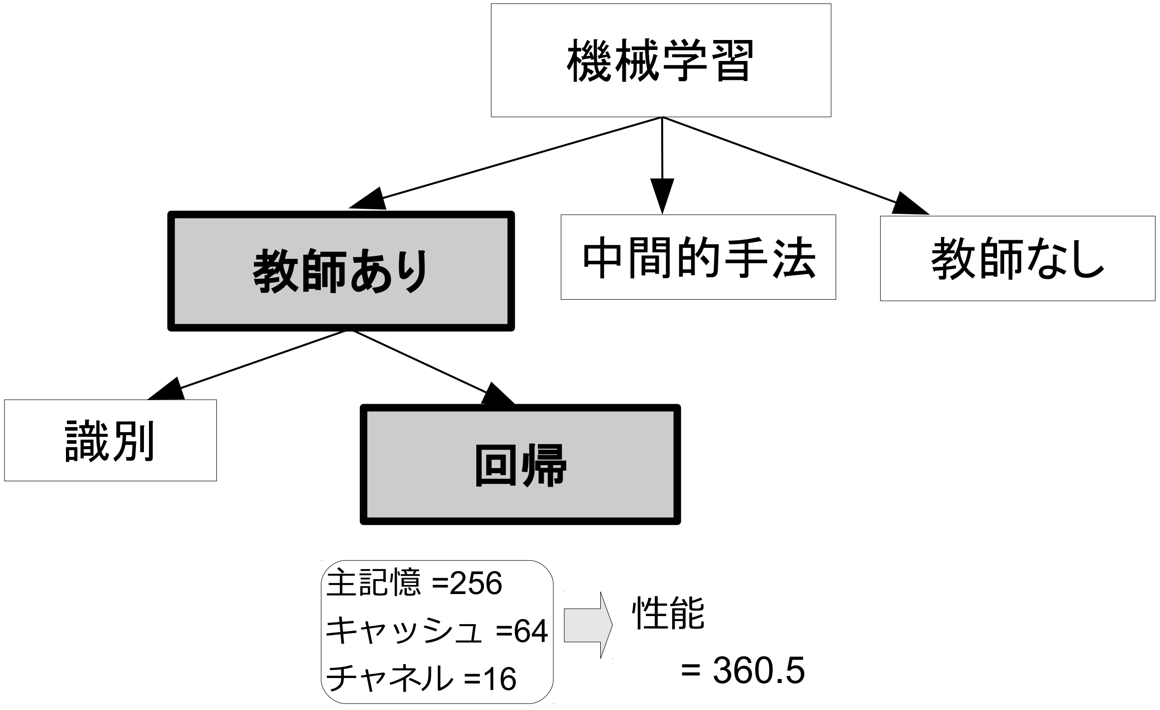
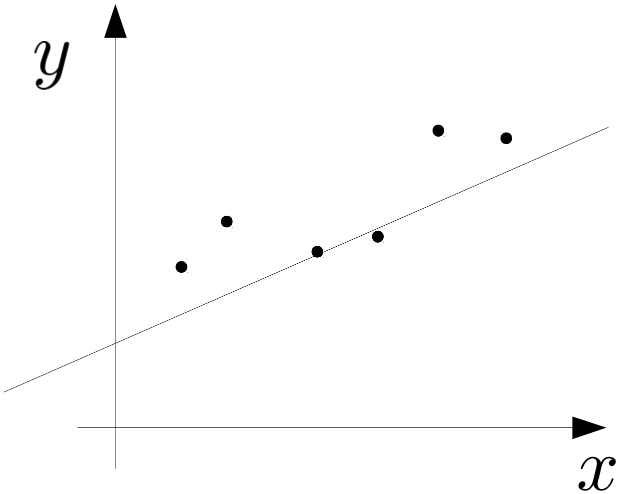
8. 回帰

- 問題設定
  - 教師あり学習
  - 数値入力　→　数値出力



8.2 線形回帰

- 目標：なるべく誤差の少ない直線を求める



- 線形回帰の定義
  - 入力 $\boldsymbol{x}$ から出力 y を求める回帰式を 1 次式に限定
  - 学習データから係数 w を求める

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

8.2 線形回帰

- 最小二乗法による係数の推定
  - 推定の基準：誤差の二乗和 *E* を最小化

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

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

X：全学習データを並べた行列

w：係数のベクトル表現

  - *w* で微分した値が 0 となるのは

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

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

w が解析的に
求まる

8.2 線形回帰

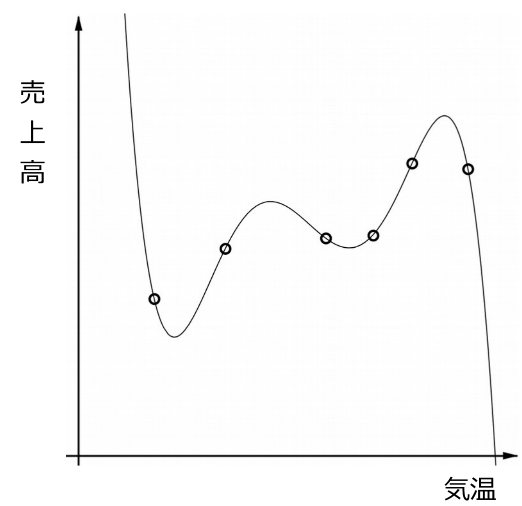
- 最小二乗法の精度向上
  - 基底関数 $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_b(\boldsymbol{x}))$ を考える

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

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

  - 係数が線形であれば、最小二乗法が適用可能

- 問題点
  - 汎化性能の低下



8.2 線形回帰

- 正則化
  - 正則化項の導入

    → 複雑なパラメータ w （過学習）の回避
    - L1 ノルム $|\boldsymbol{w}|$ ： 0 となるパラメータが多くなる Lasso
    - L2 ノルム $\|\boldsymbol{w}\|^2$ ：パラメータを 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

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

λ：誤差の二乗和と正則化項とのバランス

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

w が解析的に求まる

8.2 線形回帰

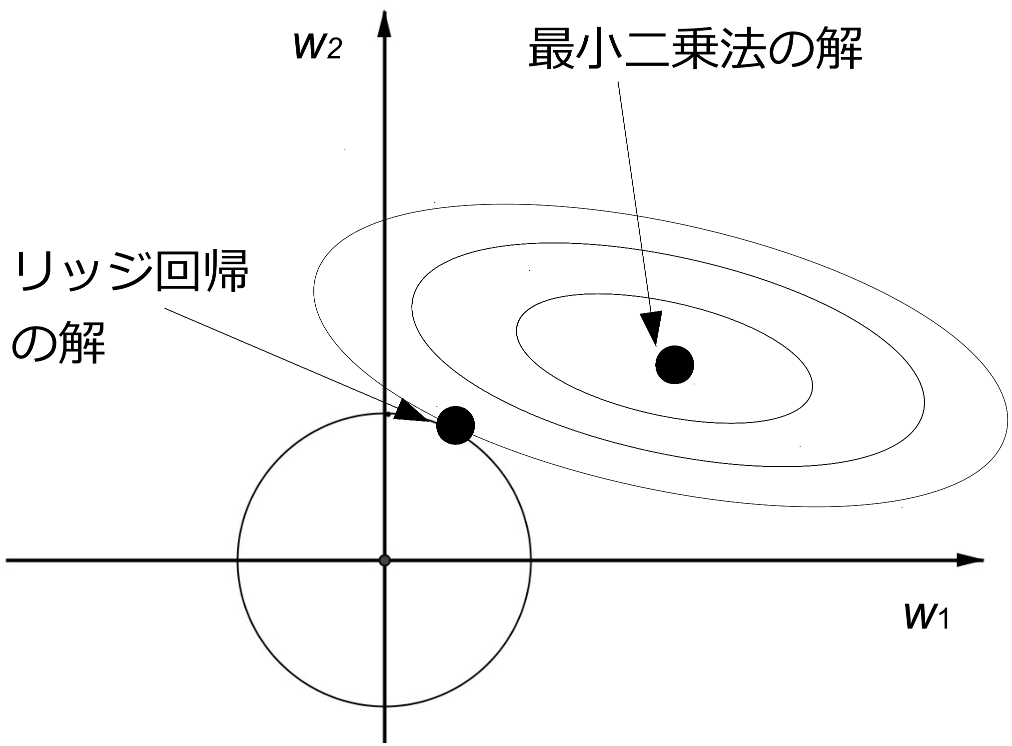
- ラッソ回帰
  - 誤差の二乗和に L1 ノルム正則化項を加える

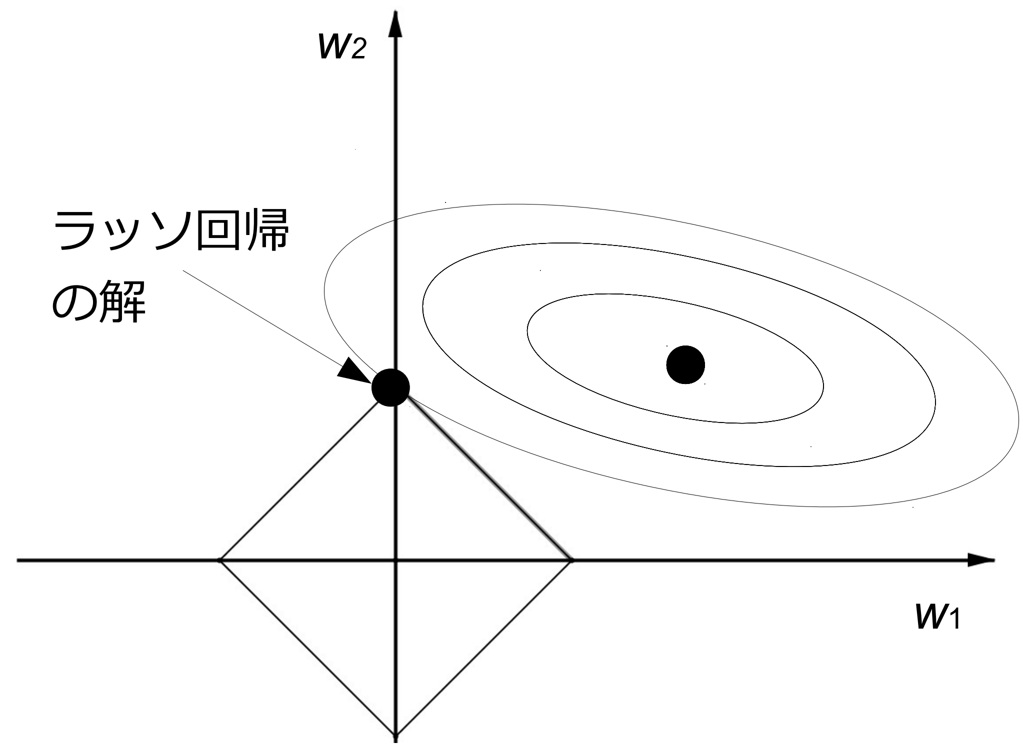
$$E(\boldsymbol{w}) = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \underline{\lambda \sum_{j=1}^{d} |w_j|}$$

  - 微分不可能な点があるため、解析的に解を求めることができない
    - 適当な初期重みから始め、リッジ回帰で上界を押さえる逐次更新アルゴリズムを用いる

8.2 線形回帰

- リッジ回帰とラッソ回帰





パラメータを 0 に
近づけている

0 となるパラメータを
多くしている

8.4 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
  - リーフを線形回帰式にしたモデル木で性能が向上