EconMLについて

参考資料集

- EconML
- EconMLのサンプルコード
- EconMLパッケージの紹介 (meta-learners編)
- 機械学習で因果推論~Meta-LearnerとEconML~
- CATEを推定するMeta-Learnersの特徴と比較
- Uplift Modelingで介入効果を最適化する
    - A/Bテスト (RCT)によって収集された学習データがあることを前提とする効果検証手法
- めっさ分かりやすい因果推論
- 異質な因果効果とその推定方法
- BART: Bayesian additive regression treesによる因果推論

EconMLとは

- EconMLは、Microsoft Researchが開発した因果推論のためのPythonパッケージ
- Heterogeneous Treatment Effect Estimation(異質処置効果推定)を行うことができる

Meta-Learners

- EconMLにおける異質処置効果推定の手法

T-Learner

- 処置群の結果を予測するモデルと非処置群の結果を予測するモデルの2つのモデルを用意する
  $$
  \hat{\tau}(x) = \hat{\mu}_{1}(x) - \hat{\mu}_{0}(x)
  $$
  ここで、$\hat{\mu}_{0} = E[Y^{(0)} | X = x], \hat{\mu}_{1} = E[Y^{(1)} | X= x]$となっている。つまり、対照群と処置群のそれぞれについて、関心のある共変量Xの下での反応の推定を行い、その結果を比較する。

- T-learnerでは処理群と対照群の観測データをプールして利用していないため、処理群と対照群のそれぞれのデータ生成過程の違いが推定性能に影響を与える。

- 処理群と対照群のそれぞれのデータ生成過程が等しい場合は、不利になる傾向になる。

- 他方で、処置効果の構造が非常に複雑で、処理群と対照群のそれぞれのデータ生成過程に共通の傾向がない場合には、特に優れた性能を発揮する傾向にある。

S-Learner

- 処置群の結果を予測するモデルと非処置群の結果を予測するモデルの2つのモデルを用意する

  $$\hat{\tau}(x) = \hat{\mu}(x, 1) - \hat{\mu}(x, 0)$$

  ここで、$\mu(x, w) := E[Y^{obs} | X = x, W = w]$ となっている。つまり、推定したモデルの対象に対して、wに0/1を代入した差分を考える。
- S-learnerでは、処置変数を他の共変量の同様に扱い、処置変数には特別な役割はない。
- そのため、lassoやRandomForestのようなアルゴリズムは、治療の割り当てを完全に無視して、治療の割り当てを選択しないこともできる。
- シミュレーション結果から、データ生成過程が等しい場合とCATEが多くの場所で0である場合において、最も良い推定を行うことが確認できる。

X-Learner

- まず、T-learner同様に$\hat{\mu}*{0} = E[Y^{(0)} | X = x],\hat{\mu}*{1} = E[Y^{(1)} | X= x]を考える。次に、\hat{\mu}*{0}
  を用いて、処置群の個人の処置を行わない場合の結果の推定を行う。この推定値と、観測された対照における結果の差を、個人の介入効果とする（\tilde D^{0}*{i}$）。同様に処置群の効果も推定する。

$$
\tilde D^{1}*{i} := Y^{1}*{i} - \hat{\mu}*{0}(X^{1}*{i}) \\
\tilde D^{0}*{i} := Y^{1}*{i} - \hat{\mu}*{0}(X^{1}*{i})
$$

- そして介入群のみからなるデータと、対照群からなるデータそれぞれを用いて、介入効果を推定するモデルを作成する（二段階目のベース学習器）。

$$
\hat{\tau}*{0} = E[\tilde D^{0}*{i} | X = x]\\
\hat{\tau}*{1} = E[\tilde D^{1}*{i} | X= x]
$$

- 最後に、得られたベース学習器について、傾向スコア($g(x)$)を用いた重み付き平均を求めることで、介入効果を推定する。
  $$
  \hat{\tau}*{X}(x) = g(x)\hat{\tau}*{0}(x) + (1-g(x))\hat{\tau}_{1}(x)
  $$

- X-learnerは，CATEに構造的な仮定がある場合や，一方の処置群が他方の処置群よりもはるかに大きい場合に特に優れた性能を発揮する。

- 期待されるCATEがほとんど0であるという強い信念がない限り、小さなデータサイズの場合はBARTを用いたX-learnerを、大きなデータサイズの場合にはRandomForestを用いるべきであるとしている。

R-Learner(githubのページ)

- Nie and Wager (2021)によって提案された方法
- HTEを推定するために、ロビンソン分解（Robinson (1988)）を用いる
  - Kaddour et al.(2021)もまた参考文献としてありそう。

$$Y_i - m(X_i) = (T_i - g(X_i))\tau(X_i) + \varepsilon_i$$

ここで$m(X_i) = E[Y_i|X_i]$はアウトカムの条件付き期待値で、$g(X_i)$は傾向スコア。
これらの$m(X_i)$及び$g(X_i)$を用いて、R-learnerは以下の式を最小化するときのX_iを用い、介入効果($\tau(X_i)$)を推定する。なお、$\Lambda_n(\tau(\cdot))$は正則化項。

$$\sum_{i=1}^{n}[(Y_i - m(X_i)) - (T_i - \pi(X_i))\tau(X_i)]^2 + \Lambda_n(\tau(\cdot))$$

- 実務上は$m(X_i)$及び$g(X_i)$は道なので、観測データから推計する。手法を提案した研究では、$m(X_i)$及び$g(X_i)$の推定と$\tau(X_i)$の推定は、それぞれ元のデータセットを分割した、別々のデータセットを用いて行うことを提案している(cross-fitting)。

その他の手法

- DA-Learner (Domain Adaptation Learner)
  - DA-Learnerは, X-Learnerにおけるμ^0,μ^1の学習に共変量シフトを用いた手法
- DR-Learner (Doubly Robust Learner)
  - Doubly Robustを用いてCATEを代替するようなsurrogate outcomeを作り, それをXに回帰する方法
- Targeted Maximum Likelihood Estimation:TMLEについて
  - なんじゃこりゃ。super learnerとかいうのもあるらしい。

Meta-Leanerの利点

- 特定のベース学習器に依存しない
  - 線形モデルではなくても良いことが利点。
  - 他方で、ベース学習器の選択をどのように行うかが重要となる。

- 以下のような特徴がある、らしい。。。
  - データの生成構造が大域的に線形である状況やデータセットが小さい場合には、BARTのように大域的に作用する推定器が大きな優位性を持つ。
    - $Y = \sum_{i=1}^{m} g(x:T,M) + \varepsilon, \varepsilon \sim N(0,\rho)$ というm個の木による加法的予測モデル（これを森と呼ぶ）をベイズで求めることを考える。初期値は適当なpriorを考え、森を作成したのちに木を一つずつMCMCで更新していく。
    - これが収束したら、事後分布に基づいたK個の森をサンプリングして、予測する際はK個の森を使うことで予測値の事後分布を得る。
    - 基本的には因果推論として特別なことをするわけではなく，treatmentも他のcovariateと同列に扱った予測モデルを構築し，その上でtreatment Z={0,1}とした予測値の差で因果効果を推定するアプローチ
    - authorのHillは，事前の知識や因果構造についての推論などをモデル化に活用すると，予測していなかった重要な結果をマスクしてしまったり，推定にバイアスを生んでしまう，という立場を取っている．なので事前知識に応じた既知の因果構造を取り込んだpriorを設定するといった方法論の記載は論文内にない．
  - 大域的な構造がない場合やデータセットが大きい場合には、Random Forestのような高次元の交互作用を用いることができるモデルが有利になる。
    - 結局これまでやってきたような介入効果は、簡単な構造でより効率的に（バリアンスにつ強い）推計をできる一方で、より複雑なデータの生成過程がある場合は、今回のようなより複雑なモデルを用いる必要があるということ？
    - 線形回帰モデルのあてはめは構造がシンプルすぎるのが問題ということらしい。概念としてはS-learnerが線形回帰モデルによる因果推論を内包していると考えられる？

Simulation 4 (global linear)

$$
\begin{aligned}
e(x) &= 0.5, \quad d = 5, \\
\mu_0(x) &= x^T\beta, \ \textit{with } \beta \sim \textit{Unif}([1,30]^5), \\
\mu_1(x) &= \mu_0(x).
\end{aligned}
$$

Simulation 5 (piecewise linear)

$$
\begin{aligned}
e(x) &= 0.5, \quad d = 20, \\
\mu_0(x) &= \begin{cases} x^T\beta_l & \textit{if } \ x_{20} < -0.4 \\ x^T\beta_m & \textit{if } \ -0.4 \le x_{20} \le 0.4 \\ x^T\beta_u & \textit{if } \ 0.4 < x_{20}, \end{cases} \\
\mu_1(x) &= \mu_0(x),
\end{aligned}
$$

with

$$
\beta_l(i) = \begin{cases} \beta(i) & \textit{if } \ i \le 5 \\ 0 & \textit{otherwise} \end{cases} \qquad
\beta_m(i) = \begin{cases} \beta(i) & \textit{if } \ 6 \le i \le 10 \\ 0 & \textit{otherwise} \end{cases} \qquad
\beta_u(i) = \begin{cases} \beta(i) & \textit{if } \ 11 \le i \le 15 \\ 0 & \textit{otherwise} \end{cases}
$$

and

$$
\beta \sim \textit{Unif}([-15,15]^d).
$$