決定木アルゴリズム: モデル集計

経済学のための機械学習入門

川田恵介

Table of contents

Model aggregation

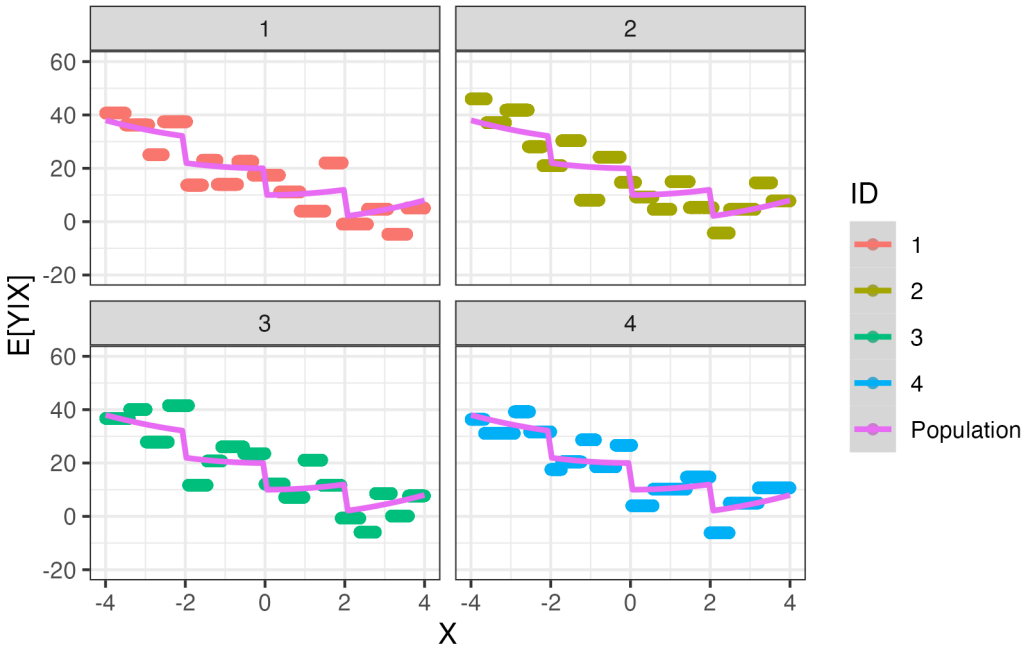
比喩: 予測屋会議

- 複数の” 専門家” の予測を集計して最終予測モデルとする
  - “エコノミスト” の見通しの平均値
  - 専門家委員会
- 一人の予測に頼るよりも、ましでは？
  - 教師付き学習にも応用可能な発想

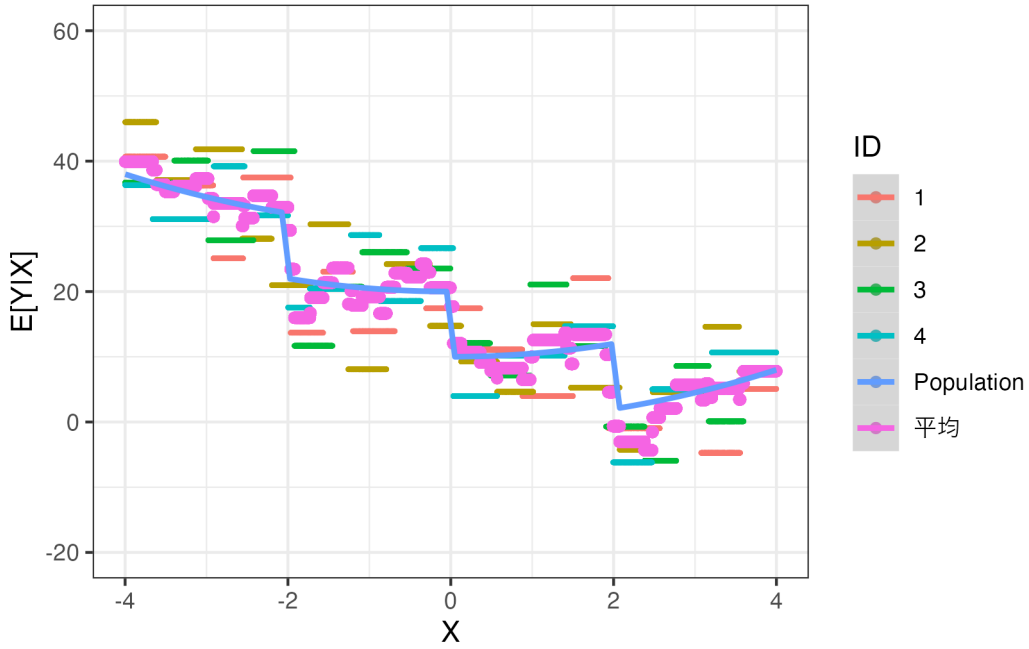
数値例

- 独立してサンプリングしたデータについて、深い予測木 (剪定なし) を推定
- 各予測値と、予測値の平均を比較

Estimation Error



Aggregation



チャレンジ

- 「独立して抽出された」有限個データから生成された予測モデル
- 「独立して抽出した複数のデータから得た」予測モデルの集計は通常不可能
  - 推定に使ったサンプルサイズが実質的に増えているので、性能改善は” 当たり前”
- 近似的に行う
  - (Nonparametric) bootstrap の活用

Bootstrap Aggregating

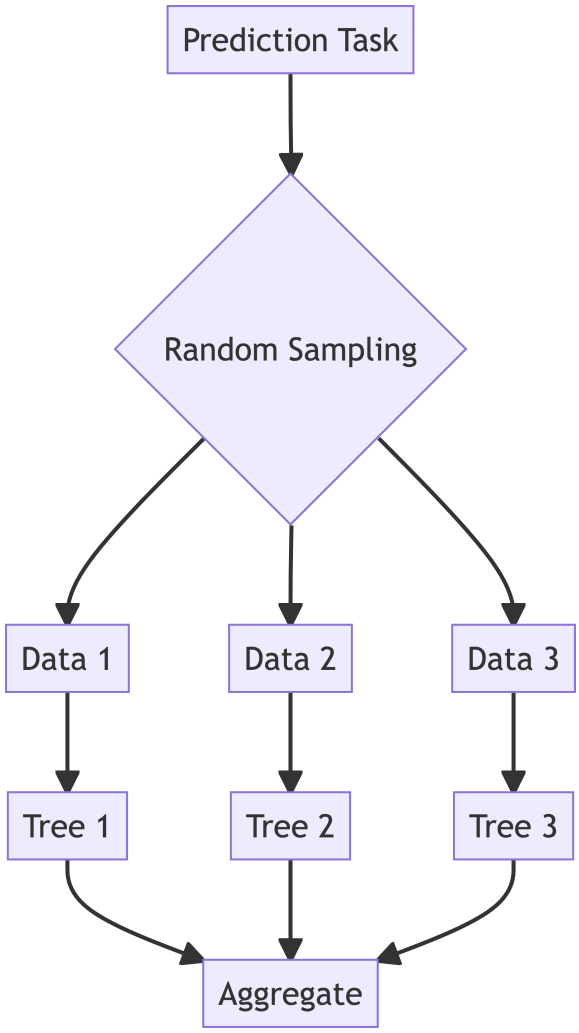
- Bagging

決定木の不安定性

- 多くの実践で、決定木推定の不安定性 (= データ依存, 大きな Estimation error) は、Reguliazation を行っても十分に緩和できない
  - 変数や分割回数の決定など、Discrete choice が避けられないことが理由の一つ

- 伝統的なアプローチ (研究者がモデルを設定する OLS, サブグループ分析) では、無意味な方法が有効
  - Bootstrap でデータを複製して、モデル集計

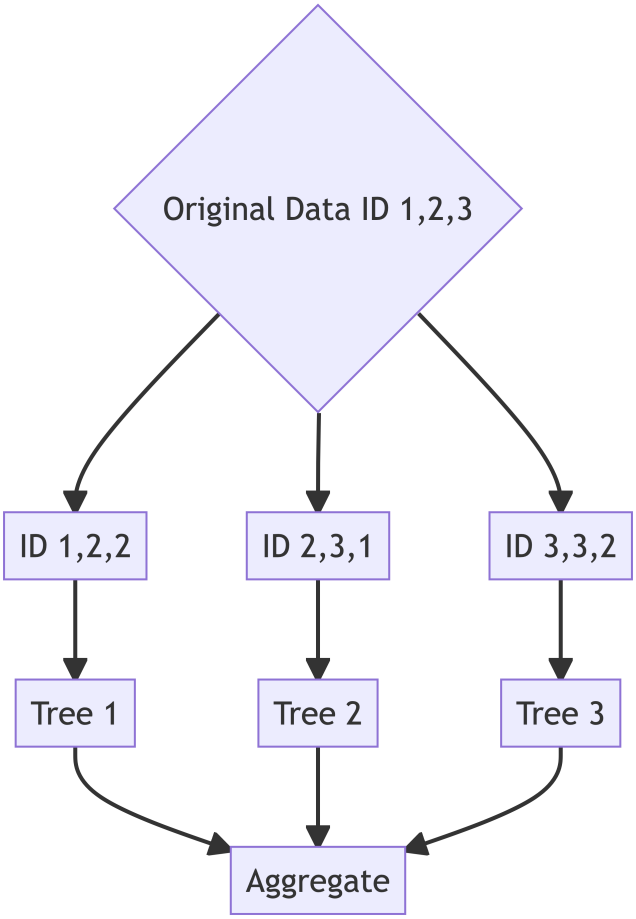
理想の Bagging



アルゴリズム

1. Nonparametric bootstrap で、データの複製を行う (500,1000,2000 など)
2. 各複製データについて、“深い” 決定木を推定
3. 各 X についての予測値の平均を最終予測値とする

補論: Bootstrap



Bagging **の発想**

- $g_b(X)$ 複製データ b から生成されたモデルの X についての予測値
  - $g_b(X)$：確率変数
- 確率変数は、一般に、

$$g_{ave}(X) := \frac{\sum_b g_b(X)}{B}$$

Bagging の発想

- 基本アイディア: 非常に深い木 $g_b(X)$ を生成すれば、Approximation error は減少する一方で、Estimation error が大きくなる
- 確率変数の平均値は一般に分散が削減できる
  - 独立・無相関であれば、無限個の**複製データ**から予測モデルを作れば、分散を 0 にできる (一致性)
  - 今の PC であれば、大量の予測モデルの生成は可能

Bagging の限界

- Bootstrap から計算した統計量は、一般に相関するので

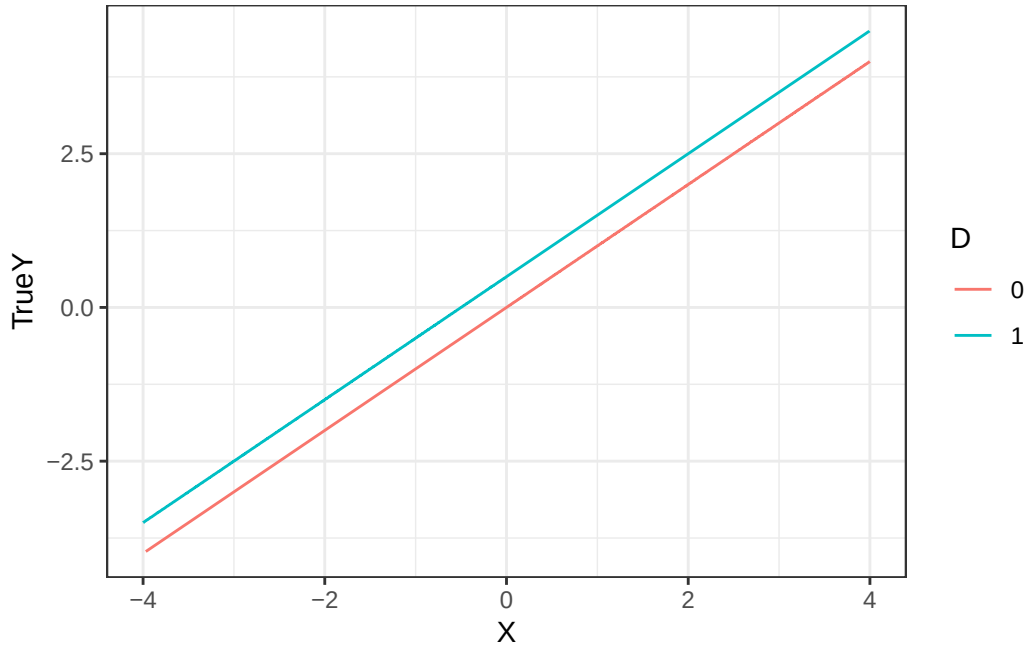
$$\lim_{B\to\infty} E[(g_{ave}(X) - E[g_{ave}(X)])^2] = \underbrace{\frac{var(g_b(X))}{B}}_{\to 0}$$

$$+ \underbrace{\frac{B-1}{B} \times corr(g_b(X), g_{b'}(X)) \times var(g_b(X))}_{\not\to 0}$$

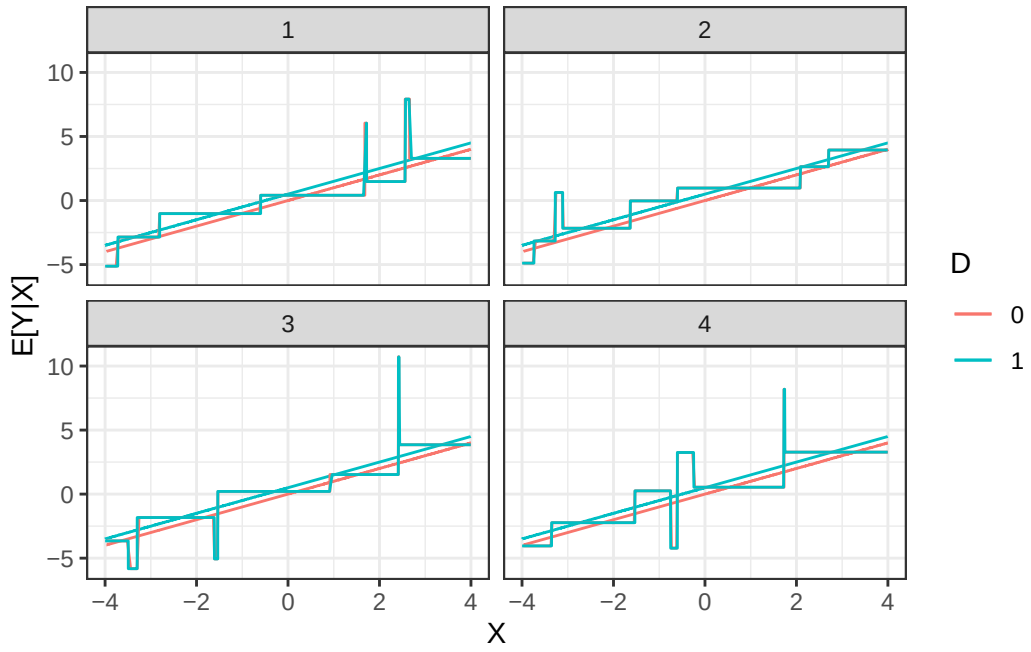
RandomForest

- データ分割に用いることができる変数群をランダムに選ぶ
- 例：ある予測木の第 n 分割を行う際に
  - Bagging: { 年齢、性別、学歴 } から選ぶ
  - Random Forest: { 年齢、性別 } から選ぶ
    - 第 $n+1$ 分割を行う際には、{ 学歴、性別 }
- 動機: 予測値同士の相関を弱める
  - 相関を強める要因 (データが多少変わっても、同じような変数を活用する) を排除
  - そこそこの予測力を持つ変数が、強力な予測力を持つ変数の陰に隠れてしまうことを避けられる

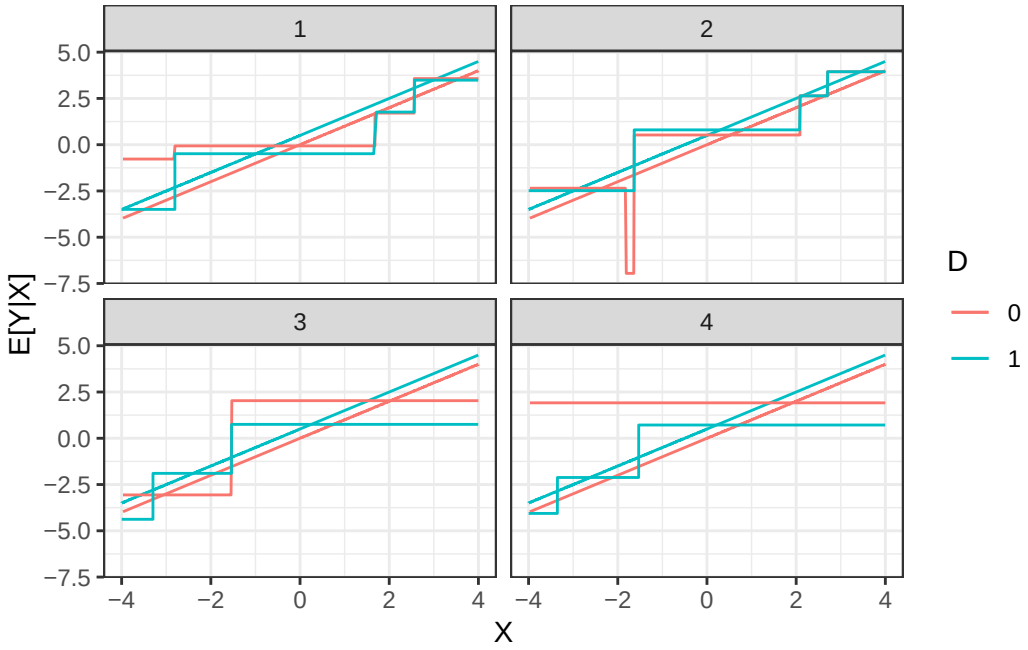
数値例



数値例



数値例



Hyper prameter tuning

- 多くの実戦で、パッケージが提供する default values をそのまま使っている
- Hyper parameter のチューニングについて、議論は存在し、実装されているパッケージもある
  - mlr3tuningspaces
  - grf
- 明確なのは、ブートストラップの回数は多ければ、多いほどよい

実例

```
   nr task_id  learner_id resampling_id iters  regr.rsq
1:  1    Data          OLS       holdout     1 0.8536216
2:  2    Data OptimalTree       holdout     1 0.7876120
3:  3    Data           RF       holdout     1 0.8606409
4:  4    Data   OptimalRF       holdout     1 0.8679677
Hidden columns: resample_result
```

まとめ

- Resampling は現代のデータ分析において、強力な手法
  - モデル評価 (Cross fitting) だけでなく、決定木の予測性能改善 (Bagging/RandomForest) にも有効
  - 伝統的な Inference への Bootstrap の応用も、もちろん重要

Resampling 法の整理