Tree Model for Prediction

川田恵介

2025-06-10

1 Regression Tree

1.1 動機

- 一般に、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 (伝統的)サブグループ法

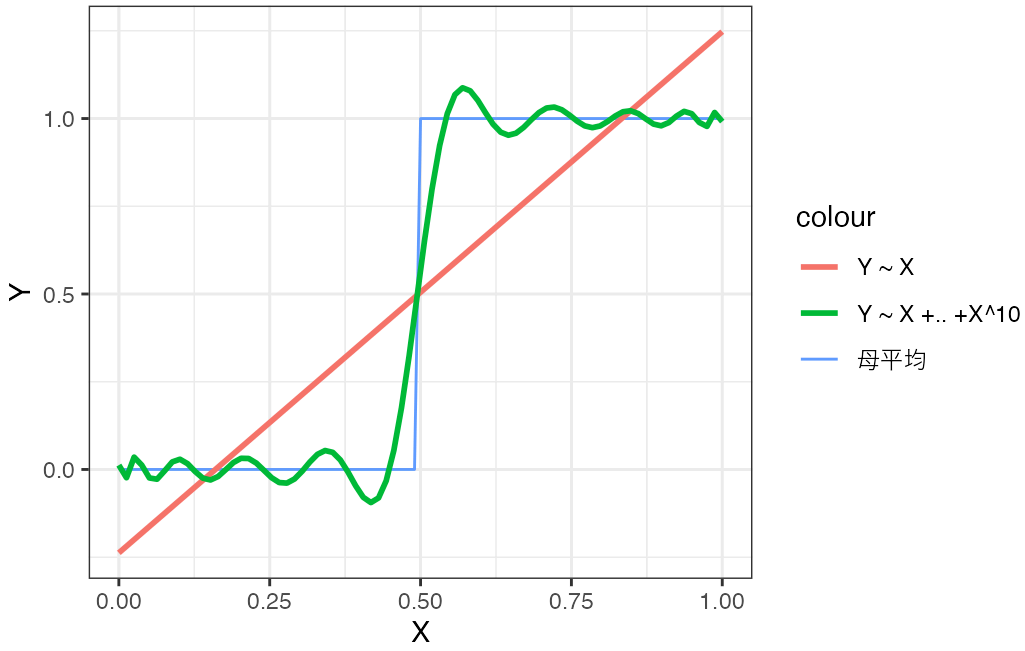
1. Y, X を指定する
2. **研究者が** X についてサブグループを定義する
3. Y のサブグループ平均を計算する

1.3 回帰木法

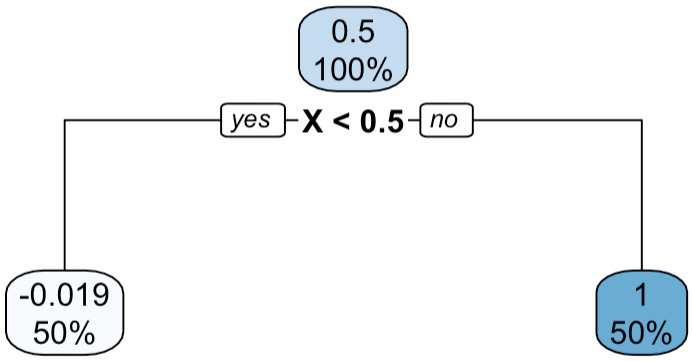
1. Y, X を指定する
2. **データによって** X についてサブグループを定義する
3. Y のサブグループ平均を計算する

1.4 線型モデルの弱点: 数値例

- Kink する母平均を近似するのが難しい

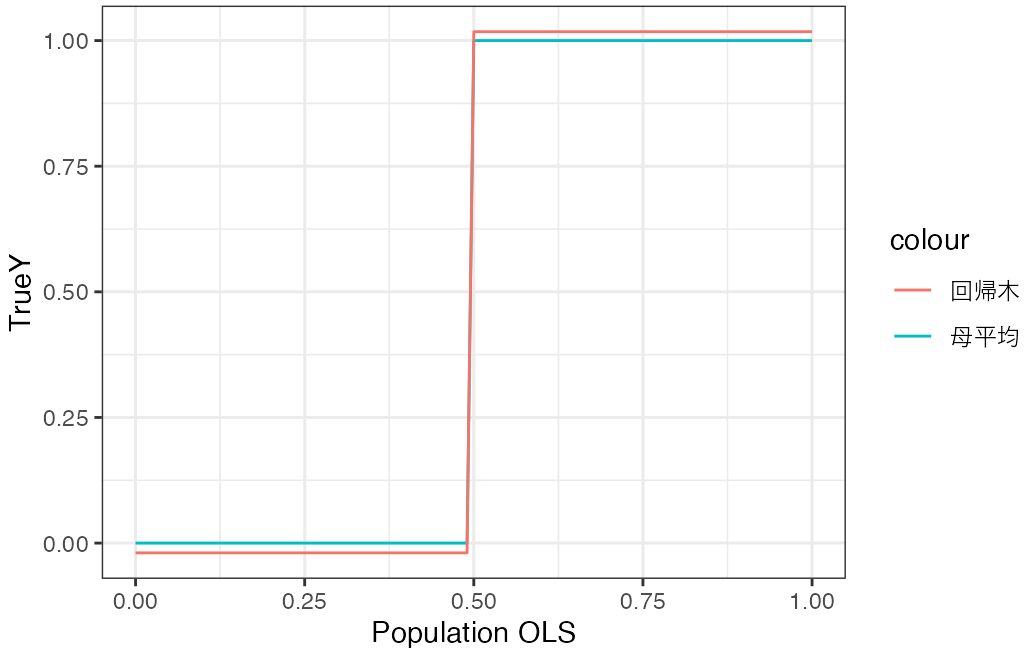


1.5 数値例: 回帰木の推定



1.6 数値例: 回帰木の推定

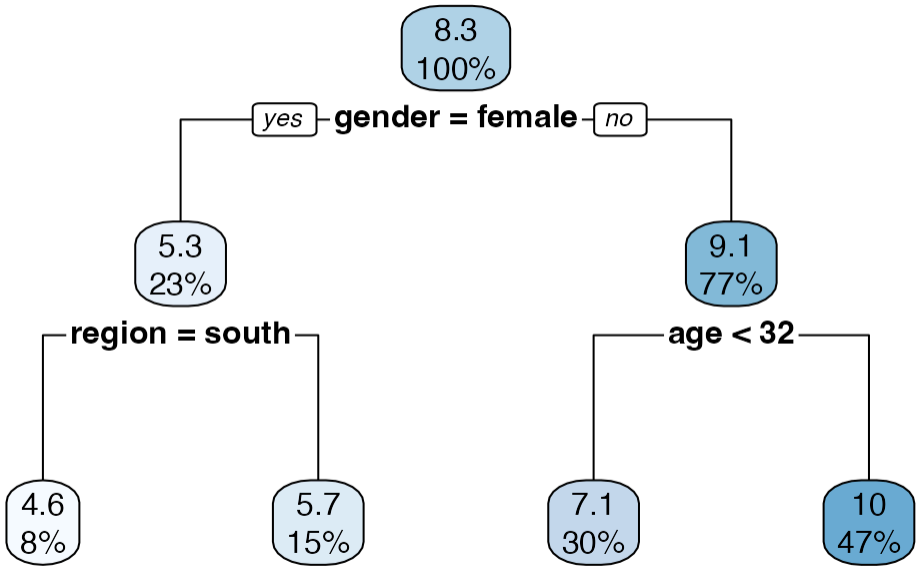
- サブサンプル平均が予測値なので、kink に強い



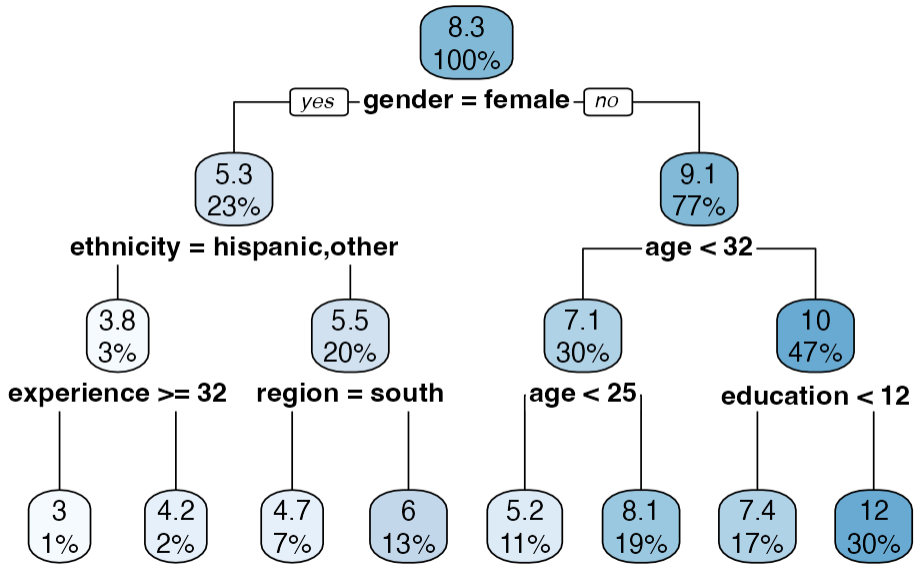
1.7 性質

- “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、サブグループの事例数が少なくなる
- $$Y - g_Y(X) = \underbrace{Y - E[Y \mid X]}_{Irreducible\ Error}$$
$$+ \underbrace{E[Y \mid X] - g^*_{Y,\infty}(X)}_{Approximation\ Error \rightarrow 減少} + \underbrace{g^*_{Y,\infty}(X) - g_Y(X)}_{Estimation\ Error \rightarrow 増加}$$

1.8 実例: 浅い木 (100 事例)



1.9 実例: 深い木



1.10 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

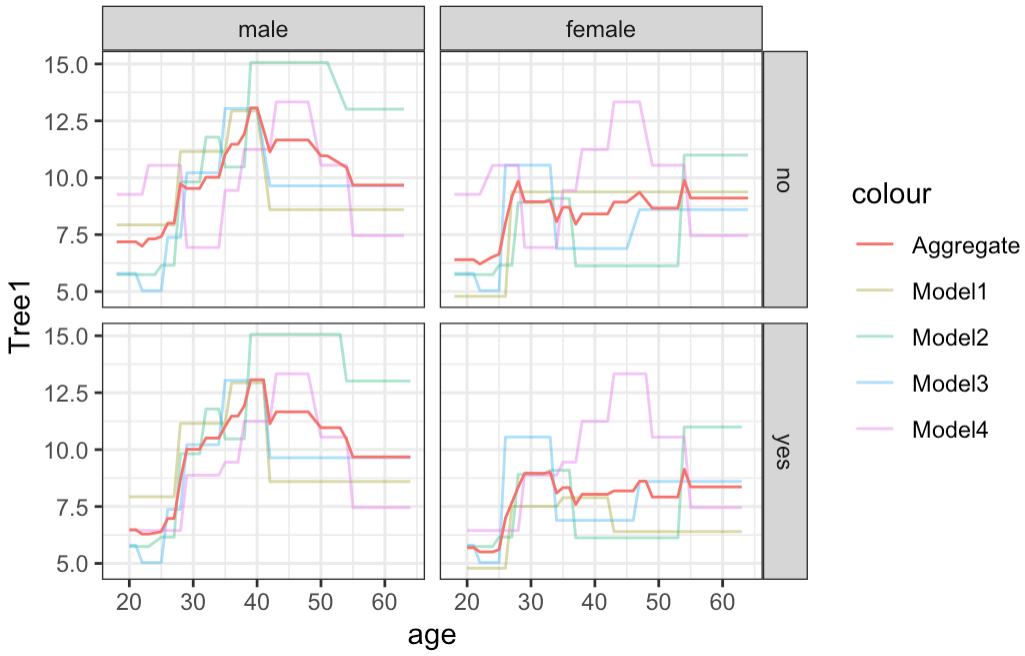
2.1 集計

- 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

2.2 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

2.3 例: wage ~ gender + married



2.4 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

2.5 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

2.6 シンプルな例

- Training データ $= [5, 6, 100]$
- 復元抽出により、同じ数(３)の事例をランダムに選ぶ
  - 複製データ 1 $= [6, 6, 100] =$ の平均値 37.3
  - 複製データ 2 $= [6, 6, 5] =$ の平均値 5.7
  - 複製データ 3 $= [5, 5, 5] =$ の平均値 5
- 最終予測 $= 16$

- ハズレ値(“100”)を反映しない予測も活用され、より頑強?

2.7 回帰木への応用

- Training データの複製を行い、各データに”回帰木”を当てはめ、予測モデルを得る
- 予測値の平均を最終予測とする
  - ハズレ値の影響を緩和できる
  - 注: 平均値や OLS ではあまり意味がない

2.8 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

2.9 Random Forest

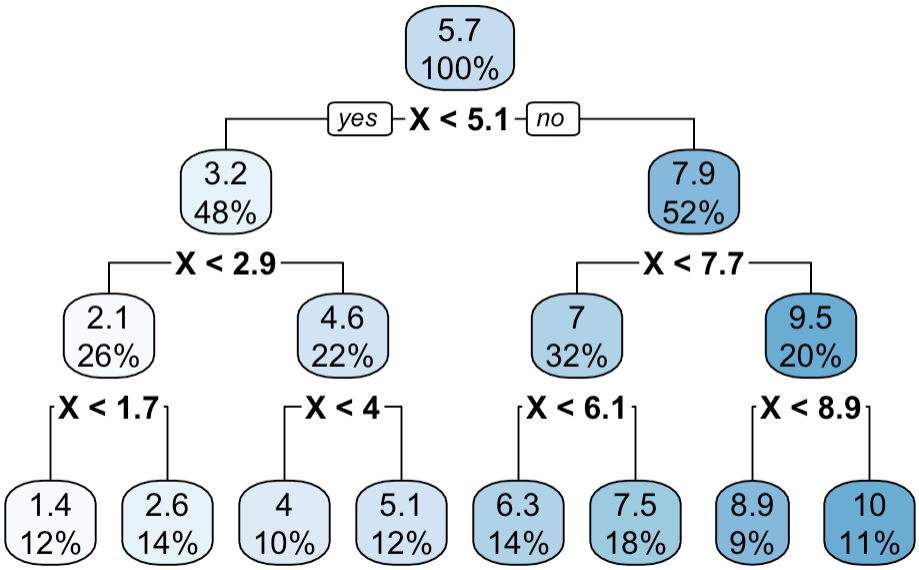
1. $\{Y, X\}$ を決める
2. ブートストラップにより、データを複製 (可能な限り多く、 ranger の default は 500)
3. 各複製データについて、Regression Tree を推定

- RandomForest では、分割時に使用できる変数はランダムに選ぶ

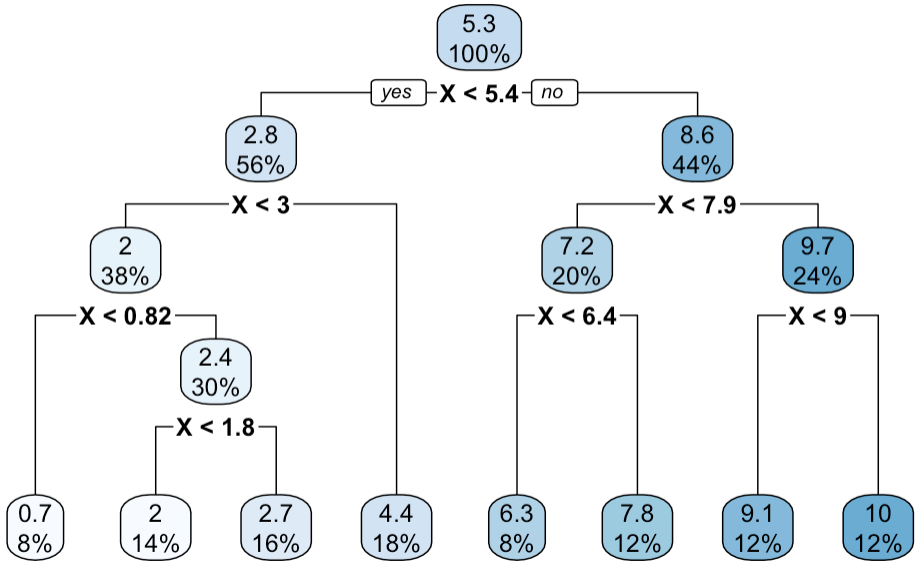
2.10 数値例

- $D \in \{0, 1\} \mid \Pr[D = 1] = \Pr[D = 0] = 0.5$
- $X \in unif(0, 10)$
- $Y = D + X + \underbrace{e}_{\sim N(0,0.1)}$

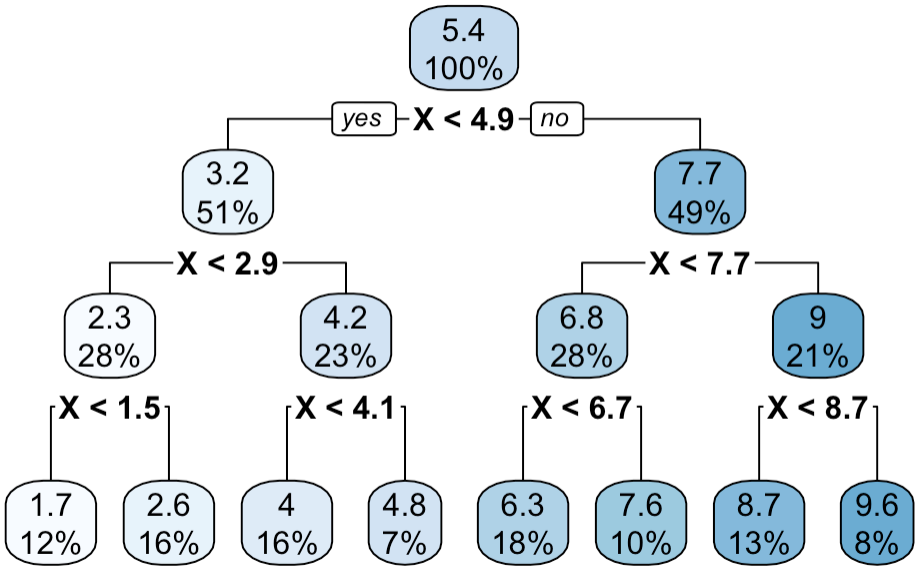
2.11 数値例: Model 1



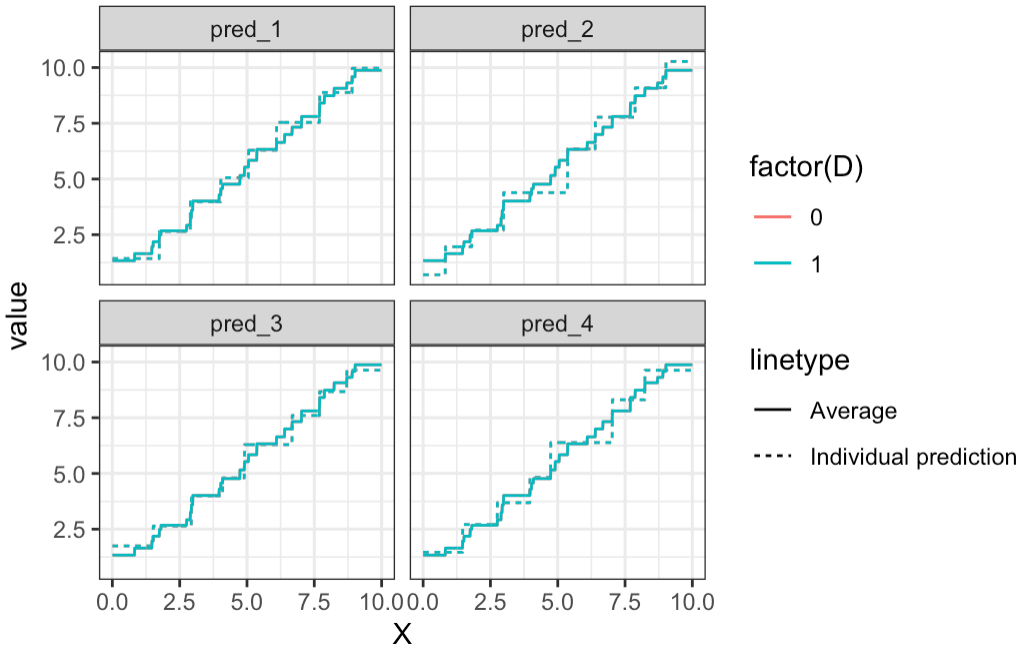
2.12 数値例: Model 2



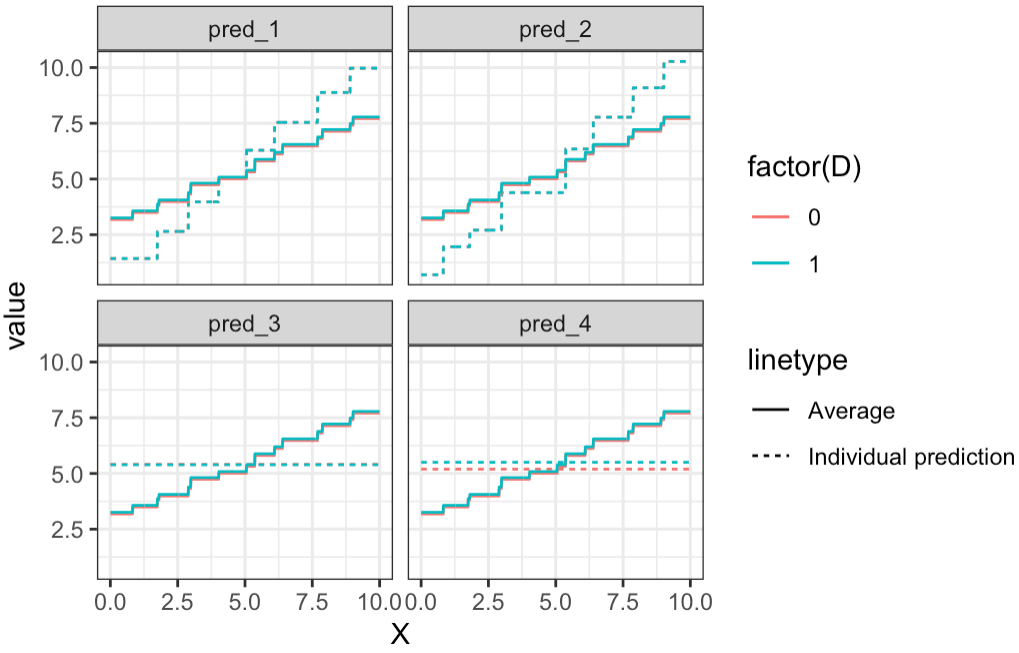
2.13 数値例: Model 3



2.14 数値例: 集計



2.15 数値例: Random Forest



3 Boosting

3.1 Boosting

- 代替的な回帰木モデルの集計方法
  - 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

3.2 推定方法

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

3.3 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

3.6 Reference

Bibliography