決定木のアンサンブル法
・ランダムフォレスト
・勾配ブースティング決定木

Decision Tree Ensemble
-Random Forest
-Gradient Boosting Decision Tree

大阪府立大学 工学研究科
清水　悠生

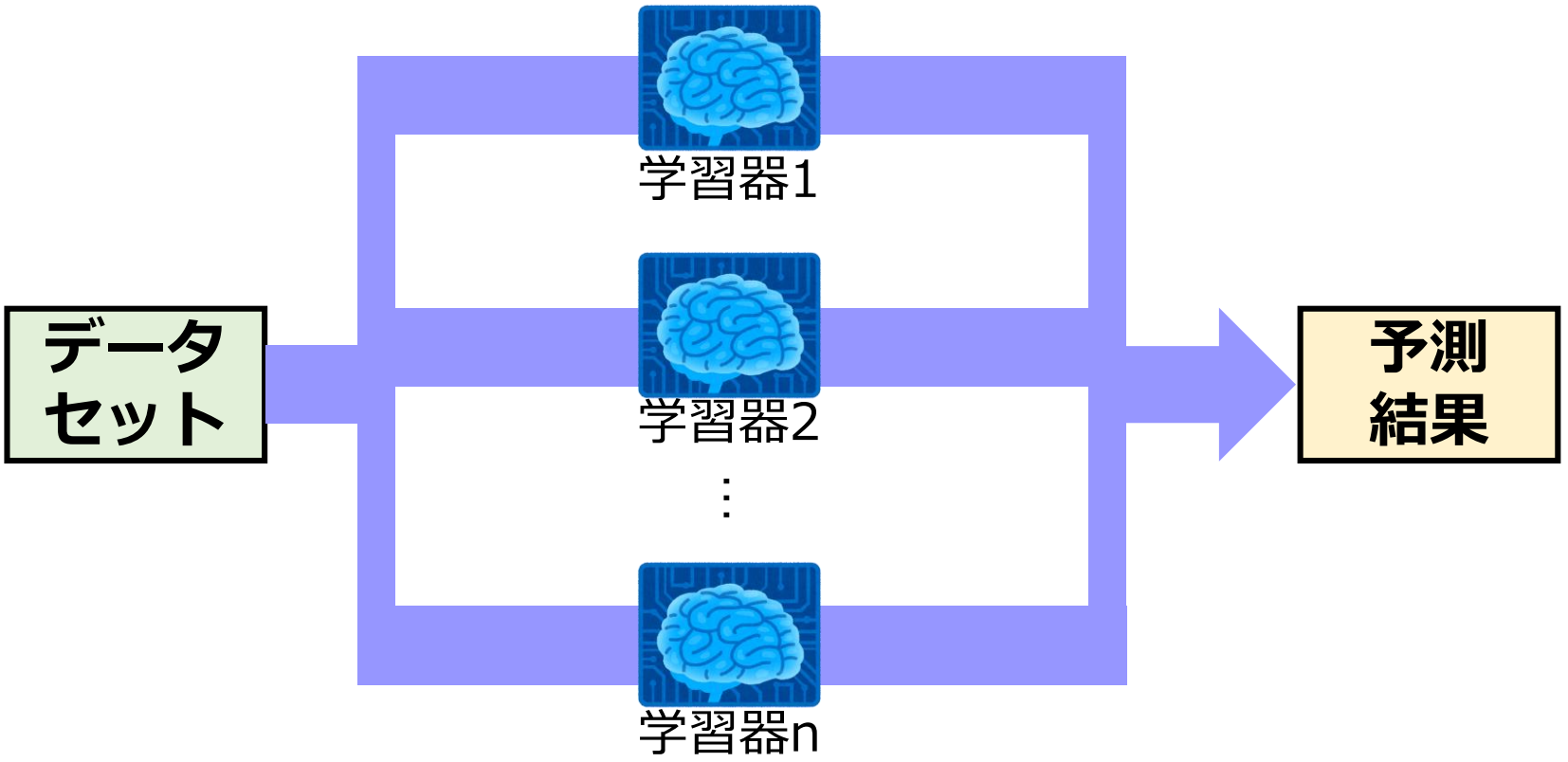
はじめに

- ✓ 本記事は決定木↓の理解を前提に書いています
- ✓ https://yuyumoyuyu.com/2021/02/07/decisiontree/

- ✓ 具体的な解析結果やPythonプログラムは GitHubを参照してください
- ✓ https://github.com/yshimizu12/EnsembleDecisionTree

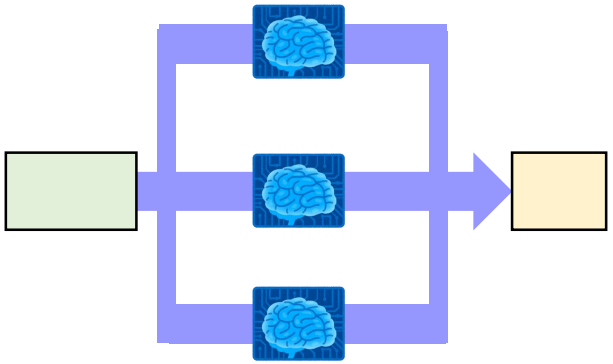
アンサンブル学習とは

✓ 同一のデータセットから複数の学習器を学習し
その結果を統合して汎化能力を向上する方法
⇒**アンサンブル学習**

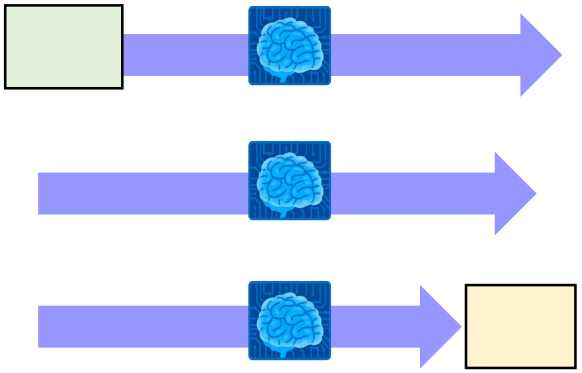


2種類の決定木アンサンブル学習

- ✓ 決定木は汎化性能が低いため，
  アンサンブル学習を利用することがほとんど
- ✓ 決定木アンサンブル学習は下記の2つが代表的
  - ランダムフォレスト

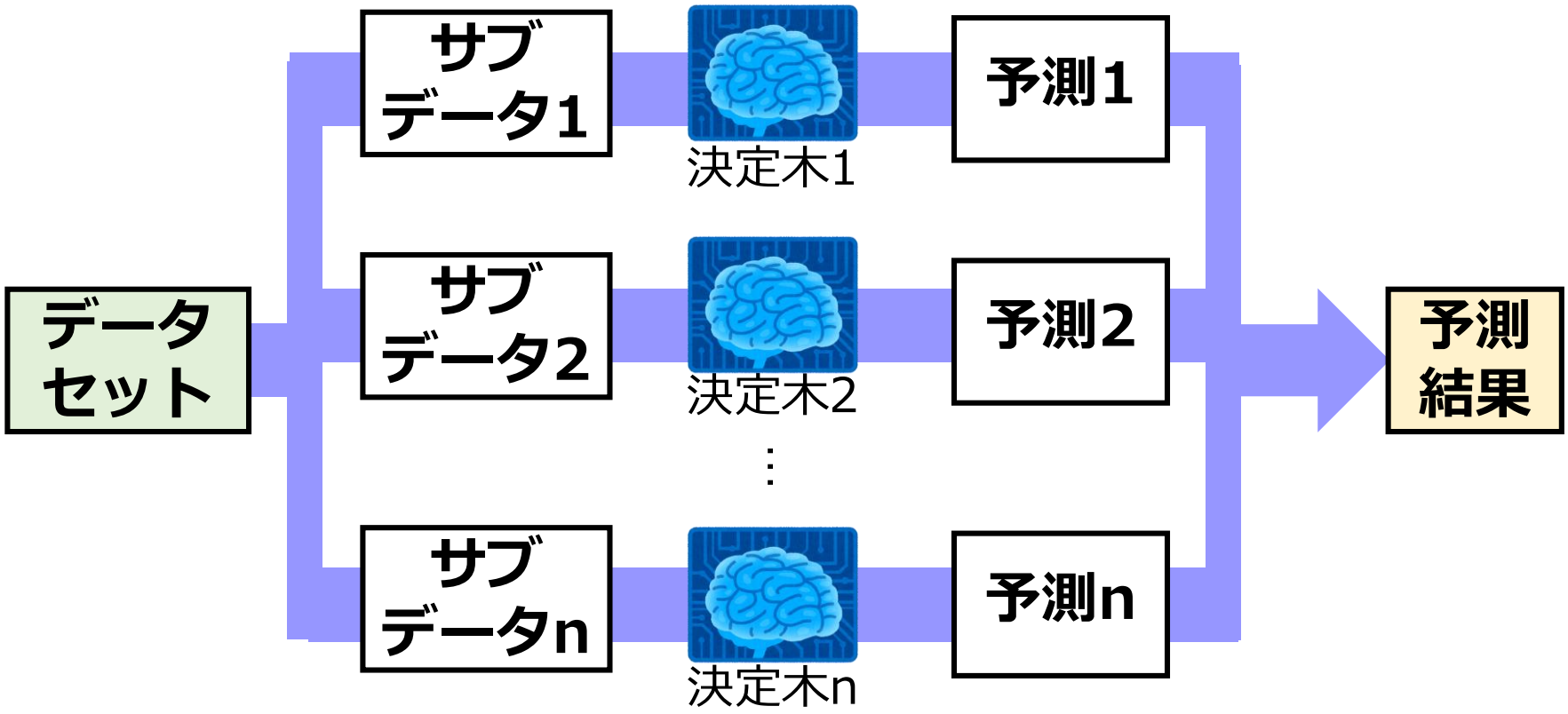


  - 勾配ブースティング決定木



ランダムフォレスト

- ✓ あるデータセットからサブデータセットを複数作成し
  それぞれを使って独立した決定木を学習し
  学習結果を統合するモデル⇒**ランダムフォレスト**

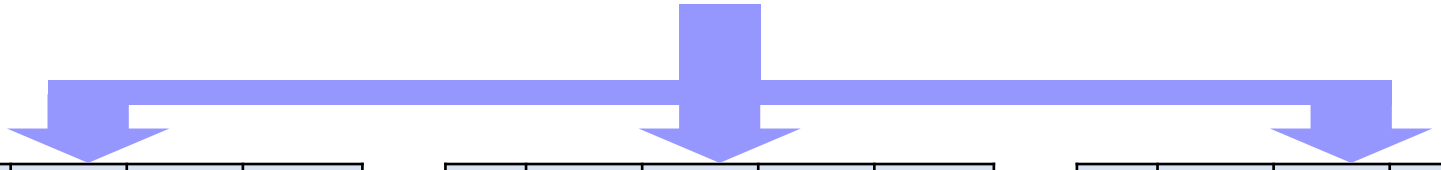


サブデータセットの作り方

- ✓ データセットから重複ありでデータを抽出し，
  データセットと同じサイズのサブデータをn個生成する
- ✓ **ブートストラップサンプリング**と呼ぶ

データセット

No.	寸法1	寸法2	寸法3	トルク
1	5 mm	6 mm	2 mm	3 Nm
2	3 mm	7 mm	1 mm	4 Nm
3	2 mm	5 mm	1 mm	2 Nm
4	4 mm	6 mm	2 mm	5 Nm



No.	寸法1	寸法2	寸法3	トルク
1	5	6	2	3
2	3	7	1	4
1	5	6	2	3
3	2	5	1	2

サブデータセット1

No.	寸法1	寸法2	寸法3	トルク
4	4	6	2	5
4	4	6	2	5
1	5	6	2	3
1	5	6	2	3

サブデータセット2

No.	寸法1	寸法2	寸法3	トルク
3	2	5	1	2
2	3	7	1	4
4	4	6	2	5
3	2	5	1	2

サブデータセット3

決定木で使用する特徴量選択

- ✓ ランダムフォレストが高い汎化性能を有するには決定木同士が異なる構造を持つほうがよい
- ✓ **特徴量をランダムに選択**して学習を行う

サブデータセット1

No.	寸法1	寸法2	寸法3	トルク
1	5	6	2	3
2	3	7	1	4
1	5	6	2	3
3	2	5	1	2

No.	寸法2	寸法3	トルク
1	6	2	3
2	7	1	4
1	6	2	3
3	5	1	2

サブデータセット2

No.	寸法1	寸法2	寸法3	トルク
4	4	6	2	5
4	4	6	2	5
1	5	6	2	3
1	5	6	2	3

No.	寸法1	寸法3	トルク
4	4	2	5
4	4	2	5
1	5	2	3
1	5	2	3

サブデータセット3

No.	寸法1	寸法2	寸法3	トルク
3	2	5	1	2
2	3	7	1	4
4	4	6	2	5
3	2	5	1	2

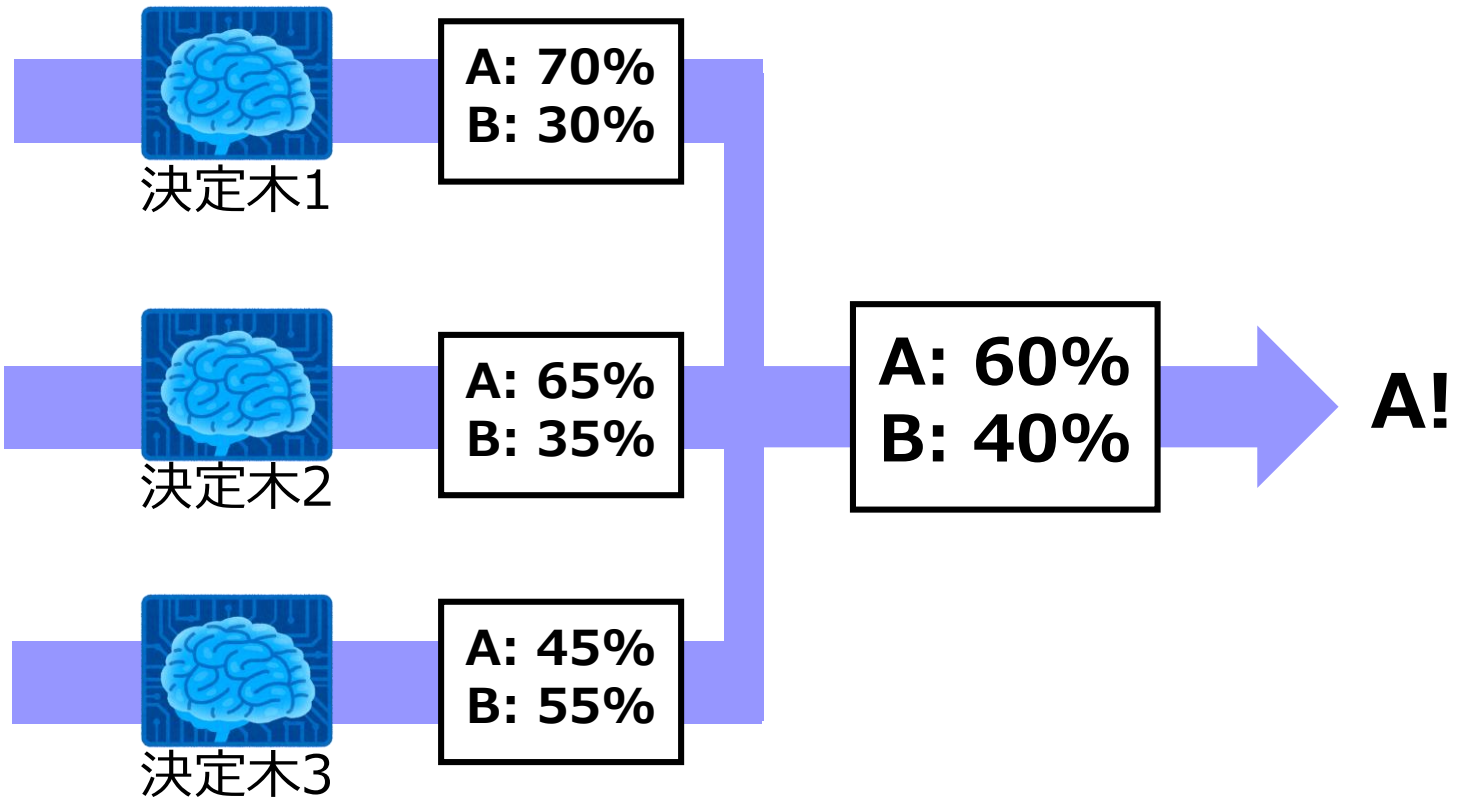
No.	寸法1	寸法2	トルク
3	2	5	2
2	3	7	4
4	4	6	5
3	2	5	2

各決定木の結果の統合

✓ 各決定木の出力を統合する方法は下記の通り

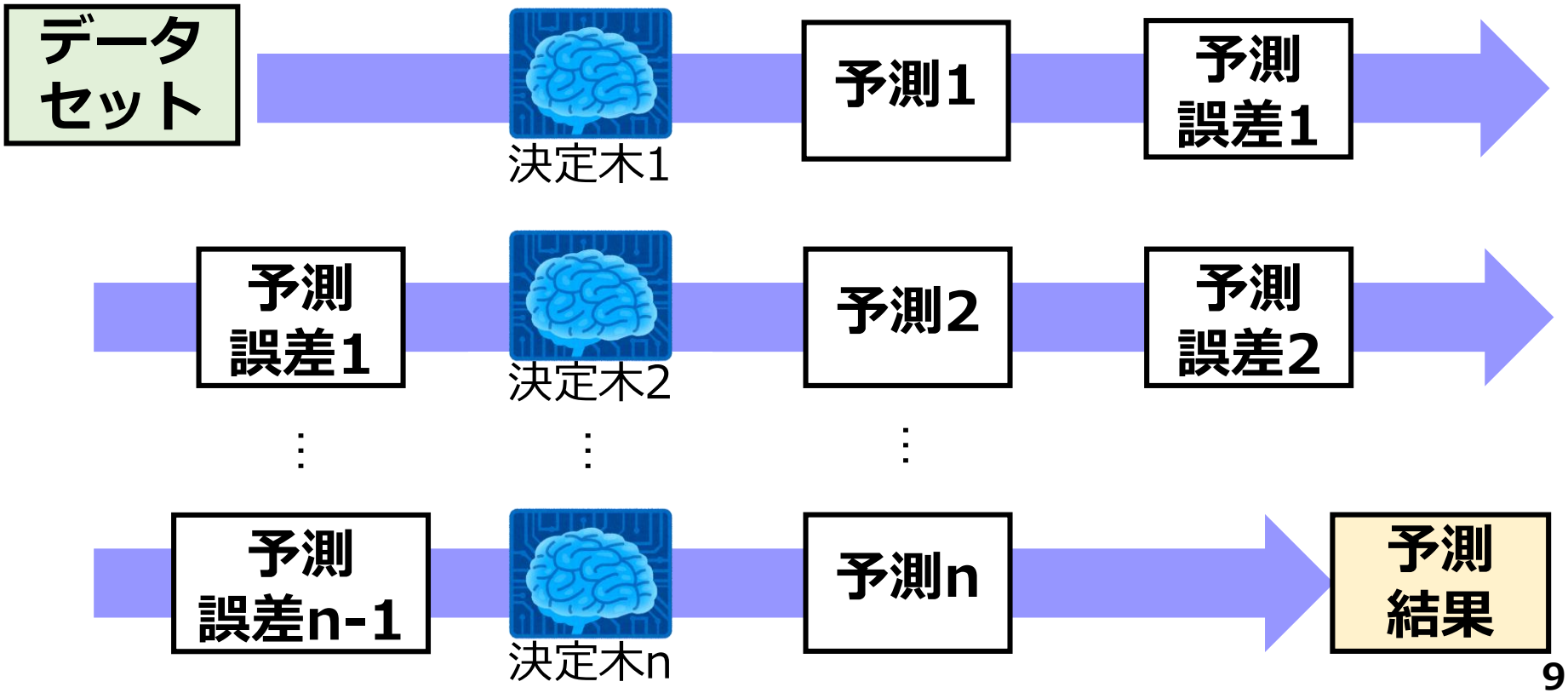
- 回帰：各決定木の平均値
- 分類：各決定木の出力ラベルの確率平均により決定

2クラス分類の例



勾配ブースティング決定木

- ✓ 構築した決定木の損失関数の結果から
  予測誤差を減らすように次々と決定木を構築・学習し
  学習結果を統合するモデル⇒**勾配ブースティング決定木**

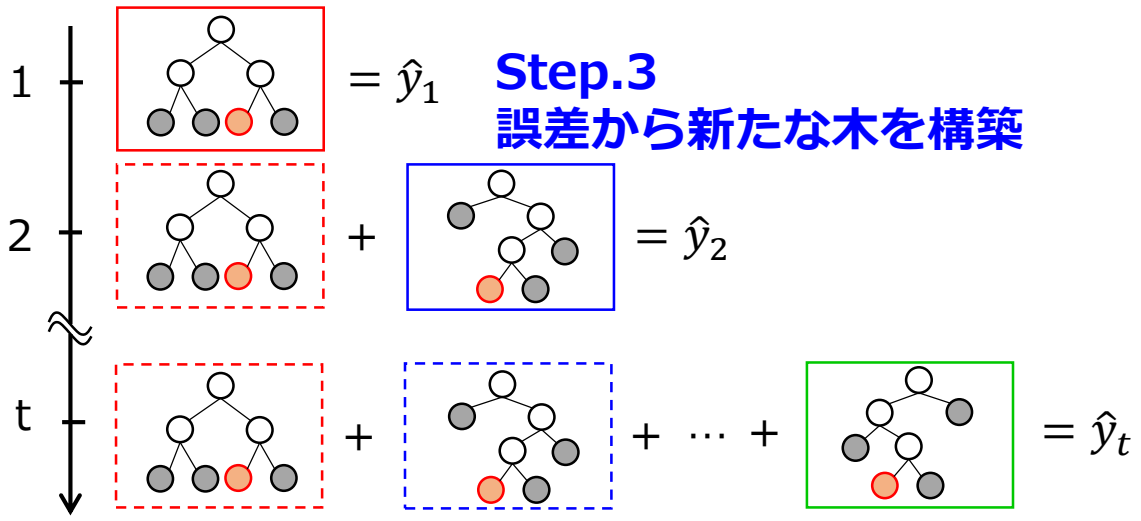


勾配ブースティング決定木のアルゴリズム

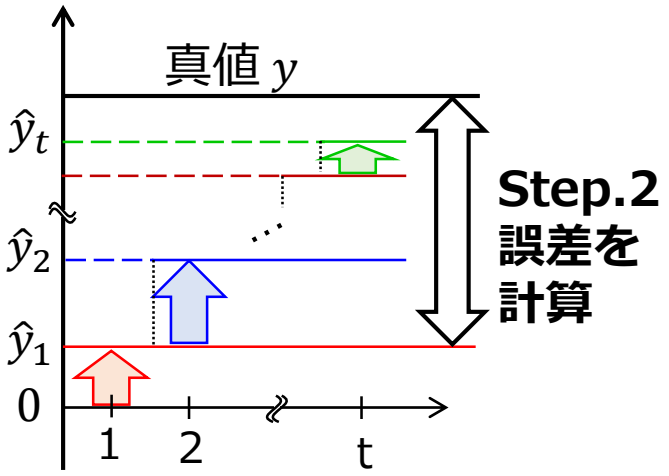
✓ 勾配ブースティング決定木のアルゴリズムは下記の通り

1. 最初の決定木を構築
2. 真値に対する誤差を計算（実際は誤差関数の勾配）
3. 誤差を利用して，誤差が小さくなるよう木を構築
4. 2,3を繰り返す

Step.1 決定木を構築



Step.3
誤差から新たな木を構築



Step.2
誤差を計算

よく使われる手法

- ✓ Pythonでは勾配ブースティングの派生として
  高精度かつ学習時間が短い下記がよく用いられる
  - XGBoost
  - LightGBM

- ✓ 実装例はこちら↓
- ✓ https://github.com/yshimizu12/EnsembleDecisionTree