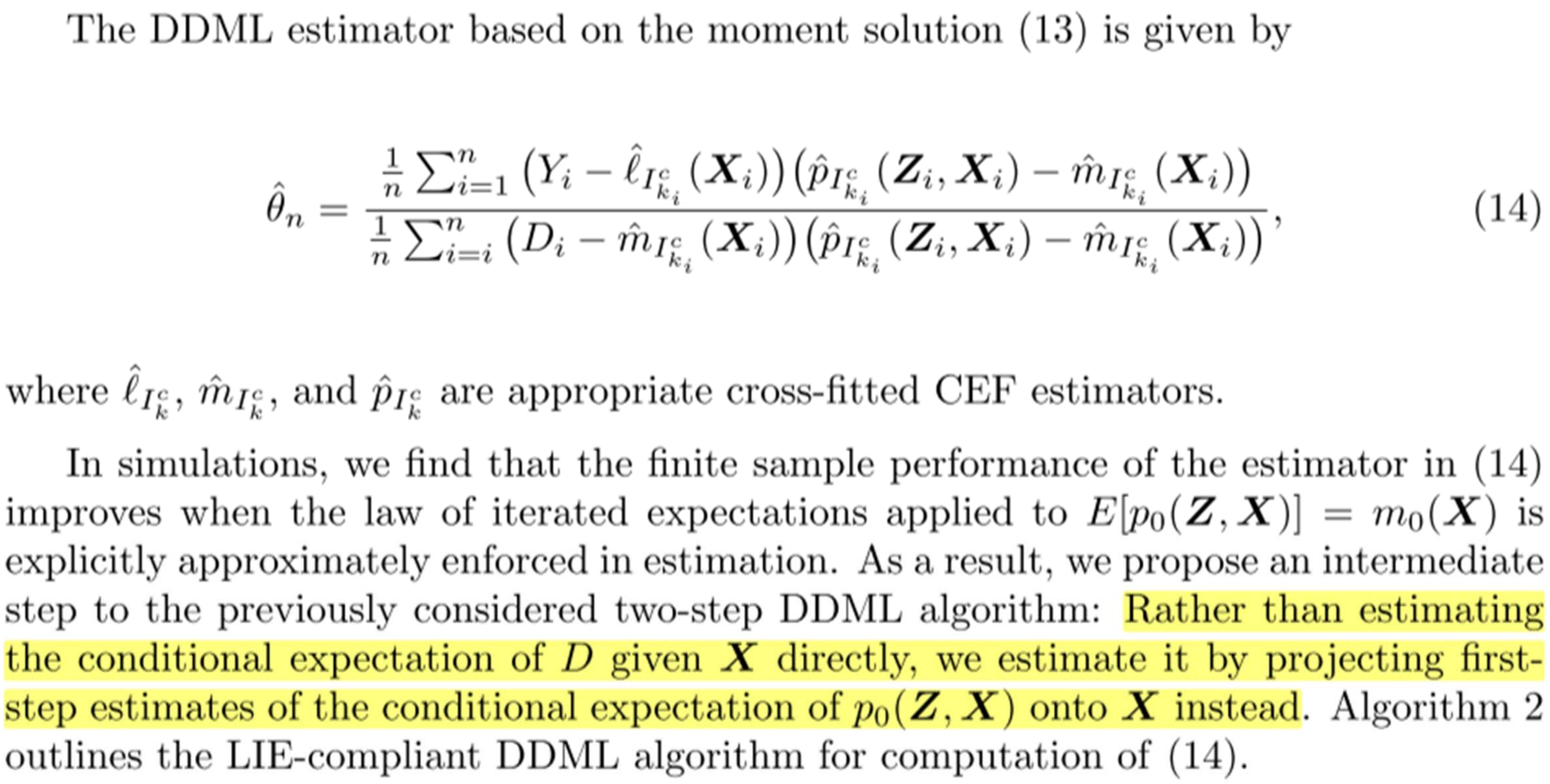
$$\psi(\boldsymbol{W};\theta,\ell,m,p) = \Big(Y - \ell(\boldsymbol{X}) - \theta(D - m(\boldsymbol{X}))\Big)\Big(p(\boldsymbol{Z},\boldsymbol{X}) - m(\boldsymbol{X})\Big),$$

where $\boldsymbol{W} \equiv (Y, D, \boldsymbol{X}, \boldsymbol{Z})$. Note that for $\ell_0(\boldsymbol{X}) \equiv E[Y|\boldsymbol{X}]$, $m_0(\boldsymbol{X}) \equiv E[D|\boldsymbol{X}]$, and $p_0(\boldsymbol{Z},\boldsymbol{X}) \equiv E[D|\boldsymbol{Z},\boldsymbol{X}]$, Assumption 6 and the law of iterated expectations imply $E[\psi(\boldsymbol{W};\theta_0,\ell_0,m_0,p_0)] = 0$ and the Gateaux differentiability condition holds. Rewriting then results in a Wald expression given by

$$\theta_0 = \frac{E\left[\big(Y - \ell_0(\boldsymbol{X})\big)\big(p_0(\boldsymbol{Z},\boldsymbol{X}) - m_0(\boldsymbol{X})\big)\right]}{E\big[\big(D - m_0(\boldsymbol{X})\big)\big(p_0(\boldsymbol{Z},\boldsymbol{X}) - m_0(\boldsymbol{X})\big)\big]}, \qquad (13)$$

where Assumption 7 ensures a non-zero denominator.

灵活部分线性IV模型的估计量



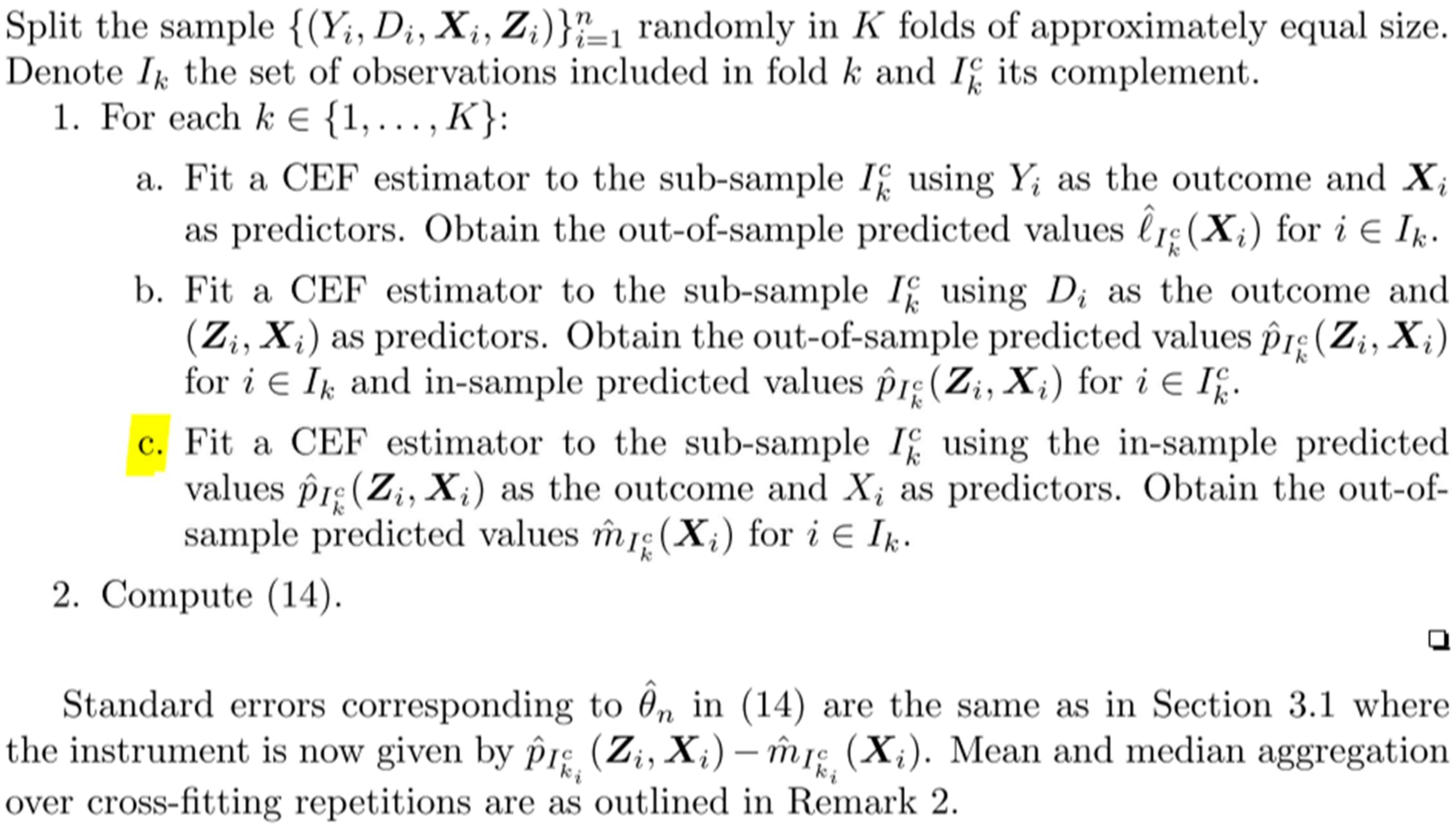
The DDML estimator based on the moment solution (13) is given by

$$\hat{\theta}_n = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{\ell}_{I^c_{k_i}}(\boldsymbol{X}_i)\right)\left(\hat{p}_{I^c_{k_i}}(\boldsymbol{Z}_i, \boldsymbol{X}_i) - \hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i)\right)}{\frac{1}{n}\sum_{i=i}^{n}\left(D_i - \hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i)\right)\left(\hat{p}_{I^c_{k_i}}(\boldsymbol{Z}_i, \boldsymbol{X}_i) - \hat{m}_{I^c_{k_i}}(\boldsymbol{X}_i)\right)}, \tag{14}$$

where $\hat{\ell}_{I^c_k}$, $\hat{m}_{I^c_k}$, and $\hat{p}_{I^c_k}$ are appropriate cross-fitted CEF estimators.

In simulations, we find that the finite sample performance of the estimator in (14) improves when the law of iterated expectations applied to $E[p_0(\boldsymbol{Z}, \boldsymbol{X})] = m_0(\boldsymbol{X})$ is explicitly approximately enforced in estimation. As a result, we propose an intermediate step to the previously considered two-step DDML algorithm: Rather than estimating the conditional expectation of D given $\boldsymbol{X}$ directly, we estimate it by projecting first-step estimates of the conditional expectation of $p_0(\boldsymbol{Z}, \boldsymbol{X})$ onto $\boldsymbol{X}$ instead. Algorithm 2 outlines the LIE-compliant DDML algorithm for computation of (14).

❑ Algorithm 2. LIE-compliant DDML for the Flexible Partially Linear IV Model.



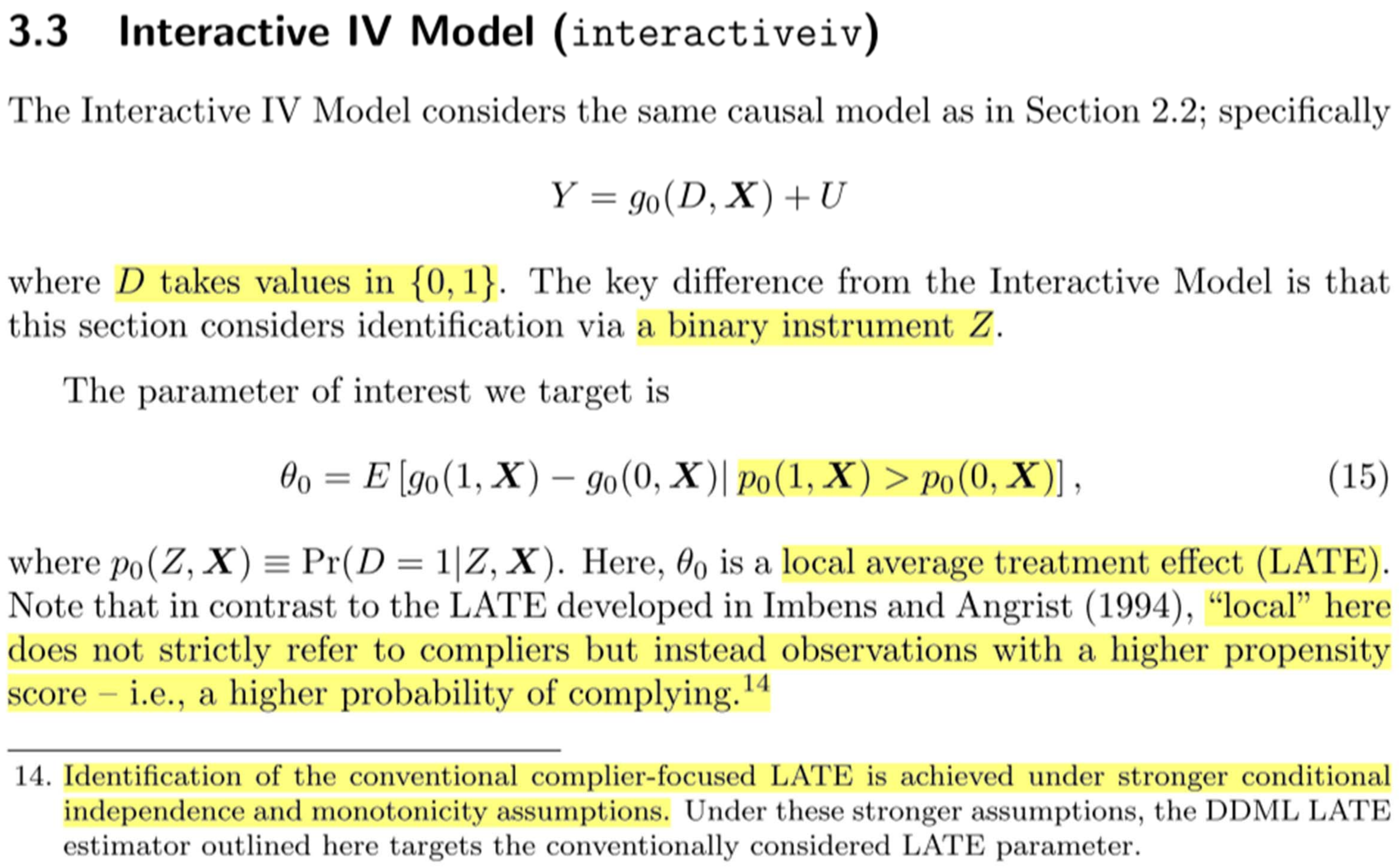
Split the sample $\{(Y_i, D_i, \boldsymbol{X}_i, \boldsymbol{Z}_i)\}_{i=1}^{n}$ randomly in K folds of approximately equal size. Denote I_k the set of observations included in fold k and I_k^c its complement.

1. For each $k \in \{1, \ldots, K\}$:
   - a. Fit a CEF estimator to the sub-sample I_k^c using Y_i as the outcome and $\boldsymbol{X}_i$ as predictors. Obtain the out-of-sample predicted values $\hat{\ell}_{I_k^c}(\boldsymbol{X}_i)$ for $i \in I_k$.
   - b. Fit a CEF estimator to the sub-sample I_k^c using D_i as the outcome and $(\boldsymbol{Z}_i, \boldsymbol{X}_i)$ as predictors. Obtain the out-of-sample predicted values $\hat{p}_{I_k^c}(\boldsymbol{Z}_i, \boldsymbol{X}_i)$ for $i \in I_k$ and in-sample predicted values $\hat{p}_{I_k^c}(\boldsymbol{Z}_i, \boldsymbol{X}_i)$ for $i \in I_k^c$.
   - c. Fit a CEF estimator to the sub-sample I_k^c using the in-sample predicted values $\hat{p}_{I_k^c}(\boldsymbol{Z}_i, \boldsymbol{X}_i)$ as the outcome and X_i as predictors. Obtain the out-of-sample predicted values $\hat{m}_{I_k^c}(\boldsymbol{X}_i)$ for $i \in I_k$.
2. Compute (14).

❑

Standard errors corresponding to $\hat{\theta}_n$ in (14) are the same as in Section 3.1 where the instrument is now given by $\hat{p}_{I_{k_i}^c}(\boldsymbol{Z}_i, \boldsymbol{X}_i) - \hat{m}_{I_{k_i}^c}(\boldsymbol{X}_i)$. Mean and median aggregation over cross-fitting repetitions are as outlined in Remark 2.

5. 交互工具变量模型



3.3 Interactive IV Model (`interactiveiv`)

The Interactive IV Model considers the same causal model as in Section 2.2; specifically

$$Y = g_0(D, \boldsymbol{X}) + U$$

where D takes values in $\{0, 1\}$. The key difference from the Interactive Model is that this section considers identification via a binary instrument Z.

The parameter of interest we target is

$$\theta_0 = E\left[g_0(1, \boldsymbol{X}) - g_0(0, \boldsymbol{X}) | \, p_0(1, \boldsymbol{X}) > p_0(0, \boldsymbol{X})\right], \tag{15}$$

where $p_0(Z, \boldsymbol{X}) \equiv \Pr(D = 1|Z, \boldsymbol{X})$. Here, θ_0 is a local average treatment effect (LATE). Note that in contrast to the LATE developed in Imbens and Angrist (1994), "local" here does not strictly refer to compliers but instead observations with a higher propensity score – i.e., a higher probability of complying.[14]

[14] Identification of the conventional complier-focused LATE is achieved under stronger conditional independence and monotonicity assumptions. Under these stronger assumptions, the DDML LATE estimator outlined here targets the conventionally considered LATE parameter.

Assumption 8 (Monotonicity) *$p_0(1,\boldsymbol{X}) \geq p_0(0,\boldsymbol{X})$ with probability 1.*

Assumption 9 (IV Overlap) *$\Pr(Z=1|\boldsymbol{X}) \in (0,1)$ with probability 1.*

Assumptions 6-9 imply that

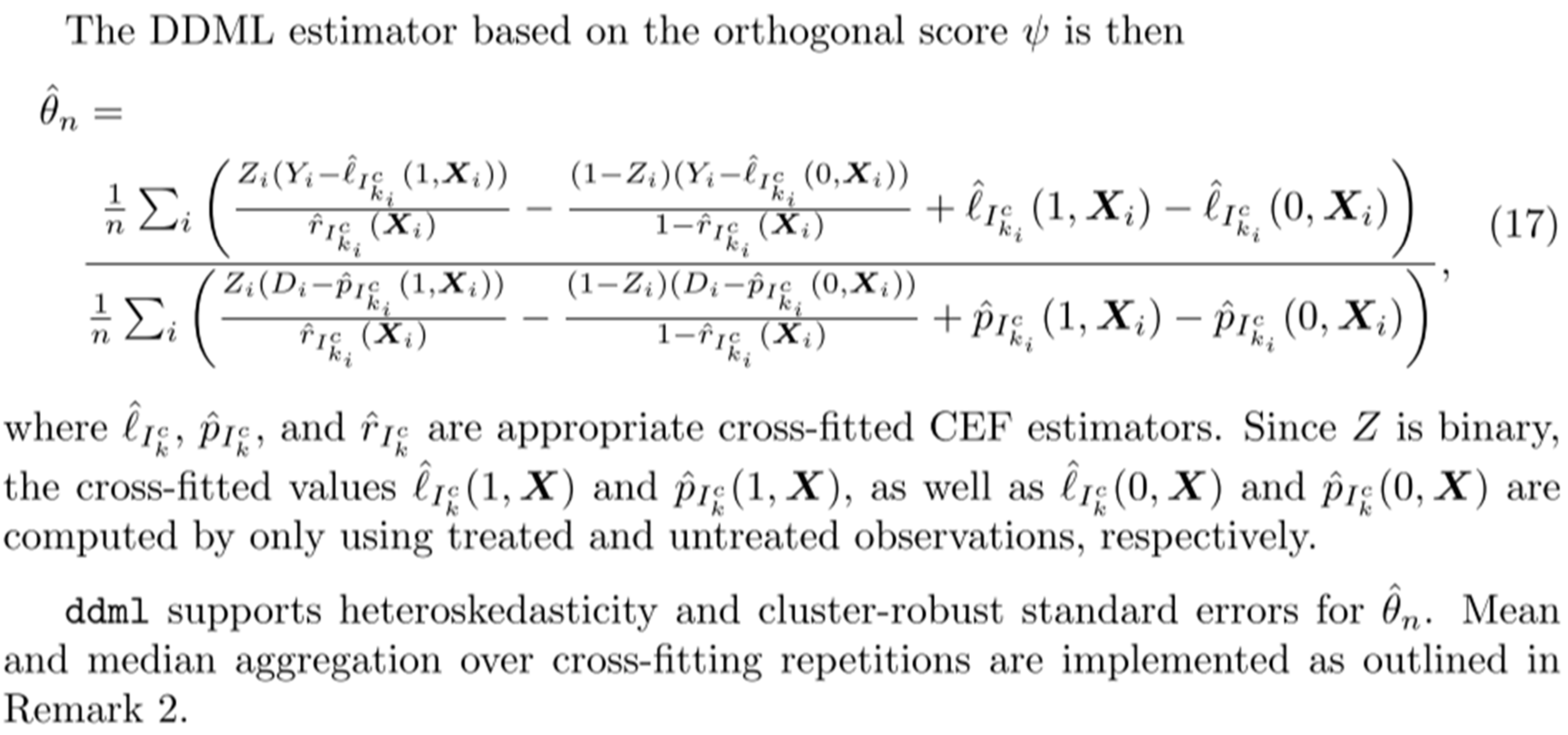
$$\theta_0 = \frac{E\left[\ell_0(1,\boldsymbol{X}) - \ell_0(0,\boldsymbol{X})\right]}{E\left[p_0(1,\boldsymbol{X}) - p_0(0,\boldsymbol{X})\right]}, \tag{16}$$

where $\ell_0(Z,\boldsymbol{X}) \equiv E[Y|Z,\boldsymbol{X}]$, verifying identification of the LATE θ_0. Akin to Section 6, however, estimators of θ_0 should not directly be based on Equation (16) because the estimating equations implicit in obtaining (16) do not satisfy Neyman-orthogonality. Hence, a direct estimator of θ_0 obtained by plugging nonparametric estimators in for nuisance functions in (16) will potentially be highly sensitive to the first step nonparametric estimation error. Rather, we base estimation on the Neyman orthogonal score function

$$\begin{aligned}\psi(\boldsymbol{W};\theta,\ell,p,r) = {} & \frac{Z(Y-\ell(1,\boldsymbol{X}))}{r(\boldsymbol{X})} - \frac{(1-Z)(Y-\ell(0,\boldsymbol{X}))}{1-r(\boldsymbol{X})} + \ell(1,\boldsymbol{X}) - \ell(0,\boldsymbol{X}) \\ & + \left[\frac{Z(D-p(1,\boldsymbol{X}))}{r(\boldsymbol{X})} - \frac{(1-Z)(D-p(0,\boldsymbol{X}))}{1-r(\boldsymbol{X})} + p(1,\boldsymbol{X}) - p(0,\boldsymbol{X})\right] \times \theta\end{aligned}$$

where $\boldsymbol{W} \equiv (Y,D,\boldsymbol{X},Z)$. Note that under Assumptions 6-9 and for $\ell_0(Z,\boldsymbol{X}) \equiv E[Y|Z,\boldsymbol{X}]$, $p_0(Z,\boldsymbol{X}) \equiv E[D|Z,\boldsymbol{X}]$, and $r_0(\boldsymbol{X}) \equiv E[Z|\boldsymbol{X}]$, we have $E[\psi(\boldsymbol{W};\theta_0,\ell_0,p_0,r_0)] = 0$ and can verify that its Gateaux derivative with respect to the nuisance functions local to their true values is also zero.

交互IV模型的估计量



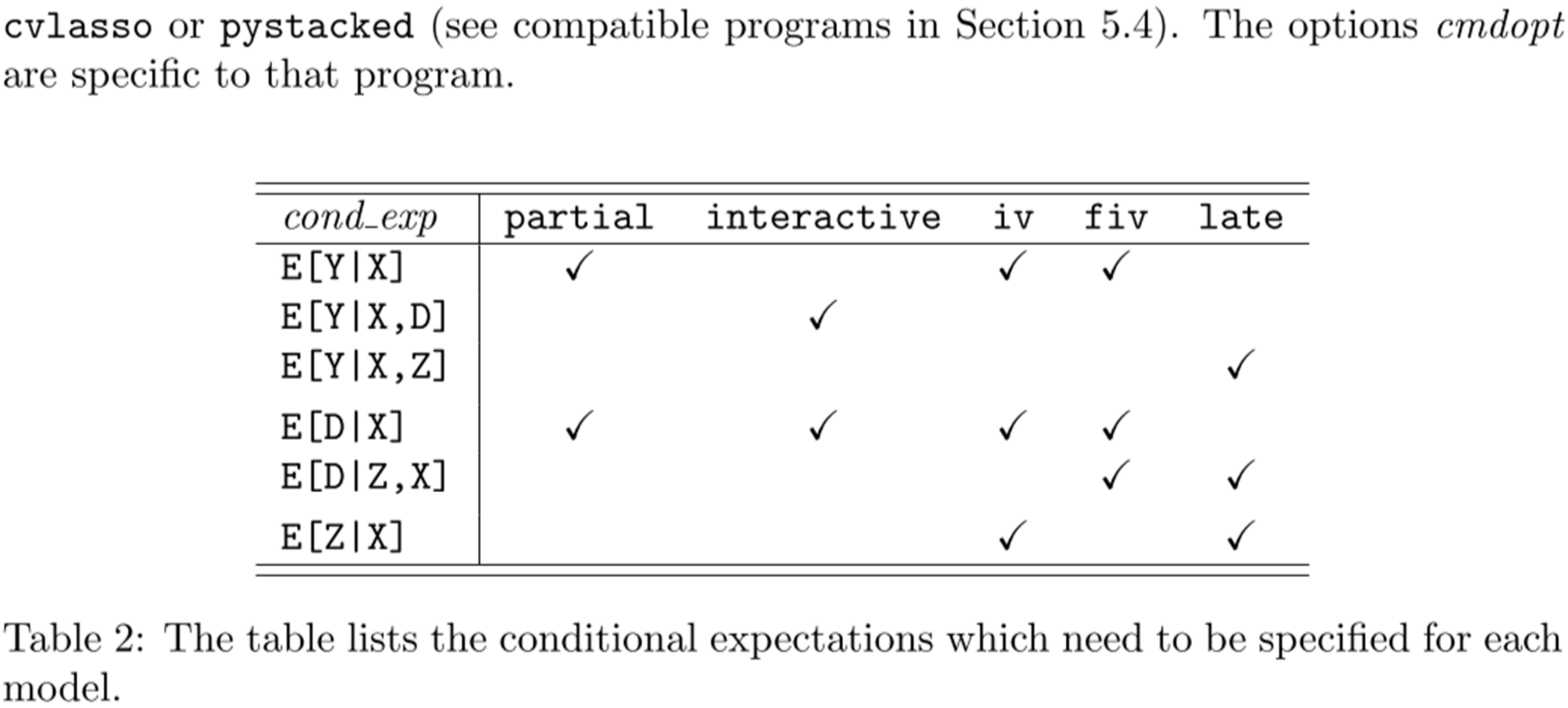
The DDML estimator based on the orthogonal score ψ is then

$$\hat{\theta}_n = \frac{\frac{1}{n}\sum_i \left(\frac{Z_i(Y_i - \hat{\ell}_{I^c_{k_i}}(1,\boldsymbol{X}_i))}{\hat{r}_{I^c_{k_i}}(\boldsymbol{X}_i)} - \frac{(1-Z_i)(Y_i - \hat{\ell}_{I^c_{k_i}}(0,\boldsymbol{X}_i))}{1-\hat{r}_{I^c_{k_i}}(\boldsymbol{X}_i)} + \hat{\ell}_{I^c_{k_i}}(1,\boldsymbol{X}_i) - \hat{\ell}_{I^c_{k_i}}(0,\boldsymbol{X}_i) \right)}{\frac{1}{n}\sum_i \left(\frac{Z_i(D_i - \hat{p}_{I^c_{k_i}}(1,\boldsymbol{X}_i))}{\hat{r}_{I^c_{k_i}}(\boldsymbol{X}_i)} - \frac{(1-Z_i)(D_i - \hat{p}_{I^c_{k_i}}(0,\boldsymbol{X}_i))}{1-\hat{r}_{I^c_{k_i}}(\boldsymbol{X}_i)} + \hat{p}_{I^c_{k_i}}(1,\boldsymbol{X}_i) - \hat{p}_{I^c_{k_i}}(0,\boldsymbol{X}_i) \right)}, \qquad (17)$$

where $\hat{\ell}_{I^c_k}$, $\hat{p}_{I^c_k}$, and $\hat{r}_{I^c_k}$ are appropriate cross-fitted CEF estimators. Since Z is binary, the cross-fitted values $\hat{\ell}_{I^c_k}(1,\boldsymbol{X})$ and $\hat{p}_{I^c_k}(1,\boldsymbol{X})$, as well as $\hat{\ell}_{I^c_k}(0,\boldsymbol{X})$ and $\hat{p}_{I^c_k}(0,\boldsymbol{X})$ are computed by only using treated and untreated observations, respectively.

`ddml` supports heteroskedasticity and cluster-robust standard errors for $\hat{\theta}_n$. Mean and median aggregation over cross-fitting repetitions are implemented as outlined in Remark 2.

机器学习所估计的条件期望函数(CEF)



`cvlasso` or `pystacked` (see compatible programs in Section 5.4). The options *cmdopt* are specific to that program.

cond_exp	`partial`	`interactive`	`iv`	`fiv`	`late`
`E[Y|X]`	✓		✓	✓	
`E[Y|X,D]`		✓			
`E[Y|X,Z]`					✓
`E[D|X]`	✓	✓	✓	✓	
`E[D|Z,X]`				✓	✓
`E[Z|X]`			✓		✓

Table 2: The table lists the conditional expectations which need to be specified for each model.

应使用何种机器学习算法？

- 机器学习算法的优越性取决于“数据生成过程”(data generating process, DGP)
- 例如：若DGP为线性，则线性模型最优
- 共识：不存在绝对最优的机器学习算法(无论在任何DGP下都最优)

引入集成学习

- 引入“集成学习”(ensemble learning)，也称“组合学习”或“堆叠”(stacking)。

- 以部分线性模型为例。假设分别使用三个“基学习器”(base learners)，通过交叉拟合估计$l_0(\mathbf{X}_i) \equiv \mathrm{E}(Y_i \mid \mathbf{X}_i)$与$m_0(\mathbf{X}_i) \equiv \mathrm{E}(D_i \mid \mathbf{X}_i)$

- 记估计结果为$\hat{l}_0^{(j)}(\mathbf{X}_i)$与$\hat{m}_0^{(j)}(\mathbf{X}_i)$，$j = 1, 2, 3$

加权平均

- 将三个预测结果进行加权平均：

$$\hat{l}_0(\mathbf{X}_i) = w_1 \hat{l}_0^{(1)}(\mathbf{X}_i) + w_2 \hat{l}_0^{(2)}(\mathbf{X}_i) + w_3 \hat{l}_0^{(3)}(\mathbf{X}_i)$$

- 其中，权重 w_1, w_2, w_3 非负，且权重之和为1

- 类似地，将 $\hat{m}_0^{(j)}(\mathbf{X}_i)\ (j = 1,2,3)$ 也进行加权平均。理论上，堆叠学习器的预测能力一定不弱于每个基学习器。

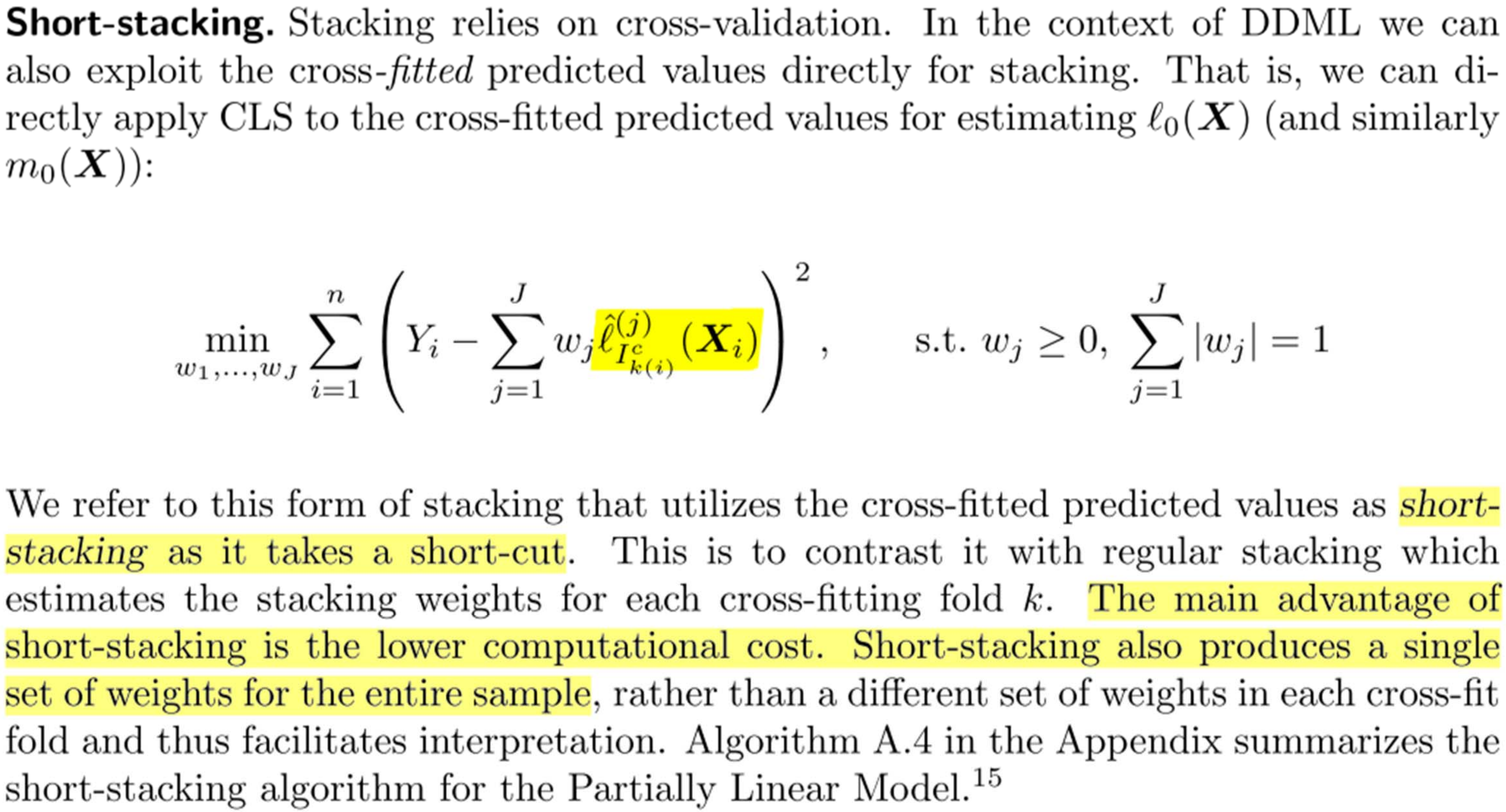
最优权重

- 堆叠学习器的预测能力取决于权重

- 可通过“约束最小二乘法”(Constrained Least Squares, CLS)选择最优权重

- 假设共有J个基学习器，对于个体i，记其第j个基学习器通过交叉拟合的预测结果为 $\hat{l}^{(j)}_{I^c_{k_i}}(\mathbf{X}_i)$

短堆叠(short-stacking)



Short-stacking. Stacking relies on cross-validation. In the context of DDML we can also exploit the cross-*fitted* predicted values directly for stacking. That is, we can directly apply CLS to the cross-fitted predicted values for estimating $\ell_0(\boldsymbol{X})$ (and similarly $m_0(\boldsymbol{X})$):

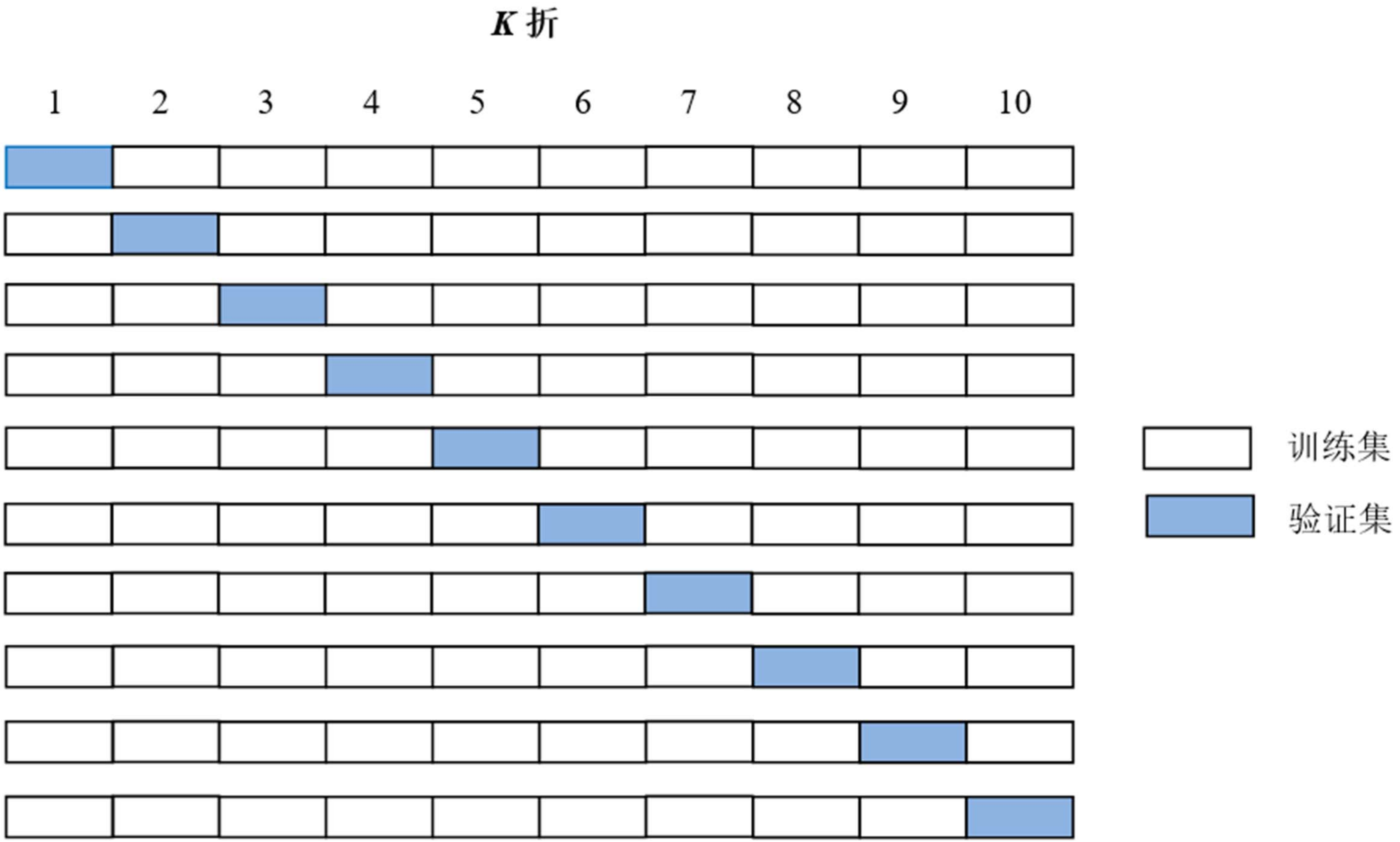
$$\min_{w_1,\dots,w_J}\sum_{i=1}^{n}\left(Y_i-\sum_{j=1}^{J}w_j\hat{\ell}^{(j)}_{I^c_{k(i)}}(\boldsymbol{X}_i)\right)^2, \qquad \text{s.t. } w_j\geq 0,\ \sum_{j=1}^{J}|w_j|=1$$

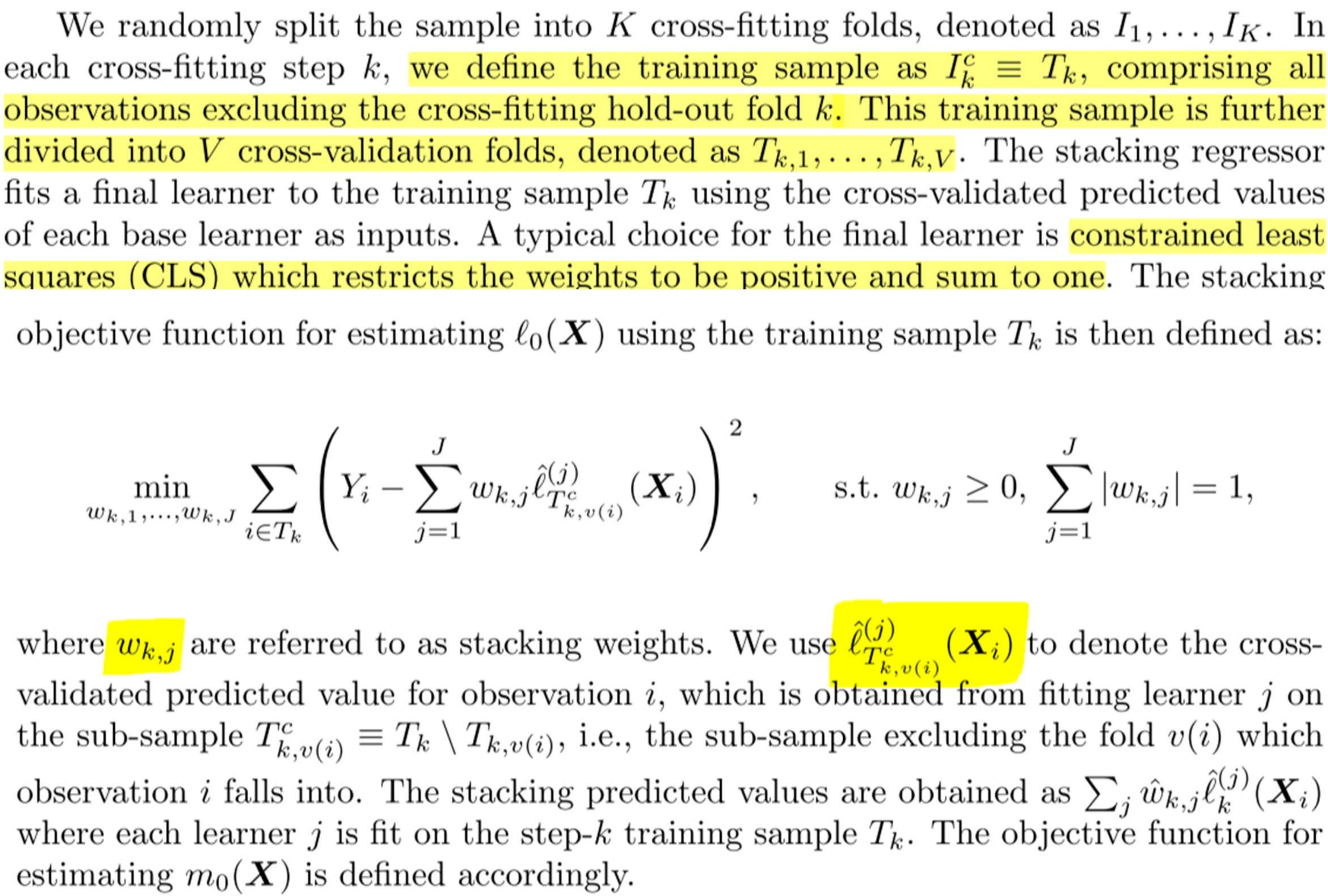
We refer to this form of stacking that utilizes the cross-fitted predicted values as *short-stacking* as it takes a short-cut. This is to contrast it with regular stacking which estimates the stacking weights for each cross-fitting fold k. The main advantage of short-stacking is the lower computational cost. Short-stacking also produces a single set of weights for the entire sample, rather than a different set of weights in each cross-fit fold and thus facilitates interpretation. Algorithm A.4 in the Appendix summarizes the short-stacking algorithm for the Partially Linear Model.[15]

常规堆叠(regular stacking)

- 短堆叠赋予每个基学习器一个统一的权重 $\left\{w_j\right\}_{j=1,\cdots,J}$
- 常规堆叠(regular stacking)则针对不同折(folds)，赋予不同的权重 $\left\{w_{kj}\right\}_{k=1,\cdots,K;\,j=1,\cdots,J}$
- 对于第*k*折数据 I_k，使用其补集 I_k^c 进行“交叉验证”(cross-validation)

交叉验证(Cross-validation)





We randomly split the sample into K cross-fitting folds, denoted as $I_1, \ldots, I_K$. In each cross-fitting step k, we define the training sample as $I_k^c \equiv T_k$, comprising all observations excluding the cross-fitting hold-out fold k. This training sample is further divided into V cross-validation folds, denoted as $T_{k,1}, \ldots, T_{k,V}$. The stacking regressor fits a final learner to the training sample T_k using the cross-validated predicted values of each base learner as inputs. A typical choice for the final learner is constrained least squares (CLS) which restricts the weights to be positive and sum to one. The stacking objective function for estimating $\ell_0(\boldsymbol{X})$ using the training sample T_k is then defined as:

$$\min_{w_{k,1},\ldots,w_{k,J}} \sum_{i \in T_k} \left(Y_i - \sum_{j=1}^{J} w_{k,j} \hat{\ell}^{(j)}_{T^c_{k,v(i)}}(\boldsymbol{X}_i) \right)^2, \qquad \text{s.t. } w_{k,j} \geq 0, \ \sum_{j=1}^{J} |w_{k,j}| = 1,$$

where $w_{k,j}$ are referred to as stacking weights. We use $\hat{\ell}^{(j)}_{T^c_{k,v(i)}}(\boldsymbol{X}_i)$ to denote the cross-validated predicted value for observation i, which is obtained from fitting learner j on the sub-sample $T^c_{k,v(i)} \equiv T_k \setminus T_{k,v(i)}$, i.e., the sub-sample excluding the fold $v(i)$ which observation i falls into. The stacking predicted values are obtained as $\sum_j \hat{w}_{k,j} \hat{\ell}^{(j)}_k(\boldsymbol{X}_i)$ where each learner j is fit on the step-k training sample T_k. The objective function for estimating $m_0(\boldsymbol{X})$ is defined accordingly.

双重机器学习的Stata实现

- `ssc install ddml, all replace`
- `ssc install lassopack, all replace`
- `ssc install rforest, all replace`
- `ssc install pystacked,all replace`

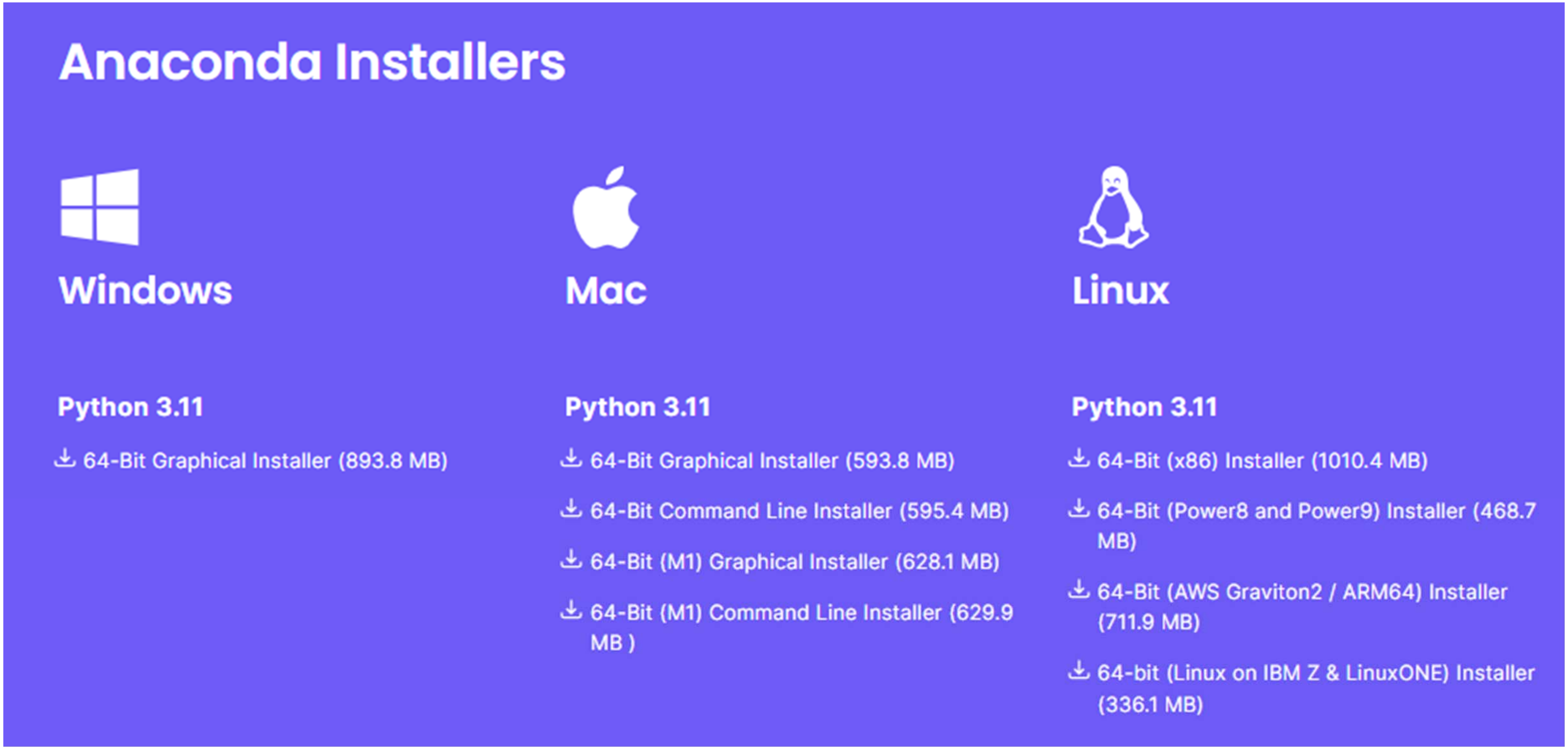
- Stata命令`ddml`调用Python实现双重机器学习，故需安装Python。

- 作为通用语言， Python本身功能有限，常需载入其他模块(modules)，比如 numpy, pandas，sklearn

- 通过Anaconda下载Python，则自带这些常用模块

通过Anaconda安装Python

- 登录Anaconda官网：https://www.anaconda.com/download



Anaconda Installers

Windows

Python 3.11

64-Bit Graphical Installer (893.8 MB)

Mac

Python 3.11

64-Bit Graphical Installer (593.8 MB)

64-Bit Command Line Installer (595.4 MB)

64-Bit (M1) Graphical Installer (628.1 MB)

64-Bit (M1) Command Line Installer (629.9 MB)

Linux

Python 3.11

64-Bit (x86) Installer (1010.4 MB)

64-Bit (Power8 and Power9) Installer (468.7 MB)

64-Bit (AWS Graviton2 / ARM64) Installer (711.9 MB)

64-bit (Linux on IBM Z & LinuxONE) Installer (336.1 MB)

设置Python

- python search

```
Python environments found:
C:\Users\DELL\anaconda3\python.exe
```

- python set exec C:\Users\DELL\anaconda3\python.exe, permanently

```
(python_exec preference recorded)
```

检查Python设置

- python query

```
--------------------------------------------------------------------------------
Python Settings
  set python_exec         C:\Users\DELL\anaconda3\python.exe
  set python_userpath

Python system information
  initialized             no
  version                 3.8.8
  architecture            64-bit
  library path            C:\Users\DELL\anaconda3\python38.dll
```

- python which sklearn

```
<module 'sklearn' from 'C:\\Users\\DELL\\anaconda3\\lib
> \\site-packages\\sklearn\\__init__.py'>
```

在Stata中调用Python的方式一

- 方式一：交互式 (可执行多行命令)。
- 在命令窗口，输入“python”进行Python状态，输入“end”结束Python状态，返回Stata

```
. python
───────────────────────────────────────────────────── python (type end to exit) ──
>>> print('Hello Stata, I am Python')
Hello Stata, I am Python
>>> 2*3
6
>>> end
──────────────────────────────────────────────────────────────────────────────────
```

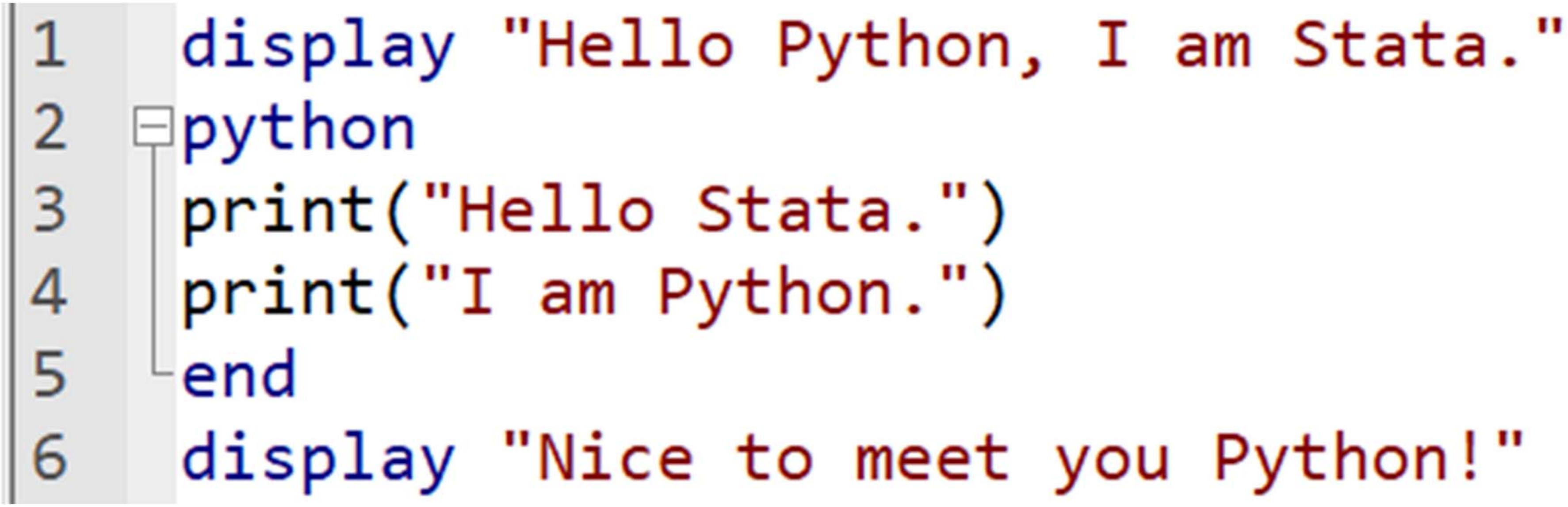
在Stata中调用Python的方式二

- 方式二：交互式 (执行单行命令)。
- 在命令窗口，输入“python: python command”，回车后得到结果，即返回Stata

```
. python: print('Hello Stata, I am Python')
Hello Stata, I am Python

.
```

在Stata中调用Python的方式三

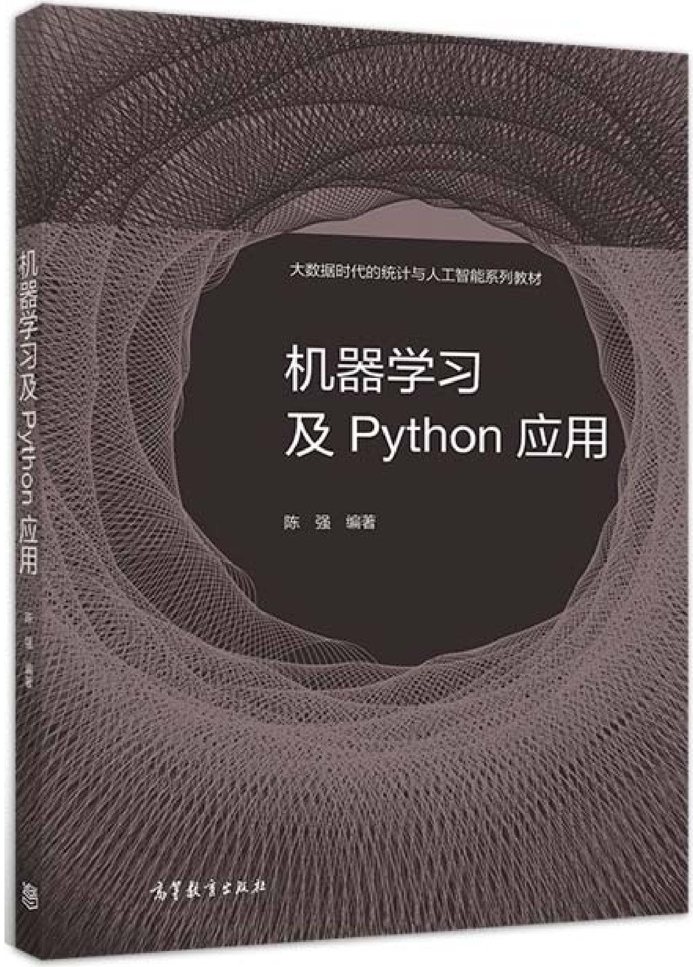
- 方式三：在do文件中调用Python



```
display "Hello Python, I am Stata."
python
print("Hello Stata.")
print("I am Python.")
end
display "Nice to meet you Python!"
```

```
. display "Hello Python, I am Stata."
Hello Python, I am Stata.

. python
─────────────────────────────────────────────── python (type end to exit) ─
>>> print("Hello Stata.")
Hello Stata.
>>> print("I am Python.")
I am Python.
>>> end
──────────────────────────────────────────────────────────────────────────

. display "Nice to meet you Python!"
Nice to meet you Python!
```

进一步学习Python

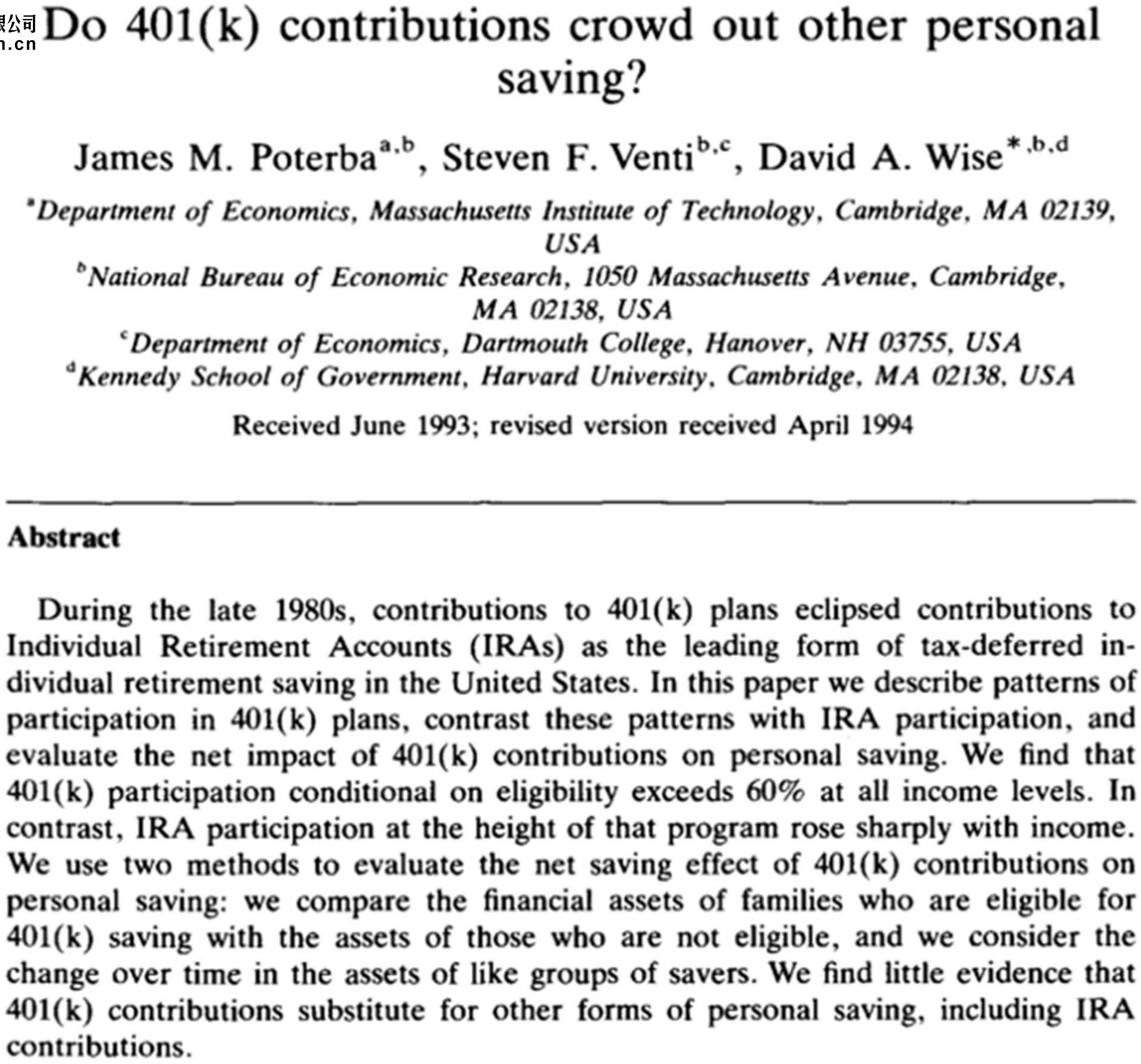


陈强，《机器学习及Python应用》，高等教育出版社，2021年

案例：参与401(k)养老金计划的资格对净金融资产的影响 (Poterba et al., 1995)

- **Outcome**: net total financial assets (*net_tfa*)
- **Treatment**: eligibility to enroll for the 401(k) pension plan (*e401*)

- **Control variables**: age, income, years of education, family size
- **Control variables (indicators)**: martial status, two-earner status, benefit pension status, IRA participation, home ownership



Do 401(k) contributions crowd out other personal saving?

James M. Poterba[a,b], Steven F. Venti[b,c], David A. Wise*[,b,d]

[a]*Department of Economics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

[b]*National Bureau of Economic Research, 1050 Massachusetts Avenue, Cambridge, MA 02138, USA*

[c]*Department of Economics, Dartmouth College, Hanover, NH 03755, USA*

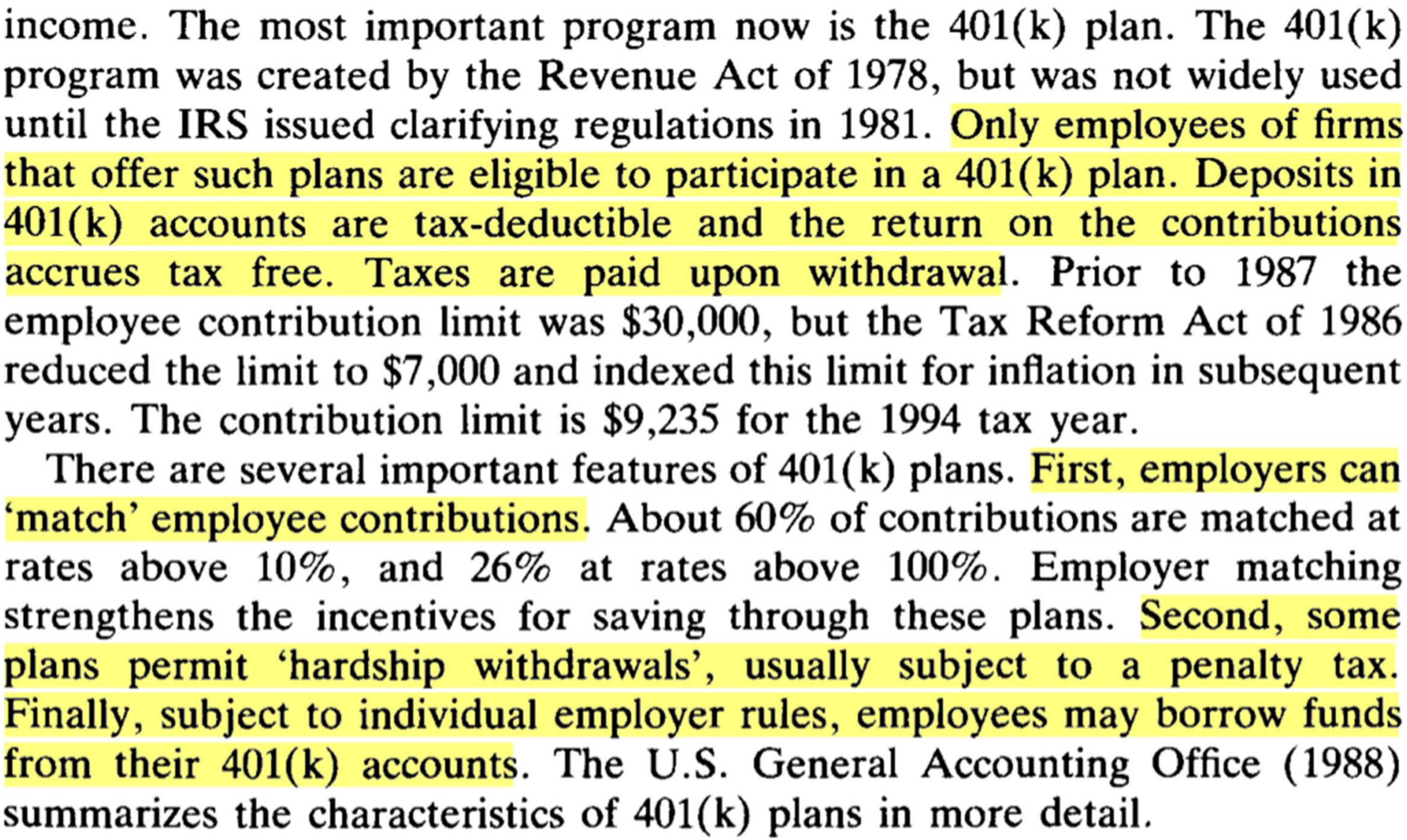
[d]*Kennedy School of Government, Harvard University, Cambridge, MA 02138, USA*

Received June 1993; revised version received April 1994

Abstract

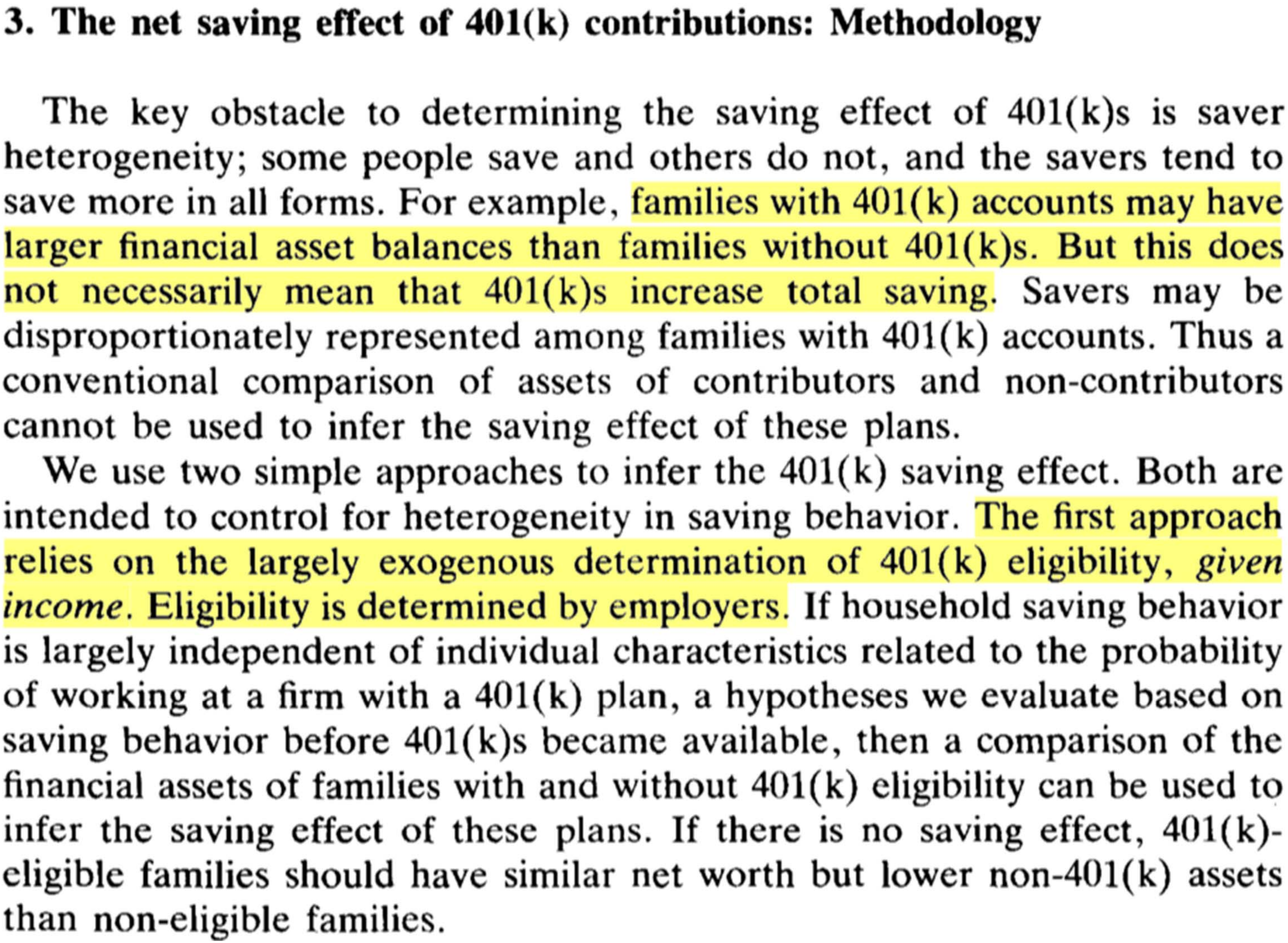
During the late 1980s, contributions to 401(k) plans eclipsed contributions to Individual Retirement Accounts (IRAs) as the leading form of tax-deferred individual retirement saving in the United States. In this paper we describe patterns of participation in 401(k) plans, contrast these patterns with IRA participation, and evaluate the net impact of 401(k) contributions on personal saving. We find that 401(k) participation conditional on eligibility exceeds 60% at all income levels. In contrast, IRA participation at the height of that program rose sharply with income. We use two methods to evaluate the net saving effect of 401(k) contributions on personal saving: we compare the financial assets of families who are eligible for 401(k) saving with the assets of those who are not eligible, and we consider the change over time in the assets of like groups of savers. We find little evidence that 401(k) contributions substitute for other forms of personal saving, including IRA contributions.

401(k)养老金计划的制度背景



income. The most important program now is the 401(k) plan. The 401(k) program was created by the Revenue Act of 1978, but was not widely used until the IRS issued clarifying regulations in 1981. Only employees of firms that offer such plans are eligible to participate in a 401(k) plan. Deposits in 401(k) accounts are tax-deductible and the return on the contributions accrues tax free. Taxes are paid upon withdrawal. Prior to 1987 the employee contribution limit was $30,000, but the Tax Reform Act of 1986 reduced the limit to $7,000 and indexed this limit for inflation in subsequent years. The contribution limit is $9,235 for the 1994 tax year.

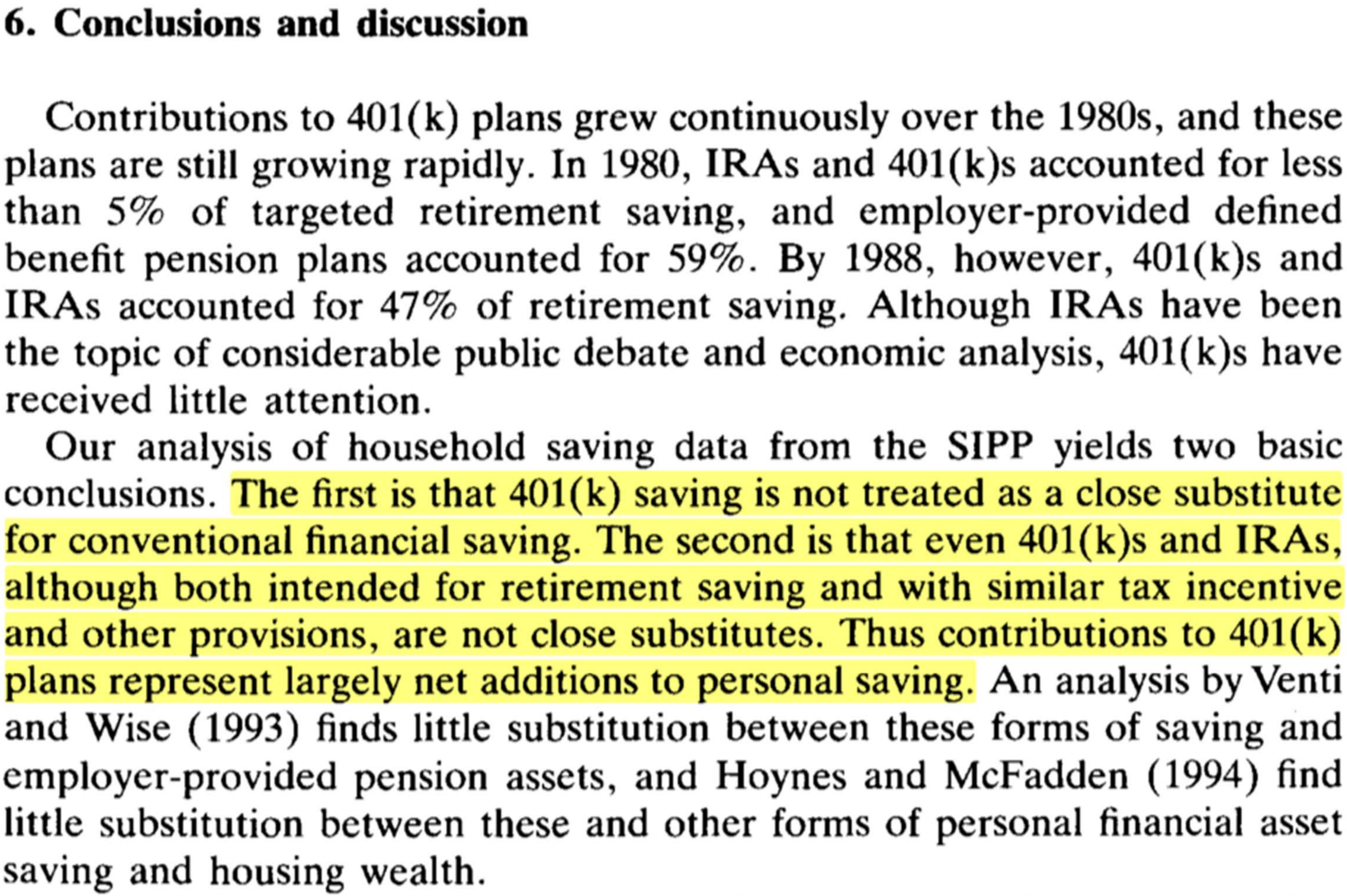
There are several important features of 401(k) plans. First, employers can 'match' employee contributions. About 60% of contributions are matched at rates above 10%, and 26% at rates above 100%. Employer matching strengthens the incentives for saving through these plans. Second, some plans permit 'hardship withdrawals', usually subject to a penalty tax. Finally, subject to individual employer rules, employees may borrow funds from their 401(k) accounts. The U.S. General Accounting Office (1988) summarizes the characteristics of 401(k) plans in more detail.



3. The net saving effect of 401(k) contributions: Methodology

The key obstacle to determining the saving effect of 401(k)s is saver heterogeneity; some people save and others do not, and the savers tend to save more in all forms. For example, families with 401(k) accounts may have larger financial asset balances than families without 401(k)s. But this does not necessarily mean that 401(k)s increase total saving. Savers may be disproportionately represented among families with 401(k) accounts. Thus a conventional comparison of assets of contributors and non-contributors cannot be used to infer the saving effect of these plans.

We use two simple approaches to infer the 401(k) saving effect. Both are intended to control for heterogeneity in saving behavior. The first approach relies on the largely exogenous determination of 401(k) eligibility, *given income*. Eligibility is determined by employers. If household saving behavior is largely independent of individual characteristics related to the probability of working at a firm with a 401(k) plan, a hypotheses we evaluate based on saving behavior before 401(k)s became available, then a comparison of the financial assets of families with and without 401(k) eligibility can be used to infer the saving effect of these plans. If there is no saving effect, 401(k)-eligible families should have similar net worth but lower non-401(k) assets than non-eligible families.



6. Conclusions and discussion

Contributions to 401(k) plans grew continuously over the 1980s, and these plans are still growing rapidly. In 1980, IRAs and 401(k)s accounted for less than 5% of targeted retirement saving, and employer-provided defined benefit pension plans accounted for 59%. By 1988, however, 401(k)s and IRAs accounted for 47% of retirement saving. Although IRAs have been the topic of considerable public debate and economic analysis, 401(k)s have received little attention.

Our analysis of household saving data from the SIPP yields two basic conclusions. The first is that 401(k) saving is not treated as a close substitute for conventional financial saving. The second is that even 401(k)s and IRAs, although both intended for retirement saving and with similar tax incentive and other provisions, are not close substitutes. Thus contributions to 401(k) plans represent largely net additions to personal saving. An analysis by Venti and Wise (1993) finds little substitution between these forms of saving and employer-provided pension assets, and Hoynes and McFadden (1994) find little substitution between these and other forms of personal financial asset saving and housing wealth.

- use https://github.com/aahrens1/ddml/raw/master/data/sipp1991.dta, clear
- describe

```
Contains data from sipp1991.dta
 Observations:         9,915
    Variables:            14                  20 Jun 2013 14:08
-------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------
nifa            float   %9.0g                 Net non-401(k) financial
                                                assets
net_tfa         float   %9.0g                 Net total financial assets
tw              float   %9.0g                 Total wealth
age             byte    %9.0g                 Age of the head of the
                                                household
inc             float   %9.0g                 Household income
fsize           byte    %9.0g                 Household size
educ            byte    %9.0g                 Years of education of the
                                                head of the household
db              byte    %9.0g                 Defined benefit pension
                                                status indicator
marr            byte    %9.0g                 Married indicator
twoearn         byte    %9.0g                 Two-earner status indicator
e401            byte    %9.0g                 401(k) eligibility
p401            byte    %9.0g                 401(k) participation
pira            byte    %9.0g                 IRA participation indicator
hown            byte    %9.0g                 House ownership indicator
```

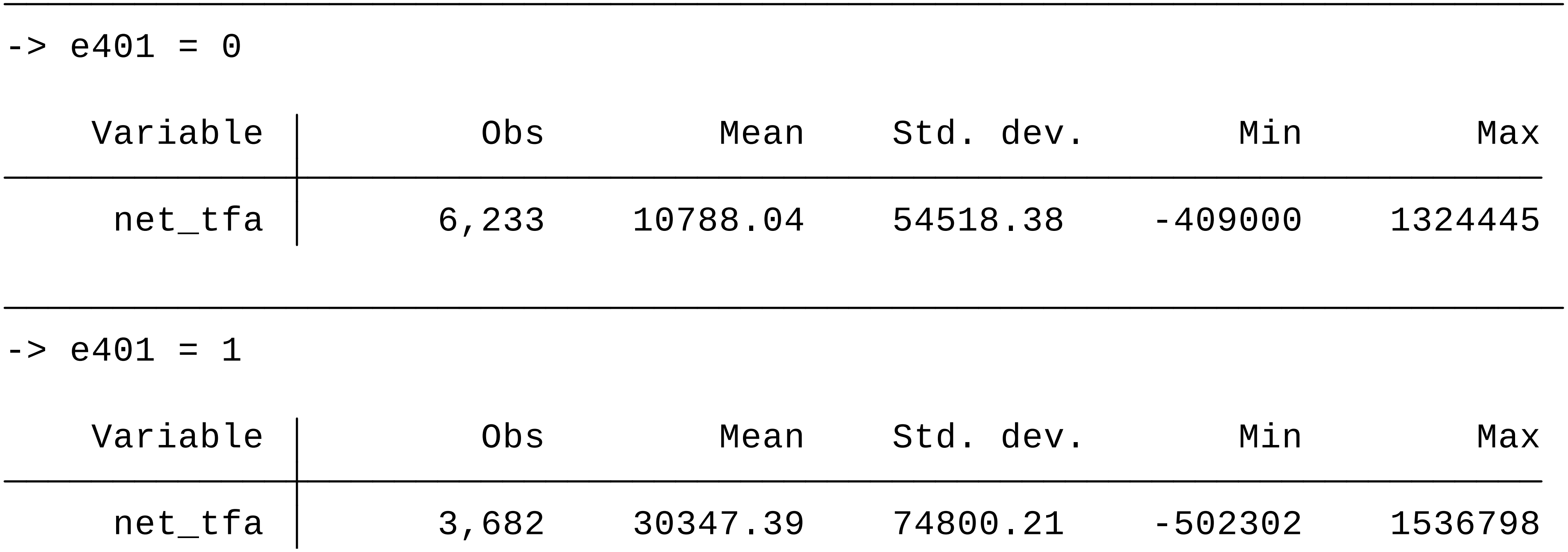
统计特征

- sum

```
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
        nifa |      9,915    13928.64    54904.88          0    1430298
     net_tfa |      9,915    18051.53     63522.5    -502302    1536798
          tw |      9,915    63816.85    111529.7    -502302    2029910
         age |      9,915    41.06021     10.3445         25         64
         inc |      9,915    37200.62    24774.29      -2652     242124
-------------+---------------------------------------------------------
       fsize |      9,915     2.86586    1.538937          1         13
        educ |      9,915    13.20625    2.810382          1         18
          db |      9,915    .2710035    .4445003          0          1
        marr |      9,915    .6048411    .4889094          0          1
     twoearn |      9,915    .3808371    .4856171          0          1
-------------+---------------------------------------------------------
        e401 |      9,915    .3713565    .4831919          0          1
        p401 |      9,915    .2616238     .439541          0          1
        pira |      9,915    .2421583    .4284112          0          1
        hown |      9,915    .6351992    .4813985          0          1
```

分组统计特征

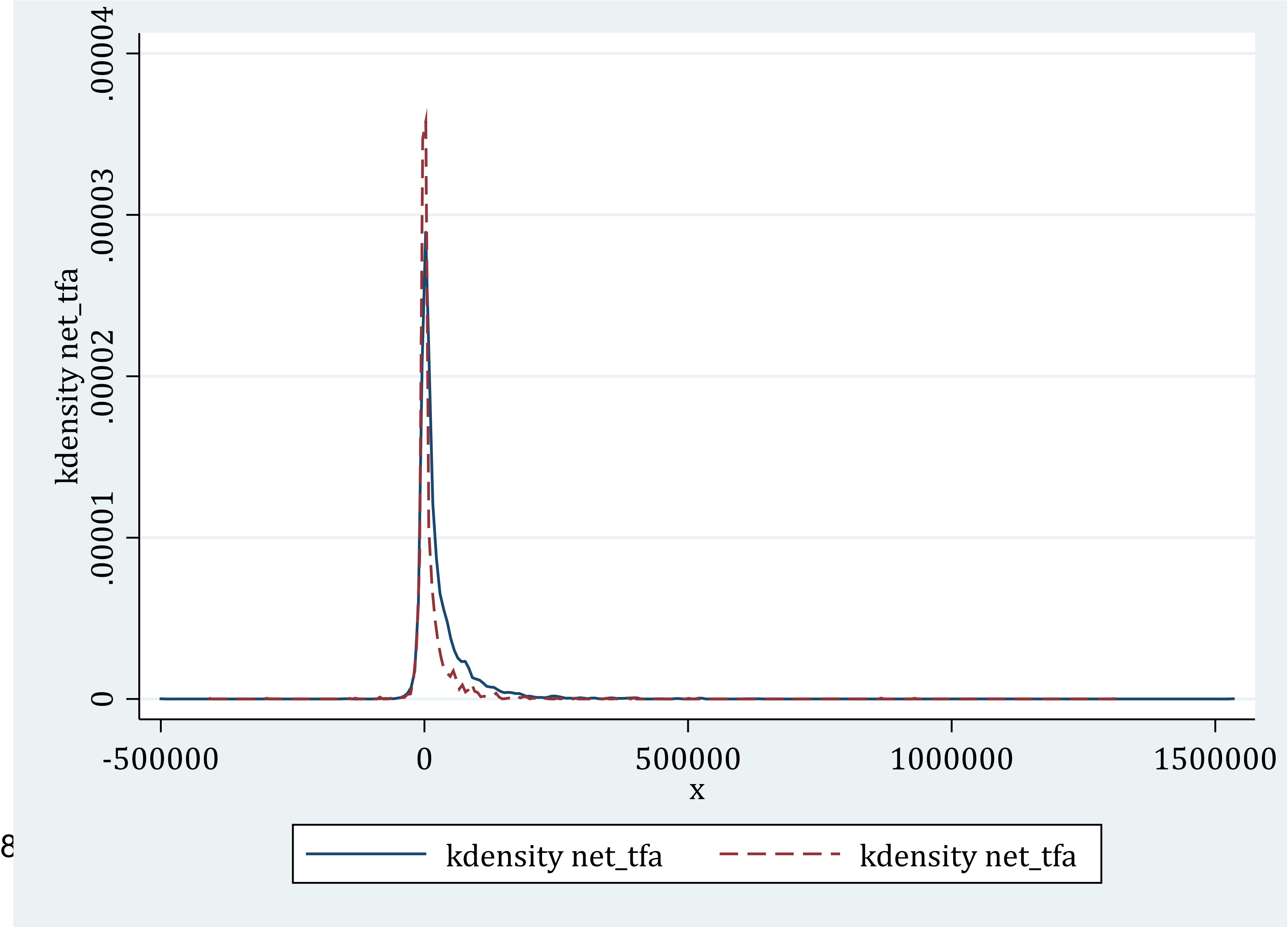
- bysort e401: sum net_tfa



```
-> e401 = 0

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
     net_tfa |      6,233    10788.04    54518.38    -409000    1324445

-> e401 = 1

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
     net_tfa |      3,682    30347.39    74800.21    -502302    1536798
```

核密度图的比较

- `twoway (kdensity net_tfa if e401==1) (kdensity net_tfa if e401==0,lp(dash))`



定义变量及OLS回归

- global Y net_tfa
- global X age inc educ fsize marr twoearn db pira hown
- global D e401
- reg $Y $D $X,r

OLS回归结果

```
Linear regression                               Number of obs     =      9,915
                                                F(10, 9904)       =     106.74
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2312
                                                Root MSE          =      55726

------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   5896.198   1524.034     3.87   0.000     2908.782    8883.615
         age |   624.1455   55.47429    11.25   0.000     515.4046    732.8864
         inc |   .9356652   .1093677     8.56   0.000     .7212823    1.150048
        educ |  -639.7538   335.7713    -1.91   0.057    -1297.934    18.42628
       fsize |  -1018.798   386.2917    -2.64   0.008    -1776.008   -261.5876
        marr |   743.3445   1732.644     0.43   0.668    -2652.991     4139.68
     twoearn |  -19226.92   2530.589    -7.60   0.000    -24187.39   -14266.45
          db |  -4904.568   1303.413    -3.76   0.000    -7459.523   -2349.614
        pira |   29533.06   1822.715    16.20   0.000     25960.17    33105.95
        hown |   1185.256    967.737     1.22   0.221    -711.7052    3082.218
       _cons |  -32907.13   4405.615    -7.47   0.000    -41543.03   -24271.23
------------------------------------------------------------------------------
```

用快捷DDML命令 (qddml)估计部分线性模型

- set seed 123
- qddml $Y $D (c.($X)##c.($X)), model(partial) cmd(cvlasso) cmdopt(lopt postresults) kfolds(4) noreg robust

- 其中，“model(partial)”表示部分线性模型，“c.($X)##c.($X)”表示生成所有X的线性项与交互项(含二次项)，“model(partial)”表示估计部分线性模型，“cmd(cvlasso)”表示用交叉验证的Lasso进行第一阶段的机器学习估计，“cmdopt(lopt)”表示“optimal lambda”(选择使MSPE最小的λ)，“cmdopt(postresults) ”表示返回估计结果，“kfolds(4)”表示进行4折交叉拟合，“noreg”表示不使用OLS作为学习器(默认使用OLS)，“robust”表示异方差稳健标准误

使用cvlasso进行DDML估计的结果

```
DDML estimation results:
spec  r     Y learner     D learner         b        SE
 opt  1     Y1_cvlasso    D1_cvlasso  9788.185(1343.972)
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]  = Y1_cvlasso_1                         Number of obs   =      9915
D-E[D|X,Z]= D1_cvlasso_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   9788.185   1343.972     7.28   0.000     7154.048    12422.32
       _cons |    84.7435   534.6942     0.16   0.874    -963.2378    1132.725
------------------------------------------------------------------------------
```

- 由于使用Stata命令cvlasso进行Lasso估计，故运行此命令很费时

使用pystacked调用Python的机器学习算法

- `set seed 123`
- `qddml $Y $D (c.($X)##c.($X)), model(partial) cmd(pystacked) cmdopt(method(lassocv)) kfolds(4) noreg robust`

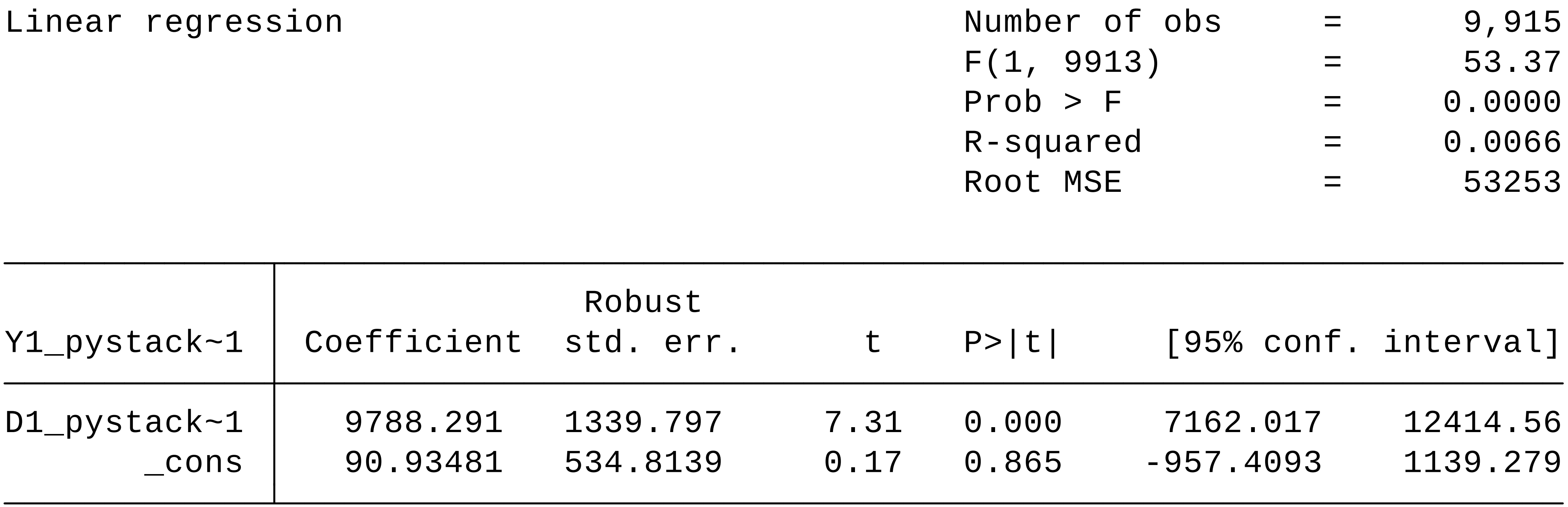
- 命令pystacked调用Python模块sklearn的机器学习算法进行估计，可大幅提高运算速度

使用pystacked进行DDML估计的结果

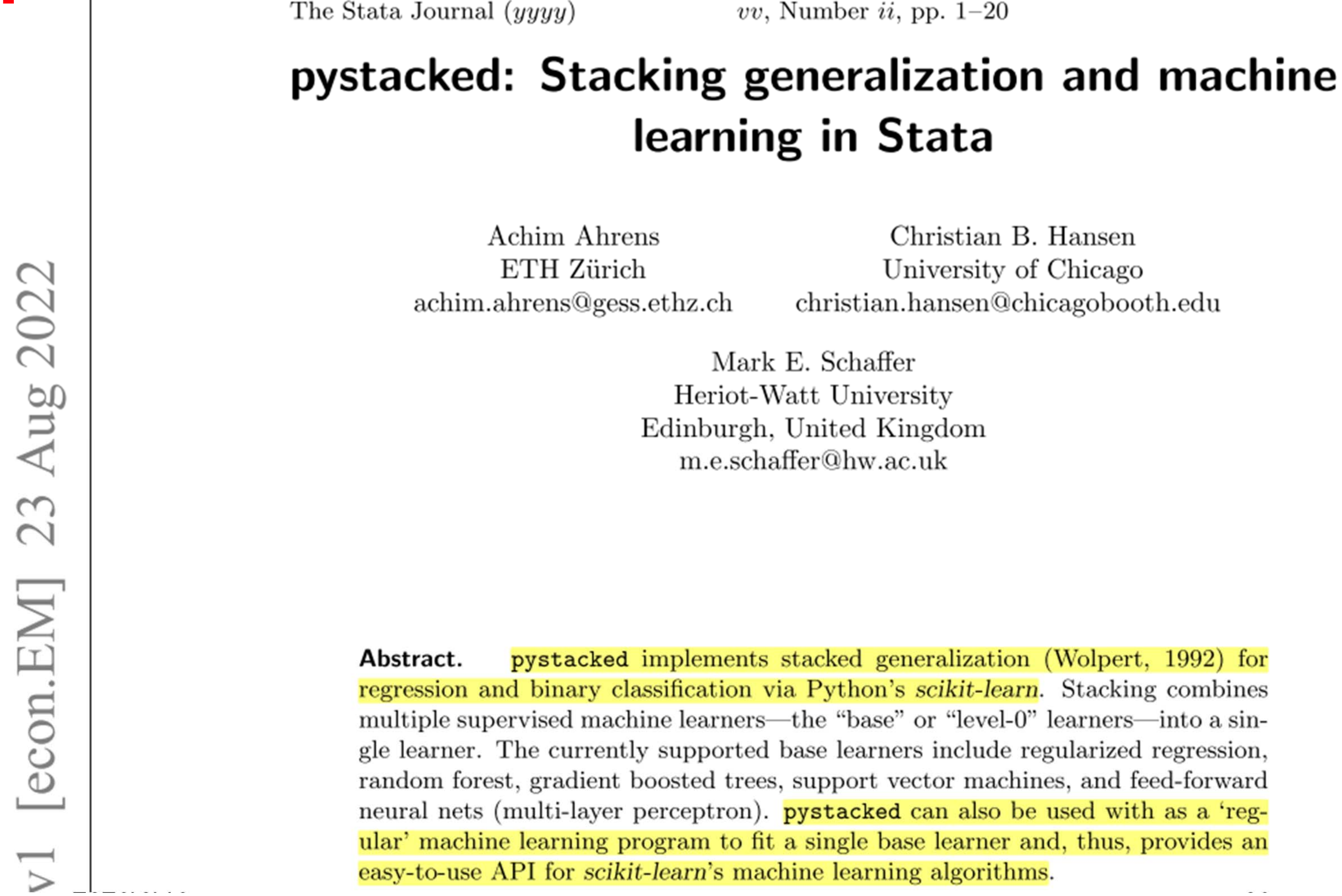
```
DDML estimation results:
spec  r     Y learner     D learner         b        SE
 opt  1  Y1_pystacked  D1_pystacked  9788.291(1339.797)
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]  = Y1_pystacked_1                        Number of obs   =      9915
D-E[D|X,Z]= D1_pystacked_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   9788.291   1339.797     7.31   0.000     7162.337    12414.24
       _cons |   90.93481   534.8139     0.17   0.865    -957.2813    1139.151
------------------------------------------------------------------------------
```

手工进行第二阶段回归

- reg Y1_pystacked_1 D1_pystacked_1,r



```
Linear regression                               Number of obs     =      9,915
                                                F(1, 9913)        =      53.37
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0066
                                                Root MSE          =      53253

------------------------------------------------------------------------------
             |               Robust
Y1_pystack~1 | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
D1_pystack~1 |   9788.291   1339.797     7.31   0.000     7162.017    12414.56
       _cons |   90.93481   534.8139     0.17   0.865    -957.4093    1139.279
------------------------------------------------------------------------------
```



v1 [econ.EM] 23 Aug 2022

The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–20

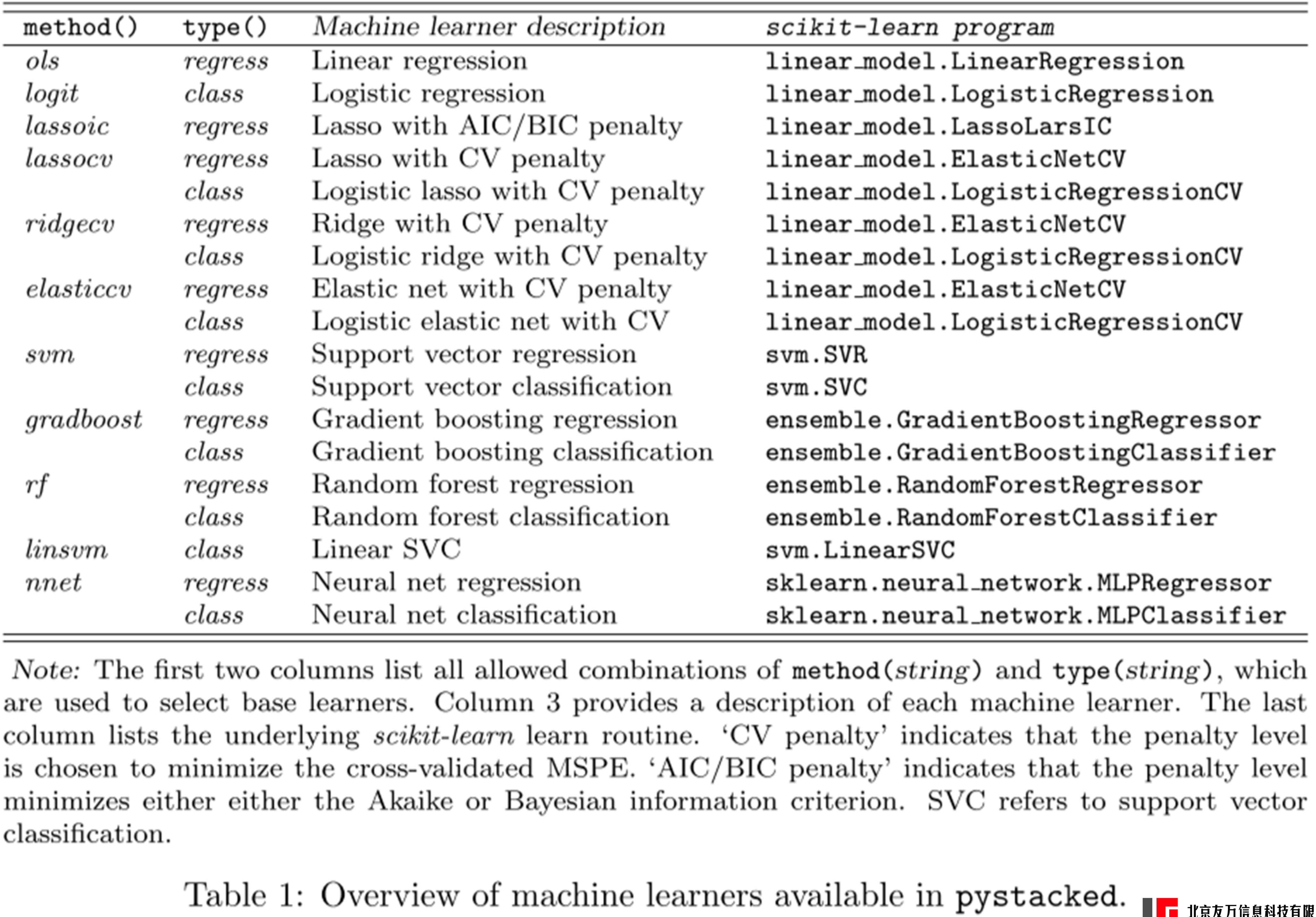
pystacked: Stacking generalization and machine learning in Stata

Achim Ahrens
ETH Zürich
achim.ahrens@gess.ethz.ch

Christian B. Hansen
University of Chicago
christian.hansen@chicagobooth.edu

Mark E. Schaffer
Heriot-Watt University
Edinburgh, United Kingdom
m.e.schaffer@hw.ac.uk

Abstract. `pystacked` implements stacked generalization (Wolpert, 1992) for regression and binary classification via Python's *scikit-learn*. Stacking combines multiple supervised machine learners—the "base" or "level-0" learners—into a single learner. The currently supported base learners include regularized regression, random forest, gradient boosted trees, support vector machines, and feed-forward neural nets (multi-layer perceptron). `pystacked` can also be used with as a 'regular' machine learning program to fit a single base learner and, thus, provides an easy-to-use API for *scikit-learn*'s machine learning algorithms.



`method()`	`type()`	*Machine learner description*	*scikit-learn program*
ols	*regress*	Linear regression	`linear_model.LinearRegression`
logit	*class*	Logistic regression	`linear_model.LogisticRegression`
lassoic	*regress*	Lasso with AIC/BIC penalty	`linear_model.LassoLarsIC`
lassocv	*regress*	Lasso with CV penalty	`linear_model.ElasticNetCV`
	class	Logistic lasso with CV penalty	`linear_model.LogisticRegressionCV`
ridgecv	*regress*	Ridge with CV penalty	`linear_model.ElasticNetCV`
	class	Logistic ridge with CV penalty	`linear_model.LogisticRegressionCV`
elasticcv	*regress*	Elastic net with CV penalty	`linear_model.ElasticNetCV`
	class	Logistic elastic net with CV	`linear_model.LogisticRegressionCV`
svm	*regress*	Support vector regression	`svm.SVR`
	class	Support vector classification	`svm.SVC`
gradboost	*regress*	Gradient boosting regression	`ensemble.GradientBoostingRegressor`
	class	Gradient boosting classification	`ensemble.GradientBoostingClassifier`
rf	*regress*	Random forest regression	`ensemble.RandomForestRegressor`
	class	Random forest classification	`ensemble.RandomForestClassifier`
linsvm	*class*	Linear SVC	`svm.LinearSVC`
nnet	*regress*	Neural net regression	`sklearn.neural_network.MLPRegressor`
	class	Neural net classification	`sklearn.neural_network.MLPClassifier`

Note: The first two columns list all allowed combinations of `method`(*string*) and `type`(*string*), which are used to select base learners. Column 3 provides a description of each machine learner. The last column lists the underlying *scikit-learn* learn routine. 'CV penalty' indicates that the penalty level is chosen to minimize the cross-validated MSPE. 'AIC/BIC penalty' indicates that the penalty level minimizes either either the Akaike or Bayesian information criterion. SVC refers to support vector classification.

Table 1: Overview of machine learners available in `pystacked`.

使用岭回归进行DDML估计

- set seed 123
- qddml $Y $D ($X), model(partial) cmd(pystacked) cmdopt(method(ridgecv)) kfolds(4) noreg robust

```
DDML estimation results:
spec  r     Y learner     D learner         b        SE
_opt  1  Y1_pystacked  D1_pystacked  9779.796(1340.818)
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]  = Y1_pystacked_1                        Number of obs    =      9915
D-E[D|X,Z]= D1_pystacked_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   9779.796   1340.818     7.29   0.000      7151.84    12407.75
       _cons |   77.45151   534.6547     0.14   0.885    -970.4524    1125.355
------------------------------------------------------------------------------
```

使用随机森林进行DDML估计

- set seed 123
- qddml $Y $D ($X), model(partial) cmd(pystacked) cmdopt(method(rf)) kfolds(4) noreg robust

```
DDML estimation results:
spec   r     Y learner     D learner          b         SE
 opt   1  Y1_pystacked  D1_pystacked  9375.423(1423.319)
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]   = Y1_pystacked_1                         Number of obs    =      9915
D-E[D|X,Z]= D1_pystacked_1

------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   9375.423   1423.319     6.59   0.000     6585.77     12165.08
       _cons |  -1129.542   591.6808    -1.91   0.056   -2289.215     30.13108
------------------------------------------------------------------------------
```

使用梯度提升法进行DDML估计

- set seed 123
- qddml $Y $D ($X), model(partial) cmd(pystacked) cmdopt(method(gradboost)) kfolds(4) noreg robust

```
DDML estimation results:
spec   r      Y learner     D learner            b        SE
 opt   1  Y1_pystacked  D1_pystacked 10519.065(1531.961)
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]  = Y1_pystacked_1                        Number of obs   =      9915
D-E[D|X,Z]= D1_pystacked_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   10519.07   1531.961     6.87   0.000     7516.477    13521.65
       _cons |   58.14207   569.7209     0.10   0.919     -1058.49    1174.774
------------------------------------------------------------------------------
```

估计交互模型(Interactive Model)：以rf为例

- `set seed 123`
- `qddml $Y $D ($X), model(interactive) cmd(pystacked) cmdopt(method(rf)) kfolds(4) noreg robust`

```
DDML estimation results (ATE):
spec   r     Y0 learner     Y1 learner      D learner          b         SE
 opt   1  Y1_pystacked  Y1_pystacked  D1_pystacked  8937.063(1690.193)
opt = minimum MSE specification for that resample.


E[y|X,D=0]   = Y1_pystacked0_1                         Number of obs   =      9915
E[y|X,D=1]   = Y1_pystacked1_1
E[D|X]       = D1_pystacked_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   8937.063   1690.193     5.29   0.000     5624.345    12249.78
------------------------------------------------------------------------------
Warning: 379 propensity scores trimmed to lower limit .01.
Warning: 4 propensity scores trimmed to upper limit .99.
Warning: . resamples had propensity scores trimmed to upper limit .99.
```

工具变量模型

- 以*p401*(是否参与401(k)养老金计划)作为内生的处理变量，以*e401*(是否有参与401(k)养老金计划的资格)作为工具变量

- 首先，使用传统的线性IV模型(2SLS)

- `ivregress 2sls $Y $X (p401=e401),r`

```
Instrumental variables 2SLS regression            Number of obs   =      9,915
                                                  Wald chi2(10)   =    1076.88
                                                  Prob > chi2     =     0.0000
                                                  R-squared       =     0.2348
                                                  Root MSE        =      55565

------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        p401 |   8502.323   2192.535     3.88   0.000     4205.034    12799.61
         age |   630.0183   55.21924    11.41   0.000     521.7906     738.246
         inc |   .9265484   .1107111     8.37   0.000     .7095586    1.143538
        educ |  -617.5012    337.431    -1.83   0.067    -1278.854    43.85145
       fsize |  -1014.935   384.9964    -2.64   0.008    -1769.514   -260.3556
        marr |   893.3085   1732.216     0.52   0.606    -2501.773     4288.39
     twoearn |  -19278.86   2525.998    -7.63   0.000    -24229.73      -14328
          db |  -4552.834   1311.777    -3.47   0.001    -7123.869   -1981.799
        pira |   29114.81   1801.951    16.16   0.000     25583.05    32646.57
        hown |   1087.157   956.7152     1.14   0.256    -787.9705    2962.284
       _cons |  -33151.53    4401.75    -7.53   0.000     -41778.8   -24524.26
------------------------------------------------------------------------------
Instrumented: p401
 Instruments: age inc educ fsize marr twoearn db pira hown e401
```

部分线性IV模型

- set seed 123
- qddml $Y ($X) (p401 = e401) , model(iv) cmd(pystacked) cmdopt(method(rf)) kfolds(4) noreg robust

```
DDML estimation results:
spec  r     Y learner     D learner           b         SE      Z learner
 opt   1  Y1_pystacked  D1_pystacked 14367.238(2092.390)
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]  = Y1_pystacked_1                               Number of obs   =      9915
D-E[D|X,Z]= D1_pystacked_1
Z-E[Z|X]  = Z1_pystacked_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        p401 |   14367.24    2092.39     6.87   0.000     10266.23    18468.25
       _cons |  -1009.768   591.2746    -1.71   0.088    -2168.645     149.109
------------------------------------------------------------------------------
```

灵活线性IV模型

- set seed 123
- qddml $Y ($X) (p401 = e401) , model(fiv) cmd(pystacked) cmdopt(method(rf)) kfolds(4) noreg robust

```
DDML estimation results:
spec   r      Y learner      D learner              b         SE     DH learner
 opt   1   Y1_pystacked  D1_pystacked 15650.240(2367.635) D1_pystacke~h
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]  = Y1_pystacked_1                          Number of obs     =      9915
E[D|X,Z]  = D1_pystacked_1
E[D|X]    = D1_pystacked_h_1
Orthogonalised D = D - E[D|X]; optimal IV = E[D|X,Z] - E[D|X].
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        p401 |   15650.24   2367.635     6.61   0.000     11009.76    20290.72
       _cons |  -1041.804   591.1342    -1.76   0.078    -2200.406    116.7976
------------------------------------------------------------------------------
```

交互IV模型

- set seed 123
- qddml $Y ($X) (p401 = e401) , model(interactiveiv) cmd(pystacked) cmdopt(method(rf)) kfolds(4) noreg robust

```
DDML estimation results (LATE):
spec  r     Y0 learner     Y1 learner     D0 learner     D1 learner          b        SE      Z learner
 opt   1  Y1_pystacked  Y1_pystacked  D1_pystacked  D1_pystacked 14013.373(2695.983)  Z1_pystacked
opt = minimum MSE specification for that resample.


E[y|X,D=0]   = Y1_pystacked0_1                          Number of obs   =      9915
E[y|X,D=1]   = Y1_pystacked1_1
E[D|X,Z=0]   = D1_pystacked0_1
E[D|X,Z=1]   = D1_pystacked1_1
E[Z|X]       = Z1_pystacked_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        p401 |   14013.37   2695.983     5.20   0.000     8729.344     19297.4
------------------------------------------------------------------------------
Warning: 398 propensity scores trimmed to lower limit .01.
Warning: 7 propensity scores trimmed to upper limit .99.
Warning: . resamples had propensity scores trimmed to upper limit .99.
```

完整的ddml命令

- qddml是快捷的双重机器学习命令，只能使用一种机器学习算法进行第一阶段估计

- 更完整的ddml命令可使用多种机器学习算法进行第一阶段估计，并进行堆叠回归(stacking regression)

- qddml命令调用ddml命令进行运算

使用ddml命令进行堆叠回归(1)

- Step 1

- `set seed 123`
- `ddml init partial, kfolds(4)`

- 初始化用于估计部分线性模型的DDML模型，其中“kfolds(4)”表示进行4折交叉拟合，默认为“kfolds(5)”

使用ddml命令进行堆叠回归 (2)

- Step 2. Add supervised machine learners for estimating conditional expectations

- Add learners for E[Y|X]

-
  ```
  ddml E[Y|X]: pystacked $Y $X, methods(ols
  lassocv ridgecv rf gradboost)
  xvars2(c.($X)##c.($X))
  xvars3(c.($X)##c.($X))
  ```

- Add learners for E[D|X]

-
  ```
  ddml E[D|X]: pystacked $D $X, methods(ols
  lassocv ridgecv rf gradboost)
  xvars2(c.($X)##c.($X))
  xvars3(c.($X)##c.($X))
  ```

察看DDML模型的设定

- ddml describe

```
Model:                      partial, crossfit folds k=4, resamples r=1
Dependent variable (Y): net_tfa
 net_tfa learners:          Y1_pystacked
D equations (1):            e401
 e401 learners:             D1_pystacked
Specifications:             1 possible specs
```

使用ddml命令进行堆叠回归 (3)

- Step 3: Perform cross-fitting
- `ddml crossfit`

```
Cross-fitting E[y|X] equation: net_tfa
Cross-fitting fold 1 2 3 4 ...completed cross-fitting
Cross-fitting E[D|X] equation: e401
Cross-fitting fold 1 2 3 4 ...completed cross-fitting
```

- `ddml extract, show(pystacked)`
- 其中，选择项“show(pystacked)”表示展示堆叠回归的权重及MSE

```
mean pystacked weights across folds/resamples for D1_pystacked (e401)
               learner  mean_weight
     ols             1    .07969215
lassocv              2    .32957375
ridgecv              3    .27780063
      rf             4    .01130101
gradboost            5    .30163246

mean pystacked MSEs across folds/resamples for D1_pystacked (e401)
             learner   mean_MSE
     ols           1  .20041698
lassocv            2  .19618448
ridgecv            3  .19647461
      rf           4  .21842719
gradboost          5  .19750322

mean pystacked weights across folds/resamples for Y1_pystacked (net_tfa)
               learner  mean_weight
     ols             1    .09715392
lassocv              2    .62458396
ridgecv              3       .15464
      rf             4    .10027424
gradboost            5    .01927898

mean pystacked MSEs across folds/resamples for Y1_pystacked (net_tfa)
             learner   mean_MSE
     ols           1  3.342e+09
lassocv            2  3.081e+09
ridgecv            3  3.095e+09
      rf           4  3.555e+09
gradboost          5  3.462e+09
```

察看交叉拟合的细节

- ddml describe, crossfit

```
Model:                  partial, crossfit folds k=4, resamples r=1
Dependent variable (Y): net_tfa
 net_tfa learners:      Y1_pystacked
D equations (1):        e401
 e401 learners:         D1_pystacked
Specifications:         1 possible specs

Crossfit results (detail):
                                     All    By fold:
Cond. exp.  Learner      rep         MSE        1         2         3         4
net_tfa    Y1_pysta~d     1       2.9e+09   2.5e+09   3.1e+09   4.0e+09   1.9e+09
e401       D1_pysta~d     1          0.20      0.20      0.19      0.20      0.19
```

- ddml extract, show(pystacked) detail
- 选择项“detail”表示展示每一折的权重(结果略)

使用ddml命令进行堆叠回归 (4)

- Step 4: Estimate causal effects
- `ddml estimate, robust`

```
DDML estimation results:
spec  r     Y learner     D learner         b        SE
 opt  1  Y1_pystacked  D1_pystacked  9632.731(1336.508)
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]  = Y1_pystacked_1                          Number of obs   =       9915
D-E[D|X,Z]= D1_pystacked_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   9632.731   1336.508     7.21   0.000     7013.222    12252.24
       _cons |   102.5663   535.9189     0.19   0.848    -947.8154    1152.948
------------------------------------------------------------------------------
```

考察机器学习算法的默认设置

- pystacked, type(reg) methods(rf) printopt

```
Machine learner: rf

 Stata syntax:
 n_estimators(integer 100)  criterion(string)  max_depth(integer -1)
> min_samples_split(real 2)  min_samples_leaf(real 1)
 min_weight_fraction_leaf(real 0)  max_features(string)  max_leaf_node
> s(integer -1)  min_impurity_decrease(real 0)  bootstrap(string)
 oob_score  n_jobs(integer 0)  random_state(integer -1)  warm_start  c
> cp_alpha(real 0)  max_samples(real -1)

 Specified options are translated to:
 n_estimators(100)  criterion(squared_error)  max_depth(None)  min_sam
> ples_split(2)  min_samples_leaf(1)  min_weight_fraction_leaf(0)
 max_features(auto)  max_leaf_nodes(None)  min_impurity_decrease(0)  b
> ootstrap(True)  oob_score(False)  n_jobs(None)
 random_state(rng)  warm_start(False)  ccp_alpha(0)  max_samples(None)
>
```

考察机器学习算法的默认设置 (续)

- pystacked, type(reg) methods(gradboost) printopt

```
Machine learner: gradboost

 Stata syntax:
 loss(string)  criterion(string)  learning_rate(real 0.1)  n_estimators(
> integer 100)  subsample(real 1)
 min_samples_split(real 2)  min_samples_leaf(real 1)  min_weight_fractio
> n_leaf(real 0)  max_depth(integer 3)
 min_impurity_decrease(real 0)  init(string)  random_state(integer -1)
> max_features(string)  alpha(real 0.9)
 max_leaf_nodes(integer -1)  warm_start  validation_fraction(real 0.1)
> n_iter_no_change(integer -1)
 tol(real 1e-4)  ccp_alpha(real 0)

 Specified options are translated to:
 loss(squared_error)  learning_rate(.1)  n_estimators(100)  subsample(1)
>   criterion(friedman_mse)  min_samples_split(2)
 min_samples_leaf(1)  min_weight_fraction_leaf(0)  max_depth(3)  min_imp
> urity_decrease(0)  init(None)  random_state(rng)
 max_features(None)  alpha(.9)  max_leaf_nodes(None)  warm_start(False)
>  validation_fraction(.1)  n_iter_no_change(None)
 tol(.0001)  ccp_alpha(0)
```

命令DDML的另一句型

```
• set seed 123
• ddml init partial, kfolds(4)
• ddml E[Y|X]: pystacked $Y $X || method(ols) ||
  method(lassocv) xvars(c.($X)##c.($X)) ||
  method(ridgecv) xvars(c.($X)##c.($X)) || method(rf)
  opt(n_estimators(500)) || method(gradboost)
  opt(n_estimators(500) learning_rate(0.01))
• ddml E[D|X]: pystacked $D $X || method(ols) ||
  method(lassocv) xvars(c.($X)##c.($X)) ||
  method(ridgecv) xvars(c.($X)##c.($X)) || method(rf)
  opt(n_estimators(500)) || method(gradboost)
  opt(n_estimators(500) learning_rate(0.01))
• ddml crossfit
• ddml extract, show(pystacked)
• ddml estimate, robust
```

```
mean pystacked weights across folds/resamples for D1_pystacked (e401)
               learner  mean_weight
      ols            1    .02265448
  lassocv            2    .16076639
  ridgecv            3    .28821582
       rf            4    .00047302
gradboost            5    .52789029

mean pystacked MSEs across folds/resamples for D1_pystacked (e401)
             learner    mean_MSE
      ols          1  .20041698
  lassocv          2  .19618448
  ridgecv          3  .19647461
       rf          4  .21683488
gradboost          5   .1959885

mean pystacked weights across folds/resamples for Y1_pystacked (net_tfa)
               learner  mean_weight
      ols            1    .09049935
  lassocv            2    .62107226
  ridgecv            3    .15976333
       rf            4    .11735187
gradboost            5            0

mean pystacked MSEs across folds/resamples for Y1_pystacked (net_tfa)
             learner    mean_MSE
      ols          1  3.342e+09
  lassocv          2  3.081e+09
  ridgecv          3  3.095e+09
       rf          4  3.545e+09
gradboost          5  3.305e+09
```

```
DDML estimation results:
spec  r     Y learner     D learner         b        SE
 opt  1  Y1_pystacked  D1_pystacked  9525.302(1319.637)
opt = minimum MSE specification for that resample.

Min MSE DDML model
y-E[y|X]  = Y1_pystacked_1                          Number of obs   =      9915
D-E[D|X,Z]= D1_pystacked_1
------------------------------------------------------------------------------
             |               Robust
     net_tfa | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        e401 |   9525.302   1319.637     7.22   0.000     6938.861    12111.74
       _cons |   128.4252   535.9389     0.24   0.811    -921.9958    1178.846
------------------------------------------------------------------------------
```

2023年Stata中国用户大会

谢谢！

陈强

山东大学经济学院
qiang2chen2@126.com
www.econometrics-stata.com
视频号：山东大学陈强教授
公众号：计量经济学及Stata应用