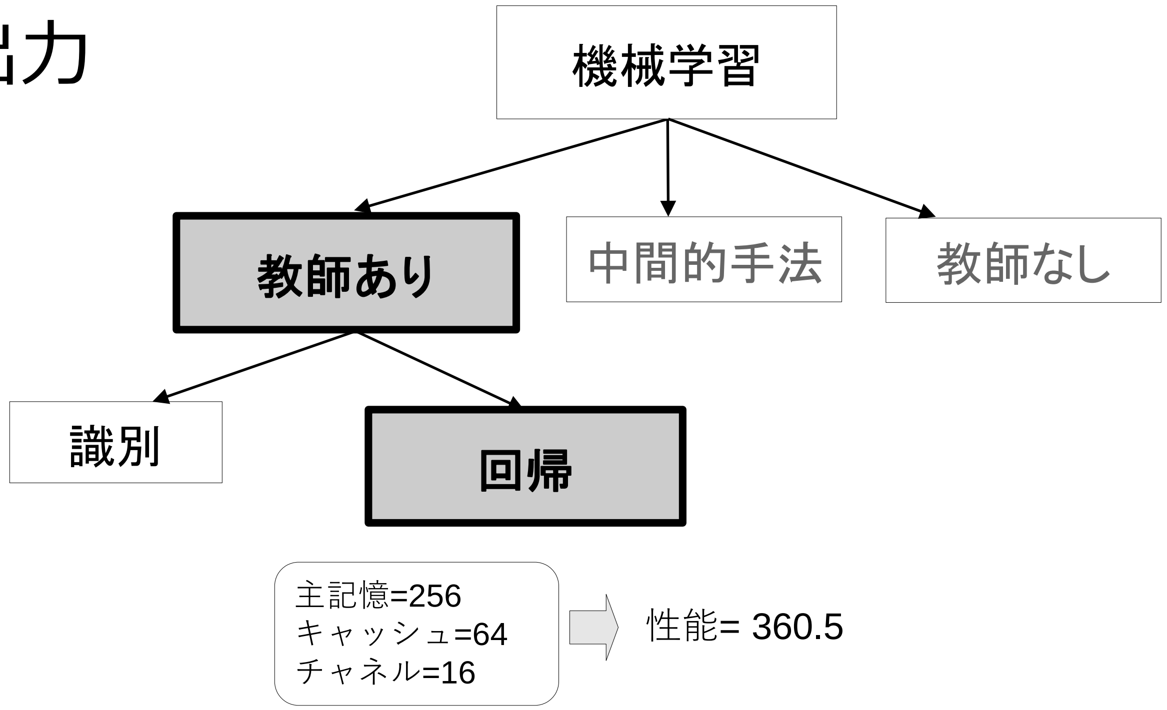
6. 回帰

- 問題設定
  - ◆ 教師あり学習
  - ◆ 数値入力　→　数値出力



6.1 数値特徴に対する「教師あり・回帰」問題の定義

- 回帰のデータ
  - 特徴ベクトル $\boldsymbol{x}$ と正解情報 y のペア

$$\{(\boldsymbol{x}_i, y_i)\}, \quad i = 1 \ldots N$$

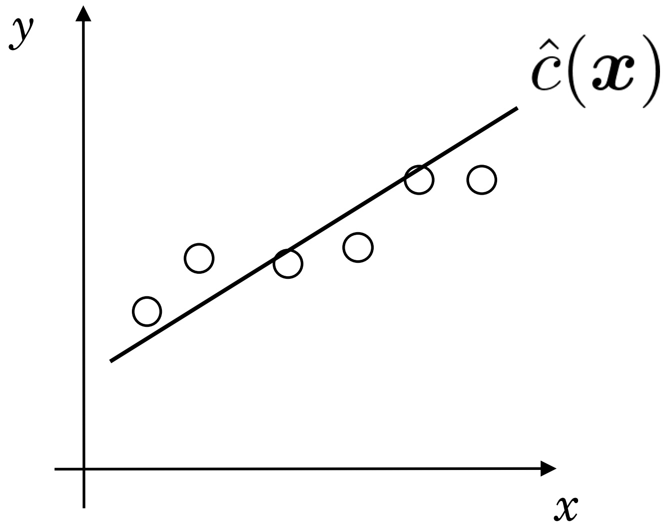
  - $\boldsymbol{x}$ は次元数 d の固定長ベクトル、 y は数値

$$\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^T$$

  - 回帰問題の出力値をターゲットとよぶ
- 回帰問題は関数 $y = \hat{c}(\boldsymbol{x})$ を求める問題と見なせる
  - 未知データに対して予測精度の高い関数を求めたい
  - どの特徴が出力値に対して大きな影響を及ぼしているかを知りたい

6.2 線形回帰

- 目標：なるべく誤差の少ない直線を求める



- 問題の定義
  - ◆ 入力 $\boldsymbol{x}$ から出力 y を求める回帰式を1次式に限定
  - ◆ 学習データから重み $\boldsymbol{w}$ を求める

$$\hat{c}(\boldsymbol{x}) = \sum_{i=0}^{d} w_i x_i$$

特徴ベクトルが1次元の場合
$\hat{c}(\boldsymbol{x}) = w_1 x_1 + w_0$

$\boldsymbol{x}$ を d+1 次元に拡張し、
$x_0 \equiv 1$ とする

6.2 線形回帰

- 最小二乗法による重みの推定
  - ◆ 損失関数：誤差の二乗和 $E(\boldsymbol{w})$　（最小化）

$$E(\boldsymbol{w}) = \sum_{i=1}^{N} (y_i - \hat{c}(\boldsymbol{x}_i))^2$$

$$= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

$\boldsymbol{X}$: 全学習データを並べた行列（パターン行列）
$\boldsymbol{y}$: 全ターゲットを並べたベクトル
$\boldsymbol{w}$: すべての重みを並べたベクトル

  - ◆ $\boldsymbol{w}$ で偏微分した値が0となるのは

$$\boldsymbol{X}^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) = 0$$

$$\Leftrightarrow \boldsymbol{X}^T \boldsymbol{y} = \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w}$$

$$\Leftrightarrow \boldsymbol{w} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

$\boldsymbol{w}$が解析的に求まる

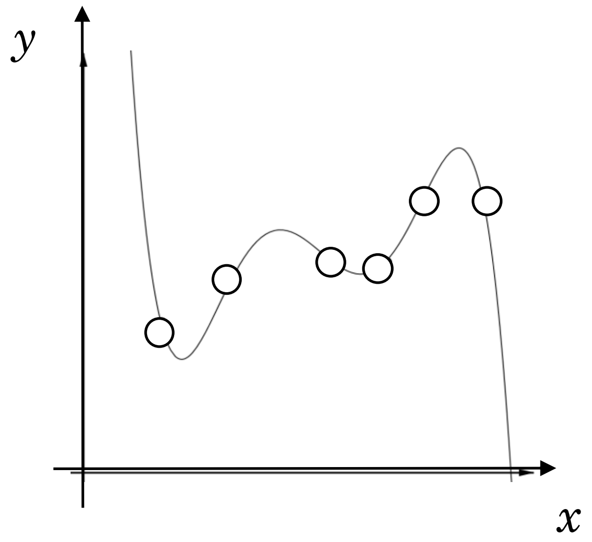
6.2 線形回帰

- 線形回帰の精度向上 (p.93)
  - 基底関数 $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_0(\boldsymbol{x}), \ldots, \phi_b(\boldsymbol{x}))$ を考える

    例 $\boldsymbol{\phi}(x) = (1, x, x^2, \ldots, x^b)$

$$\hat{c}(\boldsymbol{x}) = \sum_{j=0}^{b} w_j \phi_j(\boldsymbol{x})$$

  - 重み $\boldsymbol{w}$ が線形であれば、最小二乗法が適用可能
- 問題点
  - 汎化性能の低下



6.3 回帰モデルの評価

- 回帰モデルの評価法
  - ◆ 誤差の二乗和の平均
    - 手法間の評価に有効
  - ◆ 相関係数
    - 出力と正解とがどの程度似ているか
  - ◆ 決定係数
    - 正解との離れ具合の二乗と分散との比を1から引いたもの
    - 相関係数の二乗と等しい

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{c}(\boldsymbol{x}_i))^2}{\sum_{i=1}^{N} (y_i - \tilde{y})^2}$$

$\tilde{y}$： yの平均

6.4　正則化

- 正則化の考え方
  - ◆ 正則化項の導入

    → 複雑な重み $\boldsymbol{w}$（過学習）の回避

    - L1ノルム $|\boldsymbol{w}|$ ： 0となる重みが多くなる　Lasso
    - L2ノルム $\|\boldsymbol{w}\|^2$: 重みを0に近づける　Ridge
- リッジ回帰
  - ◆ 誤差の二乗和にL2ノルム正則化項を加える

$$E(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \underline{\alpha \boldsymbol{w}^T \boldsymbol{w}}$$

α：誤差と正則化項とのバランス

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \alpha \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$\boldsymbol{w}$が解析的に求まる

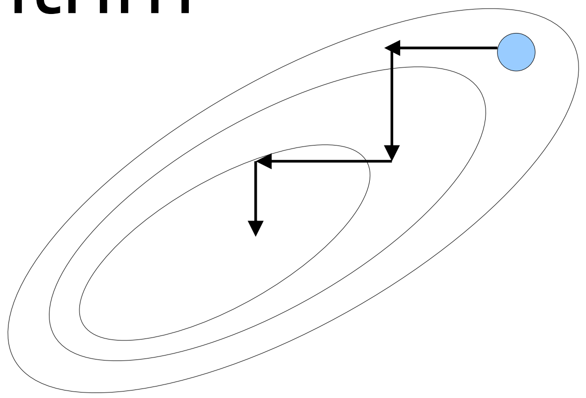
6.4 正則化

- ラッソ回帰
  - 誤差の二乗和にL1ノルム正則化項を加える

$$E(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \alpha \underline{\sum_{j=1}^{d} |w_j|}$$

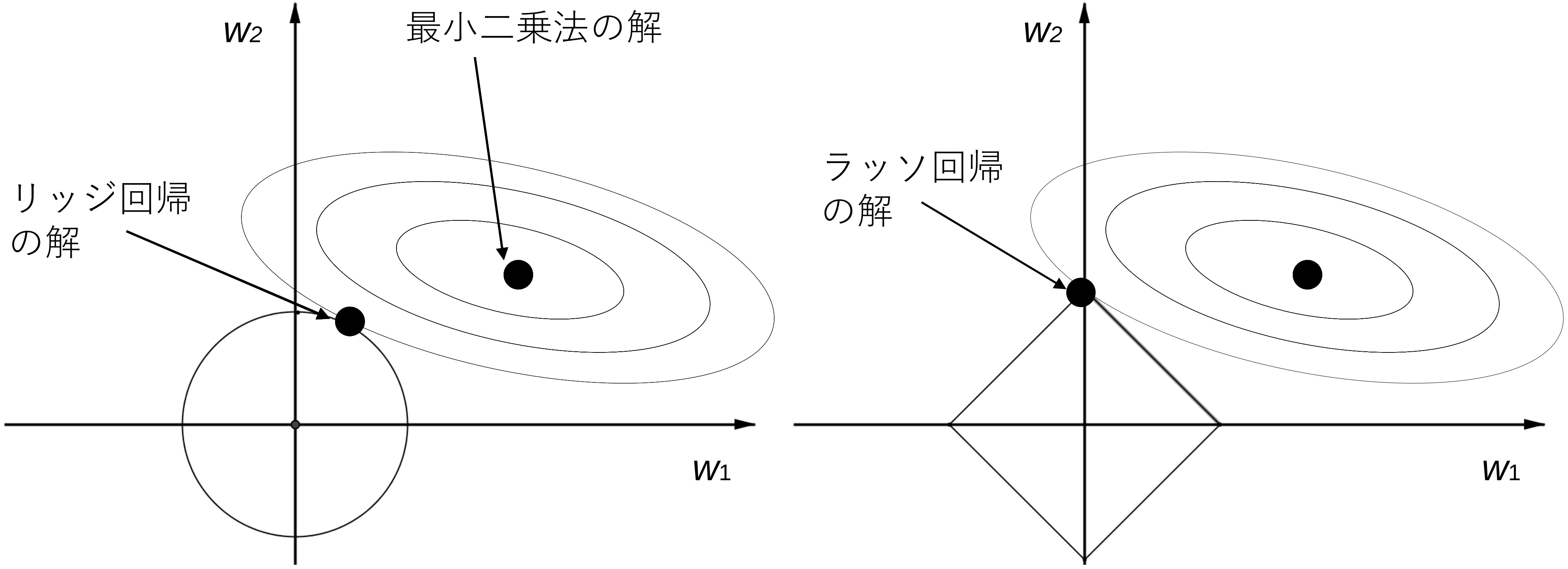
  - 微分不可能な点があるため、解析的に解を求められない
  - 解法例：coordinate descent algorithm

1つの変数（軸）の値だけを誤差が
減る方向に変更することを繰り返す



6.4 正則化

- リッジ回帰とラッソ回帰

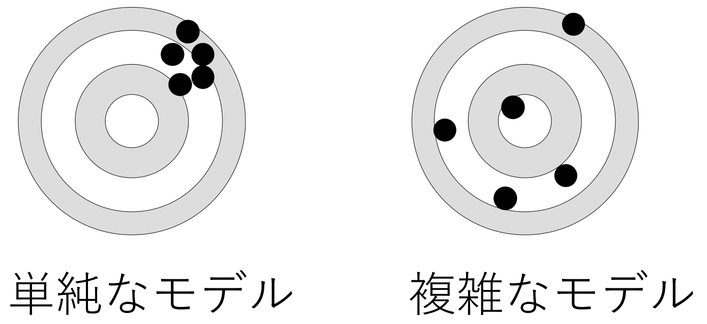


重みを0に
近づけている

0となる重みを
多くしている

6.5 バイアス－分散のトレードオフ

- バイアスと分散
  - ◆ バイアス：正解（＝真のモデル）からのズレ
  - ◆ 分散：求まるモデルの安定性

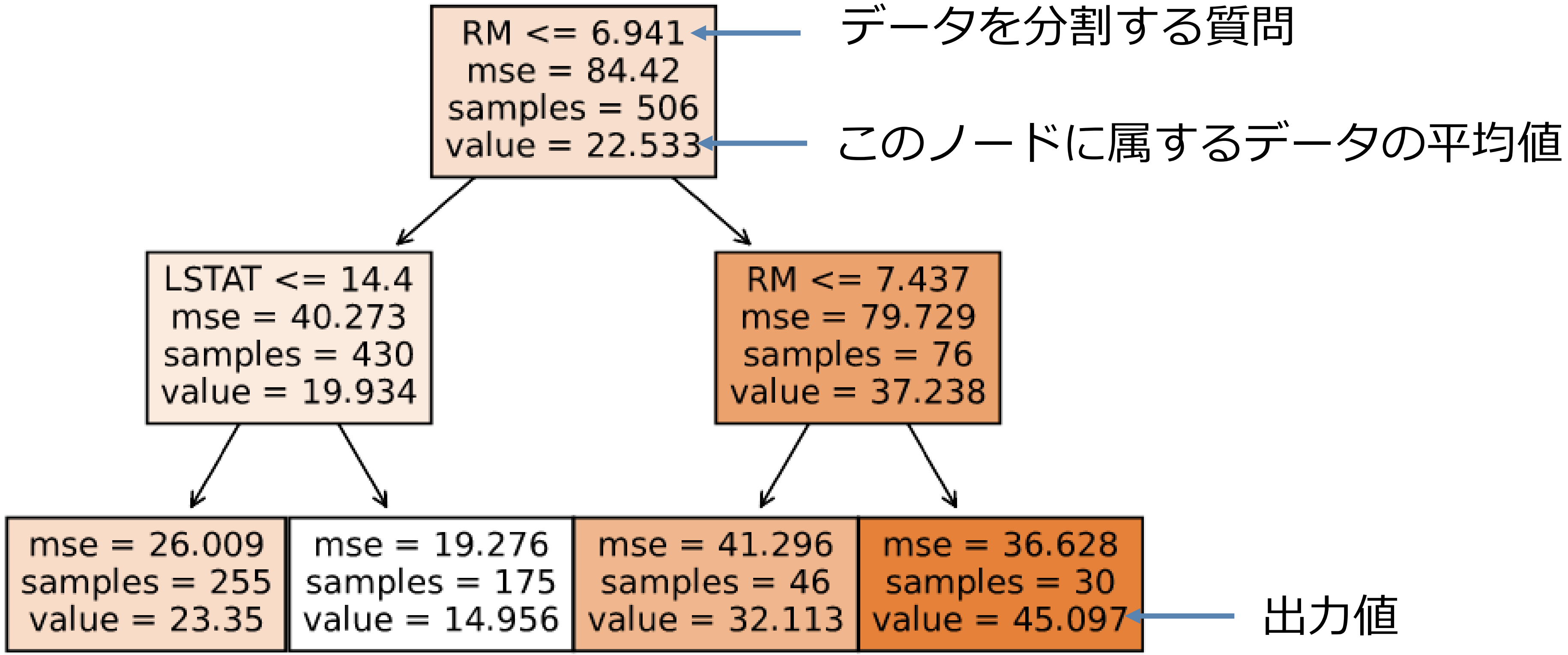


- 単純なモデル
  - ◆ 正解をカバーしていないかもしれない → バイアス大
  - ◆ データが多少ぶれても結果は似ている → 分散小
- 複雑なモデル
  - ◆ 正解をカバーしている可能性が高い → バイアス小
  - ◆ データが少し違えば結果が大きく異なる → 分散大

正則化項はこれを小さくしようとしている

6.6 回帰木

- 回帰木とは
  - ◆ 識別における決定木の考え方を回帰問題に適用
  - ◆ ターゲット値の分散が小さくなるように分割



6.6 回帰木

- CART (classification and regression tree)
  - ◆ 木の構造を二分木に限定
  - ◆ 識別の際のデータの分類基準はジニ不純度
    - 2クラスの場合のジニ不純度　$I_G(p) = 2p(1-p)$　　p: 正例の割合
    - クラスの出現が等確率のとき最大
  - ◆ 回帰に用いるときのデータの分類基準：ターゲット値の分散
    - 子ノードの重み付き分散和が最小
      ＝分割後の分散の減少量が最大　となる特徴を選ぶ

6.6 回帰木

- CARTの特徴選択基準
  - データ D の分散

$$SS(D) = \frac{1}{|D|} \sum_{y_i \in D} (y_i - \tilde{y})^2$$

$\tilde{y}$: Dの平均

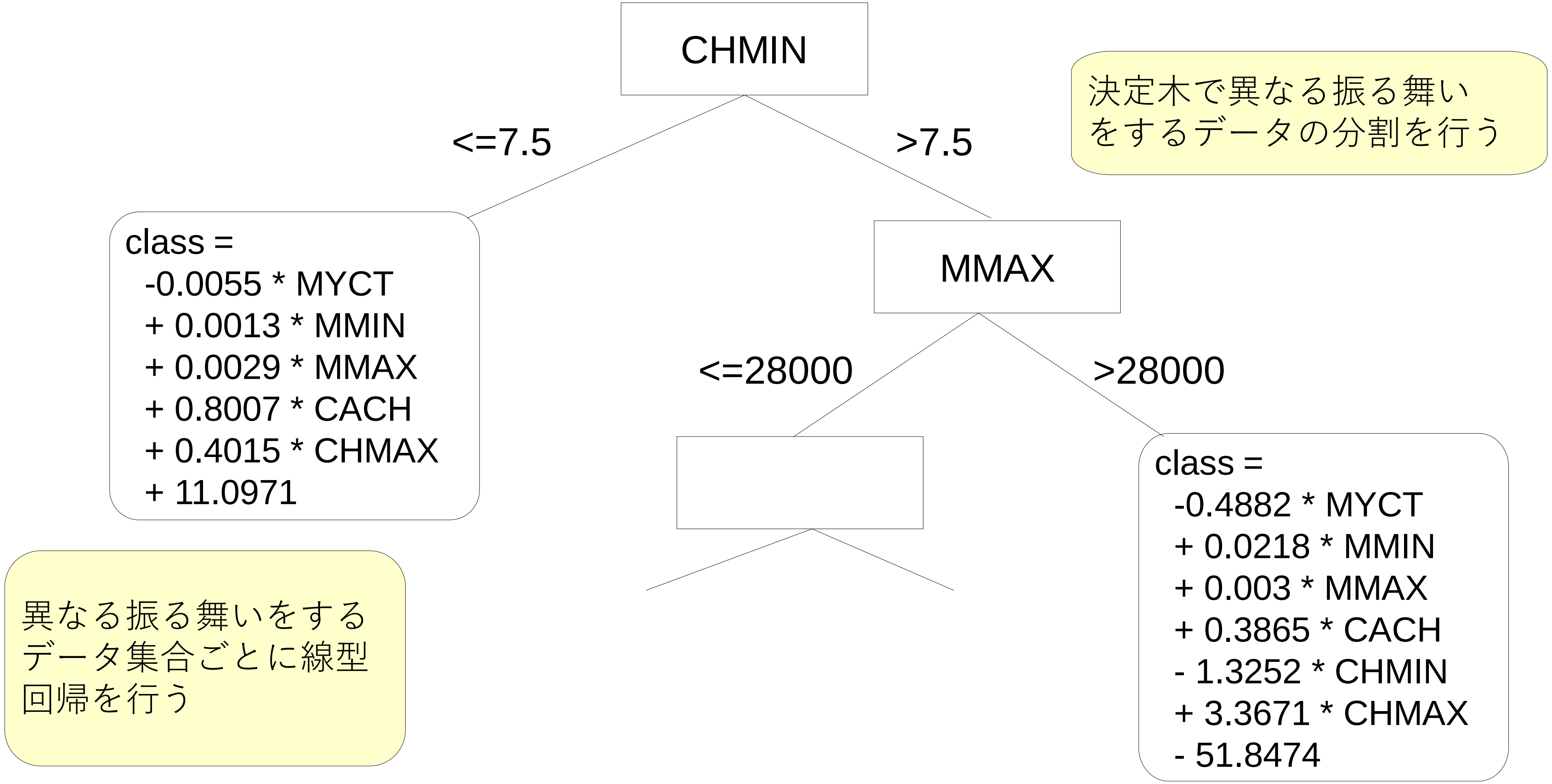
  - 分割後の分散の減少量が最大となる特徴を選ぶ

$$\Delta SS(D) = SS(D) - P_L \cdot SS(D_L) - P_R \cdot SS(D_R)$$

D_L : 左部分木のデータ
D_R : 右部分木のデータ
P_L : 左部分木のデータ数の割合
P_R : 右部分木のデータ数の割合

6.7 モデル木

- モデル木とは
  - ◆ リーフを線形回帰式にした回帰木



まとめ

- 線形回帰
  - ◆ 最小二乗法で回帰関数のパラメータを求めることができる
  - ◆ 基底関数によって複雑な関数も表現可能
  - ◆ 正則化で過学習を回避
- 回帰木
  - ◆ 決定木を用いて回帰を行う
  - ◆ 振る舞いが異なるデータが混合しているときに有効
  - ◆ リーフを線形回帰式にしたものがモデル木