ControlBurn:
Feature Selection by Sparse Forests

2022年6月7日 @ 読み会

発表者：楊明哲

論文情報と選択理由

論文情報

ControlBurn: Feature Selection by Sparse Forests

Brian Liu
Cornell University

Miaolan Xie
Cornell University

Madeleine Udell
Cornell University

選択理由

特徴量選択手法は説明性，公平性にも大きく関わるから

Kaggleの特徴量をとりあえず全部つっこむことに納得できない

TreeBaseの特徴量選択手法を提案

背景：決定木＋アンサンブル手法は精度，解釈性ともに優秀

問題：特徴量同士に相関があるデータでは，
　　特徴量重要度が当てにならない → 相関バイアスに弱い

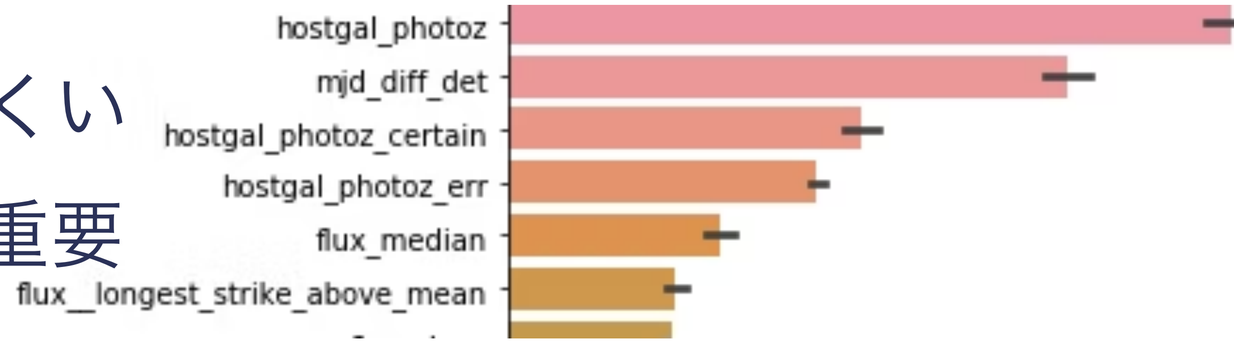
提案：特徴量選択に重み付きLASSOを適用

結果：相関があるデータで既存の特徴量選択よりも
　　ROC-AUCが高く，かつ同等の計算コストに抑えた

決定木は相関バイアスに弱い

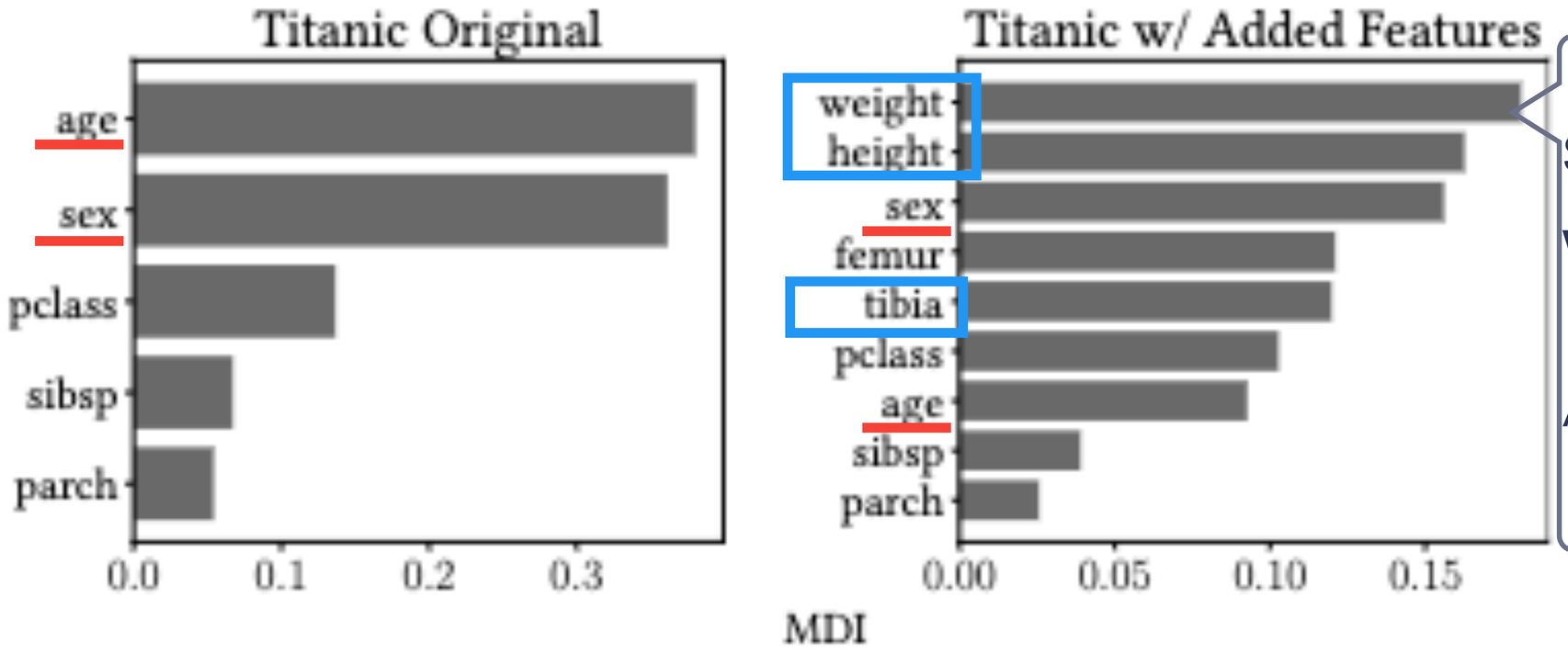
決定木ベースの手法は
精度と解釈性のバランスが良い
アンサンブルベースは特徴量数が
大きい場合でもオーバフィットしにくい
ブースティング系の手法では特徴量重要
度の計算が容易

- LightGBM, XGBoost, Scikit-learn
  の.feature_importance



LightGBMのfeature importance例

相関バイアスによって特徴量重要度が変化する



Sexから

wight, height, femur

Ageから**tibia**を人工生成

相関がある特徴量同士で重要度が薄まってしまう

使わない特徴量を後から間引く

最初に決定木による深い森（アンサンブルツリー）を作り
あとで必要ない木（特徴量）を燃やす🔥🌲🔥

応用先

- 説明可能性：必要な特徴量を選択するから解釈しやすい！
- Optimal experimental design
  - 特徴量を取得するためのコストを削減できる
  - 医療分野とかでわざわざデータをとる時など

準備：決定木のいろいろ

決定木：ジニ不純度や分類誤差を最小化するように学習

- 深さが深くなるほど表現が細かくなる（$d \to 2^d$ ）

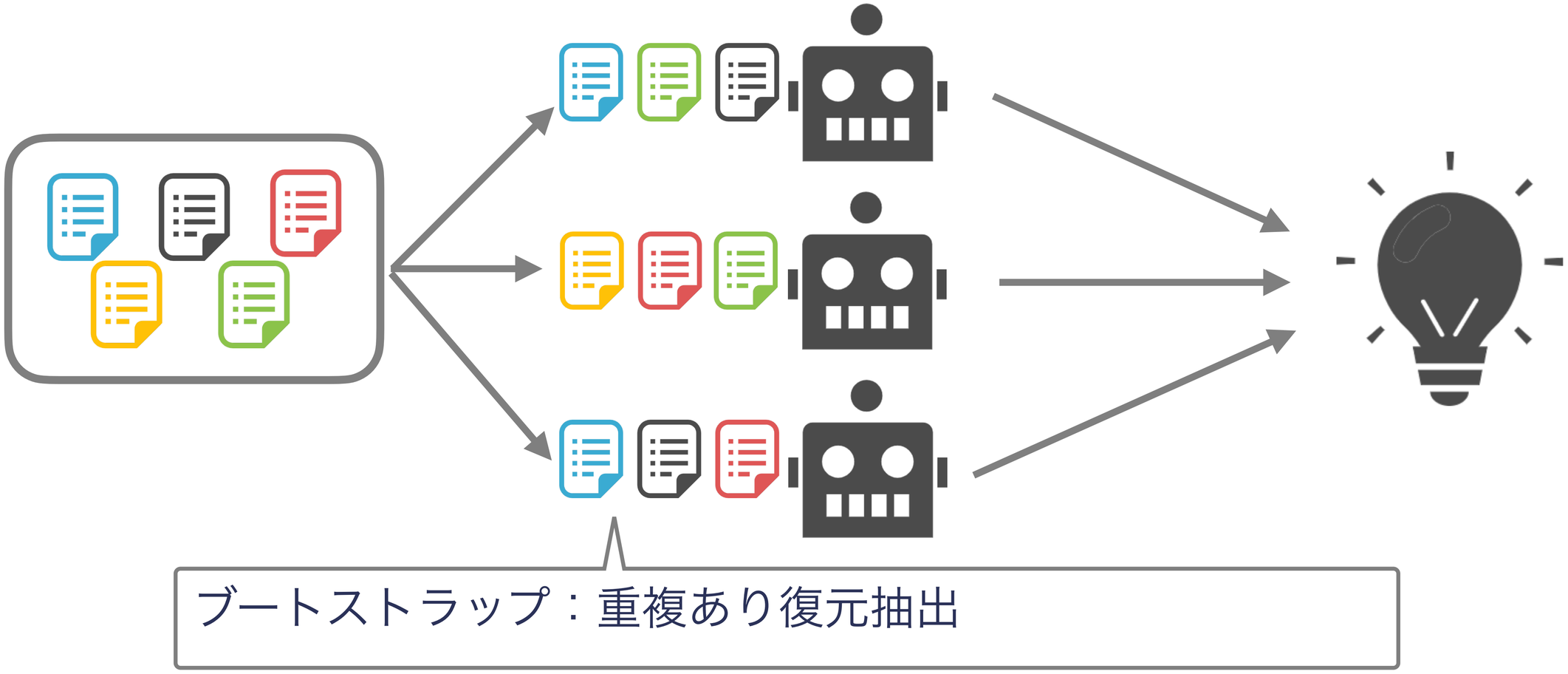
Bagging：bootstrapしたデータで
学習器を**独立**に学習，出力を統合して最終出力

Boosting：弱学習器を**直列**に繋いで学習
最終的にはそれぞれのモデルを統合して出力

Feature importance

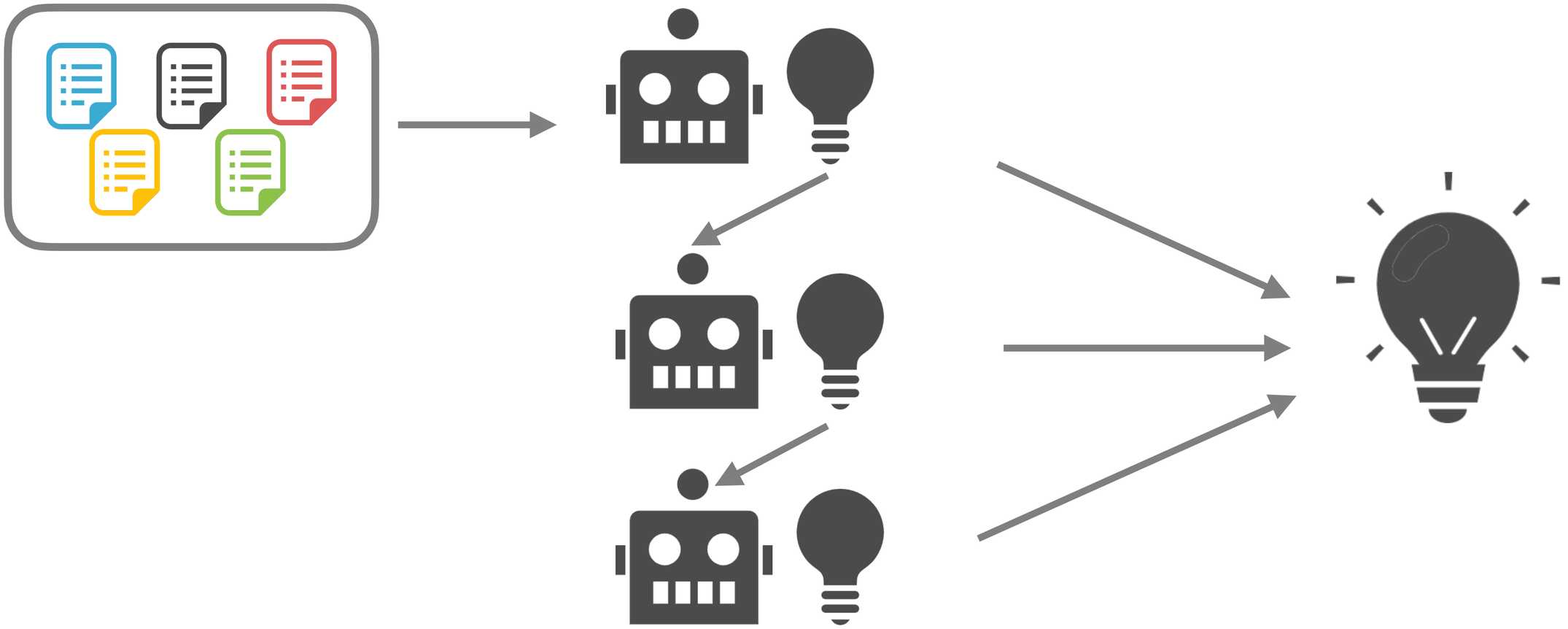
- ジニ不純度などを元に計算する → mean Decrease Impurity

Bagging：バリアンスが小さくなりやすい



- 複数で多数決するからバリアンスが抑えられる

Boosting：バイアスが小さくなりやすい



- 直前のモデルの出力を元にするからバイアスが抑えられる

準備：特徴量選択のいろいろ

データの中にはモデル性能に影響しない特徴量があるはず

モデルがノイズに対して頑健ならいいが，そうはいかない

→ 必要な特徴量を選択したい！

既存手法3パターン

Filter base：データ統計から特徴量を評価

Wrapper：機械学習モデルを使って評価

Embedding：モデル学習と同時に評価（e.g. Lasso）

提案手法：Lasso的な制約で特徴量選択

t_i for $i \in \{1,2,...,n\}$：n個の決定木

$\alpha_i \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$：分類木の$m$クラスの出力結果

$g_i \in \{0,1\}^p, G \in \mathbb{R}^{p \times n}$：p個の特徴量を使うかどうかのマスク

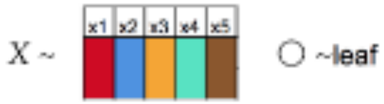
$w \in \mathbb{R}^n$：決定木の重みパラメータ

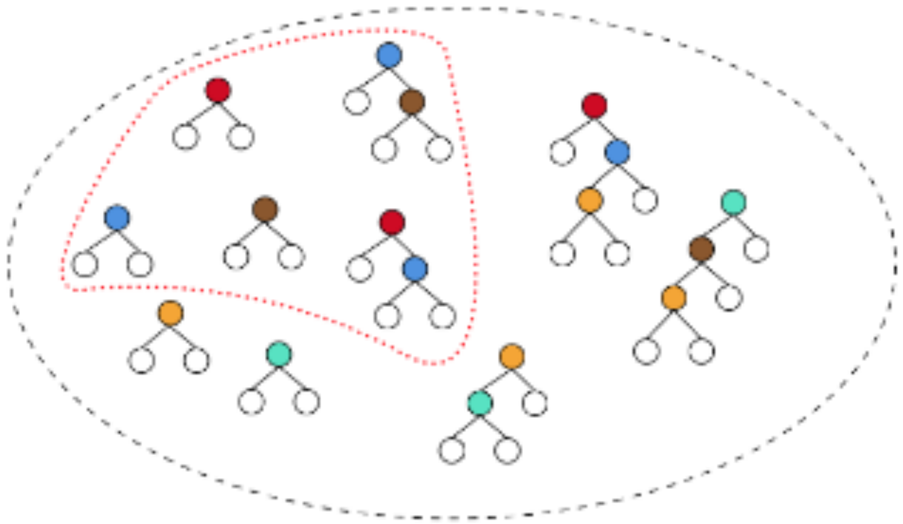
λ：正則化パラメータ

Lasso的な制約で最適化問題とする

$$\begin{aligned} \text{minimize} \quad & \frac{1}{m} L(A, w, y) + \lambda \|Gw\|_1 \\ \text{subject to} \quad & w \geq 0, \end{aligned}$$

Group Lassoみたいな罰則項





wによって赤い枠がきまる

色付きノードが使用する特徴量→Gにより決定

提案手法：森を育てて, 木を間引く

森を育てる：決定木の学習方法を2つ提案

- Incremental depth bagging
- Incremental depth bag-boosting

木を間引く：前述の最適化問題を解く

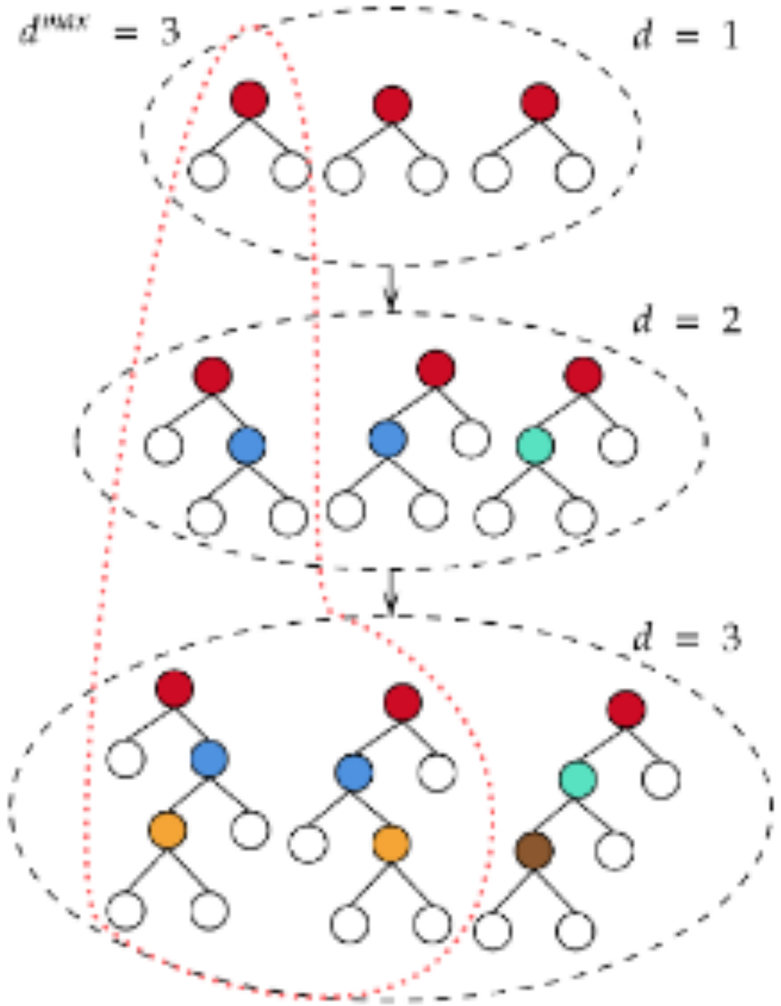
→森を育てるパートと木を間引くパートはそれぞれ独立する

Incremental depth bagging

ハイパラとして最大深さd_{max}を設定

収束したら木の深さを深くする

Train誤差が十分小さい→baggingは過学習しにくい+計算が早い



Algorithm 1 Incremental Depth Bagging.

Input: maximum depth $d^{\max}$

Initialize $d \leftarrow 1$, $F \leftarrow \emptyset$
while $d \leq d^{\max}$ **do**
    Sample bag X' from X with replacement
    Fit tree t of depth d on X', y
    Add tree t to forest: $F \leftarrow F \cup t$
    Compute train error of forest F
    If train error has converged §3.2.1, increment $d \leftarrow d+1$

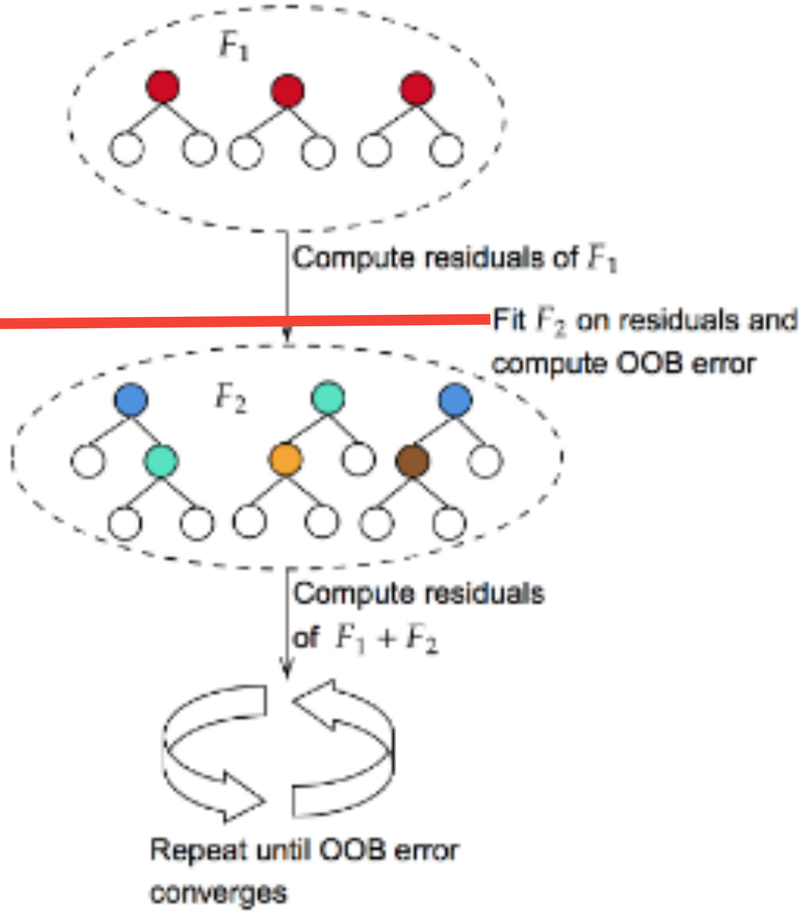
Output: Forest F

Incremental depth bag-boosting

- d_{max}は残渣から**自動調整できる** → 改善される限り深く！

Algorithm 2: Incremental Depth Bag-Boosting.

Initialize $d \leftarrow 1$, $F' \leftarrow \emptyset$, $F \leftarrow \emptyset$, $\delta > 0$
Initialize $F(x)$ by predicting the mean/majority class of y, set $e \in \mathbb{R}^m$ as the vector of residuals
while $\delta > 0$ **do**
    Sample bag X' from X with replacement
    Fit tree t of depth d on X', e
    $F' \leftarrow F' \cup t$
    Compute train error of forest $F \cup F'$
    if *train error has converged §3.2.1* **then**
        $F \leftarrow F \cup F'$
        Set e as negative gradient of loss
        Increment $d \leftarrow d + 1$
        Set δ, the improvement in OOB error from F'
        $F' \leftarrow \emptyset$
Output: Forest F



実験：提案の特徴量選択手法の妥当性を見る

Semi-synthetic：

- 人工的に相関のある特徴量を追加
- 相関バイアスを上手に扱えるかを見る

Benchmark：

- 43のデータセットで既存手法とROC-AUCの改善を見る

Case studies：

- リアルデータセットでどうなるか見る

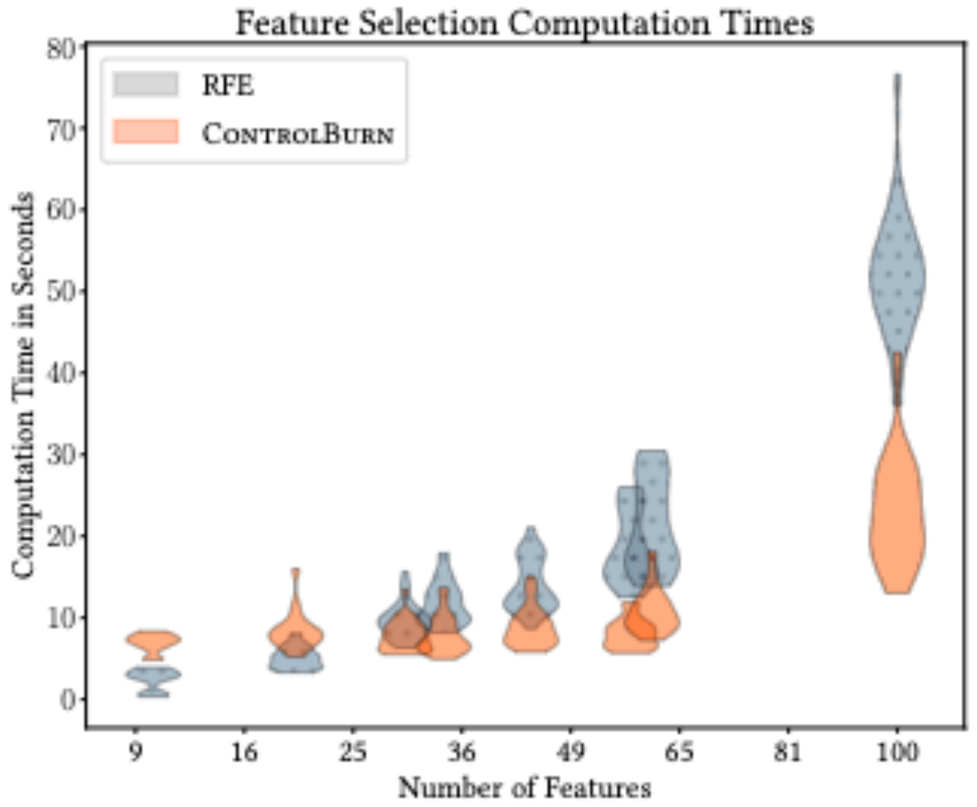
ベンチマークと計算量評価

事前に全てのデータを使ってランダムフォレストを学習

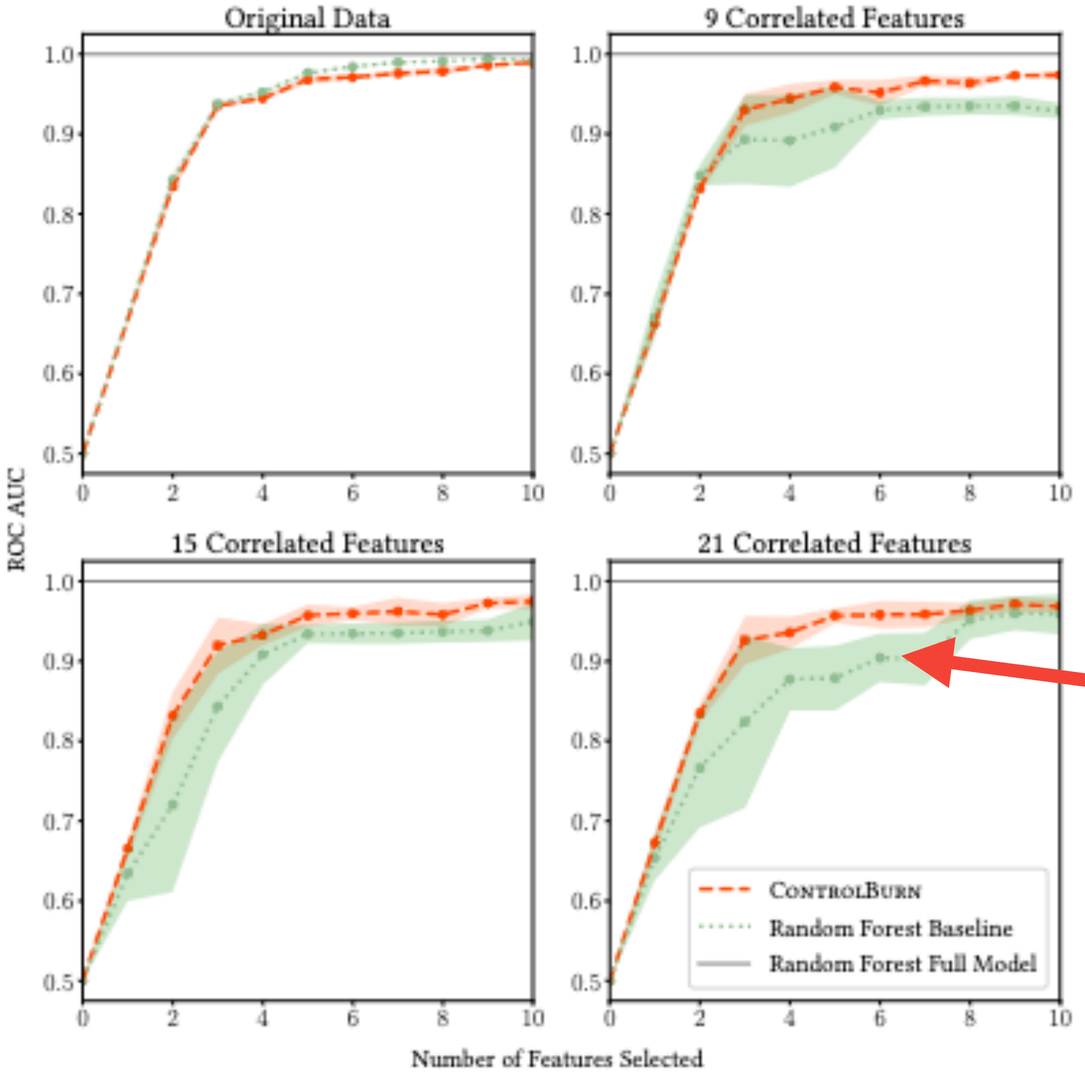
特徴量重要度を元に
上位 k個の特徴量を選択

選択された特徴量だけで再学習
→　ベンチマークとする

Wrapper-base手法のRecursive Feature Eliminationと計算量比較する



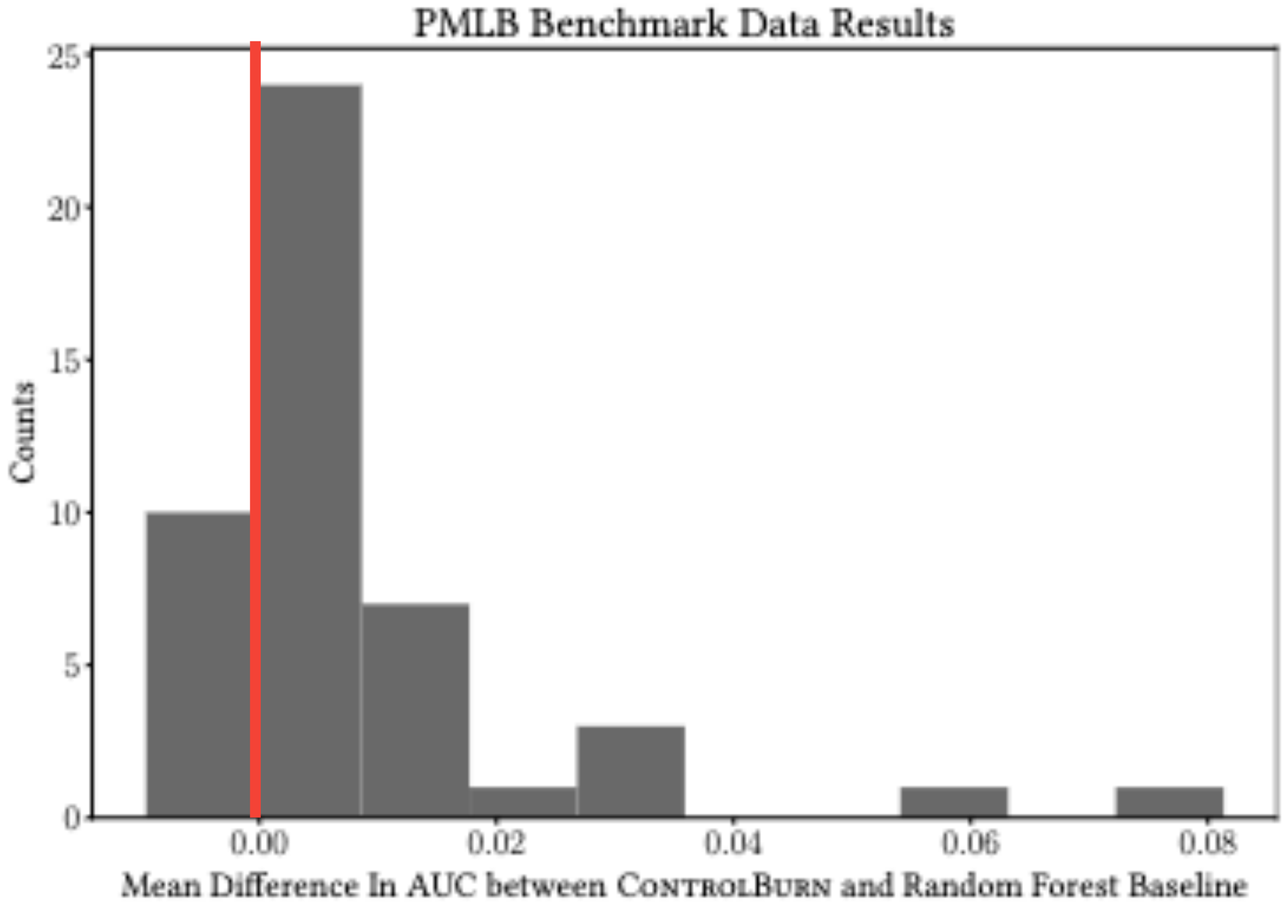
結果：相関バイアスの影響を受けにくかった



Random Forestの特徴量重要度上位にノイズを加えて，新たな特徴量として追加

ノイズ特徴量が増加しても影響が受けずに頑健

Benchmark：既存手法より精度が高め



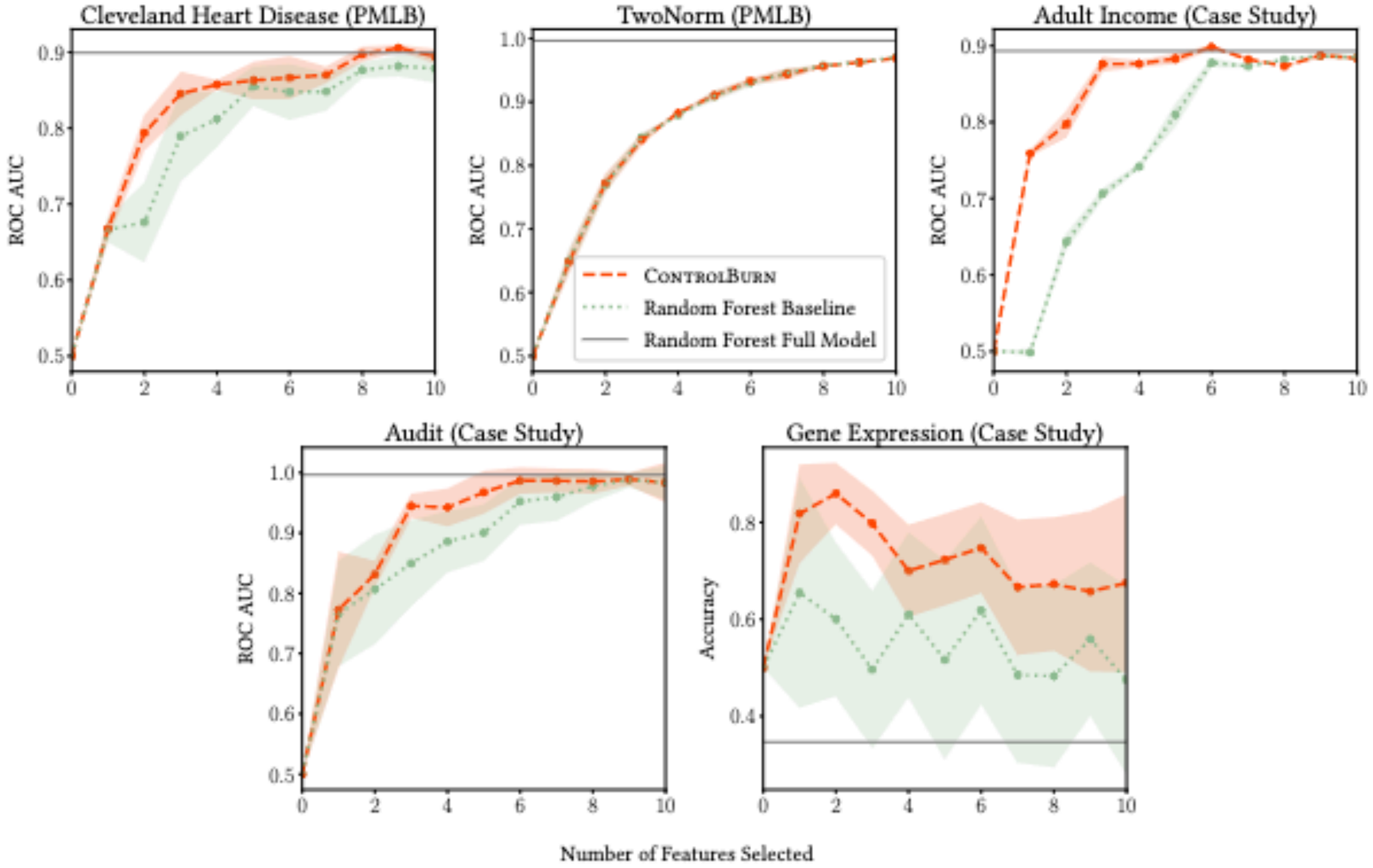
特徴量数を$k = 1,2,....,10$を

使ったROC-AUCの度数分布

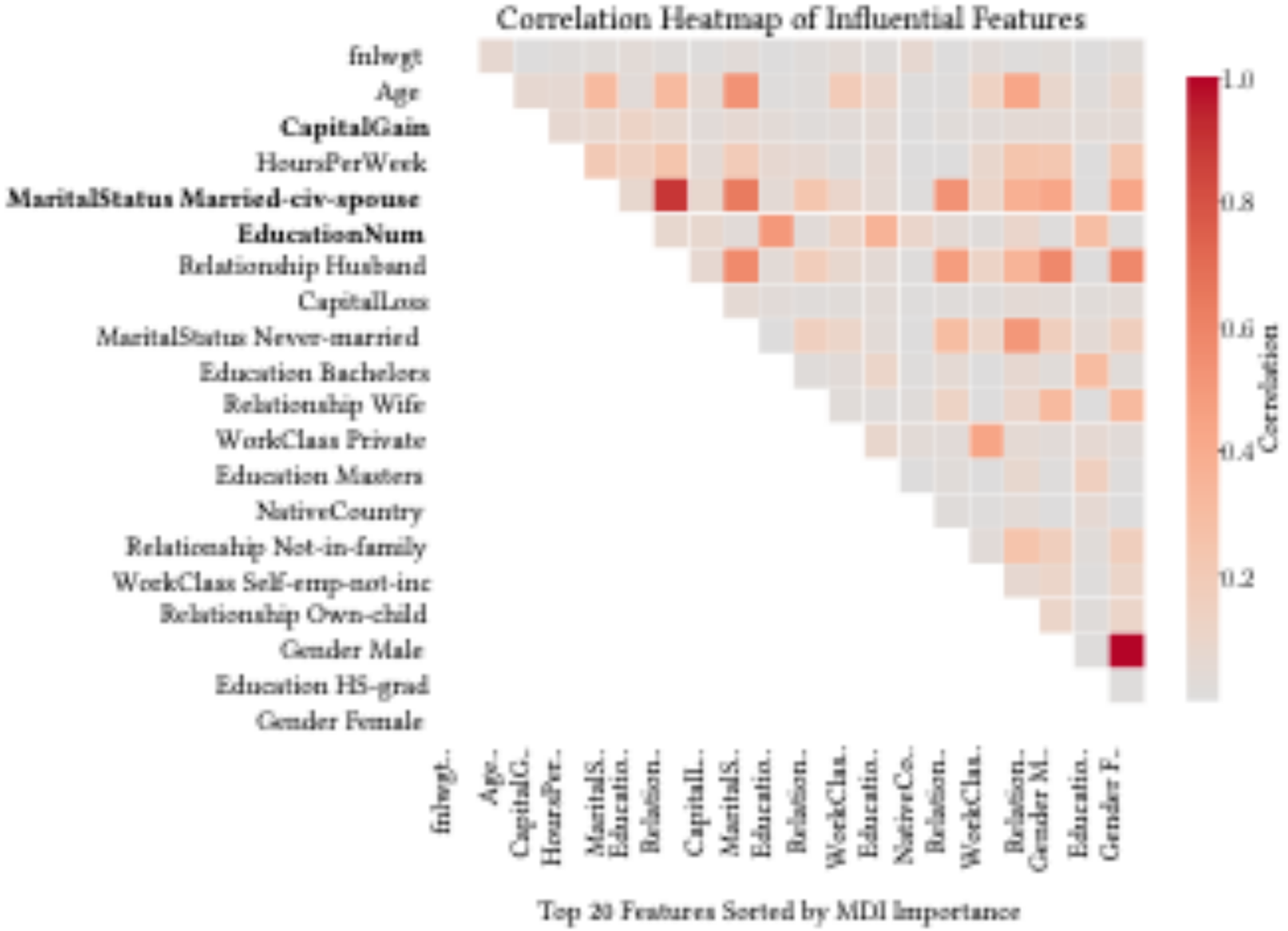
ベンチマークより精度向上が

見られた

Case Studies：
リアルデータのノイズでも利用可能だった



リアルデータセットでの相関バイアスが改善



縦軸**太字**が提案による上位特徴量

横軸が既存手法の上位特徴量

Fnlwgtがなぜか上位に来ているのが消えている

まとめ

- 重み付きLassoベースの
  特徴量選択を行うアルゴリズムを提案
- さまざまなデータセットを用いて，既存手法と比べ
  相関バイアスに対して頑健であることを示した

感想

- 特徴量選択のモチベーションもわかりやすく，手法のやりたいこともわかったが，実際の学習方法が少し曖昧
- 昔Human boosting読んだが，
  Human Stacking, Human NNとかってあるのか気になった
- 公平性に関していうと，レッドライン効果に対して使えそう
- 相関のある特徴量において
  どっちが本質か決定するの難しそう