Tree Model for Prediction

川田恵介

Table of contents

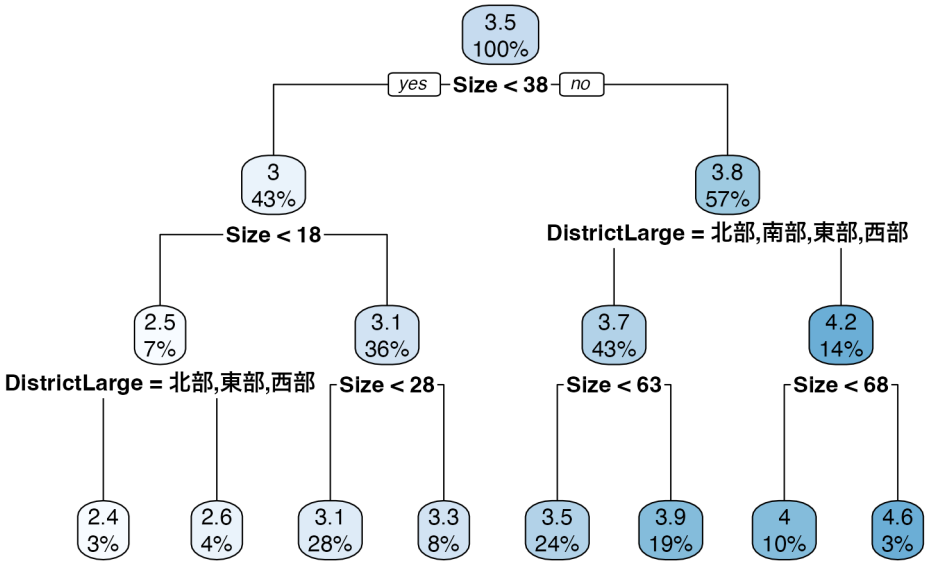
1 Regression Tree

- 一般に線形モデルとは大きく異なるモデルを構築
- サブグループの平均値を予測値とする
  - 伝統的方法: 人間がサブグループを決定
  - 本講義: データがサブグループを決定

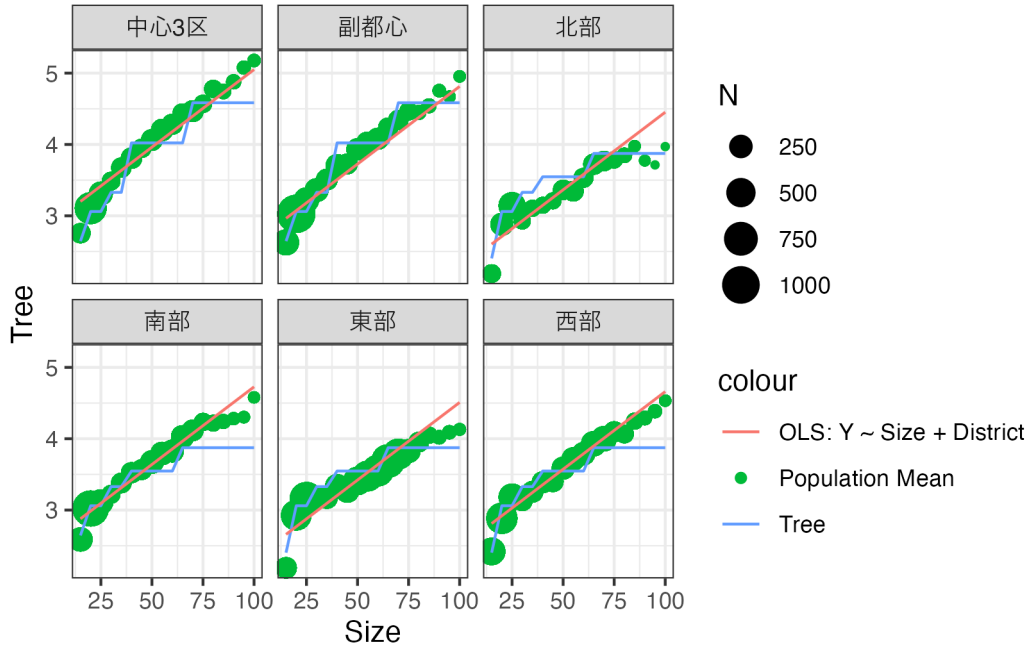
1.1 動機

- Social Outcome の平均値を良く近似するモデルを、“安定して生み出せるアルゴリズム” は、（知る限り）存在しない
  - 複数のアルゴリズムの予測結果を集計することが実践的
- Linear Model とは、異なるモデルを生み出すアルゴリズムも用いることが必要
  - Tree Model は重要な要素

1.2 実例: 価格予測



1.3 実例: VS OLS



1.4 (伝統的) サブグループ法

1. Y, X を指定する
2. **研究者が** X についてサブグループを定義する
3. Y のサブグループ平均を計算する

1.5 データ主導のサブグループ設定

1. Y, X を指定する
2. **データによって** X についてサブグループを定義する
3. Y のサブグループ平均を計算する

1.6 "貪欲な" 予想木アルゴリズム

- 2.1. 分割の停止条件 (最大分割数、サブグループの最小サンプル数など) を設定
- 2.2. データへの適合度が最大（平均二乗誤差が最小）になるように最初の分割を決定
- 2.3. 停止条件に達するまで、分割を繰り返す

1.7 停止条件の設定

- 停止条件さえ設定すれば、予測木はデータに適合するように自動構築できる
  - 停止条件をどのように設定する？
- 停止条件を" 緩く " すれば、分割は永遠と進む
  - 予測モデルが複雑化し、データと完全に適合する
    - ∗ 各サブグループは、１事例のみになるため
  - 一般に過剰適合し、予測性能が悪化

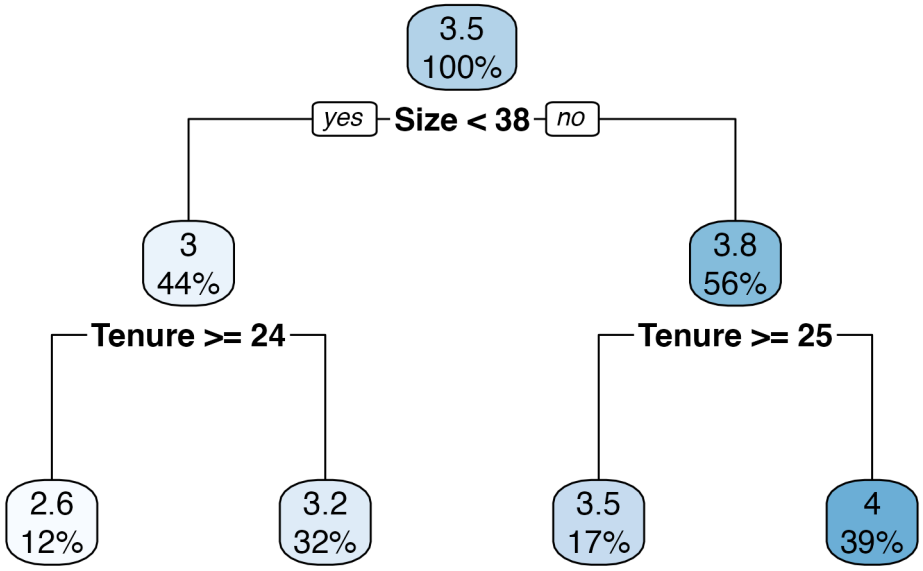
1.8 性質

- "深い木"(最大分割数多い/最小事例数が少ない) を生成すると
-

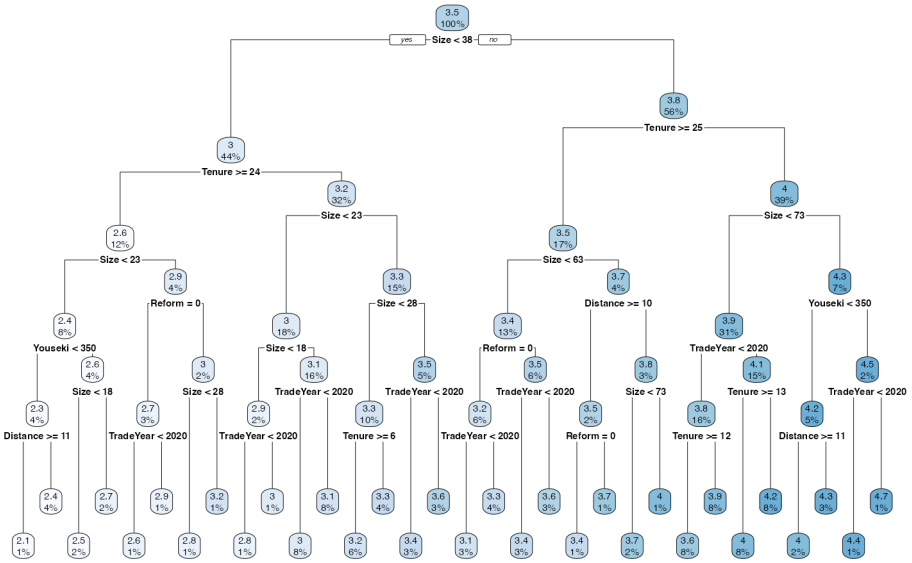
$$Y - g_Y(X) = \underbrace{Y - E[Y|X]}_{Irreducible\ Error}$$

$$+ \underbrace{E[Y|X] - g^*_{Y,\infty}(X)}_{Approximation\ Error \rightarrow 減少} + \underbrace{g^*_{Y,\infty}(X) - g_Y(X)}_{Estimation\ Error \rightarrow 増加}$$

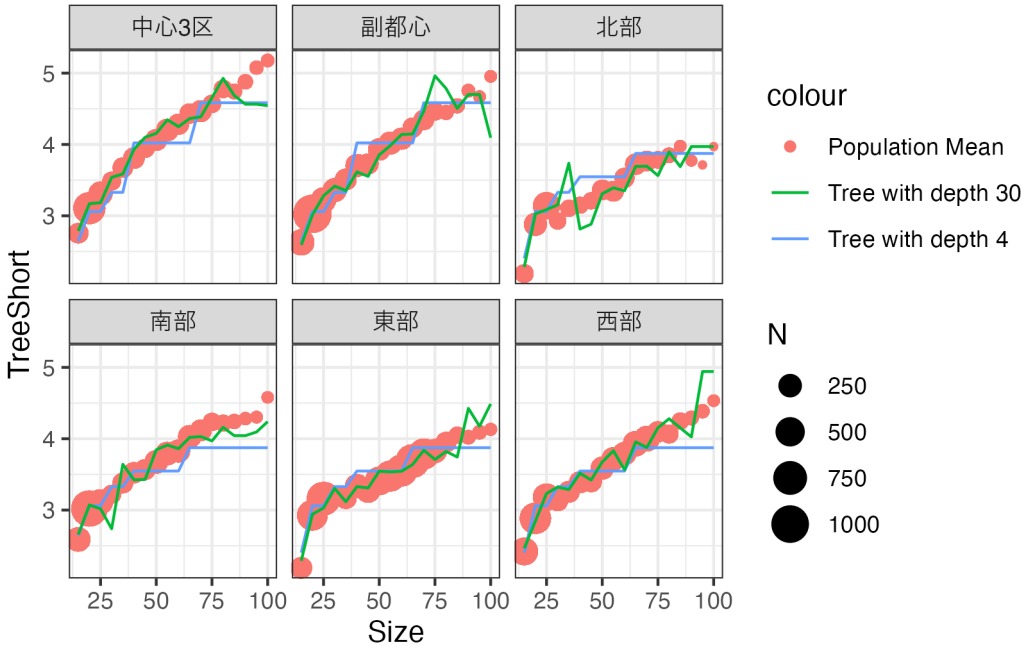
1.9 実例: 浅い木



1.10 実例: 深い木



1.11 実例: VS



1.12 まとめ

- データ主導の変数選択を導入
  - 停止条件の設定に強く依存
- 対策としては
  - LASSO と同様に、複雑なモデル (巨大な決定木) を推定し、単純化する (剪定 ISL Chap 8.1 参照)
  - 本講義では、モデル集計を紹介
    - 上手くいくことが多いため

2 Bootstrap Model Averaging

- データ分析の基本アイディア: 事例を集計することで、母集団の特徴を捉える
- 予測モデル自体も集計できる
  - シンプルかつ強力な戦略

2.1 予測木への応用

- 分割回数を増やすと
  - 母平均が大きく乖離しているサブグループへの分割が期待できる
  - サブグループの事例数が増え、データに含まれるハズレ値の影響を強く受けやすい

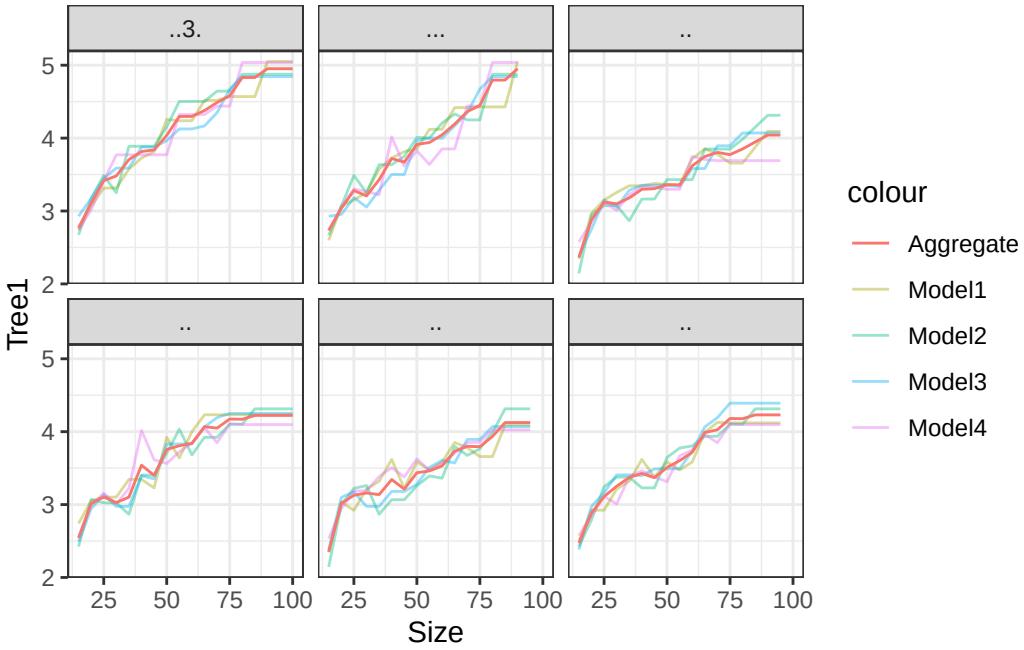
2.2 解決策

- 問題: 平均値に近づけたいのに、複雑なモデルは、(平均から大きく乖離した) ハズレ値の影響を受けやすい
- 解決: 大量の提案
  - 代表的な戦略は、モデルの適切な単純化
    - **モデル集計**

2.3 Bootstrap model averaging

- 深い決定木は、外れ値に大きな影響を受ける可能性がある
  - 外れ**予測値**が生成される可能性
- **複製データ** から大量の決定木を推定し、平均をとる
  - 外れ予測値の影響を緩和する
  - $\simeq$ 分散投資で、外れイベントの影響を緩和

2.4 例: ４つのモデルの集計



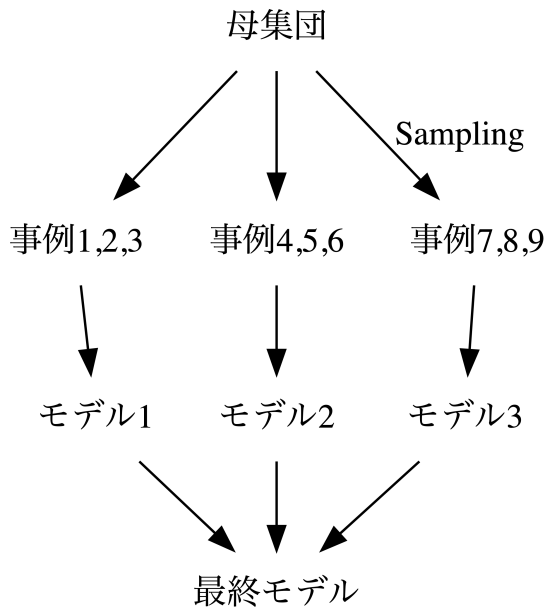
2.5 擬似的なモデル複製

- 独立して抽出したデータから得られる予測モデルを集計できれば、性能は**必ず**改善する
  - 現実には不可能
- 擬似的に行う
  - ブートストラップの活用

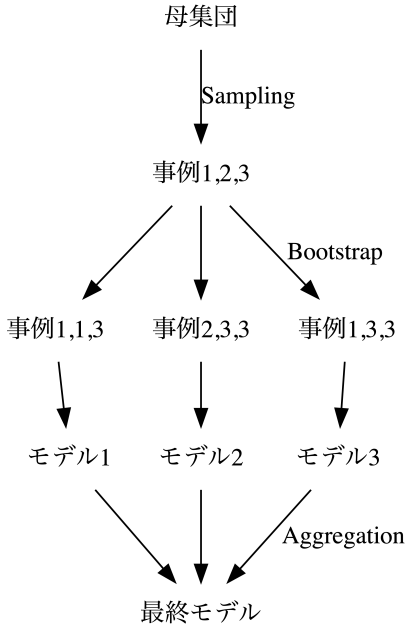
2.6 ブートストラップ

- データと同じ事例数の複製データを作成
  - 復元抽出 (被りありの抽選) を行う

2.7 理想のモデル集計



2.8 Bootstrap Model Averaging



2.9 De-correlation

- ブートストラップでは、複製データ間で同じ事例が使用されうる
  - データの特徴間に相関が生じる
  - 同じような予測値を集計したとしても、あまり予測精度は改善しない
- 事例数が限られている場合、強力な予測力をもつ変数のみが使用され、そこそこの予測力変数が使用されない
  - 分割に使用する変数もランダムに決める

2.10 Random Forest

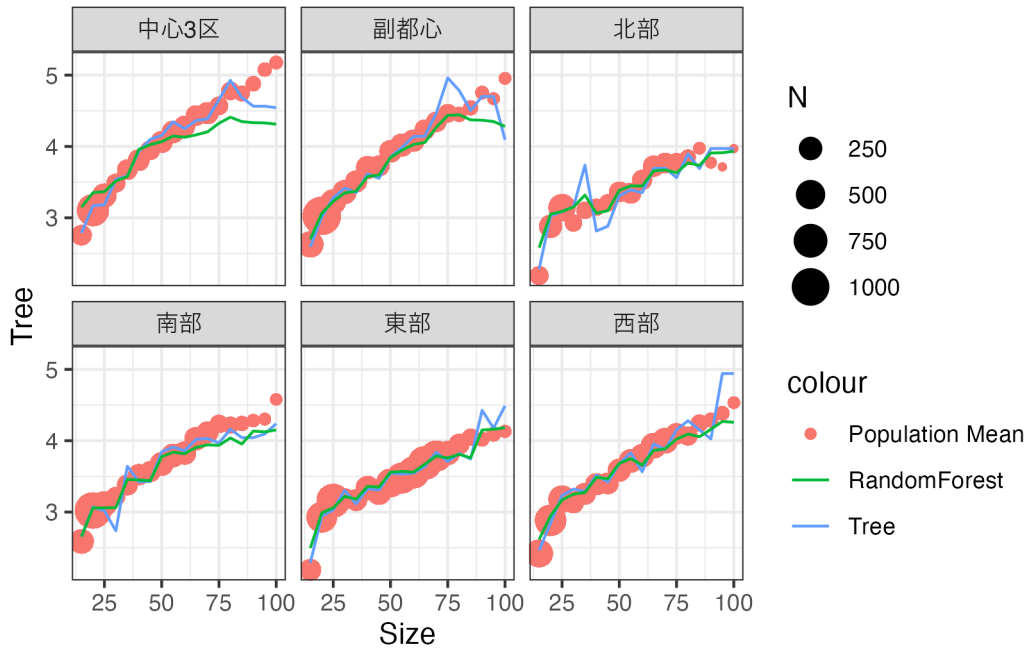
1. $\{Y, X\}$ を決める
2. ブートストラップにより、データを複製 (可能な限り多く、ranger の default は 500)
3. 各複製データについて、Regression Tree を推定

- RandomForest では、分割時に使用できる変数はランダムに選ぶ

2.11 性質

- 深すぎる Regression Tree を集計すると
- $$Y - g_Y(X) = \underbrace{Y - E[Y|X]}_{Irreducible\ Error} + \underbrace{E[Y|X] - g^*_{Y,\infty}(X)}_{Approximation\ Error \approx 0(注)} + \underbrace{g^*_{Y,\infty}(X) - g_Y(X)}_{Estimation\ Error = 大 \to 減少}$$
- 注: Tree 系のアルゴリズムについての理論的性質 (大表本性質) は、現状でも盛んに研究されている
  - Klusowski and Tian (2024) など

2.12 実例: VS Tree



3 Boosting

- 代替的なモデル集計方法
  - こちらも大人気の手法
- シンプルすぎるモデルを複雑にしていく

3.1 Algorithm: アイディア

1. X, Y を指定
2. Y を予測する” 浅い木” を推定し、予測誤差 $R = Y - g_0(X)$ を算出
3. R を予測する” 浅い木” を推定し、予測モデル $g(X)$, 予測誤差 $R = Y - g(X)$ を更新
4. 3 を一定回数繰り返し、最終予測モデル $g(X)$ を算出

3.2 性質

- 浅すぎると Regression Tree からスタートするので

- $$Y - g_Y(X) = \underbrace{Y - E[Y|X]}_{Irreducible\ Error} + \underbrace{E[Y|X] - g^*_{Y,\infty}(X)}_{Approximation=大→減少} + \underbrace{g^*_{Y,\infty}(X) - g_Y(X)}_{Estimation\ Error=小→増加?}$$

3.3 Tuning Parameter

- 繰り返す回数 = 多くし過ぎると、データに完全に (過剰) 適合する
  - Random Forest との大きな違い
- よく用いられる Tuninging 方法は、Early Stopping
  - データの一部を検証用に分割し、モデルの検証データへの当てはまりが低下したら、停止

3.4 “ゆっくり学ぶ”

- 一回でデータへの当てはまりを大きく改善すると、過剰適合する可能性が高まる
- “学習速度” を落とす
  - Regression Tree の分割回数を減らす
  - 予測モデルの更新速度を落とす
    - $g(X) = g(X) + \lambda g_0(X)$

3.5 まとめ

- Regression Tree は、Linear Model の有力な代替案
  - Stacking における重要な構成要素
  - Linear Model ほど、Data Clearning が必要ない
    - とりあえず RandomForest か Boosting を試してみる（人が企業では多いそうです）
- まだまだ理論的によくわかっていないことが多い (そうです)
  - Causal ML (Chap 9) を参照

Reference

Klusowski, Jason M, and Peter M Tian. 2024. “Large Scale Prediction with Decision Trees.” *Journal of the American Statistical Association* 119 (545): 525–37.