9. アンサンブル学習

- アンサンブル学習とは
  - 分類器を複数組み合わせ、それらの結果を統合することで個々の分類器よりも性能を向上させる方法
- アイディア
  - 訓練例集合から全く独立に L 個の分類器 (誤り率 ε, 誤りは独立) を作成

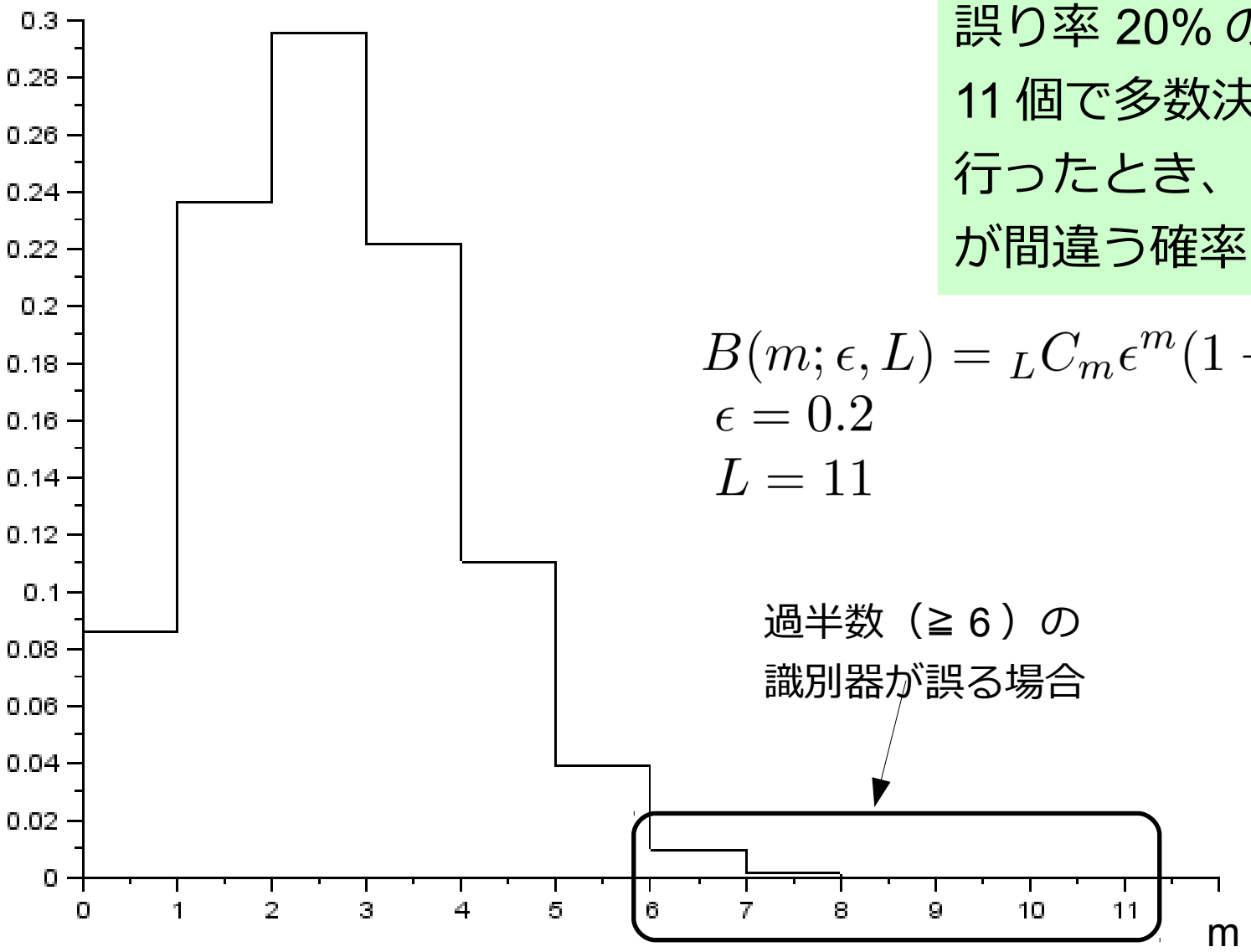
    $\rightarrow$ m 個の分類器が誤る確率は二項分布 $B(m;\epsilon,L)$

    $\rightarrow$ $\epsilon < 0.5$ のとき、$m > L/2$ となる B は小さい値

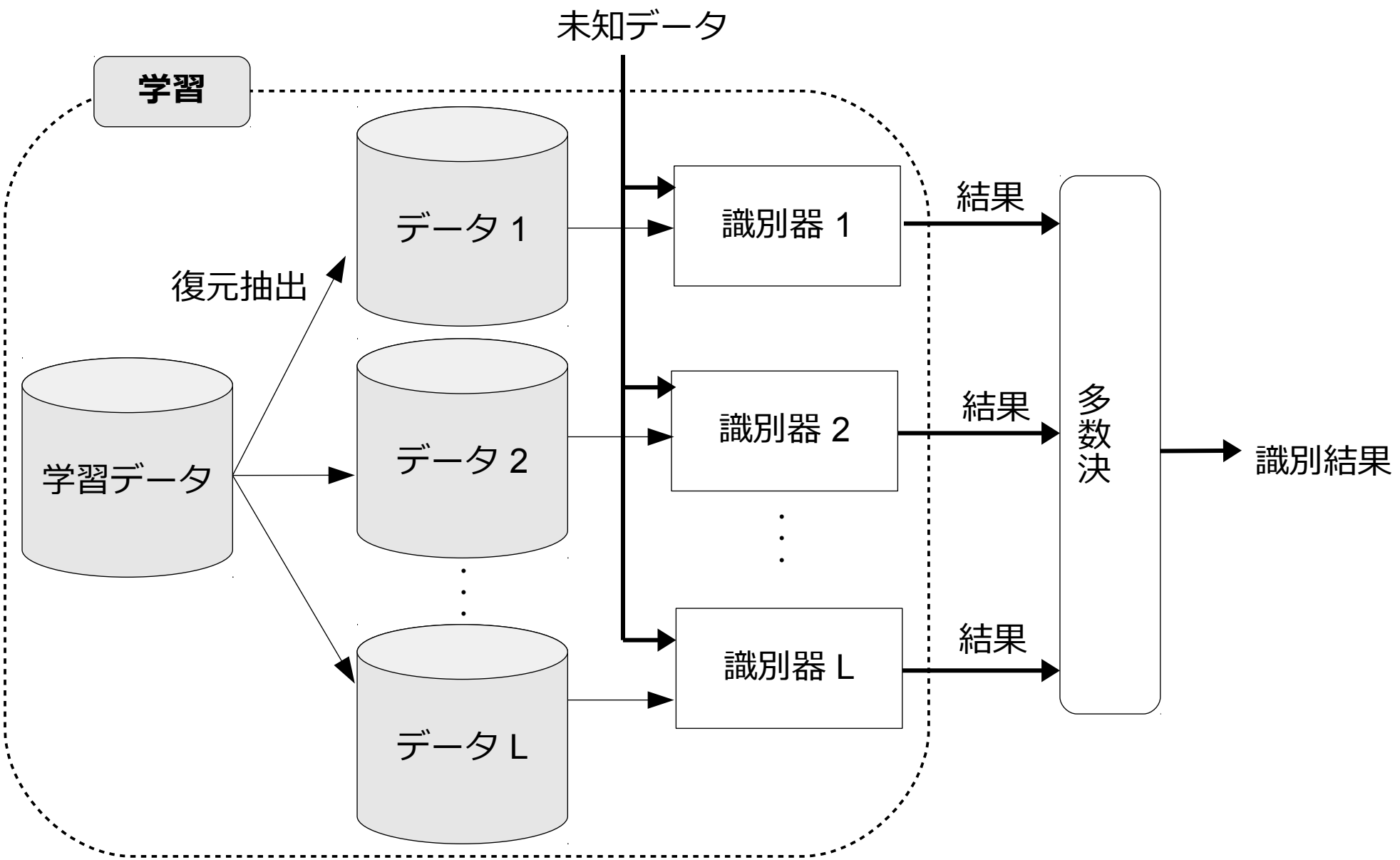
9.1 なぜ性能が向上するのか

誤り率 20% の分類器 11 個で多数決統合を行ったとき、 6 個以上が間違う確率は 1.2%

$$B(m;\epsilon,L) = {}_L C_m \epsilon^m (1-\epsilon)^{L-m}$$
$$\epsilon = 0.2$$
$$L = 11$$



9.2 バギング


未知データ
学習
データ 1
復元抽出
学習データ
データ 2
データ L
識別器 1
識別器 2
識別器 L
結果
結果
結果
多数決
識別結果

9.2 バギング

- 特徴
  - 訓練例から復元抽出することで、元のデータと同じサイズの独立なデータ集合を作成する。

    n 回行って、あるデータが抽出されない確率：$(1-\frac{1}{n})^n$

    n→∞ で
    約 0.368

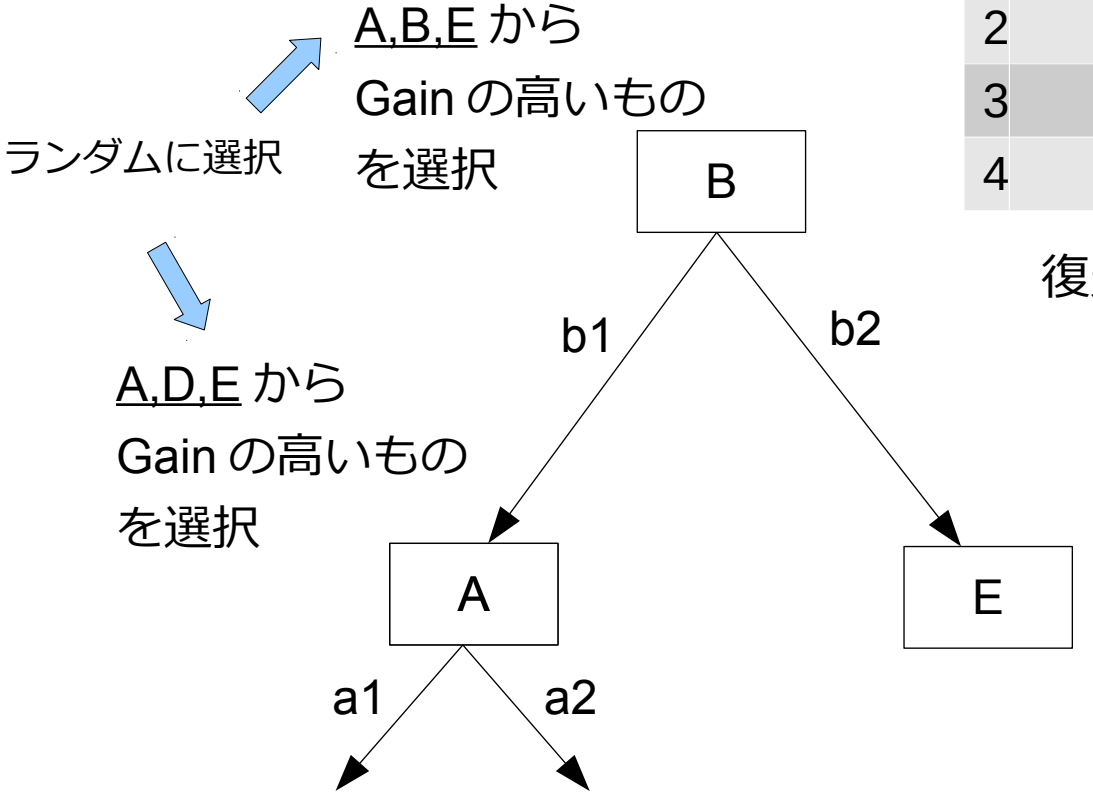
  - 各々のデータ対して同じアルゴリズムで分類器を作成する
    - アルゴリズムは不安定 (学習データの違いに敏感) な方がよい

      例) 枝刈りをしない決定木
  - 結果の統合は多数決

9.3 ランダムフォレスト

	A	B	C	D	E	class
1						
2						
3						
4						

復元抽出された学習データ

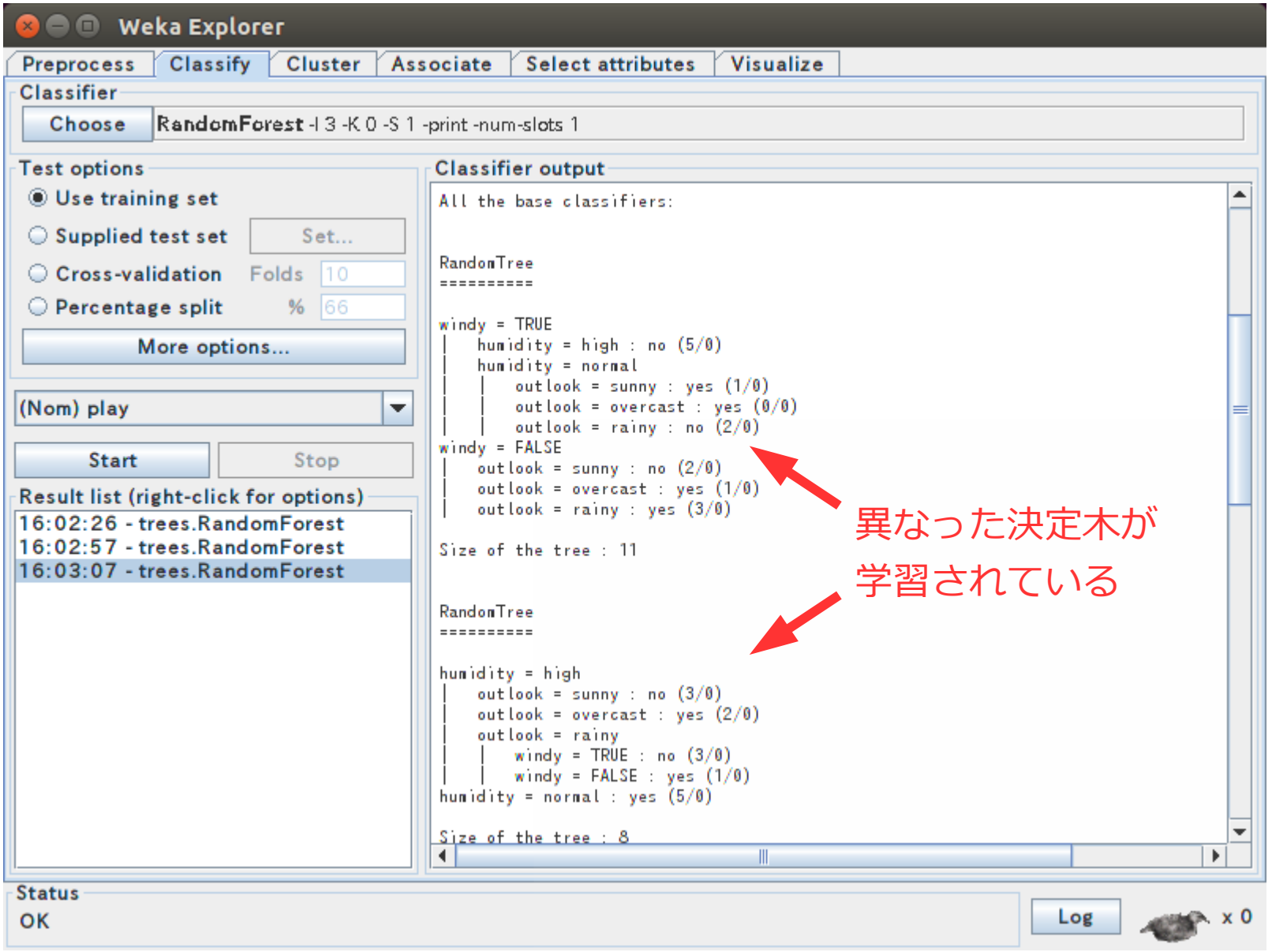


葉が単一クラスになるまで（過）学習

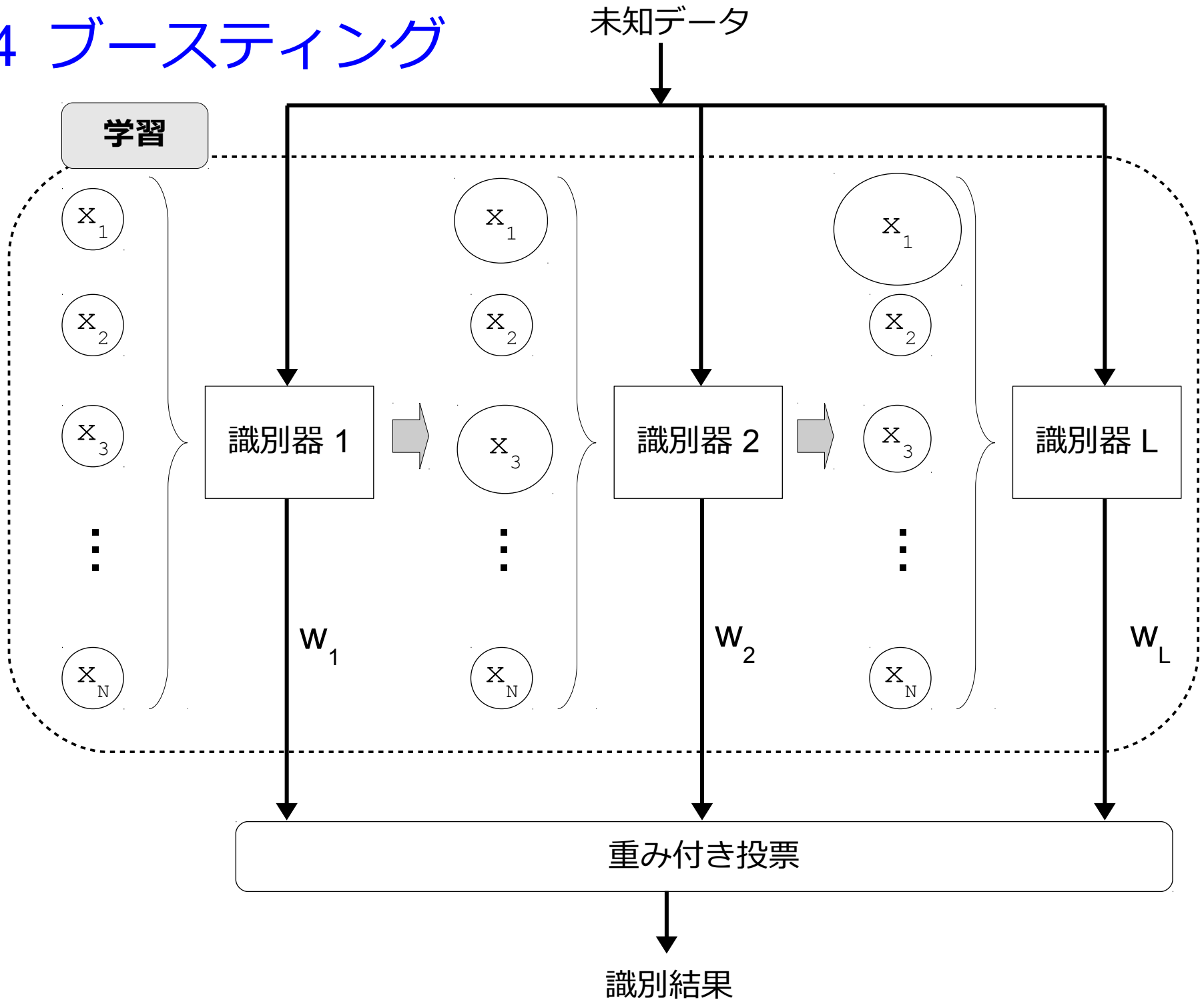
9.3 ランダムフォレスト

- 特徴
  - バギングと同様、学習データは復元抽出
  - 識別器作成に使用できる特徴をランダムに制限することで、各抽出データ毎に全く異なった識別器ができる
  - 識別器は意図的に過学習させる

Weka の RandomForest



9.4 ブースティング



9.4 ブースティング

- 特徴
  - 逐次的に相補的な分類器を作成
  - 以前の分類器が誤った事例に重みを付けて次の分類器を学習
  - 学習アルゴリズムが重みに対応していない場合は、重みに比例した数を復元抽出
  - 結果は分類器に対する重み付き投票