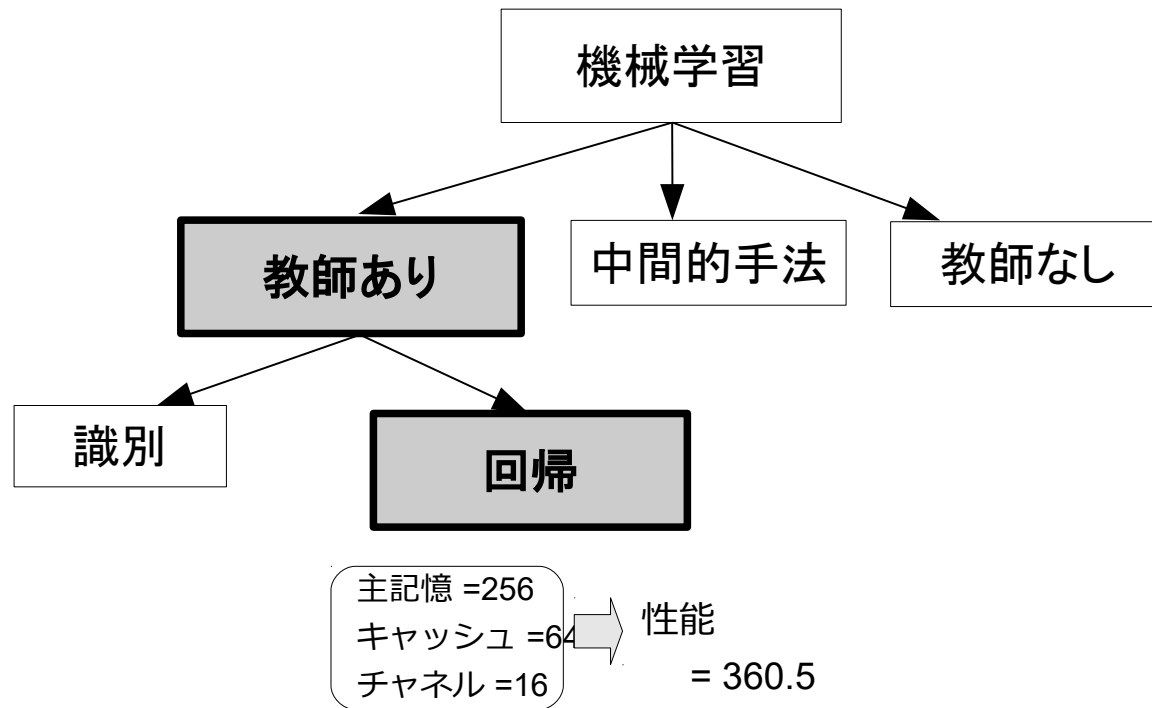


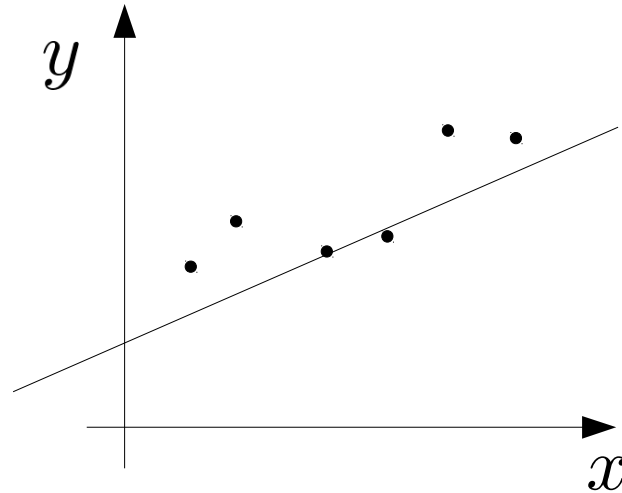
8. 回帰

- 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 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

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

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

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

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

\boldsymbol{X} : 全学習データを並べた行列

\boldsymbol{w} : 係数のベクトル表現

- \boldsymbol{w} で微分した値が 0 となるのは

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

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

\boldsymbol{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

例 $\phi(x) = (1, x, x^2, \dots, x^b)$

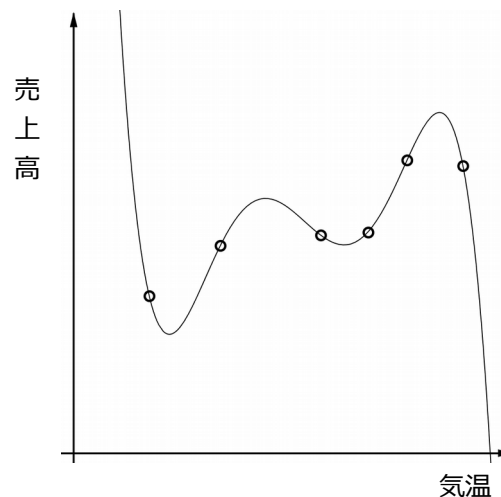
- 基底関数 $\phi(x) = (\phi_1(x), \dots, \phi_b(x))$ を考える

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

- 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

- 正則化の考え方

- 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 $|w|$: 0 となるパラメータが多くなる

Lasso

- L2 ノルム $\|w\|^2$: 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

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

λ : 誤差の二乗和と正則化項とのバランス

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

w が解析的に
求まる

8.2 線形回帰

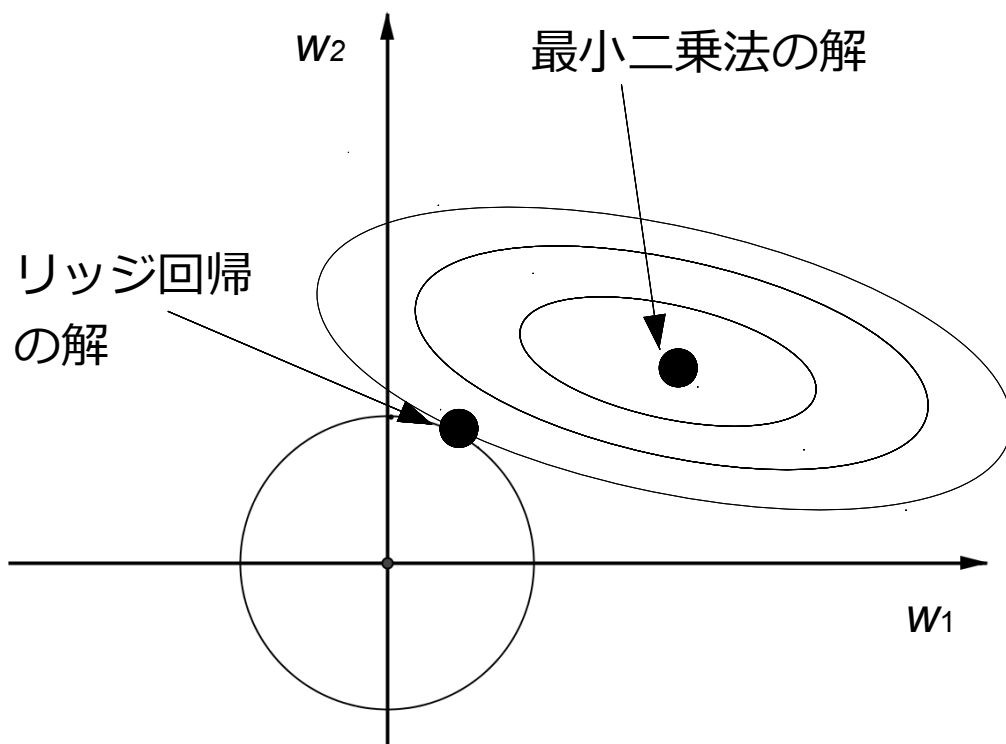
- ラッソ回帰
 - 誤差の二乗和に L1 ノルム正則化項を加える

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

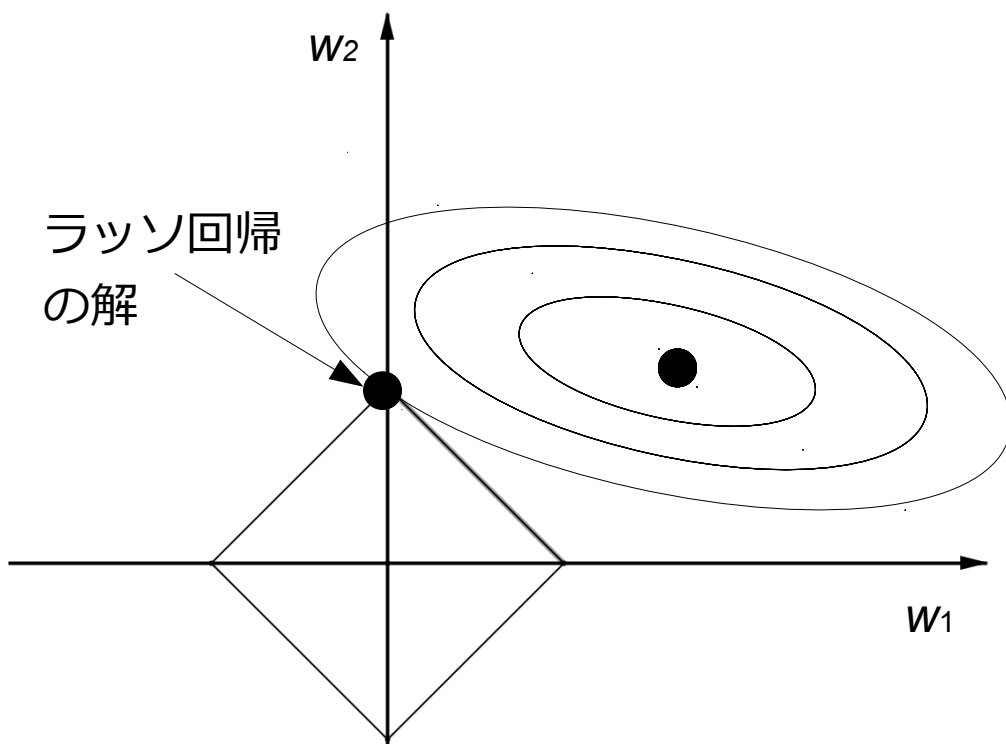
- 微分不可能な点があるため、解析的に解を求めることができない
 - 適当な初期重みから始め、リッジ回帰で上界を押さえる逐次更新アルゴリズムを用いる

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

8.2 線形回帰

- 回帰問題の評価法
 - 誤差の二乗和：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：相関係数の 2 乗

Weka の結果表示例

```
=== Cross-validation ===  
=== Summary ===
```

Correlation coefficient	0.9012
Mean absolute error	41.0886
Root mean squared error	69.556
Relative absolute error	42.6943 %
Root relative squared error	43.2421 %
Total Number of Instances	209

決定係数の式

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

\tilde{y} : y の平均

カーネル回帰

- 基底関数にカーネルを用いる

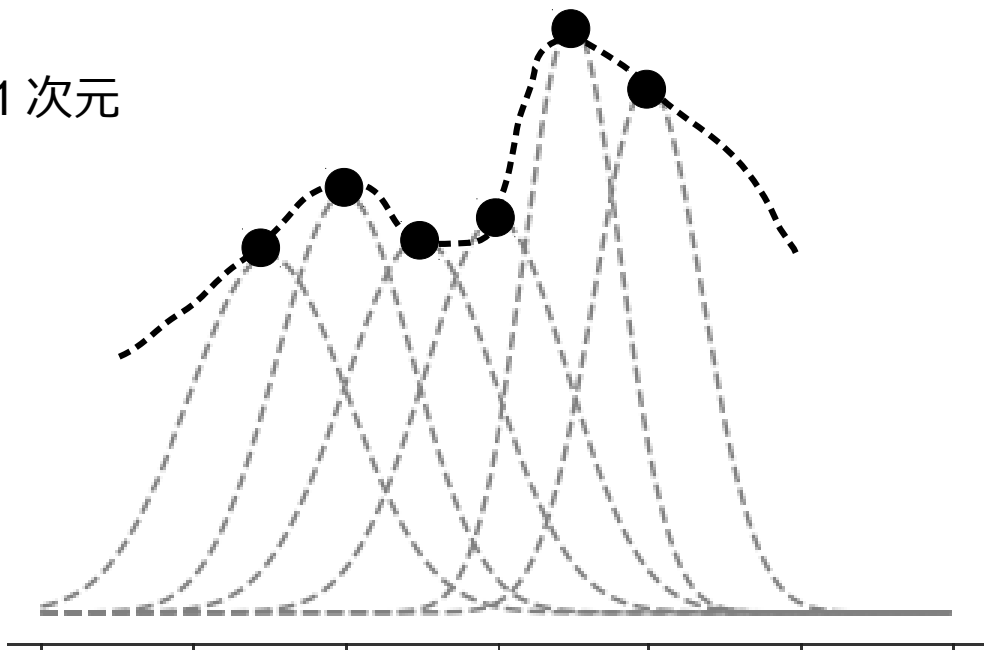
$$\hat{c}(\boldsymbol{x}) = \sum_{j=1}^N \alpha_j K(\boldsymbol{x}, \boldsymbol{x}_j)$$

- RBF カーネルを用いた場合

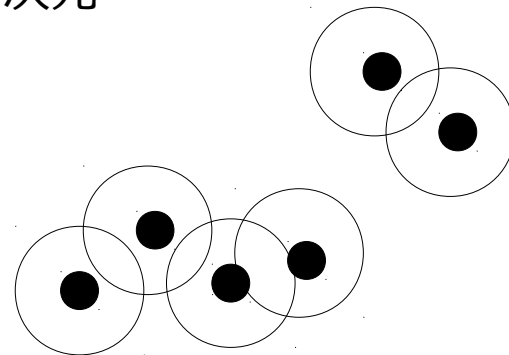
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2)$$

近くにある学習データ
とのカーネル関数の値の
重み付き和
= 学習データの近傍で
のみ関数を近似

1次元



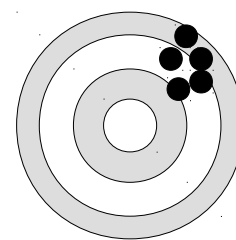
2次元



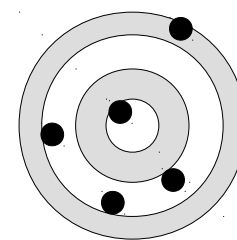
8.3 バイアスと分散のトレードオフ

- バイアスと分散

- バイアス：正解からのズレ
- 分散：求まる解の安定性



単純なモデル



複雑なモデル

- 単純なモデル

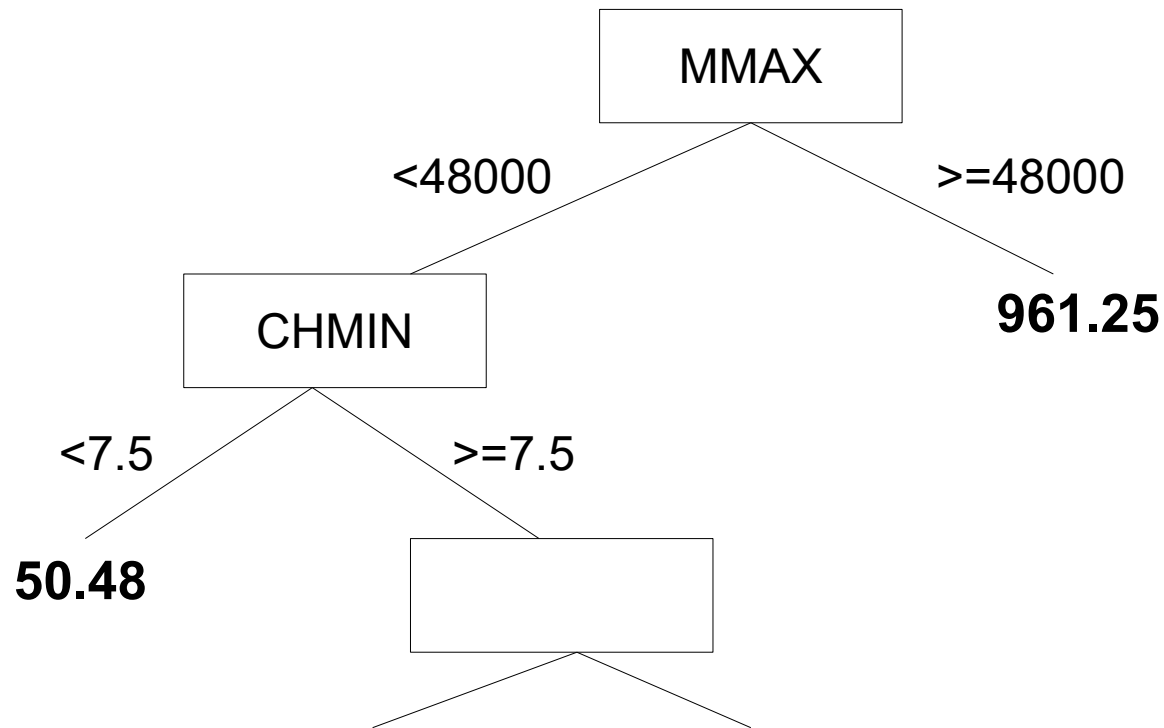
- 正解からはずれているかもしれない→バイアス**大**
- データが多少ぶれても結果は似ている→分散**小**

- 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**

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
 - ターゲット値の分散が小さくなるように分割



8.4 回帰木

- CART (classification and regression tree)
 - 木の構造を二分木に限定
 - 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 $I_G(p) = 2p(1 - p)$
 - 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はターゲット値の分散
 - 子ノードの重み付き分散和が最小となる特徴を選ぶ

8.4 回帰木

- CART (classification and regression tree)
 - 分散の計算
 - Y : あるノードに属するデータのターゲット値の集合

$$Var(Y) = \frac{1}{|Y|} \sum_{y_i \in Y} (y_i - \bar{y})^2 \quad \bar{y} : Y \text{の平均}$$

$$\begin{aligned} Var(\{Y_1, \dots, Y_l\}) &= \sum_{j=1}^l \frac{|Y_j|}{|Y|} Var(Y_j) \\ &= \sum_{j=1}^l \frac{|Y_j|}{|Y|} \left(\frac{1}{|Y_j|} \sum_{y \in Y_j} y^2 - \bar{y}_j^2 \right) \\ &= \frac{1}{|Y|} \sum_{y \in Y} y^2 - \sum_{j=1}^l \frac{|Y_j|}{|Y|} \bar{y}_j^2 \end{aligned}$$

8.4 回帰木

- モデル木とは
 - リーフを線形回帰式にした回帰木

