統計的機械学習 （応用計量分析２）第5回

ATE推定（続き）とCATE推定（参考pdf ４・5章）

- メタ学習器
- 二重機械学習

本日の内容

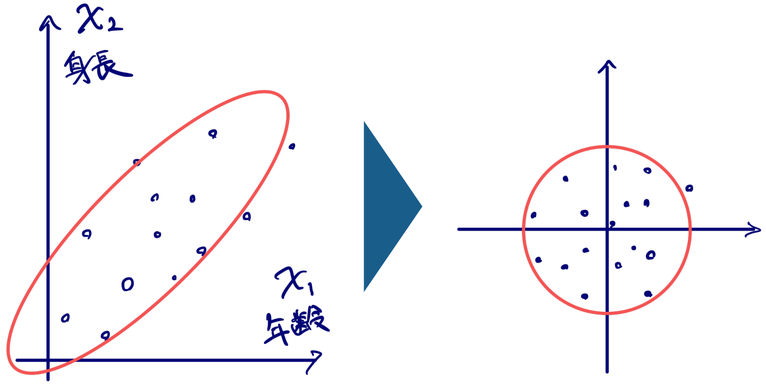
ATE推定の続きから、CATE推定へ

- 1. ガイダンス・因果推論と機械学習の概論
- 2. 意思決定理論（期待効用理論）の復習、因果推論との関係
- 3. 潜在結果モデルに基づく因果推論の枠組み
- **4. 平均因果効果の推定法**
- **5. 条件付き平均因果効果（CATE）の推定法 １：メタ学習器、２：二重機械学習**
- 6. CATEの推定法３：決定木と決定森、深層学習に基づく方法
- 7. 構造的因果モデルとバックドア基準
- 8. 因果探索
- 9. 発展的な因果推論手法：フロントドア調整、操作変数法
- 10. 発展的な因果推論手法：代理変数法、回帰不連続デザイン
- 11. 発展的な意思決定理論
- 12. 強化学習
- 13. オフライン強化学習
- 14. バンディット

（再掲）マッチングに用いる距離

ユークリッド距離・マハラノビス距離・傾向スコア差

- マッチングに使用する距離 d はいくつかの選択肢がある
  - ユークリッド距離
    - 標準化（xの各変数平均0分散1にする）しておく
  - マハラノビス距離
    - 説明変数の分散共分散行列 $\hat{\Sigma}$ を用いて
    - $d_M(x, x') = \sqrt{(x - x')^\top \hat{\Sigma}^{-1} (x - x')}$
  - 傾向スコア差・傾向スコアの対数オッズ差（なぜこれで良いかは後述）
    - $d_p(x, x') = \left| \hat{\mu}(a = 1 \mid x) - \hat{\mu}(a = 1 \mid x') \right|$
    - $d_\ell(x, x') = \left| \log \frac{\hat{\mu}(a = 1 \mid x)}{\hat{\mu}(a = 0 \mid x)} - \log \frac{\hat{\mu}(a = 1 \mid x')}{\hat{\mu}(a = 0 \mid x')} \right|$
    - 1次元なので「近傍」を見つけやすい
- 傾向スコアを推定して用いる傾向スコアマッチングは実用面で不安定という指摘もある（King & Nielsen 2019）
  - 曝露群/対照群に分ける際にも使うことになる（同じデータを二度使う）
  - 次元が高すぎなければマハラノビス距離を使っておくのが安全

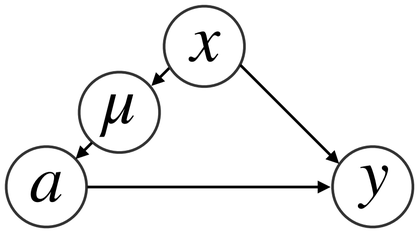


マハラノビス距離は説明変数間の相関を除去した空間でのユークリッド距離

傾向スコアのバランシング特性

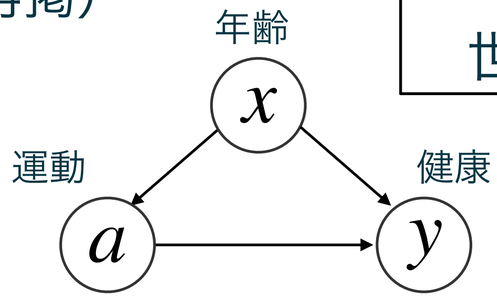
バランシングスコアが同じペアで十分、傾向スコアはその１つ

- バランシングスコアbは、xの関数であって、以下を満たすもの
  - $a \perp\!\!\!\perp x \mid b(x)$
- バランシングスコアをxの代わりに用いても無視可能性を満たせる
  - $y_{a'} \perp\!\!\!\perp a \mid b(x) \quad \forall a'$
- 定理：傾向スコア$\mu(x) = \mu(a = 1 \mid x)$ はバランシングスコアである
  - 証明略（pdf参照）



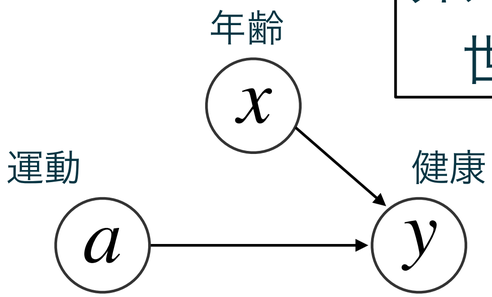
（再掲）

介入前の世界



年齢が高い人が運動する

介入後の世界



年齢が若くても運動する

傾向スコアの推定

プロパーな損失を用いて学習・クリッピングも

- 傾向スコアの学習は漸近的に真値に一致することが推定バイアスの観点では望ましい
  - $\hat{\mu} = \arg\min_{\mu \in \mathscr{F}} \frac{1}{N} \sum_i \ell(a^i, \mu(a \mid x^i))$
  - 関数クラス $\mathscr{F}$ が真の傾向スコアを含む
  - 損失 ℓ として、期待値が真の傾向スコアにおいて唯一最小値をとる損失（**プロパーな損失**）を用いる
    - 真の傾向スコア値$\bar{a} = \mu(a = 1 \mid x)$に対して以下となる
    - $\bar{a} = \arg\min_{a' \in [0,1]} \mathbb{E}_{a \sim \mu(a|x)} \ell(a, a')$
    - 交差エントロピーや二乗損失はプロパーである
- ただし傾向スコアは逆数で使うことが多いため、小さい値は後段の推定分散の観点から避けたい
  - データ点の重み付け $1/\hat{\mu}(a^i \mid x^i)$ が1点だけ100などになると$N = 100$でもほぼその1点だけを重視した学習になる
  - 20以上は20に切り捨てるクリッピングなどを行うことが多い

ATE推定の手法（2/4）

出欠コード

xxxx

IPW推定の応用：ホーヴィッツ＝トンプソン推定量

- 行動と潜在結果の積は観測できることを使う
  - $\tau = \mathbb{E}_x \mathbb{E}_{a\sim\mu(a|x)} \mathbb{E}_{y|x,a} \left[\left(\frac{a}{\mu(x)} - \frac{1-a}{1-\mu(x)} \right) y \right]$
    - 第3回でも既出の変換。最後の期待値は訓練データの分布 $p(y,a,x) = p(y\,|\,a,x)\mu(a\,|\,x)p(x)$ になっていることに注意

$$\leftarrow \mathbb{E}_{y_1,y_0,x}[y_1 - y_0] = \mathbb{E}_x \mathbb{E}_{a\sim\mu(a|x)} \mathbb{E}_{y_1,y_0|x} \left[\frac{a}{\mu(x)} y_1 - \frac{1-a}{1-\mu(x)} y_0 \right]$$

$$= \mathbb{E}_x \mathbb{E}_{a\sim\mu(a|x)} \mathbb{E}_{y|x,a} \left[\left(\frac{a}{\mu(x)} - \frac{1-a}{1-\mu(x)} \right) y \right]$$

- μを$\hat{\mu}$で、期待値を平均で置き換えたものがホーヴィッツ＝トンプソン推定量
  - $\hat{\tau}_{\mathrm{HT}} := \frac{1}{N} \sum_i \left(\frac{a^i}{\hat{\mu}(x^i)} - \frac{1-a^i}{1-\hat{\mu}(x^i)} \right) y^i$
- 上記は、結果の期待値 $\bar{y} = \mathbb{E}[y]$ が非ゼロであった場合、 ()内の重み部分のばらつきにより$\bar{y}$の影響が出て推定精度が悪化する
  - 因果推論では通常τよりも$\bar{y}$の方がスケールが大きいので重みの合計が曝露群と対照群で同じでないと$\bar{y}$の影響が大きく出る
    - yが一定値で$\tau = 0$でも重みの合計が違うと$\hat{\tau}$は非ゼロ
  - よって重みを正規化した修正版を用いることが多い（ハーイェク推定量ともいう）
  - $\hat{\tau}_{\mathrm{Haj}} := \frac{\sum_i \frac{a^i y^i}{\hat{\mu}(x^i)}}{\sum_i \frac{a^i}{\hat{\mu}(x^i)}} - \frac{\sum_i \frac{(1-a^i) y^i}{1-\hat{\mu}(x^i)}}{\sum_i \frac{1-a^i}{1-\hat{\mu}(x^i)}}$

ATE推定の手法（3/4）

出欠コード

xxxx

G計算：CATE推定量を利用する

- ATEはCATEの平均
  - $\tau = \mathbb{E}\left[y_1 - y_0\right]$
  - $= \mathbb{E}_x[\mathbb{E}[y \mid x, a = 1] - \mathbb{E}[y \mid x, a = 0]]$
- 結果予測モデル $\hat{f}(x, a)$ を（なんらか）学習して代入
  - $\hat{\tau}_{\mathrm{gc}} = \frac{1}{N} \sum_i \left[\hat{f}(x^i, a = 1) - \hat{f}(x^i, a = 0)\right]$
    - $f(x, a) = af_1(x) + (1 - a)f_0(x)$ として曝露群の関数f_1と対照群の関数f_0に分ける場合も
      - T(wo)-Learnerという（今回後半）

出欠コード

xxxx

ATE推定の手法（4/4）

IPWとG計算のハイブリッド：二重頑健推定量（Doubly robust estimator、Augmented IPW; AIPW）

- 以下の２つはそれぞれ$\mathbb{E}[a\,|\,x]$、$\mathbb{E}[y\,|\,a,x]$の推定精度に依存
  - IPW $\hat{\tau}_{\mathrm{HT}} := \frac{1}{N}\sum_i \left(\frac{a^i}{\hat{\mu}(x^i)} - \frac{1-a^i}{1-\hat{\mu}(x^i)}\right) y^i$
  - G計算 $\hat{\tau}_{\mathrm{gc}} = \frac{1}{N}\sum_i \left[\hat{f}(x^i, a=1) - \hat{f}(x^i, a=0)\right]$
- これら2つの推定量を組み合わせる
  - $\hat{\tau}_{\mathrm{dr}} = \frac{1}{N}\sum_i \left[\hat{f}(x^i, a=1) - \hat{f}(x^i, a=0) + \left(\frac{a^i}{\hat{\mu}(x^i)} - \frac{1-a^i}{1-\hat{\mu}(x^i)}\right)\left(y^i - \hat{f}(x^i, a^i)\right)\right]$
- $\hat{f}$ **または** $\hat{\mu}$ が正確であれば $\hat{\tau}_{\mathrm{dr}}$ も正確（二重頑健性）
  - 実務的には、両方がそこそこ正確であれば $\hat{\tau}_{\mathrm{dr}}$ はかなり正確
  - 理論的には、それぞれの収束率の積が効率的であれば $\hat{\tau}_{\mathrm{dr}}$ の推定も効率的

出欠コード

xxxx

ATE（ATT）推定法

- 4つの推定法
  - マッチング法
    - 距離の定義によってマッチの取り方がいくつかある
    - 高次元では距離は難しいので1次元の傾向スコアを使う方法も
      - 傾向スコアはバランシングスコアなのでそれが類似ならペアとしてよい
    - 傾向スコアの推定にはプロパーな損失を用いる
  - IPW法（ホーヴィッツ＝トンプソン推定量）
    - 行動と結果の積は観測可能、行動をかけて歪んだ分布を傾向スコアで重み付けて戻す
  - G計算（CATE推定量を用いる）
  - 二重頑健推定量（AIPW）
    - $\hat{\mu}$ と $\hat{f}$ のどちらかが正しければ正しい推定

出欠コード

xxxx

因果推論の典型的な精度は潜在結果や効果の精度

- 推定対象
  - **条件付き平均因果効果**（Conditional Average Treatment Effect; **CATE**）

    $\tau(x) = \mathbb{E}[y_1 - y_0 | x]$

    - ある程度個別化された効果。因果機械学習（機械学習分野における因果推論研究）でよく推定対象とされる
  - 条件付き平均潜在結果 $\mathbb{E}[y_a | x]$
- 評価指標
  - 因果効果の精度 $\mathrm{PEHE}(\hat{\tau}) = \mathbb{E}_x \left[\left(\tau(x) - \hat{\tau}(x) \right)^2 \right]$
  - 潜在結果の精度 $\mathrm{MSE}^u(\hat{f}) := \mathbb{E}_x \left[\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left(\mathbb{E}\left[y_a \mid x \right] - \hat{f}(x, a) \right)^2 \right]$
  - 方策価値 $V(\pi) := \mathbb{E}_x \mathbb{E}_{a \sim \pi(a|x)} \mathbb{E}\left[y_a \mid x \right]$

CATE推定のためのメタ学習器（1/4）

出欠コード

xxxx

メタ学習器は既存の機械学習器をブラックボックス的に使う推定量 aごとにモデルを分けるT-Learner、aも入力とするS-Learner

- メタ学習器（Meta learners）
  - 教師あり学習のためのベース学習器（Base learners）を用いて因果推論用の学習器を構成する手法
    - 参考）機械学習のメタ学習：「学習する方法を学習する」ことで新たなデータでの学習を効率化する手法群　とは別物
- S(ingle)-Learner

1. 行動 a と説明変数 x を区別せず一緒に入力変数として用いて学習： $f(x, a)$
2. $a = 1,0$をそれぞれ入力した予測値の差分を効果の推定とする

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

- T(wo)-Learner

1. 行動ごとにデータを分け、それぞれ別々の潜在結果予測モデルを学習
2. 学習した $\hat{f}_1, \hat{f}_0$ にそれぞれxを入力した予測値の差を効果の推定量とする：

$$\hat{\tau}(x) = \hat{f}_1(x) - \hat{f}_0(x)$$

  - もっとも基本的な選択肢

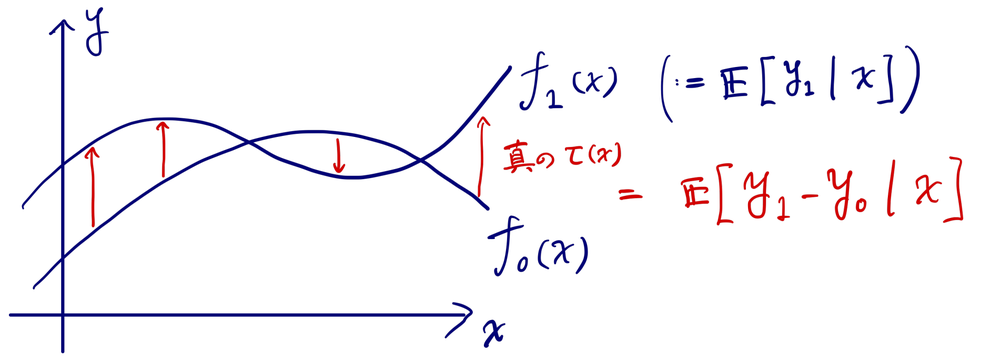
T-Learnerの課題（X-Learnerへの伏線）

出欠コード

xxxx

結果モデル f_a は複雑、曝露群が少ないとき推定精度悪化

- T-Learnerは曝露群・対照群両方の
  結果モデル推定精度に依存
  - $\hat{\tau}(x) = \hat{f}_1(x) - \hat{f}_0(x)$
- 一方で、真の因果効果 $\tau(x)$ は単純かもしれない
  - x に依存しない一定値 $\tau(x) = \tau$ かもしれない
  - x の一部の変数にしか依存しないかもしれない
  - にもかかわらず、結果モデル全体 f_a の推定精度に依存してしまう
  - とくに、曝露群は小さい（$\mu(a=1|x)$が小さい）ことが多く、$\hat{f}_1$は信頼できない
- 因果効果自体のデータはないので $\tau(x)$ を直接学習することはできない
  - しかし、$\hat{f}_0$ は比較的信頼できる。ならば、$\hat{\tau}^i = y_1^i - \hat{f}_0(x^i)$ と推定し、これを学習すればよいのではないか？
  - → これを、x ごとにサンプルの多い方を”信頼”して最終的な推定を行うように一般化したのが **X-Learner**



出欠コード

xxxx

X-Learner：群ごとの効果モデルの重み付き平均

- X-Learnerの手順

1. T-Learner と同様に、行動ごとの潜在結果予測モデルを学習

   $\hat{f}_1(x), \hat{f}_0(x)$

2. 行動ごとのデータに対し上記モデルをプラグインして
   個別の効果推定値を計算

   $\hat{\tau}_0^i = \hat{f}_1(x^i) - y_0^i$ (for $i : a^i = 0$)

   $\hat{\tau}_1^i = y_1^i - \hat{f}_0(x^i)$ (for $i : a^i = 1$)

   - 欠損している方を潜在結果モデルで補完

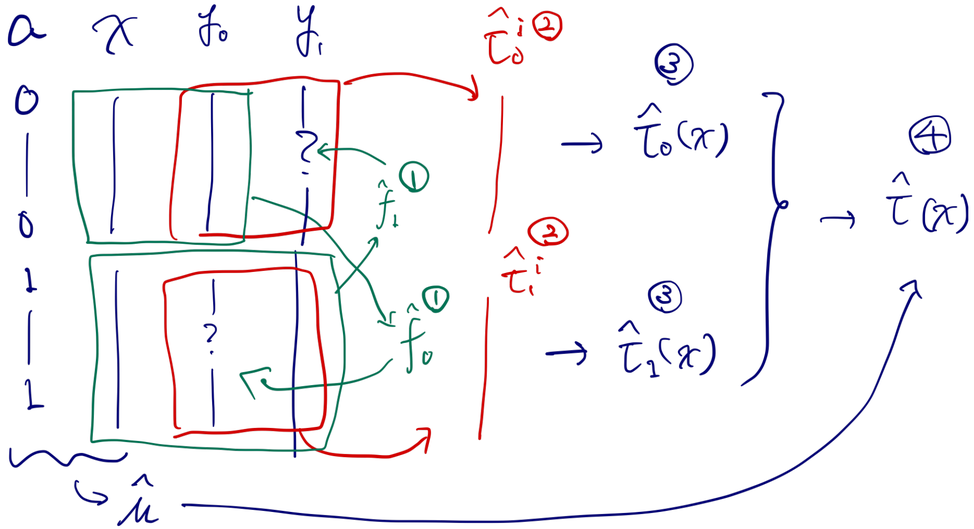
3. 行動ごとの効果モデルを学習

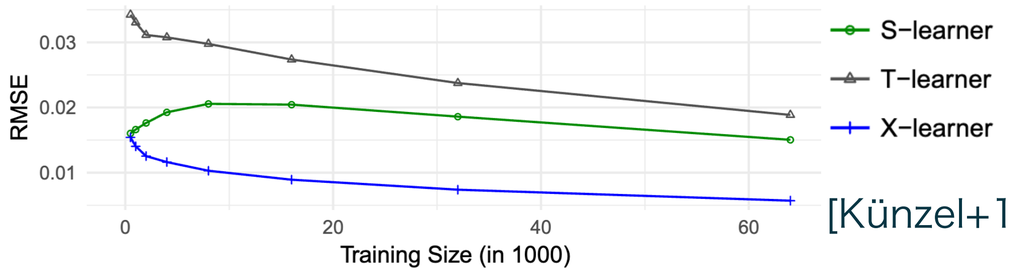
   $\hat{\tau}_0 = \mathrm{Learn}(\{\hat{\tau}_0^i, x^i\}_i),\ \hat{\tau}_1 = \mathrm{Learn}(\{\hat{\tau}_1^i, x^i\}_i)$

4. 最終的に傾向スコアで重み付きアンサンブル（足し合わせ）

   $\hat{\tau}(x) = \hat{\mu}(x)\hat{\tau}_0(x) + (1 - \hat{\mu}(x))\hat{\tau}_1(x)$

   - $\mu(x)$ が小さい＝曝露群が小さい x では $\hat{f}_0$ を途中で用いた $\hat{\tau}_1(x)$ を信頼





[Künzel+19]

CATE推定のためのメタ学習器（3/4）

目的変数変換法：ホーヴィッツ＝トンプソン推定量の応用
傾向スコアによるノイズ付き個別因果効果を用いた直接回帰

- 目的変数変換法：ATE推定のためのホーヴィッツ＝トンプソン推定の応用
  1. 傾向スコア推定モデル $\hat{\mu}(x)$ を学習
  2. 各サンプル $(x^i, a^i, y^i_{a^i})$ に対し、目的変数を以下により変換

$$z^i = y^i_1 \frac{a^i}{\hat{\mu}\left(x^i\right)} - y^i_0 \frac{\left(1 - a^i\right)}{1 - \hat{\mu}\left(x^i\right)}$$

     - $\hat{\mu}$ が正しければ、z^iの条件付き期待値はCATEになる $\mathbb{E}[z^i \mid x^i] = \tau(x^i)$
  3. z を x に回帰

$$\hat{\tau} = \mathrm{Learn}(\{x^i, z^i\}_i)$$

- 利点：結果モデル $f(x, a)$ の推定を経ることなく直接的に $\hat{\tau}$ のモデルを学習できる
- 欠点：傾向スコアの推定 $\hat{\mu}$ の値が0や1に近い場合、z^i に加わるノイズが大きく推定分散が高い

CATE推定のためのメタ学習器（4/4）

DR-Learner：二重頑健推定量の応用
結果予測モデルか傾向スコアモデルのどちらかが良ければ良い

- DR-Learner

1. T-Learnerと同様、行動ごとの潜在結果モデルを学習

   $\hat{f}_1(x), \hat{f}_0(x)$

2. DR推定量の結果変換を適用

$$z^i = \hat{f}_1(x^i) - \hat{f}_0(x^i) + \frac{y_1^i - \hat{f}_1(x^i)}{\hat{\mu}(x^i)} a^i - \frac{y_0^i - \hat{f}_0(x^i)}{1 - \hat{\mu}(x^i)}(1 - a^i)$$

   - 直接法による予測＋
     残差を目的変数変換法と同様逆傾向スコア重み付け

3. その差をxに回帰して効果モデルを学習

   $\hat{\tau} = \mathrm{Learn}(\{x^i, z^i\}_i)$

- 利点
  - 結果モデルと傾向スコアモデルの**いずれか**が正しければ後段の $\hat{\tau}$ の推定への悪影響が少ない
  - 理論的には、全体の収束レートが2つのモデルの収束レートの**積**になる
    - 片方の仮説集合が間違っていて $O(1)$ レートでも、もう片方が正しくて $O(N^{-1/2})$ レートであればよい
    - 両方のモデルがそれぞれ $O(N^{-1/4})$ 以上であれば効率的なレート $O(N^{-1/2})$ が達成される
- 欠点
  - $\hat{\mu}$ の値が0や1に近い場合、残差なのでマシとはいえやはりノイズが大きく推定が不安定

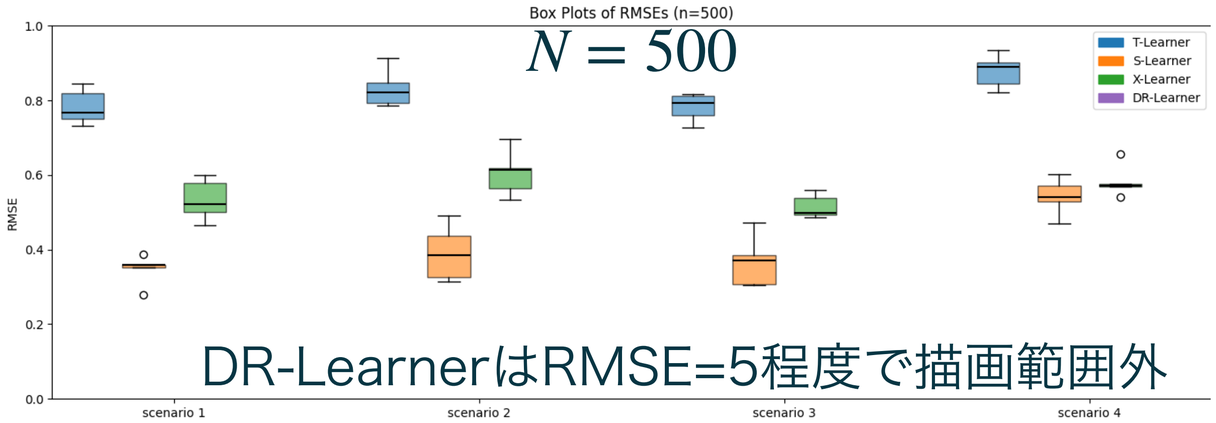
性能比較例

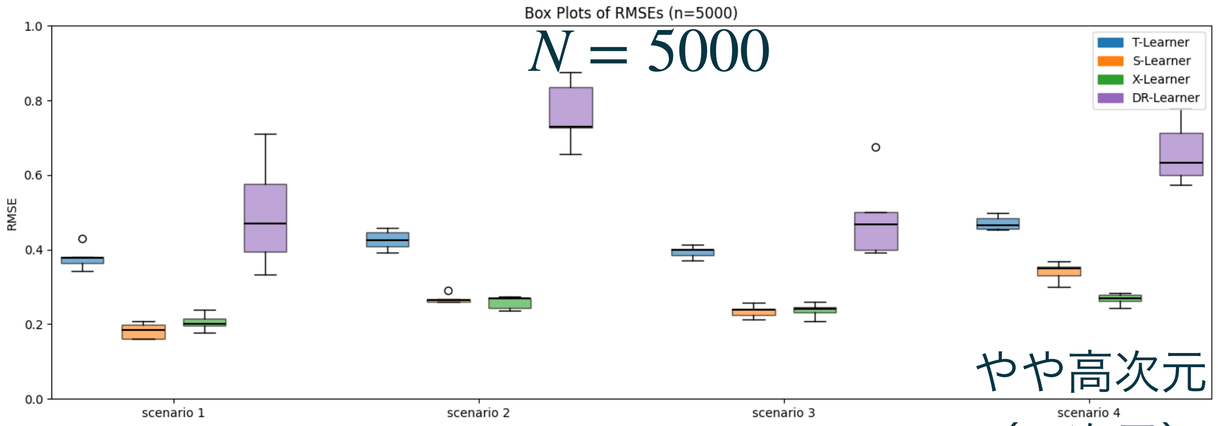
出欠コード

xxxx

DR-Learnerは逆数をとるので不安定になりがち

- 実際の精度は諸条件に依存するので参考まで
  - サンプルサイズN、次元p、μの偏り、真のモデル$f(x,a)$の複雑さ、ベース学習器、等
- 理論では $N \to \infty$ の漸近的な性質を議論しがち（理論的に言えることが他にあまりない）だが、DR-Learnerなど推定値の逆数をとる推定法はサンプルサイズが大きくないとき不安定になりがち





（30次元）

4つのデータ生成モデルで実行した例

（ベース学習器：勾配ブースティング決定木 GBDT）

二重機械学習

二値行動の一般化：行動線形モデル

二値/連続値行動向けセミパラメトリック法：DML（1/3）

出欠コード

xxxx

二重機械学習：連続的行動 $a \in \mathbb{R}$ 等にも適用可能

- **二重機械学習（Double/Debiased Machine Learning; DML、R-Learner）**
  - $y = \theta(x) \cdot a + g(x, w) + \varepsilon \qquad \big(\mathbb{E}[\varepsilon \mid x, w] = 0\big) \quad \cdots (1)$
    - 説明変数のうち、aの係数に関与する部分（**効果修飾因子**）をx、関与しない部分をw
    - 行動aに関して線形性を仮定、その他の部分 g は非線形を許容（セミパラメトリック）
  - $a = h(x, w) + \eta \qquad (\mathbb{E}[\eta \mid x, w] = 0)$
  - 部分線形モデルが正しい（誤指定ではない）場合、無視可能性から $\mathbb{E}[\eta \cdot \varepsilon \mid x, w] = 0$
- 残差を線形モデルで学習
  - (1) の期待値をとると $\mathbb{E}[y \mid x, w] = \theta(x) \cdot \mathbb{E}[a \mid x, w] + g(x, w) \cdots (2)$
  - (1) の両辺から (2) の両辺を引くと、
  - $y - \mathbb{E}[y \mid x, w] = \theta(x) \cdot (a - \mathbb{E}[a \mid x, w]) + \varepsilon$
    “普段”の売上との差　　“普段”の価格との差
- 結果を(x, w)に回帰した残差を行動の残差に回帰する

← 推定の邪魔だった $g(x, w)$ が消えている！

Robinsonの分解とも呼ばれることから
DMLによるCATE推定法のことをR-Learnerとも呼ぶ

出欠コード

xxxx

DML（2/3）イメージ

行動以外の背景因子の効果を差っ引くことで背景因子ごとに原点をずらす

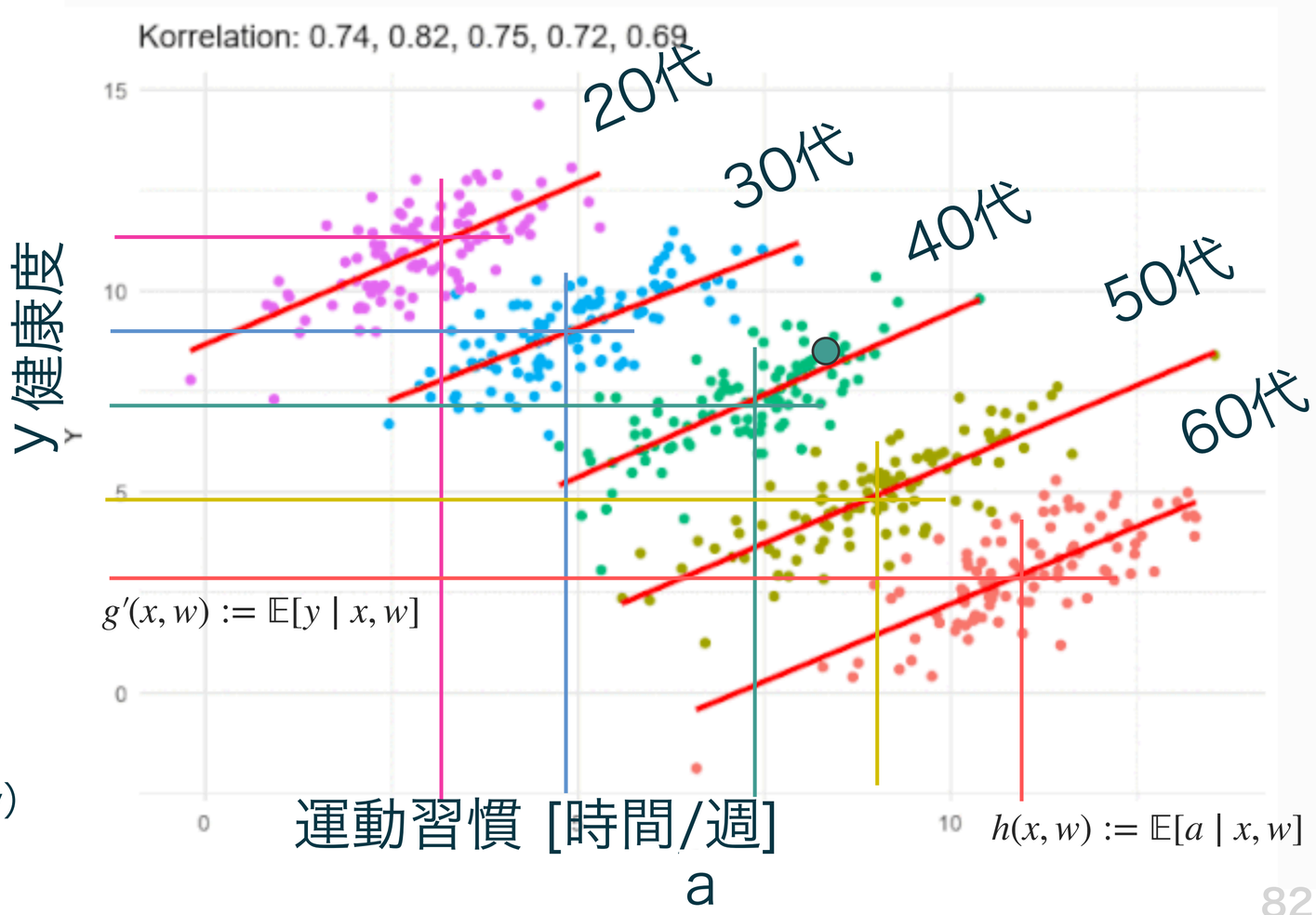
- $y - \mathbb{E}[y \mid x, w] = \theta(x) \cdot (a - \mathbb{E}[a \mid x, w]) + \varepsilon$
- 属性 (x, w)ごとに「原点」を設定

  そこからの差分にフォーカス
- g' と h の推定誤差が θ の推定誤差

  に影響する度合いが小さい
  - セミパラメトリック推論の理論
  - 誤差 Δ_θ を $(\Delta_h, \Delta_{g'})$ についてテイラー

    展開した１次の係数がゼロ
    - ネイマン直交性（Neyman orthogonality）

      という



DML（3/3）データ分割法（クロスフィット）

出欠コード

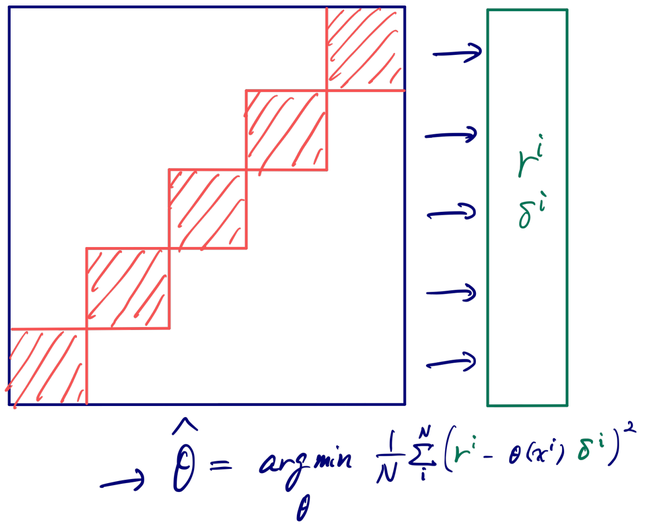
xxxx

残差を偏りなく推定するため、K-foldに分割して推定

- 学習用データと同じデータで残差を評価すると過小評価になる
  - 訓練誤差はテスト誤差より通常小さい
- データをK分割し、未学習データで残差 r^i, δ^i を算出
- これらを集め、$\left(x^i, r^i, \delta^i\right)$から$\theta(x)$ を推定
  - $\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{i}^{N} \left(r^i - \theta(x^i)\delta^i\right)^2$
- （または各foldで学習したモデルを平均する）

$$r^i = y^i - \hat{g}'(x^i, w^i)$$

$$\delta^i = a^i - \hat{h}(x^i, w^i)$$



（参考）Neyman直交性

モーメント条件が迷惑モデルの推定誤差に関して直交

- 真のパラメータ θ_0, 真の迷惑モデル $\eta_0 = (g', h)$ に対して、

  損失関数がモーメント条件 $E_P\left[\psi\left(o; \theta_0, \eta_0\right)\right] = 0$ を満たすと仮定する

  - ψ はスコア関数。損失関数のθ微分だと思えばOK
- Neyman直交性の定義
  - $\partial_\eta E_P\left[\psi\left(o; \theta_0, \eta_0\right)\right]\left[\eta - \eta_0\right] = 0, \quad \text{for all } \eta \in \mathcal{T}_N$
  - 微小なズレ $\Delta_\eta \ll 1$ では期待モーメントは0から変化しない

    →真のパラメータへの影響も小さい
- なお、Doubly RobustならNeyman直交だが、その逆は必ずしも真ではない
  - R-Learnerは二重頑健ではない

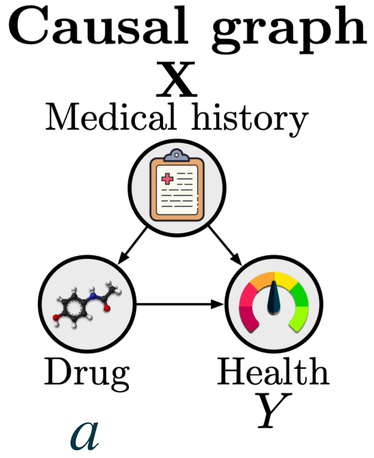
（参考）DMLの構造的行動への拡張

出欠コード

xxxx

構造のある行動空間の特徴付けに対して線形なモデル

- Causal Effect Inference for Structured Treatments (NeurIPS '21)
- 積効果モデル: $y = g(\mathbf{x})^\top h(\mathbf{a}) + \varepsilon$
  - cf) DML: $y = \theta(x) \cdot a + g(x, w) + \varepsilon$　→ aは二値か1次元か1-hotベクトルくらい
  - 行動を特徴付ける空間 h を同時学習する
  - $e^h(x) := \mathbb{E}\left[h(a) \,|\, x\right]$
  - 一般化Robinson分解
    - $y - m(x) = g(\mathbf{x})^\top \left(h(\mathbf{a}) - e^h(x)\right) + \varepsilon$
- 学習ステップ
  - $m(x) \approx \mathbb{E}[y \,|\, x]$ を学習
  - while not converged
    - g, h をK回更新
    - e^h を更新



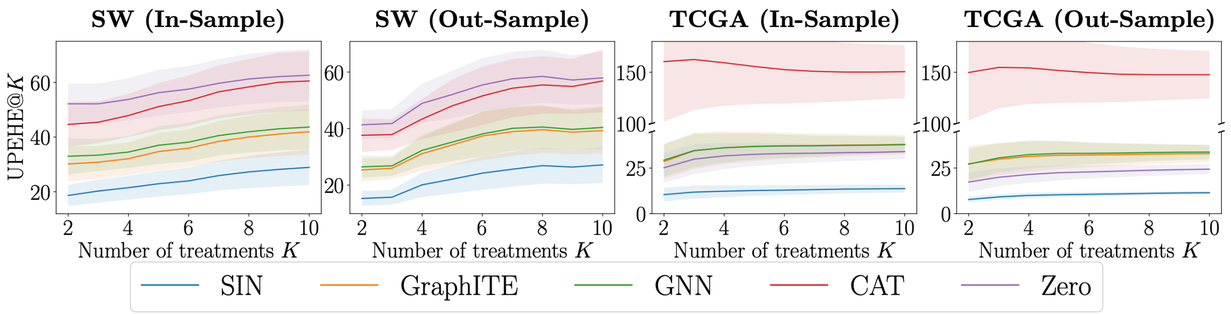


Figure 3: UPEHE@K for $K \in \{2, \ldots, 10\}$.

出欠コード

xxxx

CATE推定のためのメタ学習とDML

- メタ学習器
  - S-Learnerは行動を入力変数に含めてモデル化
  - T-Learnerは行動ごとに分けてモデル化
  - X-LearnerはT-Learnerをベースに曝露/対照群間の偏りによって結果モデルの推定精度がネックにならないよう群ごとに$\hat{\tau}$を学習して重みつき平均
  - DR-LearnerはAIPW法の擬似結果を回帰する方法で二重頑健性を保証
- 二重機械学習（DML）
  - 行動が二値$a \in \{0,1\}$の場合に加えて連続値の場合に対して線形モデルとした

    セミパラメトリックモデル（a以外に関しては非線形を許容）
  - 結果yを説明変数（a以外）に回帰したモデルの残差を、行動を説明変数に回帰したモデルの残差に回帰
  - 残差が過小推定とならないようにデータ分割、クロスフィットを行う