A combination of Transfer learning and Double machine learning

19331053 纪传宇

2022 年 8 月 21 日

1 Pre-knowledge

Interactive Regression Model (IRM) is a model to estimate causal effects

1.1 Interactive Regression Model (IRM)

We consider estimation of average treatment effects when treatment effects are fully heterogeneous and the treatment variable is binary, $D \in \{0, 1\}$. We consider vectors (Y, D, X) such that

$$\begin{aligned} Y &= g_0(D, X) + U, \quad \mathbb{E}(U \mid X, D) = 0 \\ D &= m_0(X) + V, \quad \mathbb{E}(V \mid X) = 0 \end{aligned} \tag{1}$$

Since D is not additively separable, this model is more general than the partially linear model for the case of binary D . A common target parameter

of interest in this model is the average treatment effect (ATE),

$$\theta_0 = \mathbb{E}\left[g_0(1, X) - g_0(0, X)\right]. \tag{2}$$

Another common target parameter is the average treatment effect for the treated (ATTE),

$$\theta_0 = \mathbb{E}\left[g_0(1, X) - g_0(0, X) \mid D = 1\right]. \tag{3}$$

1.2 Key Ingredients of the Double Machine Learning Inference Approach

The confounding factors X affect the treatment variable D via the propensity score, $m_0(Z) := E[D \mid X]$, and the outcome variable via the function $g_0(D, Z)$. Both of these functions are unknown and potentially complicated, and we consider estimating these functions via the use of ML methods.

We proceed to set up moment conditions with scores that obey a type of orthogonality with respect to nuisance functions. Specifically, we make use of scores $\psi(W, \theta, \eta)$ that satisfy the identification condition

$$E\psi\left(W, \theta_0, \eta_0\right) = 0 \tag{4}$$

The first key input of the inference procedure is using a score function $\psi(W; \theta; \eta)$ that satisfies (4) , with $\overline{\theta_0}$ being the unique solution, and that obeys the Neyman orthogonality condition

$$\partial_\eta E\psi\left(W, \theta_0, \eta\right)|_{\eta=\eta_0} = 0, \tag{5}$$

where W=(Y, D, X), θ_0 is the parameter of interest, η denotes nuisance functions with population value η_0 and $\partial_\eta f|_{\eta=\eta_0}$ denotes the derivative of f with respect to η (the Gateaux derivative operator).

Neyman-orthogonal scores are readily available for both the ATE and ATTE one can use the doubly robust/efficient scores of Robins and Rotnitzky (1995) and Hahn (1998) which are automatically Neyman orthogonal.

For estimating the ATE, we employ

$$\psi(W,\theta,\eta) := g(1,X)-g(0,X)+\frac{D(Y-g(1,X))}{m(X)}-\frac{(1-D)(Y-g(0,X))}{1-m(X)}-\theta \tag{6}$$

with

$$\begin{aligned}\eta(X) &:= (g(0,X), g(1,X), m(X))' \\ \eta_0(X) &:= (g_0(0,X), g_0(1,X), m_0(X))'\end{aligned} \tag{7}$$

where $\eta(X)$ is the nuisance parameter with true value denoted by $\eta_0(X)$ consisting of P -square integrable functions.

The second key input is to use a form of sample splitting at the stage of producing the estimator of the main parameter θ_0, which allows us to avoid biases arising from overfitting.

1.3 Double Machine Learning for Estimation of a Causal Parameter

We describe the estimator of θ_0 using random sample $(W_i)_{i=1}^N$. The algorithm makes use of a form of sample splitting, which we call cross-fitting. It builds on the ideas in, e.g., Angrist and Krueger (1995). The use of sample-splitting is a crucial ingredient to the approach that helps avoid overfitting which can easily result from the application of complex, flexible methods such as boosted linear and tree models, random forests, and various ensemble and hybrid ML methods. A. Algorithm: Estimation by K -fold Cross-Fitting

Step 1: Let K be a fixed integer. Form a K -fold random partition of $\{1,\ldots,N\}$ by dividing it into equal parts $(I_k)_{k=1}^K$ each of size n:=N / K , assuming that N is a multiple of K . For each set $I_k, let I_k^c$ denote all observation indices that are not in I_k.

Step 2: Construct K estimators

$$\check{\theta}_0\left(I_k, I_k^c\right), \quad k=1,\ldots,K, \tag{8}$$

that employ the machine learning estimators

$$\begin{aligned}\hat{\eta}_0\left(I_k^c\right) = \big(&\hat{g}_0\left(0, Z; I_k^c\right), \hat{g}_0\left(1, Z; I_k^c\right) \\ &\hat{m}_0\left(Z; I_k^c\right), \tfrac{1}{N-n}\textstyle\sum_{i \in I_k^c} D_i\Big)',\end{aligned} \tag{9}$$

of the nuisance parameters

$$\eta_0(X) = \left(g_0(0, X), g_0(1, X), m_0(X), E[D]\right)', \tag{10}$$

and where each estimator $\check{\theta}_0\left(I_k, I_k^c\right)$ is defined as the root θ of

$$\frac{1}{n}\sum_{i \in I_k} \psi\left(W, \theta, \hat{\eta}_0\left(I_k^c\right)\right) = 0, \tag{11}$$

for the score ψ defined in (6) for the ATE.

Step 3: Average the K estimators to obtain the final estimator:

$$\tilde{\theta}_0 = \frac{1}{K}\sum_{k=1}^{K} \check{\theta}_0\left(I_k, I_k^c\right) \tag{12}$$

An approximate standard error for this estimator is $\hat{\sigma}/\sqrt{N}$, where

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} \hat{\psi}_i^2, \hat{\psi}_i := \psi\left(W_i, \tilde{\theta}_0, \hat{\eta}_0\left(I_{k(i)}^c\right)\right), \tag{13}$$

and

$$k(i) := \{k \in \{1, \ldots, K\} : i \in I_k\}. \tag{14}$$

An approximate (1-α) $\times$ 100 percent confidence interval is

$$\mathrm{CI}_n := \left[\tilde{\theta}_0 \pm \Phi^{-1}(1 - \alpha/2)\hat{\sigma}/\sqrt{N}\right]. \tag{15}$$

1.4 Transfer learning method and algorithm

The main strategy is to first transfer the information from those sources by pooling all the data to obtain a rough estimator, then correct the bias in the second step using the target data. More specifically, we fit a GLM with ℓ_1 -penalty by pooled samples first, then fit the contrast in the second step

using only the target by another ℓ_1 -regularization. The detailed algorithm ($\mathcal{A}$ -Trans-GLM) is presented in Algorithm 1. The transferring step could be understood as to solve the following equation w.r.t. $\boldsymbol{w} \in \mathbb{R}^p$:

$$\sum_{k\in\{0\}\cup\mathcal{A}} \left[\left(\boldsymbol{X}^{(k)}\right)^T \boldsymbol{y}^{(k)} - \sum_{i=1}^{n_k} \psi'\left((\boldsymbol{w})^T \boldsymbol{x}_i^{(k)} \right) \boldsymbol{x}_i^{(k)} \right] = \mathbf{0}_p \tag{16}$$

which converges to the solution of its population version under certain conditions

$$\sum_{k\in\{0\}\cup\mathcal{A}} \alpha_k \mathbb{E}\left\{ \left[\psi'\left(\left(\boldsymbol{w}^{\mathcal{A}}\right)^T \boldsymbol{x}^{(k)} \right) - \psi'\left(\left(\boldsymbol{w}^{(k)}\right)^T \boldsymbol{x}^{(k)} \right) \right] \boldsymbol{x}^{(k)} \right\} = \mathbf{0}_p \tag{17}$$

where $\alpha_k = \frac{n_k}{n_{\mathcal{A}}+n_0}$. Notice that in the linear case, $\boldsymbol{w}^{\mathcal{A}}$ can be explicitly expressed as a linear transformation of the true parameter $\boldsymbol{w}^{(k)}$, i.e. $\boldsymbol{w}^{\mathcal{A}} = \boldsymbol{\Sigma}^{-1} \sum_{k\in\{0\}\cup\mathcal{A}} \alpha_k \boldsymbol{\Sigma}^{(k)} \boldsymbol{w}^{(k)}$, where $\boldsymbol{\Sigma}^{(k)} = \mathbb{E}\left[\boldsymbol{x}^{(k)} \left(\boldsymbol{x}^{(k)}\right)^T \right]$ and $\boldsymbol{\Sigma} = \sum_{k\in\{0\}\cup\mathcal{A}} \alpha_k \mathbb{E}\left[\boldsymbol{x}^{(k)} \left(\boldsymbol{x}^{(k)}\right)^T \right]$ (Li et al., 2021).

We can use Algorithm 1 if we are certain about which sources to transfer.

Algorithm 1: $\mathcal{A}$-Trans-GLM

Input: target data $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)})$, source data $\{(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})\}_{k=1}^K$, penalty parameters $\lambda_{\boldsymbol{w}}$ and $\lambda_{\boldsymbol{\delta}}$, transferring set $\mathcal{A}$

Output: the estimated coefficient vector $\hat{\boldsymbol{\beta}}$

1 **Transferring step**: Compute

$$\hat{\boldsymbol{w}}^{\mathcal{A}} \leftarrow \arg\min_{\boldsymbol{w}} \left\{ \frac{1}{n_{\mathcal{A}}+n_0} \sum_{k\in\{0\}\cup\mathcal{A}} \left[-(\boldsymbol{y}^{(k)})^T \boldsymbol{X}^{(k)} \boldsymbol{w} + \sum_{i=1}^{n_k} \psi(\boldsymbol{w}^T \boldsymbol{x}_i^{(k)}) \right] + \lambda_{\boldsymbol{w}} \|\boldsymbol{w}\|_1 \right\}$$

2 **Debiasing step**: Compute

$$\hat{\boldsymbol{\delta}}^{\mathcal{A}} \leftarrow \arg\min_{\boldsymbol{\delta}} \left\{ -\frac{1}{n_0} (\boldsymbol{y}^{(0)})^T \boldsymbol{X}^{(0)} (\hat{\boldsymbol{w}}^{\mathcal{A}} + \boldsymbol{\delta}) + \frac{1}{n_0} \sum_{i=1}^{n_0} \psi((\hat{\boldsymbol{w}}^{\mathcal{A}} + \boldsymbol{\delta})^T \boldsymbol{x}_i^{(0)}) + \lambda_{\boldsymbol{\delta}} \|\boldsymbol{\delta}\|_1 \right\}$$

3 Let $\hat{\boldsymbol{\beta}} \leftarrow \hat{\boldsymbol{w}}^{\mathcal{A}} + \hat{\boldsymbol{\delta}}^{\mathcal{A}}$

4 Output $\hat{\boldsymbol{\beta}}$

Fig. 1. Transfer Learning $\mathcal{A}$-Trans-GLM

We propose a simple, algorithm-free, and data-driven method to determine an informative transferring set $\mathcal{A}$. We call this approach a transferable

source detection algorithm and refer to it as Trans-GLM

Algorithm 2: Trans-GLM

Input: target data $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)})$, all source data $\{(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})\}_{k=1}^{K}$, a constant $C_0 > 0$, penalty parameters $\{\{\lambda^{(k)[r]}\}_{k=0}^{K}\}_{r=1}^{3}$

Output: the estimated coefficient vector $\hat{\boldsymbol{\beta}}$, and the determined transferring set $\widehat{\mathcal{A}}$

1 **Transferable source detection**: Randomly divide $(\boldsymbol{X}^{(0)}, \boldsymbol{y}^{(0)})$ into three sets of equal size as $\{(\boldsymbol{X}^{(0)[i]}, \boldsymbol{y}^{(0)[i]})\}_{i=1}^{3}$

2 **for** $r = 1$ **to** 3 **do**

3 $\quad \hat{\boldsymbol{\beta}}^{(0)[r]} \leftarrow$ fit the Lasso on $\{(\boldsymbol{X}^{(0)[i]}, \boldsymbol{y}^{(0)[i]})\}_{i=1}^{3} \backslash (\boldsymbol{X}^{(0)[r]}, \boldsymbol{y}^{(0)[r]})$ with penalty parameter $\lambda^{(0)[r]}$

4 $\quad \hat{\boldsymbol{\beta}}^{(k)[r]} \leftarrow$ run step 1 in Algorithm 1 with $(\{(\boldsymbol{X}^{(0)[i]}, \boldsymbol{y}^{(0)[i]})\}_{i=1}^{3} \backslash (\boldsymbol{X}^{(0)[r]}, \boldsymbol{y}^{(0)[r]})) \cup (\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})$ and penalty parameter $\lambda^{(k)[r]}$ for all $k \neq 0$

5 $\quad$ Calculate the loss function $\hat{L}_0^{[r]}(\hat{\boldsymbol{\beta}}^{(k)[r]})$ on $(\boldsymbol{X}^{(0)[r]}, \boldsymbol{y}^{(0)[r]})$ for $k = 1, \ldots, K$

6 **end**

7 $\hat{L}_0^{(k)} \leftarrow \sum_{r=1}^{3} \hat{L}_0^{[r]}(\hat{\boldsymbol{\beta}}^{(k)[r]})/3$, $\hat{L}_0^{(0)} \leftarrow \sum_{r=1}^{3} \hat{L}_0^{[r]}(\hat{\boldsymbol{\beta}}^{(0)[r]})/3$, $\hat{\sigma} = \sqrt{\sum_{r=1}^{3}(\hat{L}_0^{[r]}(\hat{\boldsymbol{\beta}}^{(0)[r]}) - \hat{L}_0^{(0)})^2/2}$

8 $\widehat{\mathcal{A}} \leftarrow \{k \neq 0 : \hat{L}_0^{(k)} - \hat{L}_0^{(0)} \leq C_0(\hat{\sigma} \vee 0.01)\}$

9 $\widehat{\mathcal{A}}$**-Trans-GLM**: $\hat{\boldsymbol{\beta}} \leftarrow$ run Algorithm 1 using $\{(\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})\}_{k \in \{0\} \cup \widehat{\mathcal{A}}}$

10 Output $\hat{\boldsymbol{\beta}}$ and $\widehat{\mathcal{A}}$

Fig. 2. Transfer Learning Trans-GLM

2 Combination of Transfer learning and Double Machine Learning

In IRM model, We consider the case where treatment effects are fully heterogeneous and the treatment variable, D , is binary, $D \in \{0, 1\}$. We let Y denote the outcome variable of interest and X denote a set of control variables. We then model random vector (Y, D, X) as

$$\begin{aligned} (1) Y &= g_0(D, X) + \zeta, \quad E[\zeta \mid X, D] = 0, \\ (2) D &= m_0(X) + \nu, \quad E[\nu \mid X] = 0. \end{aligned} \tag{18}$$

The second equation keeps track of confounding, namely the dependence of D on covariates/controls. The characteristics X affect variable D via the function $m_0(X)$ and the outcome variable via the function $g_0(X)$.

Here are three hypothesis

(1)X is a high-dimensional vector that $X = (X_1, \ldots, X_p)$ consists of other confounding covariates

(2) m_0 is a logistic model with respect to a dictionary of basis functions with respect to X and the end value is binary.

(3)g_0 is linear with respect to a dictionary of basis functions with respect to X and D

2.1 Method 1

We can use the second key of Double Machine which is cross fitting to estimate the parameter of m_0.

In "Online Appendix for "Double/Debiased/Neyman Machine Learning of Treatment Effects"" Then report results based on five simple methods for estimating the nuisance functions used in forming the orthogonal estimating equations. They consider three tree-based methods, labeled "Random Forest", "Reg. Tree", and "Boosting", one ℓ_1 -penalization based method, labeled "Lasso", and a neural network method, labeled "Neural Net".

Because we assume that m_0 is logistic with respect to a dictionary of basis functions with respect to X. We choose the "Lasso" method to estimate the parameter of m_0.

Through using "rlassologit" in "Lasso" method combining corss-fitting, we get the estimation of parameters .

We then use the parameters and original X to fix D and get $\hat{D}$

Because g_0 is linear with respect to a dictionary of basis functions with respect to X and D. So g_0 can be represented as $g_0(X, D) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \theta d$

We use Transfer Learning under Generalized Linear Models and R pack-

age glmtrans to estimate the parameters $\hat{\beta}_i$ i=1,2,...,n and $\hat{\theta}$

2.2 Method 2

D is binary, so we can consider D is a reponse variable then in order to estimate the parameters in m_0 we use Transfer Learning under Generalized Linear Models in Logistic case which the reponse variable in this model is 0-1 variable.

Then we use the parameters and original X to fix D to get $\hat{D}$

Because g_0 is linear with respect to a dictionary of basis functions with respect to X and D. So g_0 can be represented as $g_0(X, D) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n + \theta d$

We use Transfer Learning under Generalized Linear Models and R package glmtrans to estimate the parameters $\hat{\beta}_i$ i=1,2,...,n and $\hat{\theta}$.

3 Simulations

MSEtheta denote through estimating m_0, the MSE value of $\hat{\beta}$ and real β

MSEerror1 denote through estimating m_0, the different quantity between $\hat{D}$ and D

MSEtotal denote through estimating m_0 and Transfer learning to estimate g_0, the MSE value of $\hat{\alpha}$ and setting α plus $\hat{\theta}$ and setting θ

MSEerror2 denote through estimating m_0 and Transfer learning to estimate g_0, the MSE value of $\hat{\theta}$ and setting θ

3.1 Dataset 5

The data generating process is defined as

$$\begin{aligned} d_i &= 1\left\{\frac{\exp\left(x_i'\beta\right)}{1+\exp\left(x_i'\beta\right)} > v_i\right\} \\ y_i &= \theta d_i + x_i'\alpha + \zeta_i \end{aligned} \tag{19}$$

with $v_i \sim \mathcal{U}(0.5, 0.2), \zeta_i \sim \mathcal{N}(0,1)$ and covariates $\boldsymbol{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(\mathbf{0}_p, \mathbf{1}_p)$, $\beta = (3 \cdot \mathbf{1}_s, \mathbf{0}_{p-s})^T$ where s is set to be 5 , $\theta = 2$ and $\alpha = (2 \cdot \mathbf{1}_t, \mathbf{0}_{p-t})^T$ where t is set to be 5, p is set to be 30.

We set the target sample size and source sample sample siz for each $k = 1$ 5 are variable. The dimension p=31 for both target and source data. The sample size is set to be 100,150,...450,500.

For each sample size, we run 5 simulations and average the results.

3.1.1 Method 1

The four indicators are as follow:

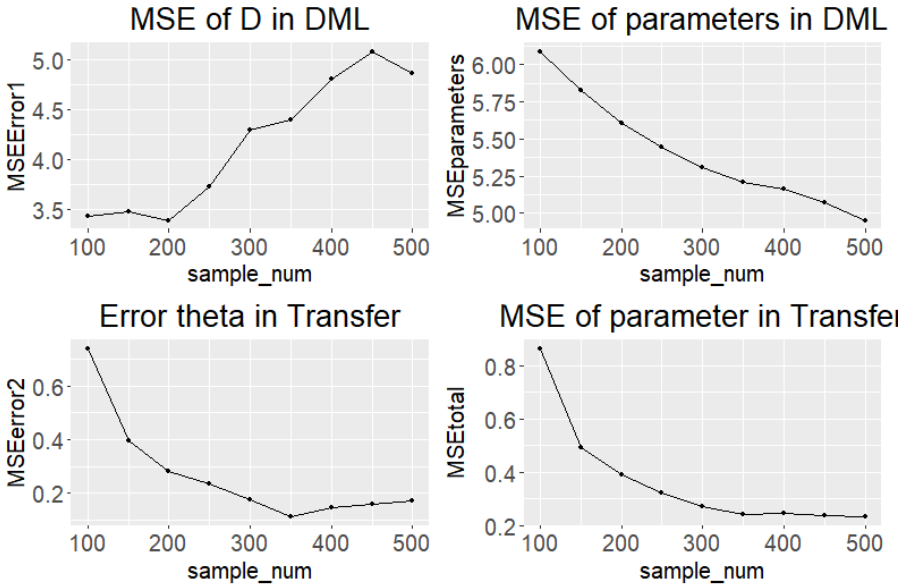

Fig. 3. MSEtheta MSEerror1 MSEtotal and MSEerror2 in dataset 5

From Fig.3, it can be seen that when fold is set to be 5,6,7, the MSE of D and MSE of parameters in DML are smaller than other fold.But after Transfer learning,the eroor in θ are very large, by contrast, the other parameters are smaller.