



データの前処理・モデリング・ 変数選択

明治大学 理工学部 応用化学科

データ化学工学研究室 専任講師 金子弘昌

Website: <http://datachemeng.com/>

Twitter: @hirokaneko226

2017年11月5日 (日)

ケモメトリックス

1

データベース

X: 説明変数

構造記述子*など

y: 目的変数

物性・活性など

*化学構造の情報を数値化したもの
例) 分子量、炭素原子の数、
ベンゼン環の数

モデリング

予測

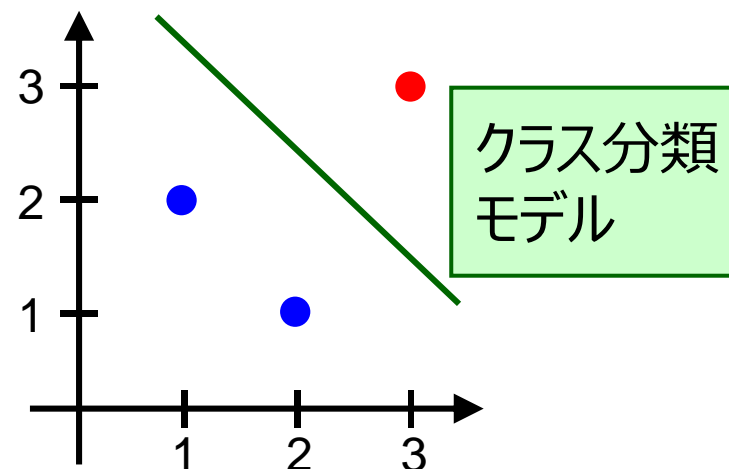
新しいデータ x_{new}

回帰モデル
クラス分類モデル
 $y = f(X)$

yの推定値

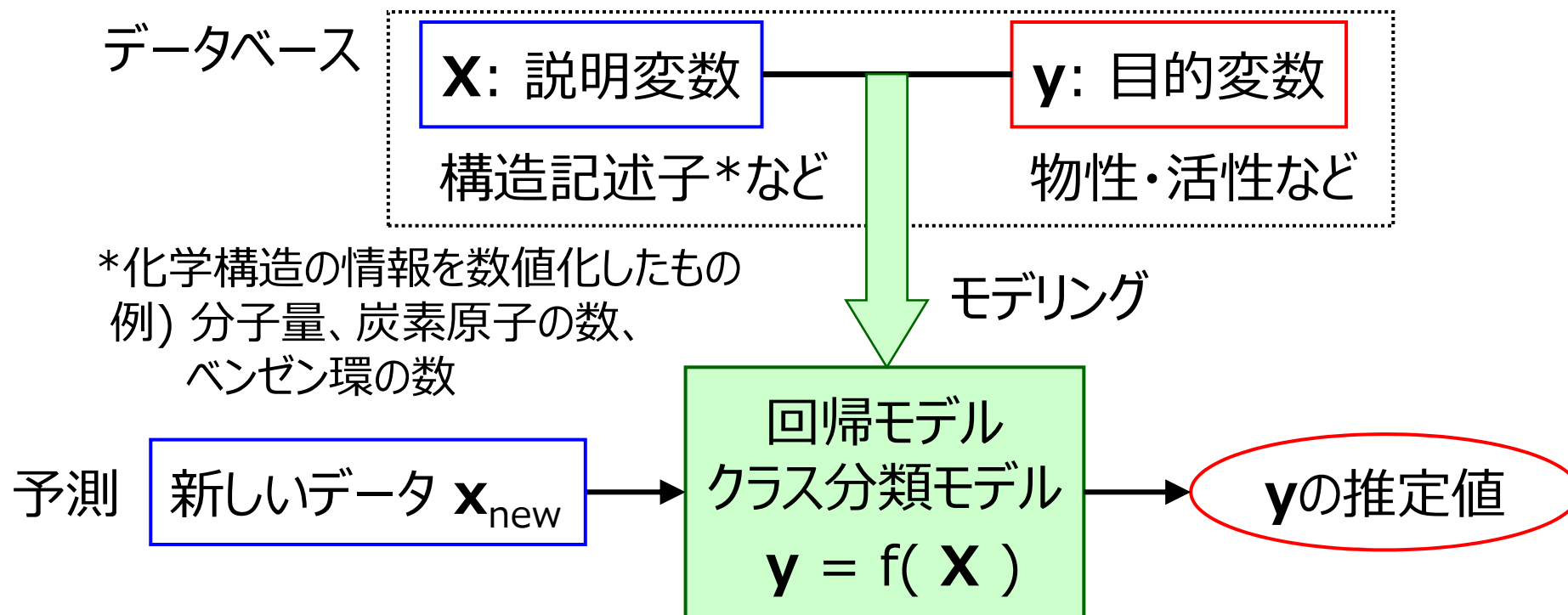
例) **X**: 2変数
データ数: 3
線形モデル

	x_1	x_2	y
データ1	1	2	活性なし
データ2	2	1	活性なし
データ3	3	3	活性あり



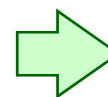
ケモメトリックス

2



例) **X**: 2変数
データ数: 3
線形モデル

	\mathbf{x}_1	\mathbf{x}_2	\mathbf{y}
データ1	1	2	5.1
データ2	2	1	3.9
データ3	3	3	9.2



回帰モデル

$$\mathbf{y} = \mathbf{x}_1 + 2\mathbf{x}_2 + \text{誤差}$$

データセットの定義

- ✓ トレーニングデータ (training dataset)
 - (回帰・クラス分類) モデルを構築するためのデータ
- ✓ バリデーションデータ (validation dataset)
 - モデルのハイパーパラメータを決めるためのデータ
- ✓ テストデータ (test dataset)
 - y の値を隠しておき、トレーニングデータ・バリデーションデータで作られたモデルの性能を最終的に確認するためのデータ

内容 1/2

✓データの前処理

- 標準化 (オートスケーリング)
- 情報量の小さい変数の削除

✓モデリング

- 入門編の復習
- 決定木 (Decision Tree, DT)
- ランダムフォレスト (Random Forests, RF)
- リッジ回帰 (Ridge Regression, RR)
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net (EN)
- Support Vector Regression (SVR)
 - モデルの検証

内容 2/2

✓変数選択

- Stepwise (ステップワイズ) 法
 - LASSO、EN、RFでも変数選択可能

注意点

✓手法のすべてを説明するわけではありません

- 説明すること

- 概要（どんな手法か？）
- 目的（手法で何を達成したいのか？数式で表わすと？）

- 説明しないこと

- 式変形

✓質問はいつでも構いません

どうしてデータの前処理をするの？

✓単位系が異なる場合など、各変数(記述子)が同等に扱われない

- 長さ: km, m, cm, mm, nm など
- 温度: °C, K など

✓データ分布の中心が 0 であると、何かとうれしい



オートスケーリング (標準化)

✓情報量のない変数はいらない (かえって邪魔になるときもある)

- ほぼすべてのサンプルで値が同じ変数



分散が0の変数の削除、同じ値を多くもつ変数の削除

- 似た変数の組の 1 つ



相関係数の高い変数の組の 1 つを削除

オートスケーリング (標準化)

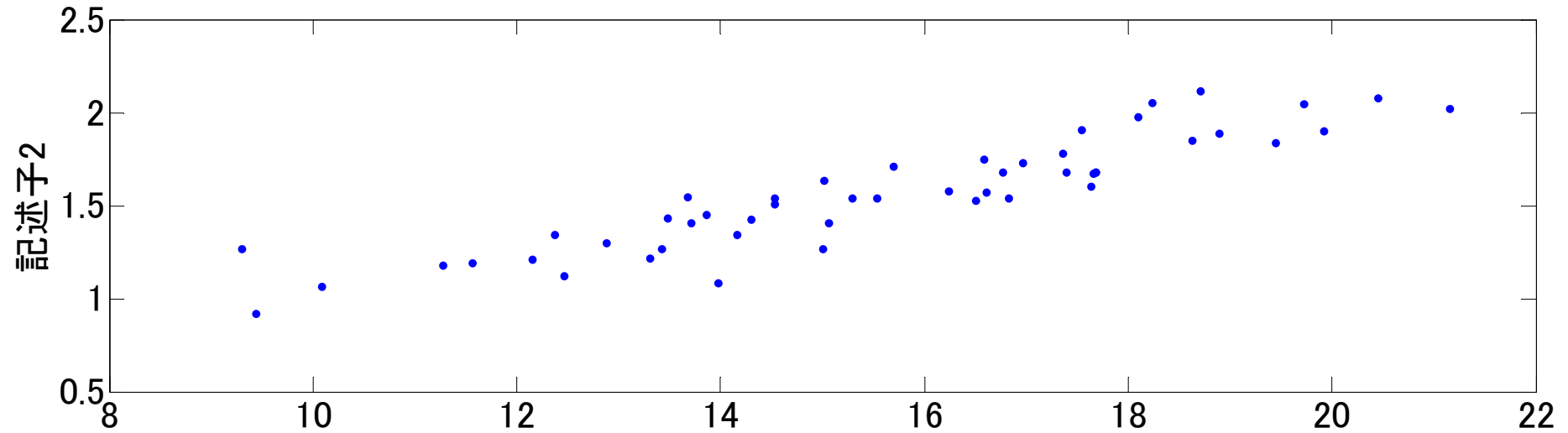
✓データ解析・ケモメトリックスにおける一般的な前処理の方法

✓オートスケーリング = センタリング + スケーリング

- センタリング: 変数(記述子)ごとにその平均を引き、平均を 0 にする
- スケーリング: 変数(記述子)ごとにその標準偏差で割り、標準偏差を 1 にする

✓各変数(記述子)が同等の重みを持つようになる

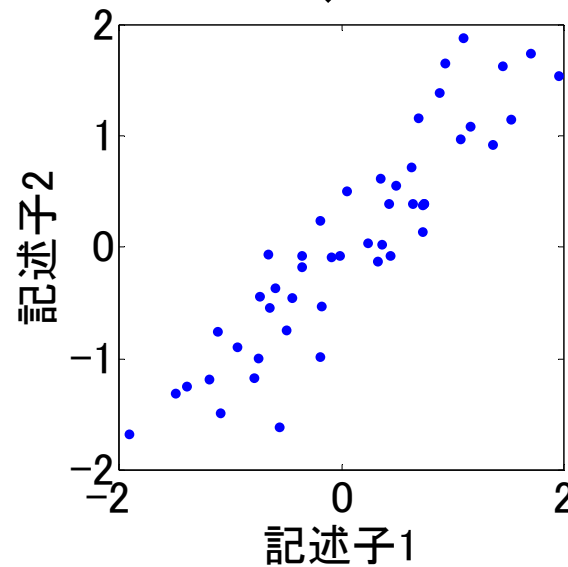
オートスケーリングの例



記述子1



オートスケーリング



$x_i^{(k)}$: k 個目のサンプルにおける、 i 番目の変数(記述子) の値

✓センタリング 各変数(記述子)の平均を0にする
(それぞれのサンプルから平均を引く)

$$x_i^{(k)'} = x_i^{(k)} - \mu_i$$
$$\mu_i = \frac{\sum_{k=1}^n x_i^{(k)}}{n}$$

n : サンプル数

スケーリング

$x_i^{(k)}$: k 個目のサンプルにおける、 i 番目の変数(記述子) の値

✓スケーリング 各変数(記述子)の標準偏差を1にする
(それぞれのサンプルを標準偏差で割る)

$$x_i^{(k)''} = \frac{x_i^{(k)'}}{\sigma_i}$$

$$\sigma_i = \sqrt{\frac{\sum_{k=1}^n (x_i^{(k)} - \mu_i)^2}{n-1}}$$

モデル検証用(テスト)データのオートスケーリング¹²

- ✓モデル検証用データ(テストデータ)のオートスケーリングには、
モデル構築用データ(トレーニングデータ)の平均・標準偏差を使用
- テストデータの平均・標準偏差ではないので注意
 - テストデータの平均・標準偏差を使うとトレーニングデータのスケールと変わってしまう

$$x_{\text{test},i}^{(k)} = \frac{x_{\text{test},i}^{(k)} - \mu_i}{\sigma_i}$$

$x_{\text{test},i}^{(k)}$: テストデータの k 個目のサンプルにおける、 i 番目の変数(記述子) の値

μ_i : トレーニングデータの i 番目の変数(記述子) の平均

σ_i : トレーニングデータの i 番目の変数(記述子) の標準偏差

分散が0の変数の削除

- ✓ 分散が0、つまりすべてのサンプルで同じ値をもつ変数は、意味がない
- ✓ 分散が0ということは、標準偏差が0なので、スケーリングができない (0で割ることになってしまう)

✓ 最初に、分散 $\frac{\sum_{k=1}^n (x_i^{(k)} - \mu_i)^2}{n-1}$ が 0 の変数を削除しましょう！

$x_i^{(k)}$: k 個目のサンプルにおける、 i 番目の変数(記述子) の値

n : サンプル数

$$\mu_i = \frac{\sum_{k=1}^n x_i^{(k)}}{n}$$

同じ値を多くもつ変数の削除

- ✓ 分散が 0 の変数を削除するだけで十分か？
- ✓ 1 つのサンプルの値が 1 で、他のサンプルの値がすべて 0 のような変数もいらないそう
 - (注意！) 分散の値が小さい、ということではない。
分散の小さい、たとえば 0.01 未満の、変数を削除してしまうと、すべて小さい値でばらつきは小さいが重要な変数を削除する危険性がある
 - クロスバリデーション(交差検定)をするときに、サンプルを分割したあとに分散が 0 になってしまうとよくない
(クロスバリデーションを知らない人は意味がわからなくてOKです)
- ✓ 同じ値を多くもつ変数も削除しましょう！
 - わたし(金子)は、よく 5-fold クロスバリデーションを行うため、8割以上が同じ値である変数を削除しています

- ✓ 1つのサンプルの値が1で、他のサンプルの値がすべて0のような変数
 - ノイズで1になった変数のときは、過学習してしまうため変数を削除すべき
 - その変数で1をとるサンプルが y に対して意味をもつときもある
 - ベンゼン環をもつ分子が一つだけあり、
 - y が毒性の有無で、ベンゼン環によって毒性が発生するとき
- ✓ 削除しないときと、削除するときの両方モデリングして比較するとよい
 - クロスバリデーションでは注意が必要

相関係数の高い変数の組の 1 つの削除

- ✓ 同じ変数が 2 つあっても意味がない
- ✓ ちょっとしか違わないが (誤差 ? というレベルで) 似ている変数も、どちらか 1 つで OK
- ✓ 最初に変数の数を減らしておくことで、
 - 「次元の呪い」を低減できる
 - あとのデータ解析がやりやすくなる
- ✓ 相関係数が高い変数の組の 1 つを削除しましょう !

$$\frac{\sum_{k=1}^n (x_i^{(k)} - \mu_i)(x_j^{(k)} - \mu_j)}{\sqrt{\sum_{k=1}^n (x_i^{(k)} - \mu_i)^2 \sum_{k=1}^n (x_j^{(k)} - \mu_j)^2}} : i \text{ 番目の変数 と } j \text{ 番目の変数 との相関係数}$$

しきい値は？どちらを消す？

✓しきい値は？

- 0.8, 0.9, 0.95, 0.99など、いろいろな候補があります
- たとえば、0.99 のように思い切って決めてしまうか、
細かく最適化したい場合は試行錯誤的に決めることになります

✓ 2つのうち どちらを消す？

- どちらでもあまり変わりませんが、その他の変数との相関係数を調べて、その絶対値の和の大きい方が他の変数との重複が大きいと考え、そちらを削除するようにしています

- ✓主成分分析 (Principal Component Analysis, PCA) や部分的最小二乗法 (Partial Least Squares, PLS) をすれば、基本的に変数間の相関関係には対処できる
- ✓相関係数の高い変数の組の 1 つを削除したからといって、その後の解析結果があまり変わらないこともある

✓データの前処理

- 標準化 (オートスケーリング)
- 情報量の小さい変数の削除

✓モデリング

- 入門編の復習
- 決定木 (Decision Tree, DT)
- ランダムフォレスト (Random Forests, RF)
- リッジ回帰 (Ridge Regression, RR)
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net (EN)
- Support Vector Regression (SVR)
 - モデルの検証

✓回帰分析

- 最小二乗法による重回帰分析 [1,2,4]
- Partial Least Squares (PLS) [1,2,3]

✓クラス分類

- 線形判別分析 (Linear Discriminant Analysis, LDA) [4,6]
- Support Vector Machine (SVM) [2,5]

- [1] 宮下芳勝・佐々木慎一, コンピュータ・ケミストリー シリーズ3 ケモメトリックスー化学パターン認識と多変量解析ー, 共立出版 (1995)
- [2] 船津公人・金子弘昌, ソフトセンサー入門～基礎から実用的研究例まで～, コロナ社 (2014)
- [3] S. Wold, et. al., Chemom. Intell. Lab. Syst., 58, 109–130, 2001.
- [4] C.M. ビショップ, パターン認識と機械学習 上, 丸善出版 (2012)
- [5] C.M. ビショップ, パターン認識と機械学習 下, 丸善出版 (2012)
- [6] 金 明哲, 金森 敬文, 竹之内 高志, 村田 昇, Rで学ぶデータサイエンス<5>パターン認識, 共立出版 (2009)

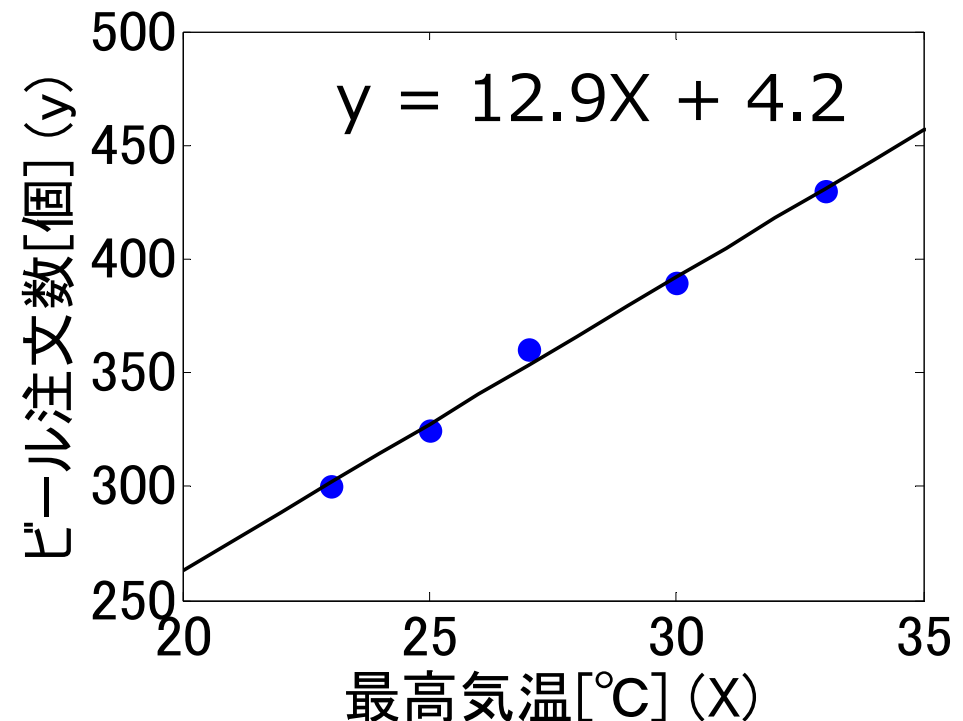
回帰分析ってなに？

21

目的変数 (y) と説明変数 (X) の関係をモデル化し、
Xによってyがどれだけ説明できるのかを**定量的**に分析すること

✓例

- 目的変数 (y)
 - ビール注文数[個]
- 説明変数 (X)
 - 最高気温[°C]



どうやってモデル化する (式を作る) のか？

最小二乗法による線形重回帰分析

- ✓ Multiple Linear Regression (MLR)
- ✓ Ordinary Least Squares (OLS)
- ✓ Classical Linear Regression (CLS)

などと呼ばれます

最小二乗法による重回帰分析

✓線形の式を仮定

- $y = b_1x_1 + b_2x_2 + \cdots + b_mx_m$

- m : 説明変数 (記述子) の数
- b_i : (標準) 回帰係数
- 標準化したあとのため、定数項は 0

✓誤差の二乗和が小さくなるように b_1, b_2, \cdots, b_m を決定

どうして PLS を使うの？ ～多重共線性～

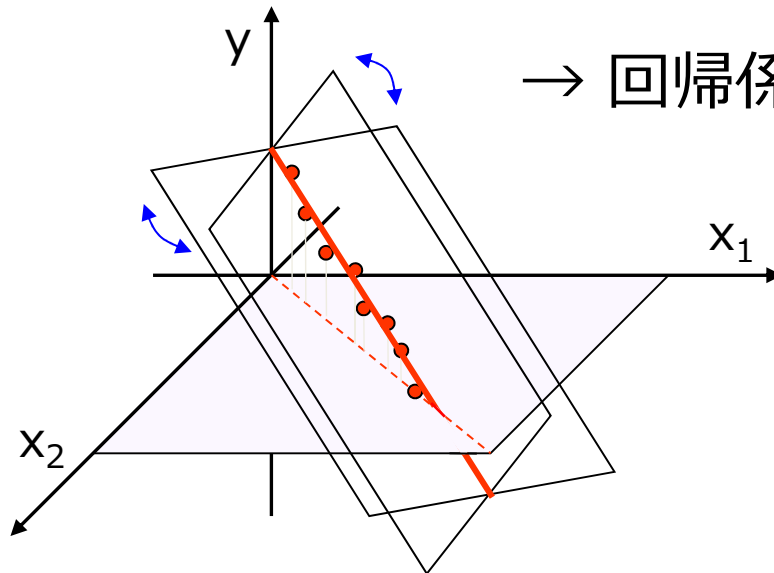
24

✓ 多重共線性の問題

- 説明変数の間に強い相関がある場合、回帰係数が不安定になる
- わずかなデータの変化（追加、削除）で回帰係数が大きく変わってしまう

赤い線を中心に回帰平面が回りやすい

→ 回帰係数が変わりやすい

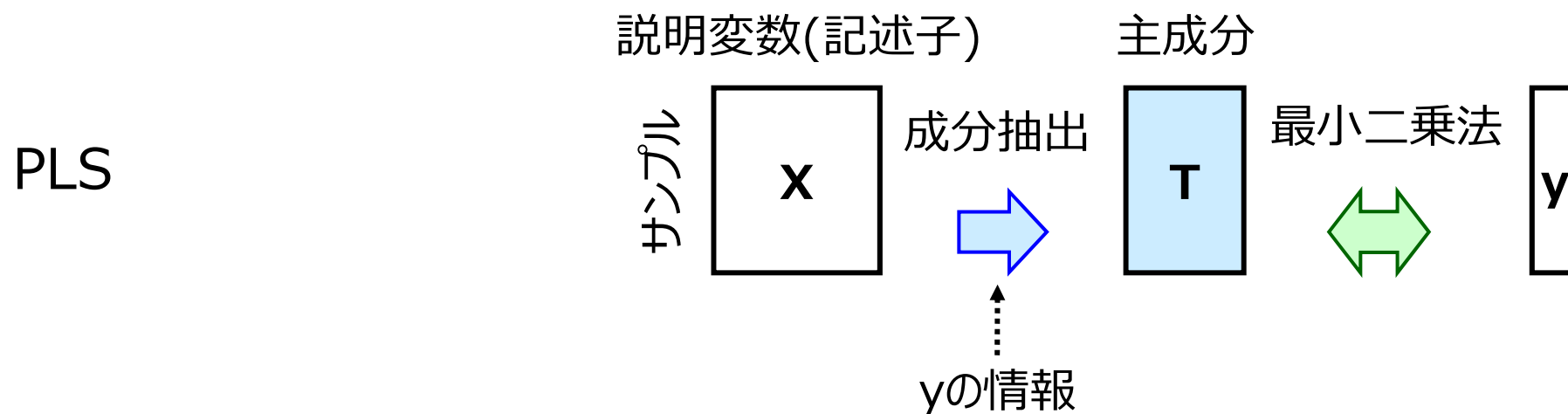
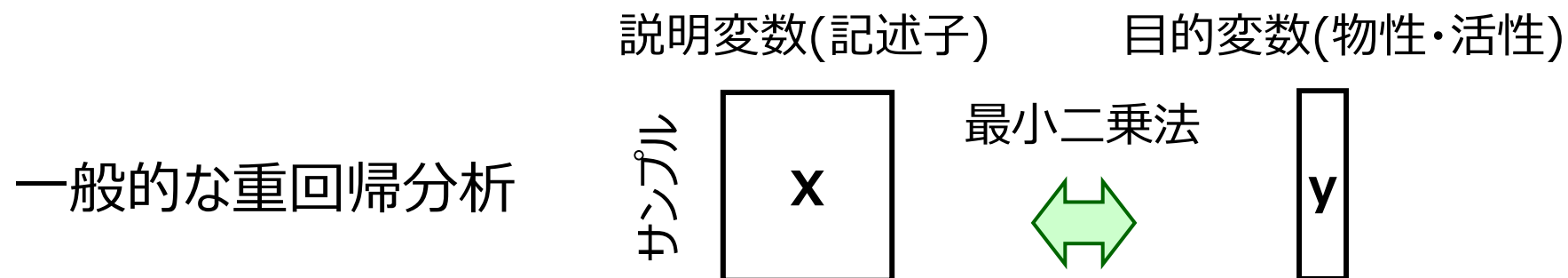


PLS とは？

- ✓線形の回帰分析手法の1つ
- ✓説明変数(記述子)の数がサンプルの数より多くても計算可能
- ✓回帰式を作るときにノイズの影響を受けにくい
- ✓説明変数の間の相関が高くても対応可能
- ✓主成分分析をしたあとの主成分と目的変数との間で最小二乗法を行うのは主成分回帰 (PCR) であり、PLSとは異なるので注意

PLSと一般的な重回帰分析

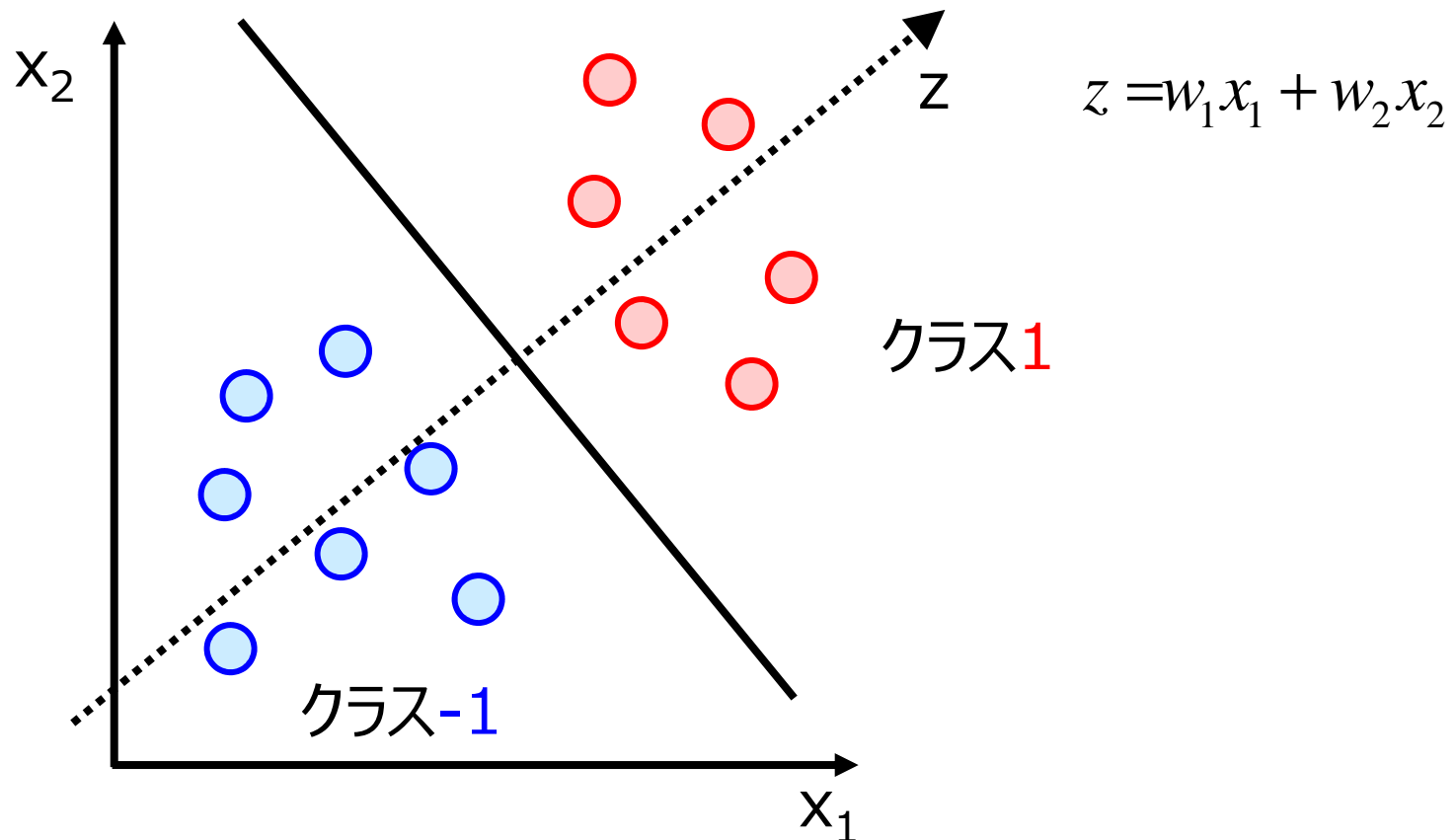
26



線形判別分析 (LDA) とは？

✓ 線形判別分析 (Linear Discriminant Analysis, LDA)

- 2つのクラスを “最もよく判別する” 直線を引く
- 1次元(z)に線形写像し、 z で2つのクラスを識別する
- クラスが3つ以上あるときにも対応できる



“最もよく判別する” とは？

✓① 各クラスのサンプルは固まっている

z でのクラス**内**のばらつき V_{Wz}

$$V_{Wz} = \sum_{i \in \text{クラス}1} \left(z^{(i)} - \bar{z}_{[1]} \right)^2 + \sum_{i \in \text{クラス}-1} \left(z^{(i)} - \bar{z}_{[-1]} \right)^2 \quad \bar{z}_{[k]} : \text{クラス } k \text{ のみの } z \text{ の平均}$$

✓② クラス**1**(**赤**)とクラス**-1**(**青**)は散らばっている

z でのクラス**間**のばらつき V_{Bz}

$$V_{Bz} = \left(\bar{z}_{[1]} - \bar{z}_{[-1]} \right)^2$$

重み w の求め方

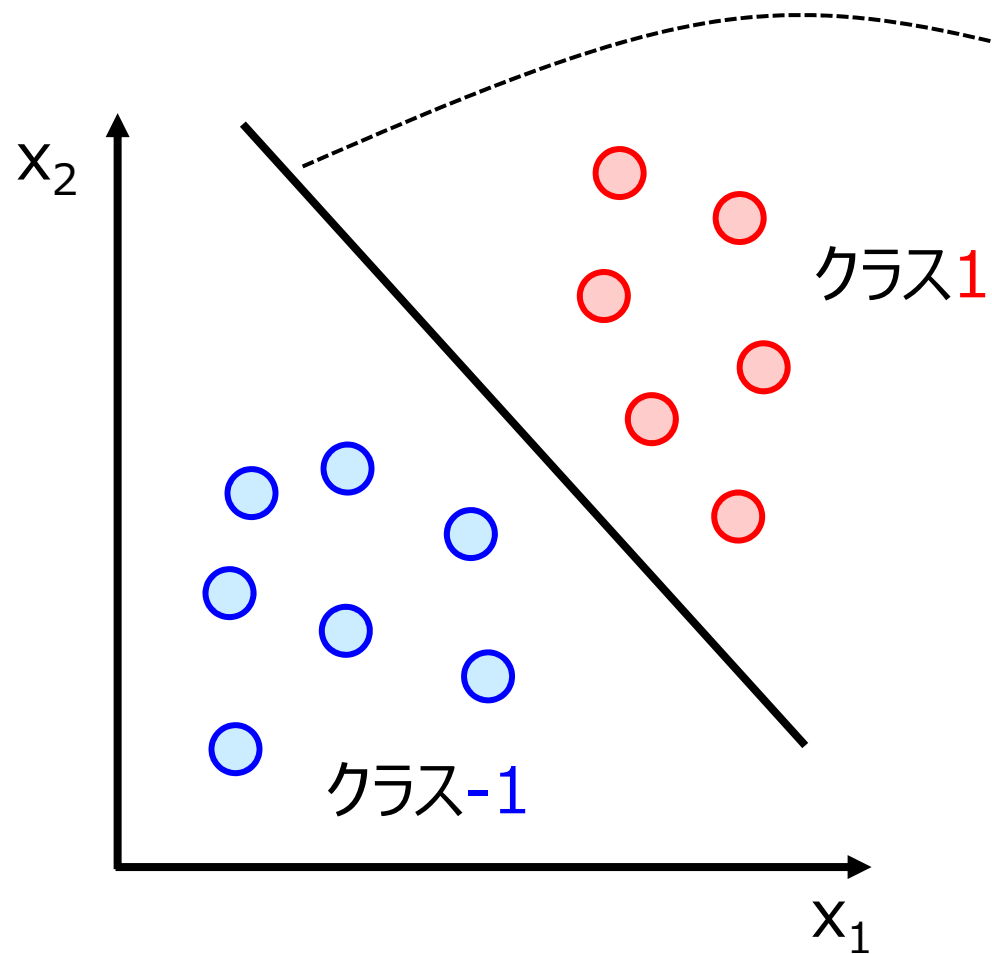
- ✓① 各クラスのサンプルは固まっている
 - z でのクラス**内**のばらつき V_{Wz}
- ✓② クラス**1**(赤) とクラス**-1**(青)は散らばっている
 - z でのクラス**間**のばらつき V_{Bz}

V_{Wz} が小さく(①)、 V_{Bz} が大きくなる(②) 直線を引く (w_1, w_2 を求める)

 $J = \frac{V_{Bz}}{V_{Wz}}$ が最大になる w_1, w_2 を求める

サポートベクターマシン (SVM) とは？

- ✓ 線形判別関数によるクラス分類
- ✓ 2つのクラス (1のクラス・-1のクラス) のどちらに属するか決定
- ✓ 予測能力の高いモデルを作成可能
- ✓ カーネルトリックにより非線形の判別モデルに



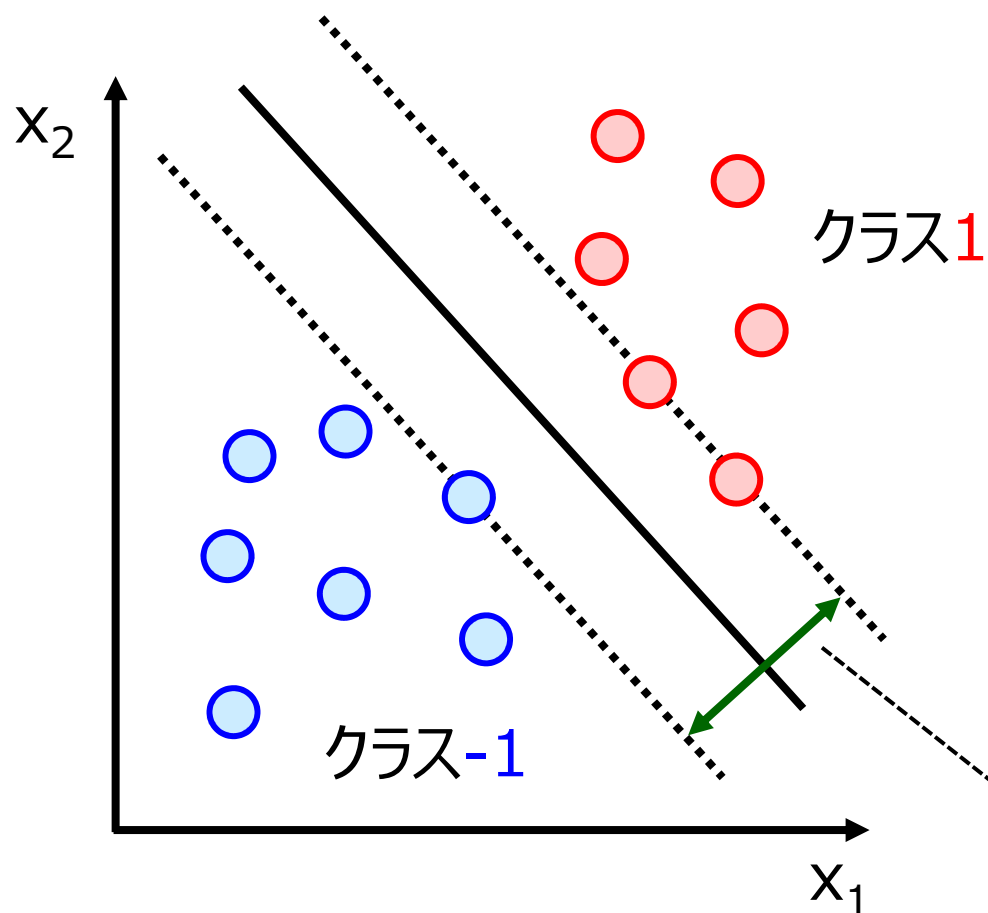
線形判別関数：

$$f(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$
$$= \mathbf{x}\mathbf{w} + b$$

$$\mathbf{x} = [x_1 \ x_2], \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

SVMの基本的な考え方

32



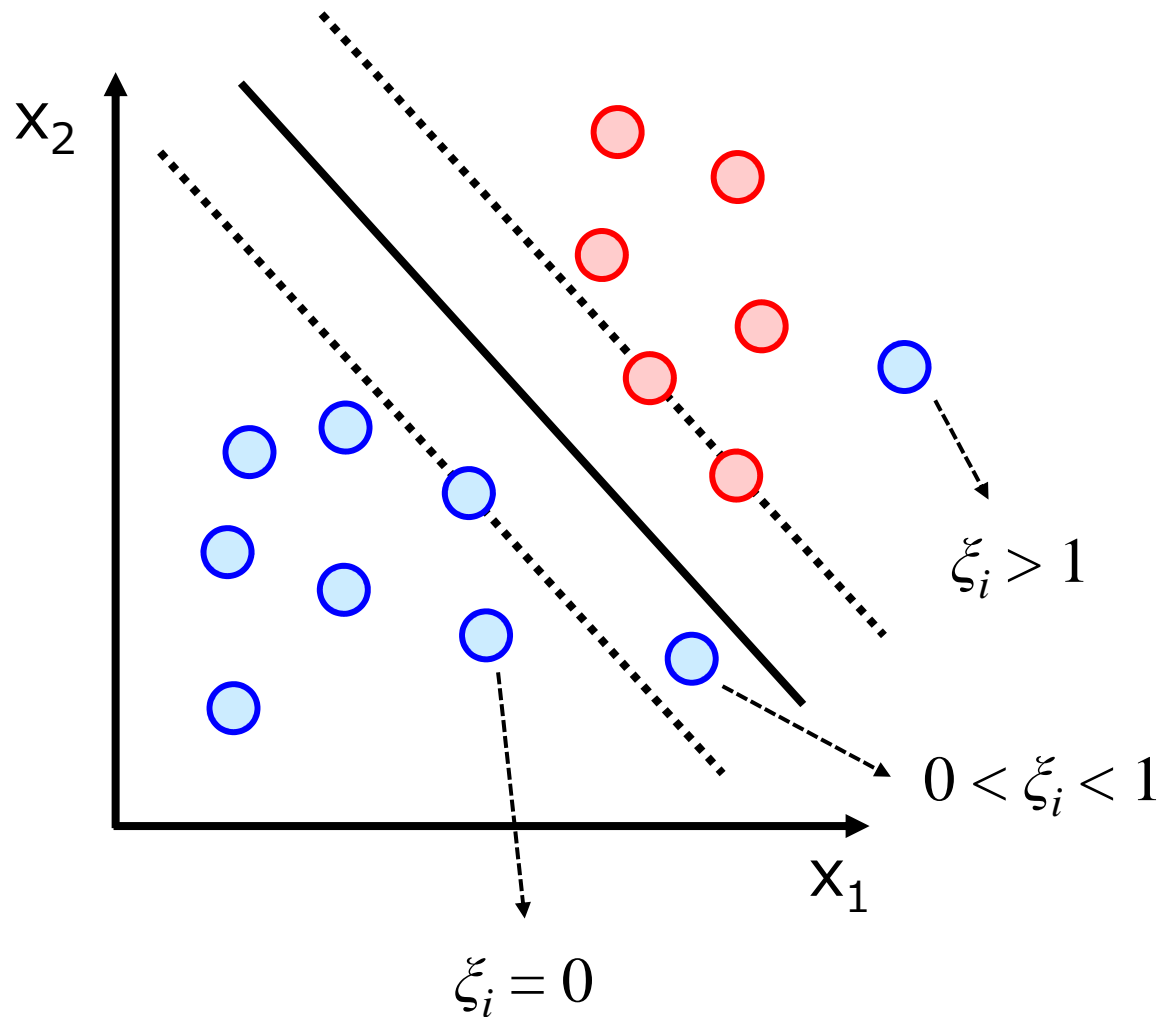
マージンを最大化するように
判別関数を決める！

$$\begin{aligned} f(x_1, x_2) &= w_1 x_1 + w_2 x_2 + b \\ &= \mathbf{xw} + b \end{aligned}$$

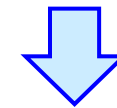
$$\text{マージン} = \frac{2}{\|\mathbf{w}\|} = \frac{2}{\sqrt{w_1^2 + w_2^2}}$$

(点と直線との距離で計算)

きれいに分離できないときは？



スラック変数 ξ を導入！



サンプルごとの ξ_i の和

$$\sum_{i=1}^n \xi_i$$

を最小化

n : モデル構築用
サンプル数

2つの項を一緒に最小化

✓ $\|\mathbf{w}\| / 2$ の最小化 → 計算の都合上、 $\|\mathbf{w}\|^2 / 2$ の最小化

✓ ξ_i の和の最小化

➡ $\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$ の最小化

ただし、 $\xi_i \geq 0, \quad y^{(i)} f(\mathbf{x}^{(i)}) \geq 1 - \xi_i$

C : 2つの項のバランスを決める係数

$\mathbf{x}^{(i)}$: i 番目のサンプルの説明変数

$y^{(i)}$: i 番目のサンプルの値 (1 もしくは -1)

$$f(\mathbf{x}) = \mathbf{x}\mathbf{w} + b$$

$$= \mathbf{x} \sum_{i=1}^n \alpha_i y^{(i)} \mathbf{x}^{(i)\mathrm{T}} + b = \sum_{i=1}^n \alpha_i y^{(i)} \mathbf{x} \mathbf{x}^{(i)\mathrm{T}} + b$$

$$b = \frac{1}{n_S} \sum_{i \in S} \left(y^{(i)} - \sum_{j \in S} \alpha_j y^{(j)} \mathbf{x}^{(i)} \mathbf{x}^{(j)\mathrm{T}} \right)$$

S : サポートベクター ($\alpha_i \neq 0$ のサンプル) の集合

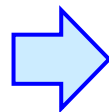
n_S : サポートベクターの個数

線形判別関数は判別能力に限界

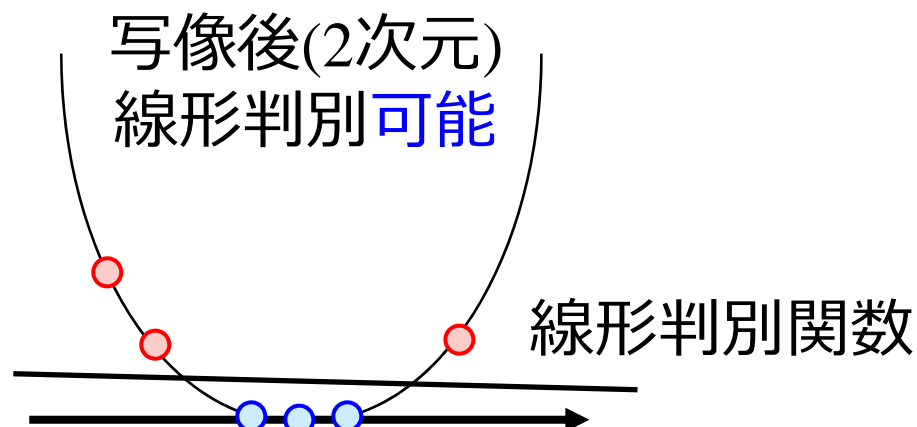
- ➡
- ✓ 元の空間より高次元に写像
 - ✓ 高次元空間上で線形判別関数を構築

高次元空間への写像の例： $x \rightarrow (x, x^2)$

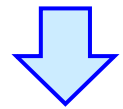
写像前(1次元)
線形判別不能



写像後(2次元)
線形判別可能



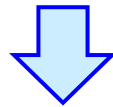
線形判別関数 (元の空間) : $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y^{(i)} \mathbf{x} \mathbf{x}^{(i)T} + b$



高次元空間への写像 : $\mathbf{x} \rightarrow \phi(\mathbf{x})$

非線形判別関数 (高次元空間) : $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y^{(i)} \underbrace{\phi(\mathbf{x}) \phi(\mathbf{x}^{(i)})^T}_{\text{カーネル関数}} + b$

$\phi(\mathbf{x})$ を求める必要はなく、内積 $\phi(\mathbf{x}) \phi(\mathbf{x}^{(i)})^T$ が分かればOK !



高次元空間への写像 $\phi(\mathbf{x})$ ではなく、内積を指定 (カーネル関数 K)

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \phi(\mathbf{x}^{(i)}) \phi(\mathbf{x}^{(j)})^T$$

カーネル関数の例

✓線形カーネル

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \mathbf{x}^{(i)\top} \mathbf{x}^{(j)}$$

✓ガウシアンカーネル (使われることが多い)

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left(-\frac{\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2\right)$$

✓多項式カーネル

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(1 + \lambda \mathbf{x}^{(i)\top} \mathbf{x}^{(j)}\right)^d$$

グリッドサーチ + クロスバリデーション

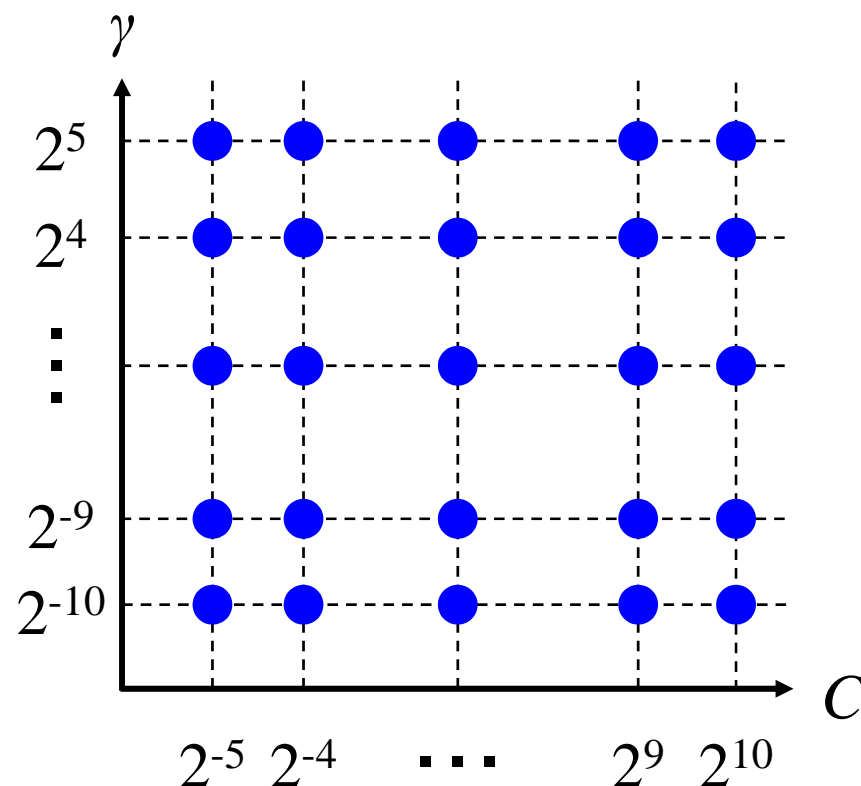
39

✓ C と γ の候補を設定し、すべての組合せ (グリッド, 下図の ●) で
クロスバリデーションを行う

✓ C と γ の候補の例

- $C : 2^{-5}, 2^{-4}, \dots, 2^9, 2^{10}$
- $\gamma : 2^{-10}, 2^{-9}, \dots, 2^4, 2^5$

✓ 例) クロスバリデーション後の
正解率が最も高い
 $C \cdot \gamma$ の組を選択



✓データの前処理

- 標準化 (オートスケーリング)
- 情報量の小さい変数の削除

✓モデリング

- 入門編の復習
- 決定木 (Decision Tree, DT)
- ランダムフォレスト (Random Forests, RF)
- リッジ回帰 (Ridge Regression, RR)
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net (EN)
- Support Vector Regression (SVR)
 - モデルの検証

“良い”回帰モデル・クラス分類モデルとは何か？⁴¹

✓新しいサンプルの目的変数の値・ラベルを、正確に推定できるモデルが
良い回帰モデル・クラス分類モデル

- 回帰モデル・クラス分類モデルを構築したサンプルではないことに注意

✓そのような良いモデルを選ぶために、
いろいろなモデルを評価・比較しなければならない

✓モデルを評価・比較するための、モデルの検証の話です

データセットの呼び方

✓トレーニングデータ (キャリブレーションデータ)

- 回帰モデル・クラス分類モデルの構築に用いるデータ
- 目的変数の値・ラベルは分かっている

✓バリデーションデータ・テストデータ

- 回帰モデル・クラス分類モデルの検証に用いるデータ
- 実際には目的変数の値・ラベルは分かっているが、わからないものとして (目隠し・ブラインドして) モデルから推定し、実際と推定結果とがどれくらいあっているか確認する
 - バリデーションデータで、モデルのハイパーパラメータ (PLSの最適成分数など) を最適化する
 - テストデータで、最終的にモデルの優劣を比較する
 - バリデーションデータはなく、トレーニングデータとテストデータだけのときもある (このときのモデルのハイパーパラメータの最適化については後述)

比較指標

- ✓モデルの性能を評価し、**比較**するための指標
 - 基本的には**比較**だけに用いるのがよく、絶対的な値に意味はない
- ✓トレーニングデータ・バリデーションデータ・テストデータそれぞれについて、実際の目的変数の値・ラベルと、推定された値・ラベルとが揃うと計算できる
- ✓回帰分析
 - 決定係数 r^2
 - 根平均二乗誤差 (Root Mean Squared Error, RMSE)
 - 平均絶対誤差 (Mean Absolute Error, MAE)
など
- ✓クラス分類
 - 混同行列 (confusion matrix) を計算したのちの、正解率、精度、検出率、誤検出率、Kappa係数など

回帰分析 決定係数 r^2

- ✓ 目的変数のばらつきの中で、回帰モデルによって説明できた割合
- ✓ 1に近いほど回帰モデルの“性能”が高い
 - どんな“性能”かは、 r^2 を計算したデータセット・推定値による
- ✓ 相関係数 r を二乗したものとは異なる
- ✓ 異なるデータセットの間で r^2 を比較してはいけない

$$r^2 = 1 - \frac{\sum_{i=1}^n \left(y^{(i)} - y_{\text{EST}}^{(i)} \right)^2}{\sum_{i=1}^n \left(y^{(i)} - y_A \right)^2}$$

$y^{(i)}$: i 番目のサンプルにおける
目的変数の値

$y_{\text{EST}}^{(i)}$: i 番目のサンプルにおける
目的変数の推定値

y_A : 目的変数の平均値

n : サンプル数

回帰分析 RMSE

- ✓ 平均的な誤差の大きさ
- ✓ 0 に近いほど回帰モデルの“性能”が高い
 - どんな“性能”かは、RMSE を計算したデータセット・推定値による
- ✓ 異なるデータセットの間で RMSE を比較してはいけない
- ✓ データセットが同じであれば、 r^2 が大きいほど RMSE は小さい
- ✓ 外れ値（異常に誤差が大きいサンプル）があると、その値の影響を受けやすく、RMSE が大きくなりやすい

$$RMSE = \sqrt{\frac{\sum_{i=1}^n \left(y^{(i)} - y_{\text{EST}}^{(i)} \right)^2}{n}}$$

回帰分析 MAE

- ✓ 平均的な誤差の大きさ
- ✓ 0 に近いほど回帰モデルの“性能”が高い
 - どんな“性能”かは、MAE を計算したデータセット・推定値による
- ✓ 異なるデータセットの間で MAE を比較しないほうがよい
- ✓ 外れ値（異常に誤差が大きいサンプル）の影響を受けにくい

$$MAE = \sqrt{\frac{\sum_{i=1}^n |y^{(i)} - y_{\text{EST}}^{(i)}|}{n}}$$

クラス分類 混同行列・正解率・精度・検出率 ⁴⁷

✓混同行列 (confusion matrix)

		予測されたクラス	
		1 (Positive, 陽性)	-1 (Negative, 陰性)
実際の クラス	1 (Positive, 陽性)	True Positive (TP)	False Negative (FN)
	-1 (Negative, 陰性)	False Positive (FP)	True Negative (TN)

$$\text{正解率} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{検出率} = \frac{TP}{TP + FN}$$

$$\text{精度} = \frac{TP}{TP + FP}$$

$$\text{誤検出率} = \frac{FP}{FP + TN} \quad \text{など}$$

クラス分類 Kappa係数

- ✓ 実際と予測結果の一致度を評価する指標
- ✓ Positive(陽性)データとNegative(陰性)データの偏りがある時に有効

$$\text{Kappa係数} = \frac{\text{正解率} - \text{偶然による一致率}}{1 - \text{偶然による一致率}}$$

$$\text{偶然による一致率} = \frac{\text{TP} + \text{FN}}{A} \times \frac{\text{TP} + \text{FP}}{A} + \frac{\text{FP} + \text{TN}}{A} \times \frac{\text{FN} + \text{TN}}{A}$$

$$(A = \text{TP} + \text{FN} + \text{FP} + \text{TN})$$

http://en.wikipedia.org/wiki/Cohen%27s_kappa

		予測されたクラス	
		1 (Positive, 陽性)	-1 (Negative, 陰性)
実際の クラス	1 (Positive, 陽性)	True Positive (TP)	False Negative (FN)
	-1 (Negative, 陰性)	False Positive (FP)	True Negative (TN)

モデルの評価・比較 ハイパーパラメータの決定

49

✓ハイパーパラメータ

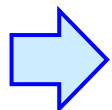
- PLSの最適成分数
- LASSOの λ
- SVMの C 、 γ

など

✓良いモデル (p.1 参照) になるようにハイパーパラメータを決めたい

どのようなハイパーパラメータを用いるか？

- ✓ トレーニングデータの比較指標の値がよくなるようなハイパーパラメータ
 - そもそもモデルがトレーニングデータを用いて構築されているため、トレーニングデータには合うが、新しいサンプルの目的変数を推定できないようなハイパーパラメータが選ばれてしまう
 - 基本的に用いられない
- ✓ バリデーションデータの比較指標の値がよくなるようなハイパーパラメータ
 - 新しいサンプルに対する推定性能を考慮できる
 - データに偏りがないようにトレーニングデータとバリデーションデータとを分けるよう注意する
 - トレーニングデータが少なくなってしまう
 - ハイパーパラメータを決めた後、バリデーションデータも合わせて再度モデルを構築する
 - 十分にデータ数が多いとき以外は、あまり用いられない

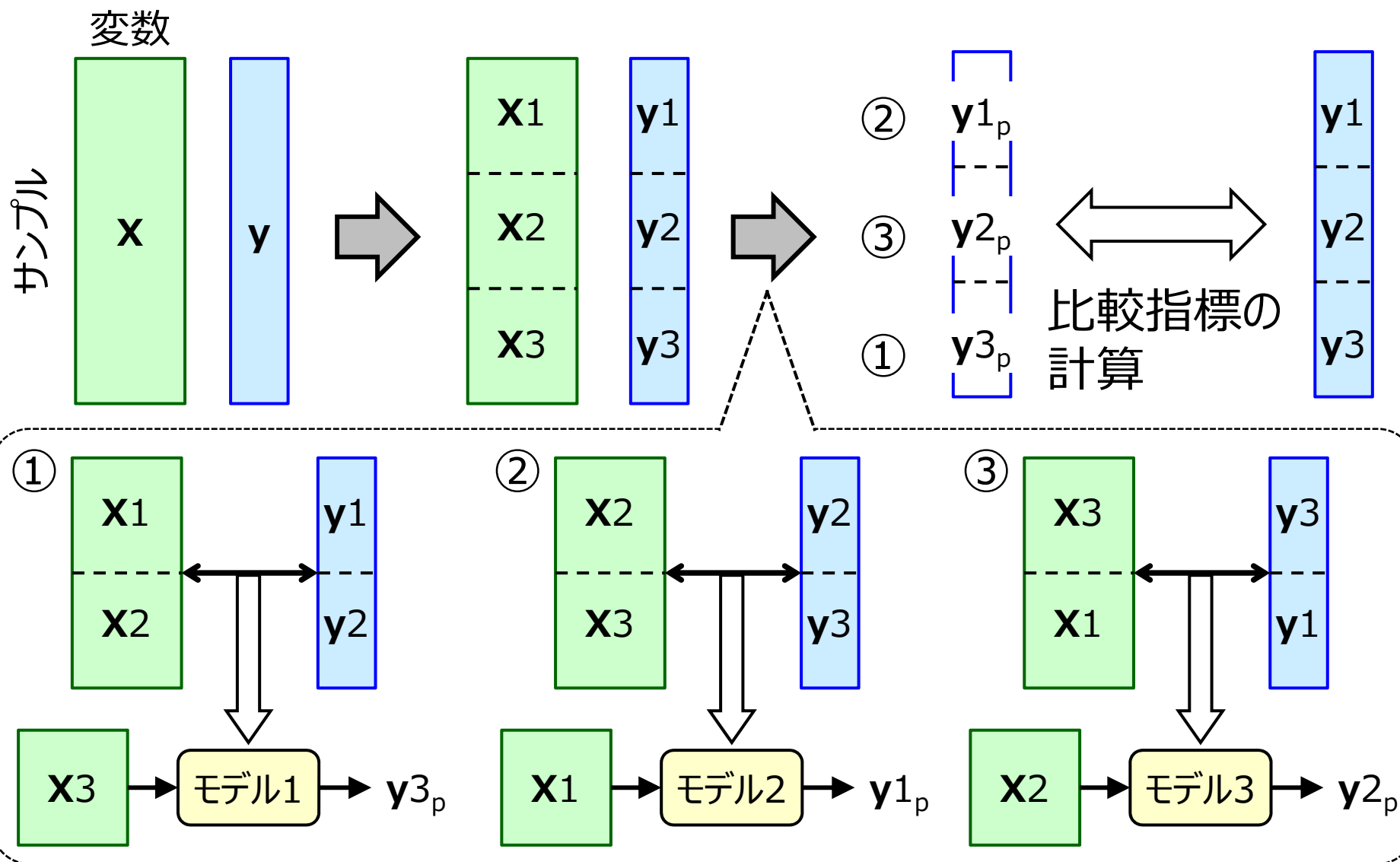


クロスバリデーション

クロスバリデーション

51

✓例) 3-fold クロスバリデーション



クロスバリデーションの補足

✓Leave-one-out クロスバリデーション

- サンプルを1つ除いて、残りのサンプルでモデルを構築し、除いたサンプルを推定する、ということをサンプル数だけ繰り返す
- 特にサンプル数が多いときに、すべてのサンプルでモデルを構築し、すべてのサンプルを推定することと似てしまうため、望ましくない

✓2-fold, 5-fold, 10-foldが一般的

✓データ数が多すぎると、計算時間がかかりすぎてしまうときは、 トレーニングデータとバリデーションデータとを分ける方法を用いる

どのようにデータセットを分けるか？

- ✓ トレーニングデータ・バリデーションデータ・テストデータで、サンプルに偏りが無い方がよい
 - 基本的にランダムに分けるのでOK

- ✓ トレーニングデータはなるべくばらついている方がよい
 - Kennard-Stone (KS) アルゴリズムにより、トレーニングデータ・バリデーションデータ・テストデータの順に選ぶ
 1. データセットの説明変数の平均を計算
 2. 平均とのユークリッド距離が一番大きいサンプルを選択
 3. 選択されていない各サンプルにおいて、これまで選択されたすべてのサンプルとの間でユークリッド距離を計算し、その中の最小値を代表距離とする
 4. 代表距離が最も大きいサンプルを選択する
 5. 3. と 4. とを繰り返す

Y-randomization (Yランダムイゼーション)

- ✓特に、サンプル数が少なく説明変数（記述子）の数が多いとき、
本当は X と y の間に相関関係がなくても、 r^2 , r^2_{cv}
(クロスバリデーションのときの r^2) の値が大きくなってしまうことがある
 - たまたま X のノイズと y との間で相関がでてしまう
 - 偶然の相関
- ✓偶然の相関かどうかを見分けるため、Y-randomizationが行われる
 - Y のみ値をランダムに並べかえて、おかしいデータセットにする
 - モデリングして、 r^2 , r^2_{cv} の値が 0 付近になることを確認する

✓データの前処理

- 標準化 (オートスケーリング)
- 情報量の小さい変数の削除

✓モデリング

- 入門編の復習
- 決定木 (Decision Tree, DT)
- ランダムフォレスト (Random Forests, RF)
- リッジ回帰 (Ridge Regression, RR)
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net (EN)
- Support Vector Regression (SVR)
 - モデルの検証

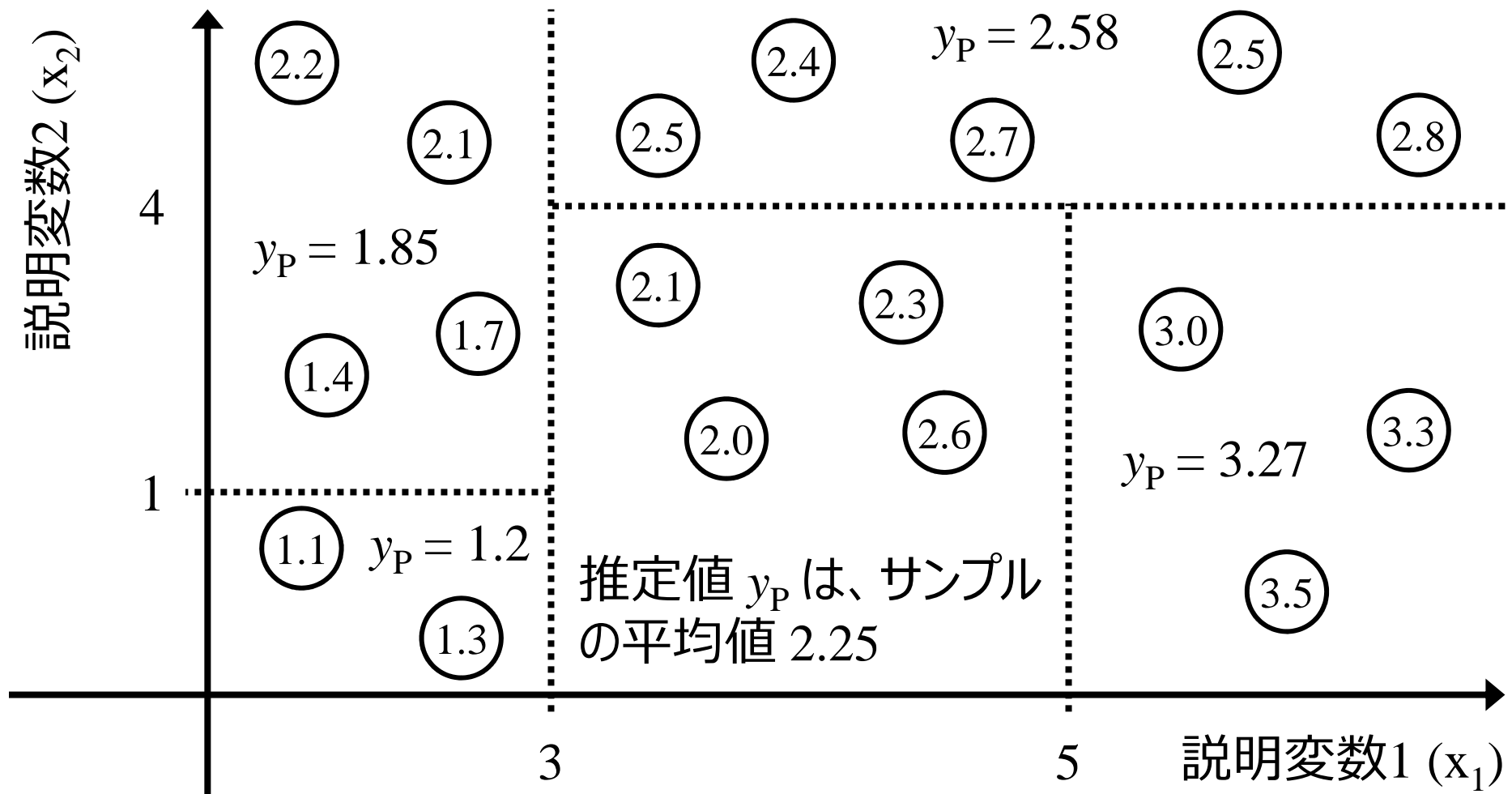
決定木 (Decision Tree, DT) とは？

- ✓ 回帰分析にもクラス分類にも使える
- ✓ 回帰モデル・クラス分類モデルが、木のような構造で与えられるため、モデルを直感的に理解しやすい
- ✓ 理解しやすい反面、モデルの精度は他の手法と比べて低くなってしまふことが多い
- ✓ 今回説明するのは CART (Classification and Regression Tree)

決定木でできることのイメージ (回帰分析)

57

(n) ... 目的変数 y の値が n のサンプル

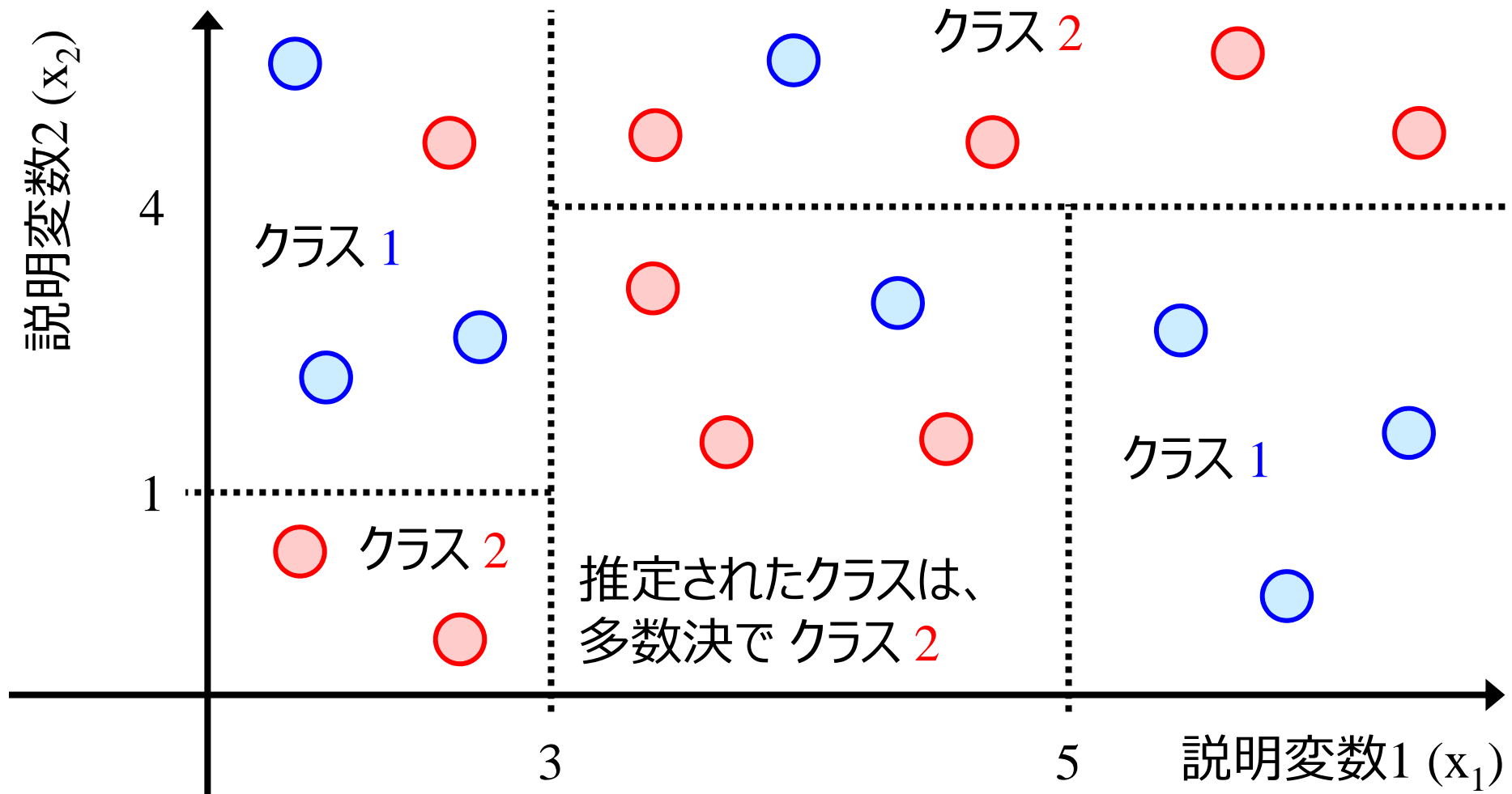


決定木のでできることのイメージ (クラス分類)

58

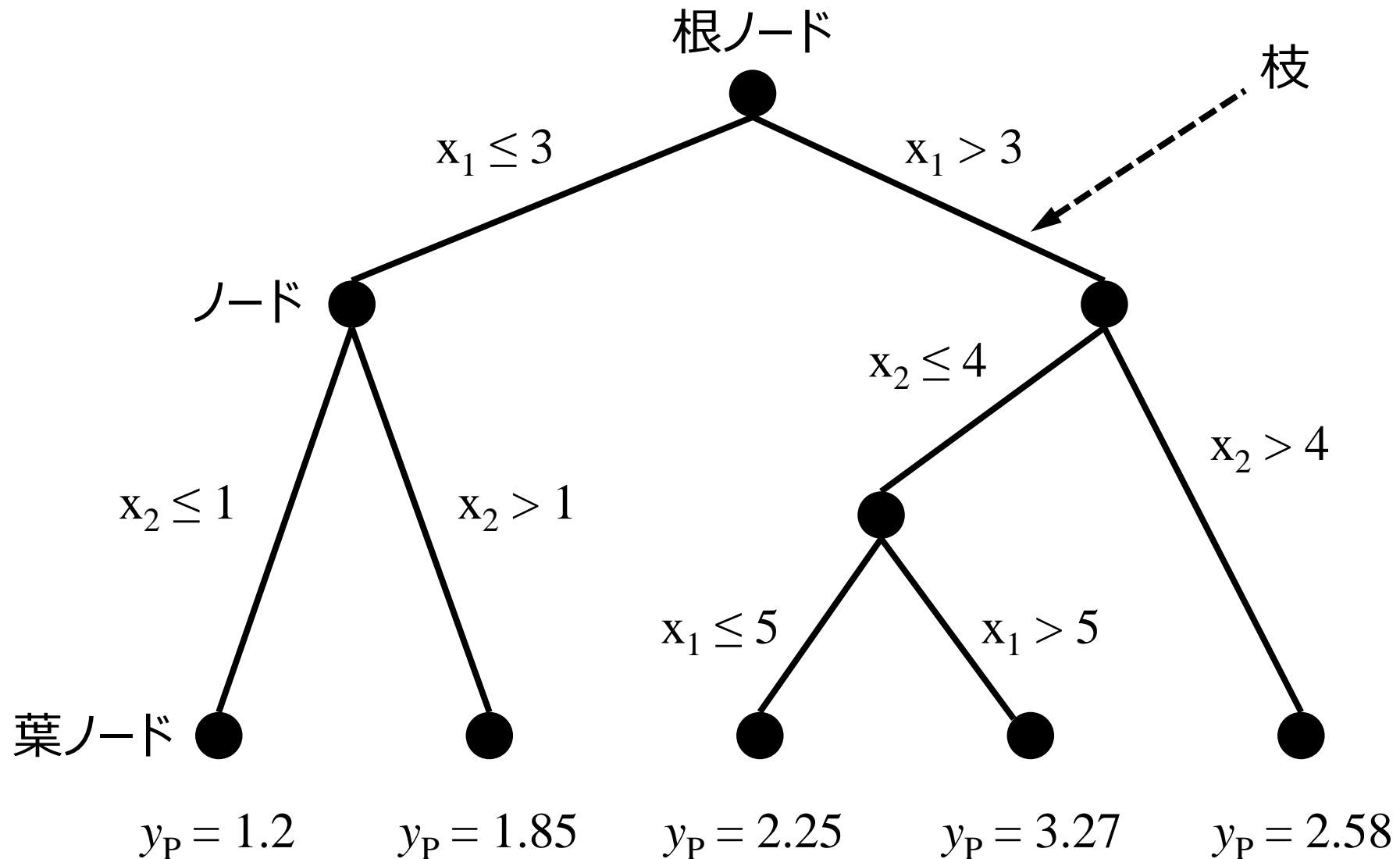
○ … クラスが 1 のサンプル

○ … クラスが 2 のサンプル



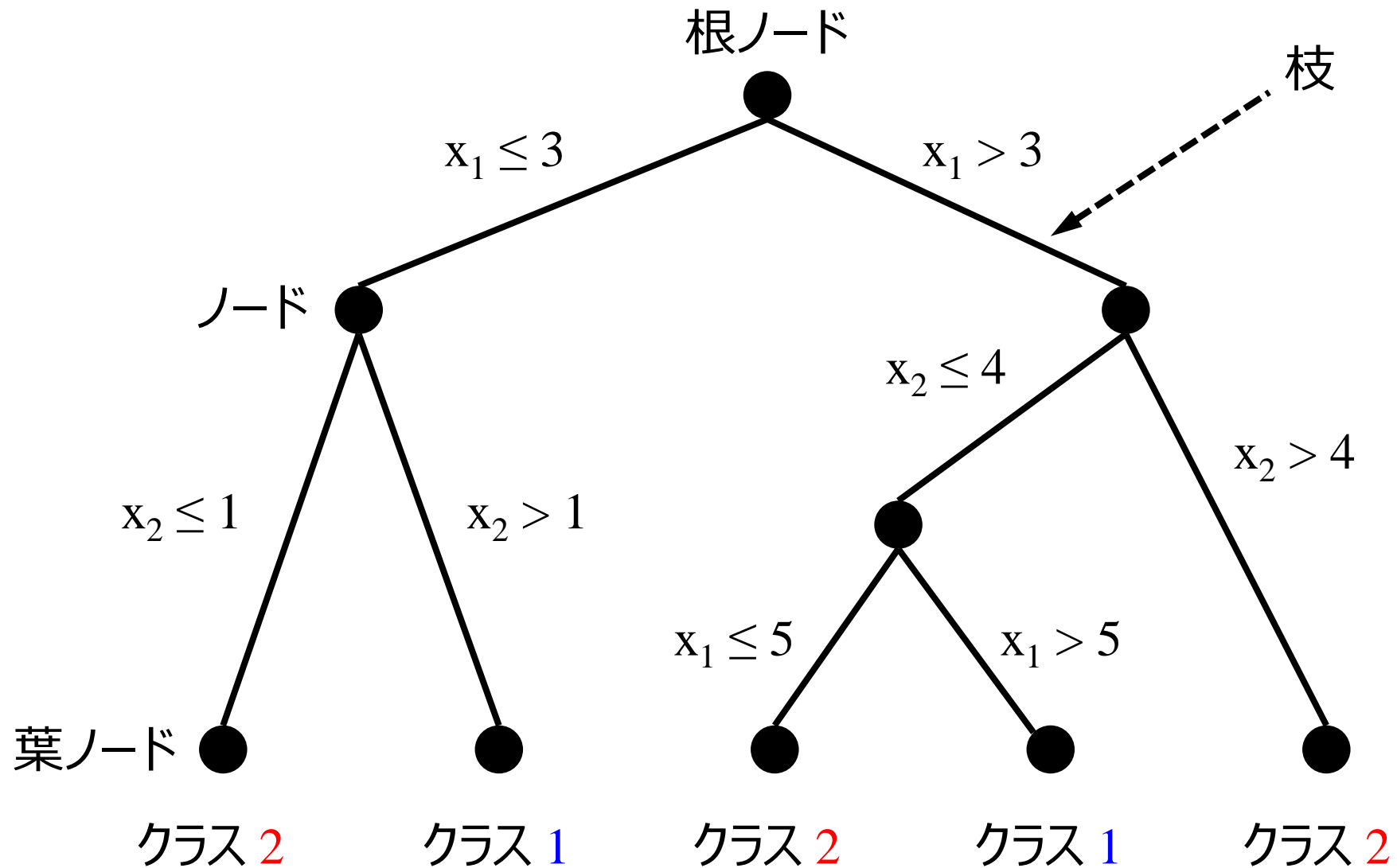
決定木モデルの木構造 (回帰分析)

59



決定木モデルの木構造 (クラス分類)

60



決定木のアルゴリズム

✓どのように木を作るか？

- 根ノードから、2つずつノードを追加していき、木を成長させる

✓どのように2つのノードを追加するか？

✓つまり、どのように説明変数を選んで、どのようにしきい値を選ぶか？

- 説明変数としきい値とのすべての組み合わせにおいて、
評価関数 E の値を計算し、それが最も小さい組み合わせにする

回帰分析における評価関数 E

✓ 目的変数の誤差の二乗和

- それぞれの葉ノードにおける目的変数の推定値は、同じ葉ノードにあるサンプルの平均値で与えられる

$$E = \sum_{i=1}^n E_i$$

$$E_i = \sum_{j=1}^{m_i} \left(y_i^{(j)} - y_{Pi} \right)^2$$

$$y_{Pi} = \frac{1}{m_i} \sum_{j=1}^{m_i} y_i^{(j)}$$

n : 葉ノードの数

E_i : 葉ノード i の評価関数

m_i : 葉ノード i におけるサンプル数

$y_j^{(i)}$: 葉ノード i における、 j 番目のサンプルの目的変数の値

y_{Pi} : 葉ノード i における目的変数の推定値

クラス分類における評価関数 E

✓交差エントロピー誤差関数

$$E_i = -\sum_{k=1}^K p_{ik} \ln p_{ik}$$

K : クラスの数

p_{ik} : 葉ノード i における、クラス k の
サンプルの割合

✓ジニ係数

$$E_i = \sum_{k=1}^K p_{ik} (1 - p_{ik})$$

いずれも、

$$E = \sum_{i=1}^n E_i$$

(ジニ係数のほうが
よく使われるかな・・・)

いつ木の成長を止めるか？

✓ クロスバリデーションの誤差が最小になるように深さを決める

✓ 1つの葉ノードにおける最小サンプル数を決め（3とか）、
とりあえずすべて木を生成させる

✓ 葉ノードを2つずつ枝刈りしていく

- 下の基準 C が大きくなったら枝刈りストップ

$$C = E + \lambda n$$

E : 評価関数

n : 葉ノードの数

λ : 木の精度と複雑度との間の
トレードオフを決める重み

- λ はクロスバリデーションで決める

内容 1/2

✓データの前処理

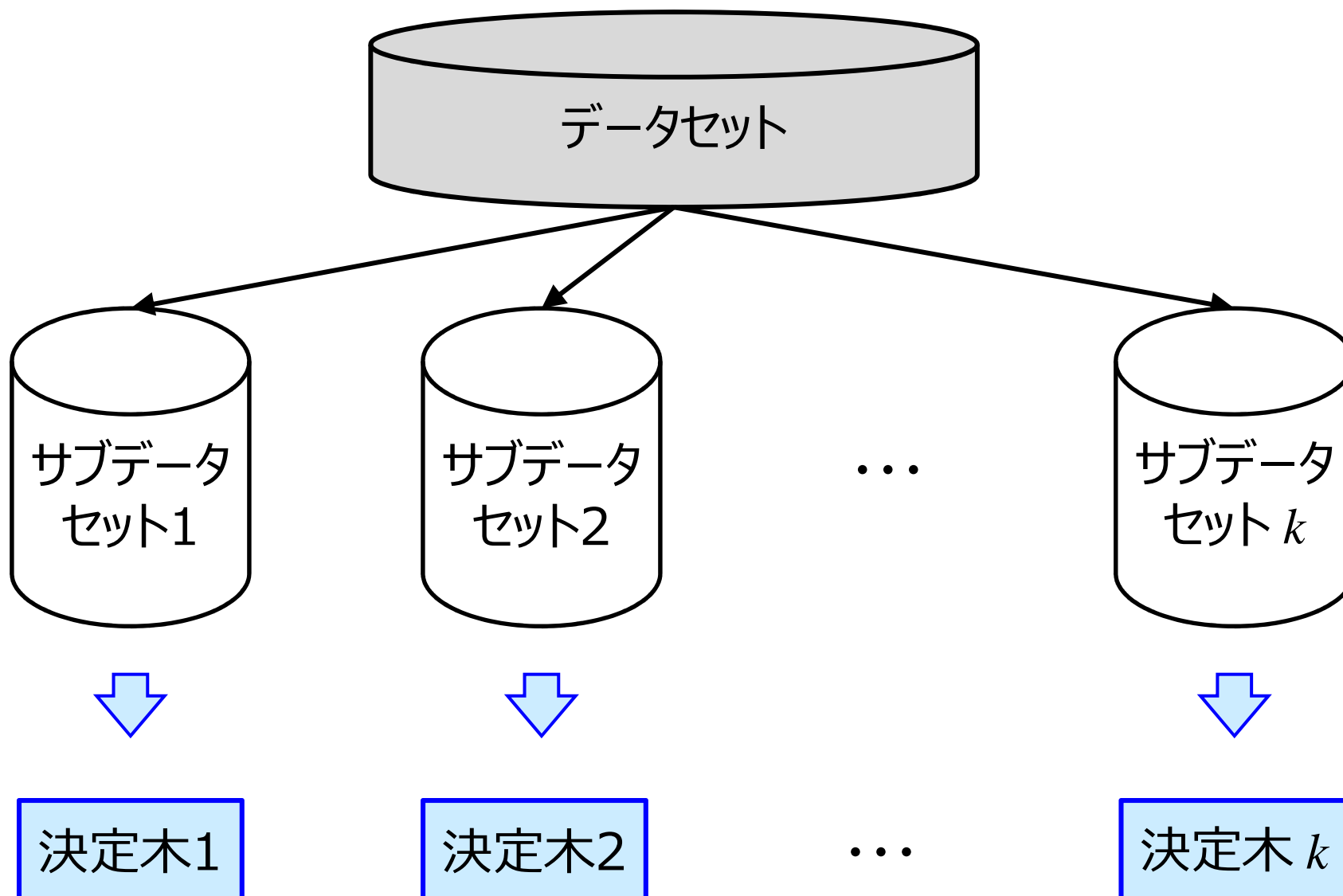
- 標準化 (オートスケーリング)
- 情報量の小さい変数の削除

✓モデリング

- 入門編の復習
- 決定木 (Decision Tree, DT)
- ランダムフォレスト (Random Forests, RF)
- リッジ回帰 (Ridge Regression, RR)
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net (EN)
- Support Vector Regression (SVR)
 - モデルの検証

Random Forest (RF) とは？

- ✓ サンプルと説明変数とをランダムにサンプリングして、決定木をたくさん作る
- ✓ 複数の決定木の推定結果を統合して、最終的な推定値とする
- ✓ アンサンブル(集団)学習 (Ensemble learning) の 1 つ
- ✓ 決定木と比べて精度は高くなることが多いが、モデルを解釈することは難しい
- ✓ 回帰分析にもクラス分類にも使える
- ✓ 説明変数の重要度を議論できる



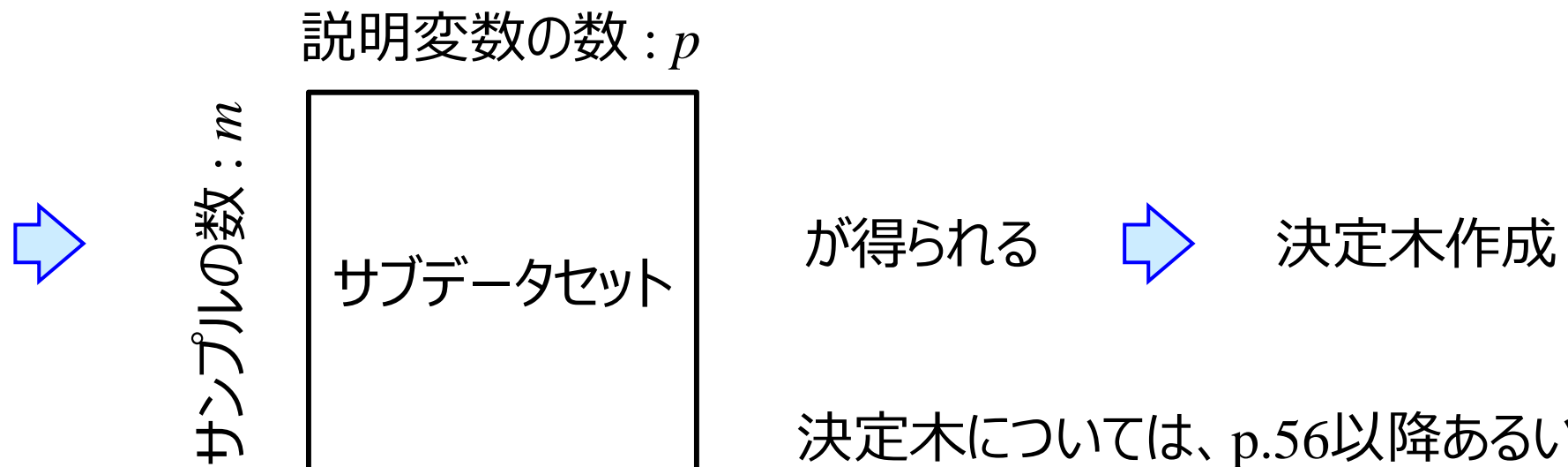
どのようにサブデータセットを作るか？

✓ データセットのサンプル数: m

- サンプルを重複を許してランダムに m 個選択

✓ データセットの説明変数(記述子)の数: n

- 説明変数を重複を許さずランダムに p 個選択



決定木については、p.56以降あるいは

<http://atachemeng.com/decisiontree/>

サブデータセットの数・説明変数の数はどうする？⁶⁹

✓グリッドサーチ + クロスバリデーション

✓サブデータセットの数の候補 例

- 100, 200, 300, 400, 500

✓説明変数の数の候補 例

- データセットにおける説明変数(記述子)の数の
10, 20, ..., 80, 90 %

どのように推定結果を統合するか？

✓回帰分析

- k 個の推定値の平均値

✓クラス分類

- k 個のクラス分類結果で多数決

Out-Of-Bag (OOB)

✓サブデータセットを作るとき、 m 個のサンプルから重複を許して m 個のサンプルを選択

- ➡ サブデータセットごとに、選ばれなかったサンプル (Out-Of-Bag, OOB) が存在
- ➡ OOBにより、各決定木の予測性能を検討可能

OOBを用いた説明変数（記述子）の重要度 ⁷²

✓説明変数（記述子）の重要度 I_j

$$I_j = \frac{1}{k} \sum_{i=1}^k (F_i - E_i)$$

k : サブデータセットの数 (決定木の数)

E_i : i 番目の決定木において、OOBを推定したときの

✓ 平均二乗誤差 (回帰分析)

✓ 誤分類率 (クラス分類)

F_i : i 番目の決定木を**作成した後に**、説明変数を**ランダムに並び替えて**、OOBを推定したときの

✓ 平均二乗誤差 (回帰分析)

✓ 誤分類率 (クラス分類)

E_i が小さいほど、 F_i が大きいほど、 I_j が大きい
→ j 番目の説明変数（記述子）の重要度が高い

✓データの前処理

- 標準化 (オートスケーリング)
- 情報量の小さい変数の削除

✓モデリング

- 入門編の復習
- 決定木 (Decision Tree, DT)
- ランダムフォレスト (Random Forests, RF)
- リッジ回帰 (Ridge Regression, RR)
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net (EN)
- Support Vector Regression (SVR)
 - モデルの検証

RR・LASSO・EN とは？

- ✓線形の回帰分析手法
- ✓目的変数の誤差の二乗和に加えて、それぞれ以下の項を最小化することで、過学習を防ぐ
- ✓RR: 回帰係数の二乗和
- ✓LASSO: 回帰係数の絶対値の和
- ✓EN: 回帰係数の二乗和と絶対値の和 (RRとLASSOとの中間)
- ✓LASSOとENは回帰係数の値が0になりやすく、変数選択としても利用できる

- ✓ 最小二乗法による線形重回帰分析
(Ordinary Least Squares, OLS)
- ✓ リッジ回帰 (Ridge Regression, RR)
- ✓ Least Absolute Shrinkage and Selection Operator (LASSO)
- ✓ Elastic Net (EN)
- ✓ サポートベクター回帰
(Support Vector Regression, SVR)

✓線形の回帰分析手法

- たとえば説明変数が2つのとき、目的変数・説明変数をオートスケーリングしたあと、

$$y = x_1 b_1 + x_2 b_2 + f$$

$$= y_C + f$$

$$(y_C = x_1 b_1 + x_2 b_2)$$

y : 目的変数

x_1, x_2 : 説明変数 (記述子)

b_1, b_2 : (標準)回帰係数

y_C : y の、 x で表すことができる部分

f : y の、 x で表すことができない部分
(誤差、残差)

と表わされる

- ✓ある関数 G を最小化することで回帰係数を求める

OLS・RR・LASSO・EN・SVRの違い 1/2

77

✓OLS: G は誤差の二乗和

$$G = \sum_{i=1}^n f_i^2 = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$$

n : サンプル数

f_i : i 番目のサンプルの誤差

行列の表し方については[こちら](#)

✓RR: G は誤差の二乗和と回帰係数の二乗和

$$G = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda \sum_{i=1}^m b_i^2$$

m : 説明変数の数

b_i : i 番目の説明変数の回帰係数

λ : 重み

✓LASSO: G は誤差の二乗和と回帰係数の絶対値の和

$$G = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda \sum_{i=1}^m |b_i|$$

OLS・RR・LASSO・EN・SVRの違い 2/2

78

✓EN: G は誤差の二乗和と回帰係数の二乗和と絶対値の和

$$G = \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \lambda \left(\alpha \sum_{i=1}^m b_i^2 + (1-\alpha) \sum_{i=1}^m |b_i| \right)$$

α : 重み
($\alpha=1 \rightarrow$ RR,
 $\alpha=0 \rightarrow$ LASSO)

✓SVR: G はある誤差関数 h と回帰係数の二乗和

$$G = h(\mathbf{y} - \mathbf{X}\mathbf{b}) + \lambda \sum_{i=1}^m b_i^2$$

- h についてはSVRの資料のときに

回帰係数の求め方

79

G が最小値を取る



G が極小値を取る



G を 各 b_i で偏微分したものが 0

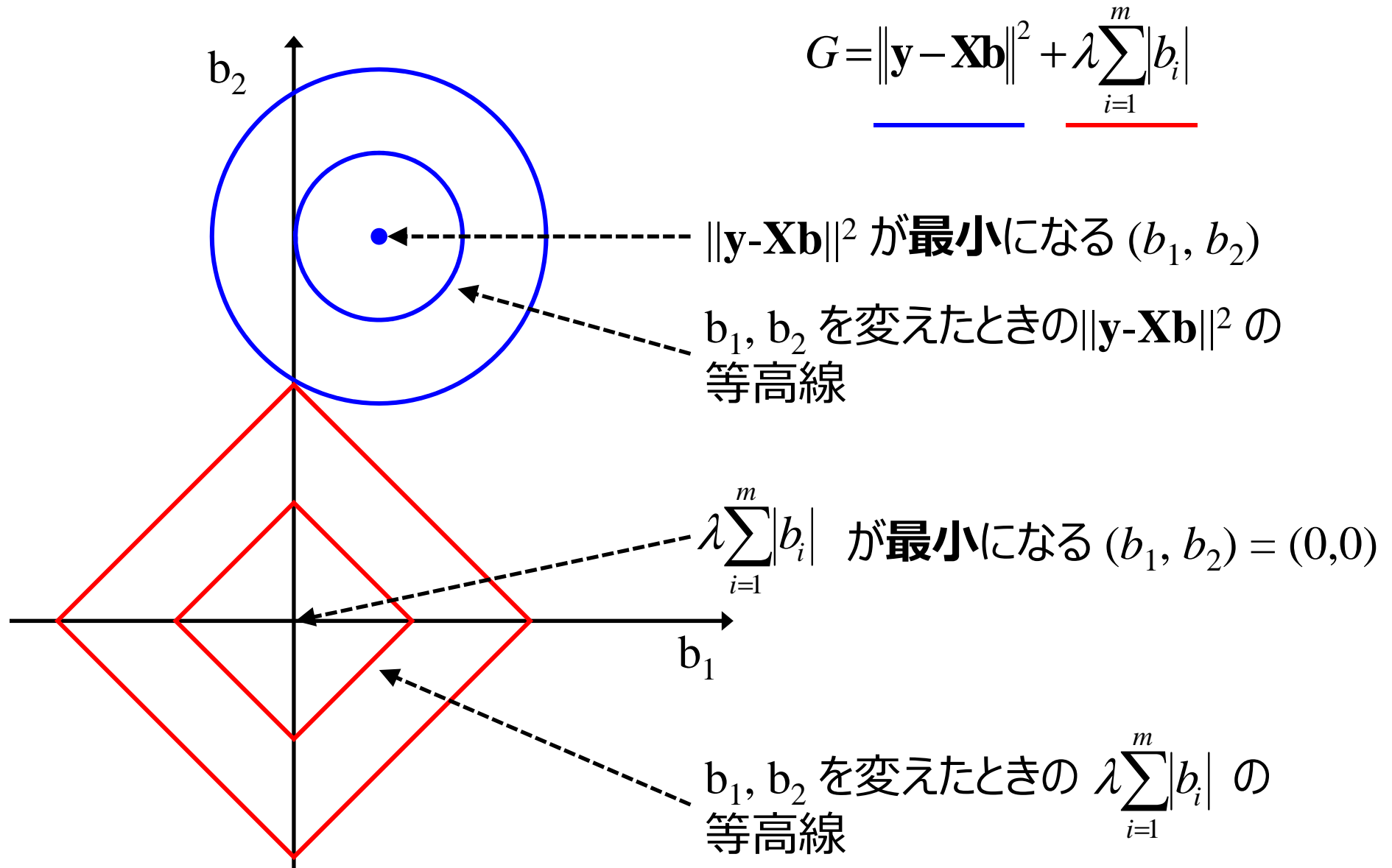
$$\frac{\partial G}{\partial b_i} = 0$$



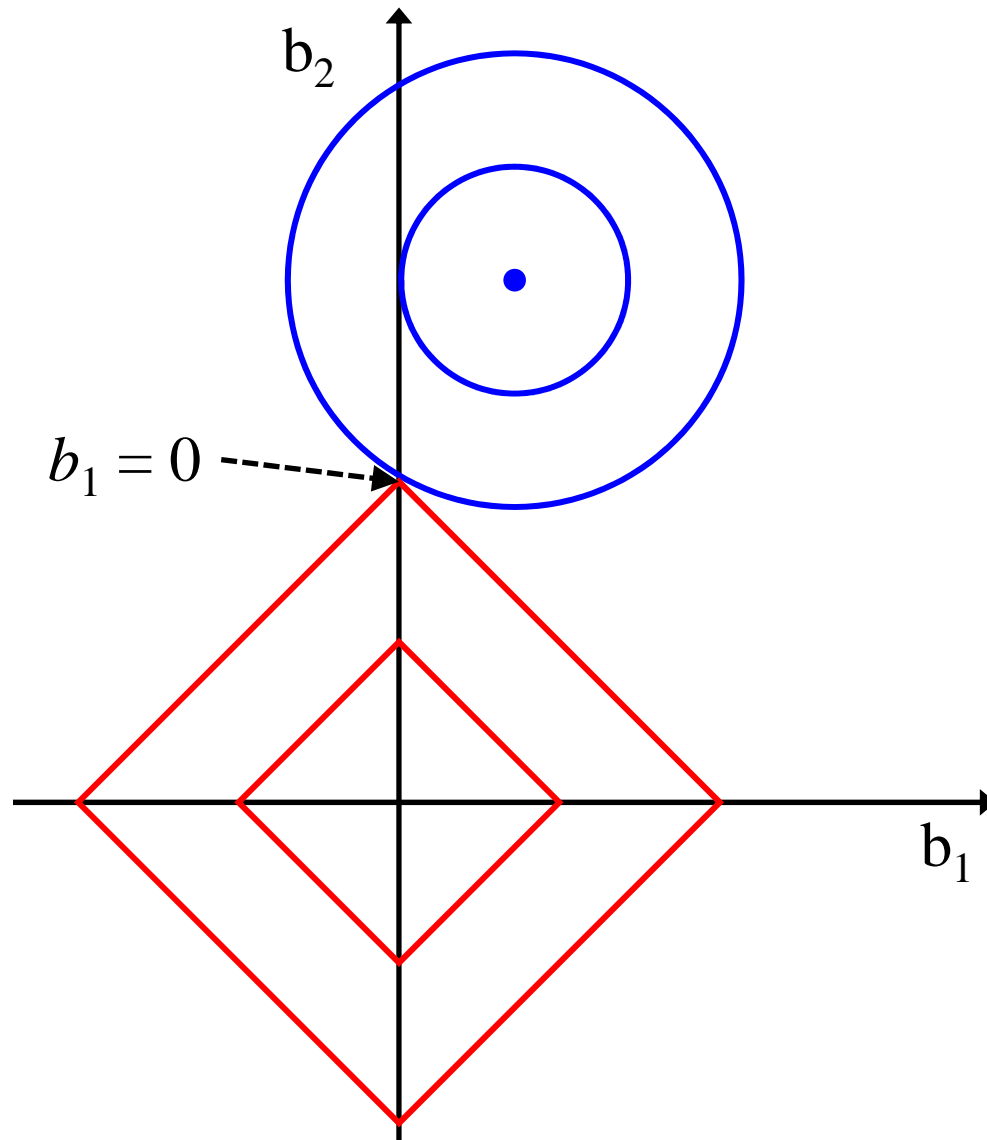
必要に応じて繰り返し計算により、

$\frac{\partial G}{\partial b_i} = 0$ を満たす各 b_i を求める

どうしてLASSOは回帰係数が0になりやすいの？⁸⁰



どうしてLASSOは回帰係数が0になりやすいの？⁸¹



$$G = \underbrace{\|y - \mathbf{X}\mathbf{b}\|^2}_{\text{blue line}} + \underbrace{\lambda \sum_{i=1}^m |b_i|}_{\text{red line}}$$

○ と ◇ との交点が、

G が最小になる (b_1, b_2)



◇ の角が軸上にあるため

b_1 もしくは b_2 が 0 になりやすい

(ENも回帰係数が0になりやすい)

重み λ , α の決め方

- ✓ グリッドサーチによって、クロスバリデーションの後の r^2 の値がもっとも高い λ (RR, LASSO) もしくは λ と α の組み合わせ (EN) とする
- ✓ RRにおける λ の候補の例: 0.01, 0.02, ..., 0.69, 0.7
- ✓ LASSOにおける λ の候補の例: 0.01, 0.02, ..., 0.69, 0.7
- ✓ ENにおける λ の候補の例: 0.01, 0.02, ..., 0.69, 0.7
- ✓ ENにおける α の候補の例: 0, 0.01, ..., 0.99, 1

✓データの前処理

- 標準化 (オートスケーリング)
- 情報量の小さい変数の削除

✓モデリング

- 入門編の復習
- 決定木 (Decision Tree, DT)
- ランダムフォレスト (Random Forests, RF)
- リッジ回帰 (Ridge Regression, RR)
- Least Absolute Shrinkage and Selection Operator (LASSO)
- Elastic Net (EN)
- Support Vector Regression (SVR)
 - モデルの検証

サポートベクター回帰 (SVR) とは？

- ✓ 線形の回帰分析手法
- ✓ サポートベクターマシン (SVM) を回帰分析に応用
- ✓ 目的変数の誤差に加えて、それぞれ以下の項を最小化することで、過学習を防ぐ
 - リッジ回帰 (RR)・LASSO・Elastic Net (EN) と同じ
 - RR・LASSO・ENとの共通点は[こちら](#)
- ✓ 誤差に不感帯を設けることでノイズの影響を受けにくい
- ✓ カーネルトリックにより非線形の回帰モデルに

基本的にSVRは線形の回帰分析手法

✓線形の回帰分析手法

- たとえば説明変数が2つのとき、目的変数・説明変数をオートスケーリングしたあと、

$$y = x_1 b_1 + x_2 b_2 + f$$
$$= y_c + f$$

$$(y_c = x_1 b_1 + x_2 b_2)$$

y : 目的変数

x_1, x_2 : 説明変数 (記述子)

b_1, b_2 : (標準)回帰係数

y_c : y の、 x で表すことができる部分

f : y の、 x で表すことができない部分
(誤差、残差)

と表わされる

回帰係数 \mathbf{b}

✓回帰係数のベクトル \mathbf{b} を

$$\mathbf{b} = [b_1 \quad b_2 \quad \cdots \quad b_m] \quad m : \text{説明変数(記述子)の数}$$

とする

✓あるサンプル (i 番目のサンプル) の目的変数の推定値 $f(\mathbf{x}^{(i)})$ は

$$f(\mathbf{x}^{(i)}) = \mathbf{x}^{(i)} \mathbf{b} \quad \mathbf{x}^{(i)} : \text{あるサンプル (} i \text{ 番目の} \\ \text{サンプル) の説明変数 (記述子)}$$

と表わされる

非線形の回帰モデルへ

線形回帰モデル (元の空間) : $f(\mathbf{x}^{(i)}) = \mathbf{x}^{(i)} \mathbf{b}$



高次元空間への写像 (非線形写像) : $\mathbf{x} \rightarrow \phi(\mathbf{x})$

非線形回帰モデル関数 (高次元空間) : $f(\mathbf{x}^{(i)}) = \phi(\mathbf{x}^{(i)}) \mathbf{w} + c$

$$\mathbf{w} = [w_1 \quad w_2 \quad \cdots \quad w_k]$$

c : 定数項

w_i : 重み

k : 高次元空間での次元数

w_i, k は、とりあえずこのように設定しておくだけで、
後に考えなくてもよくなるため、気にしなくて問題ない

SVMとSVRとの比較

88

$$\checkmark \text{SVM} \quad \underbrace{\frac{1}{2} \|\mathbf{w}\|^2}_{\text{マージンの最大化 (重みの最小化)}} + \underbrace{C \sum_{i=1}^n \xi_i}_{\text{誤分類するサンプル数の最小化 (誤差の最小化)}} \quad \text{の最小化}$$

マージンの最大化
(重みの最小化)

誤分類するサンプル数の
最小化 (誤差の最小化)

$$\checkmark \text{SVR} \quad \underbrace{\frac{1}{2} \|\mathbf{w}\|^2}_{\text{重みの最小化}} + \underbrace{C \sum_{i=1}^n h\left(y^{(i)} - f\left(\mathbf{x}^{(i)}\right)\right)}_{\text{誤差の最小化}} \quad \text{の最小化}$$

重みの最小化

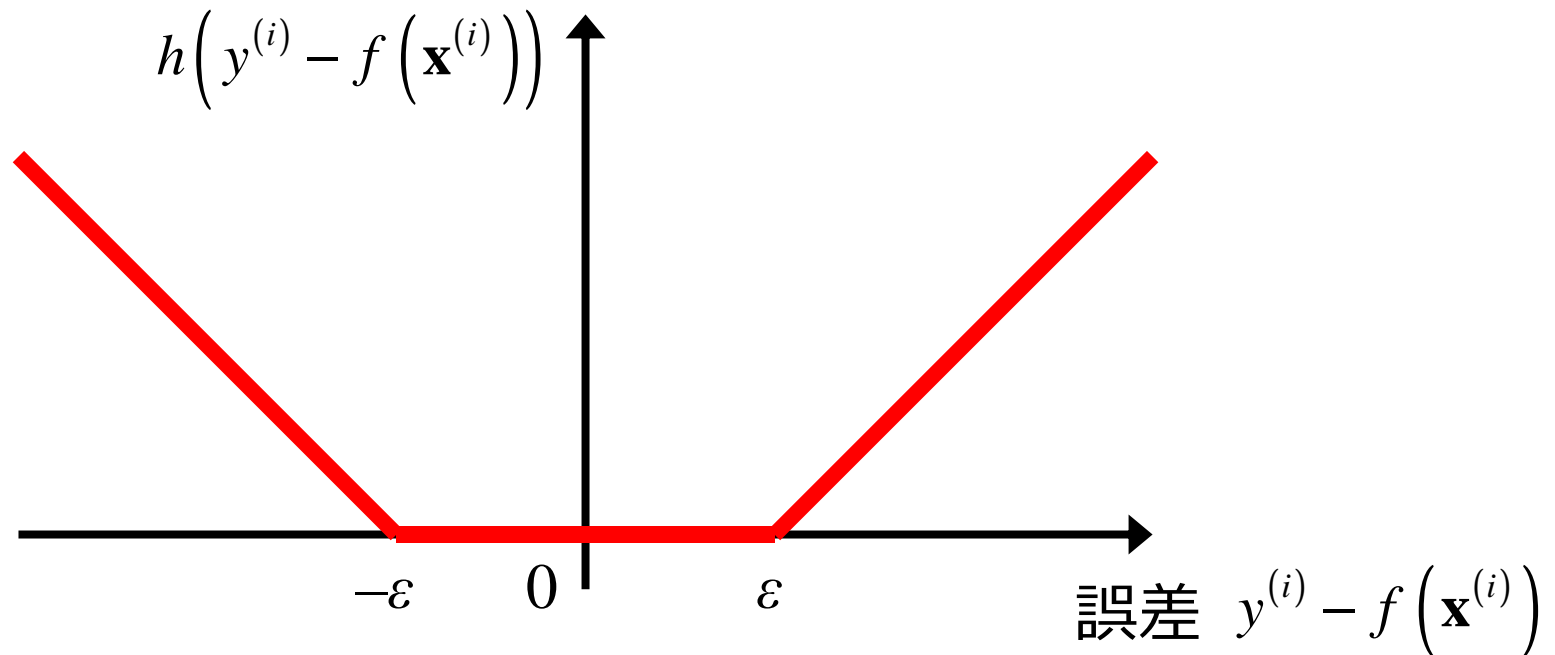
誤差の最小化

C : 2つの項のバランスを決める係数

h : 誤差関数

n : モデル構築用サンプル数

$$h\left(y^{(i)} - f\left(\mathbf{x}^{(i)}\right)\right) = \max\left(0, \left|y^{(i)} - f\left(\mathbf{x}^{(i)}\right)\right| - \varepsilon\right)$$



誤差の不感帯・・・ ε チューブと呼ぶ

$-\varepsilon \leq \text{誤差} \leq \varepsilon$ のとき、誤差 = 0 となる

✓SVMと同様にスラック変数 ξ, ξ^* を導入

$$y^{(i)} \leq f(\mathbf{x}^{(i)}) + \varepsilon + \xi_i$$

$$y^{(i)} \geq f(\mathbf{x}^{(i)}) - \varepsilon - \xi_i^*$$

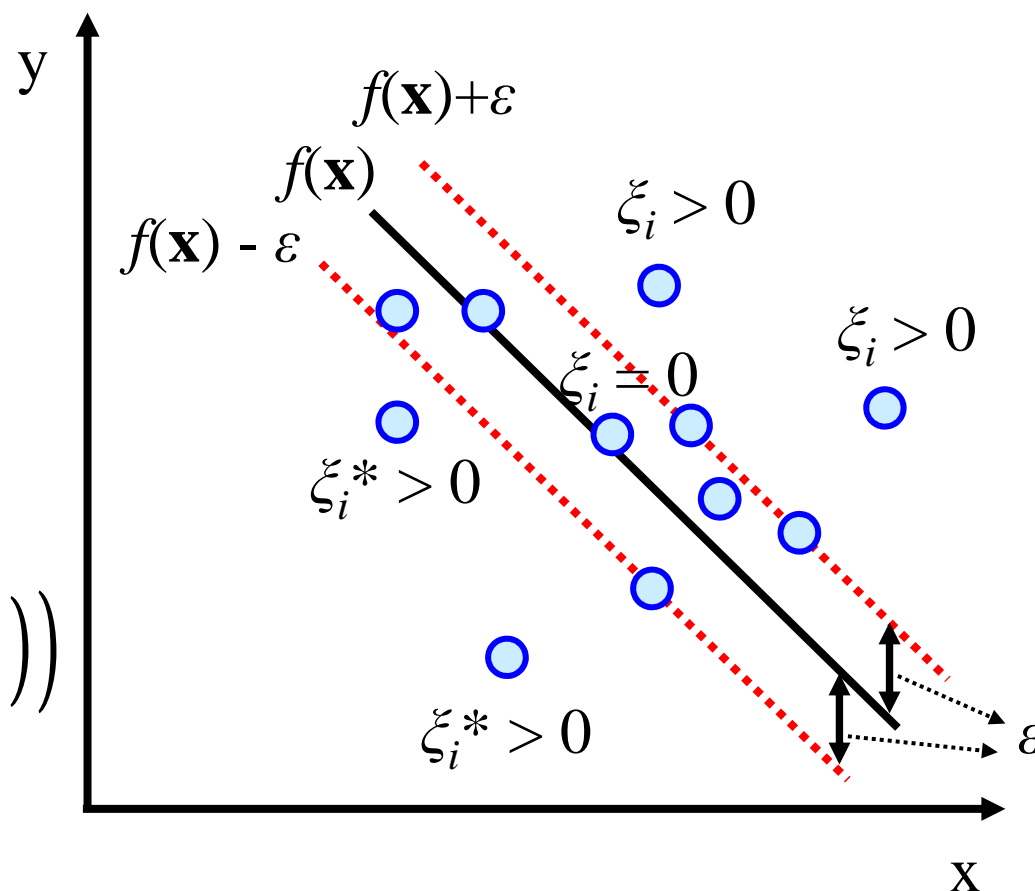
ただし、 $\xi_i \geq 0$

$$\xi_i^* \geq 0$$

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n h\left(y^{(i)} - f(\mathbf{x}^{(i)})\right)$$



$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*)$$



RR・LASSO・EN との関係

- ✓SVRでもRR・LASSO・ENと同様にして、
誤差だけでなく重み(回帰係数)も一緒に最小化することで、
過学習を防ぐ

$$\underbrace{\frac{1}{2}\|\mathbf{w}\|^2}_{\text{重みの最小化}} + \underbrace{C\sum_{i=1}^n(\xi_i + \xi_i^*)}_{\text{誤差の最小化}} \quad \text{の最小化}$$

Lagrangeの未定乗数法

✓ラグランジュ乗数 α_i 、 α_i^* 、 β_i 、 β_i^* ($i=1, 2, \dots, n$) を導入

$$G = C \sum_{i=1}^n (\xi_i + \xi_i^*) + \frac{1}{2} \|\mathbf{w}\|^2 - C \sum_{i=1}^n (\beta_i \xi_i + \beta_i^* \xi_i^*) \\ - \sum_{i=1}^n \alpha_i (\varepsilon + \xi_i + f(\mathbf{x}^{(i)}) - y_i) - \sum_{i=1}^n \alpha_i^* (\varepsilon + \xi_i^* - f(\mathbf{x}^{(i)}) + y_i)$$

\mathbf{w} 、 b 、 ξ_i 、 ξ_i^* に関して G を最小化し、 α_i 、 α_i^* 、 β_i 、 β_i^* に関して G を最大化



\mathbf{w} 、 b 、 ξ_i 、 ξ_i^* に関して G が極小



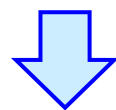
G を \mathbf{w} 、 b 、 ξ_i 、 ξ_i^* それぞれで偏微分して 0 とする

$$G \text{ を } \mathbf{w} \text{ で偏微分して0} \quad \mathbf{w} = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \phi(\mathbf{x}^{(i)})^T$$

$$G \text{ を } b \text{ で偏微分して0} \quad \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0$$

$$G \text{ を } \xi_i \text{ で偏微分して0} \quad \alpha_i + \beta_i = C \quad (i = 1, 2, \dots, n)$$

$$G \text{ を } \xi_i^* \text{ で偏微分して0} \quad \alpha_i^* + \beta_i^* = C \quad (i = 1, 2, \dots, n)$$



これらを使って
 G を変形すると...

$$G = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$
$$- \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n (\alpha_i - \alpha_i^*) y_i$$

K : カーネル関数 $K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \phi(\mathbf{x}^{(i)}) \phi(\mathbf{x}^{(j)})^T$

カーネル関数の例

✓線形カーネル

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \mathbf{x}^{(i)\top} \mathbf{x}^{(j)}$$

✓ガウシアンカーネル (使われることが多い)

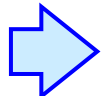
$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left(-\frac{\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{2\sigma^2}\right) = \exp\left(-\gamma\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2\right)$$


✓多項式カーネル

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \left(1 + \lambda \mathbf{x}^{(i)\top} \mathbf{x}^{(j)}\right)^d$$

$$G = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

$$- \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n (\alpha_i - \alpha_i^*) y_i$$

$\alpha_i, \alpha_i^*, \beta_i, \beta_i^*$ は Lagrange 定数  $\alpha_i \geq 0, \alpha_i^* \geq 0, \beta_i \geq 0, \beta_i^* \geq 0$

p.10より $\alpha_i + \beta_i = C \quad (i = 1, 2, \dots, n)$  $\alpha_i \leq C, \alpha_i^* \leq C$

$\alpha_i^* + \beta_i^* = C \quad (i = 1, 2, \dots, n)$

$$G = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n (\alpha_i - \alpha_i^*) y_i$$

制約 $0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C$ のもとで、

G を α_i に対して最大化する二次計画問題を解くと α_i, α_i^* が求まる

$$f(\mathbf{x}^{(j)}) = \phi(\mathbf{x}^{(j)})\mathbf{w} + c \quad \mathbf{w} = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \phi(\mathbf{x}^{(i)})^T \quad (\text{p.10})$$

$$\Rightarrow f(\mathbf{x}^{(j)}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \phi(\mathbf{x}^{(j)}) \phi(\mathbf{x}^{(i)})^T + c$$

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \phi(\mathbf{x}^{(i)}) \phi(\mathbf{x}^{(j)})^T \quad \text{より、}$$

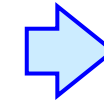
$$f(\mathbf{x}^{(j)}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}) + c$$

・・・ α_i, α_i^* は求まっているため、SVRの回帰式も求まる

サポートベクターとは

99

ε チューブ内 (誤差の絶対値が ε 未満) のサンプル



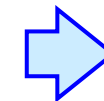
$$\alpha_i - \alpha_i^* = 0$$



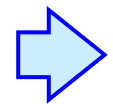
SVRの回帰式に寄与しない

ε チューブ上 (誤差の絶対値が ε) のサンプル

ε チューブ外 (誤差の絶対値が ε 以上) のサンプル



$$\alpha_i - \alpha_i^* \neq 0$$



これらのサンプルでSVRの回帰式がつくられる

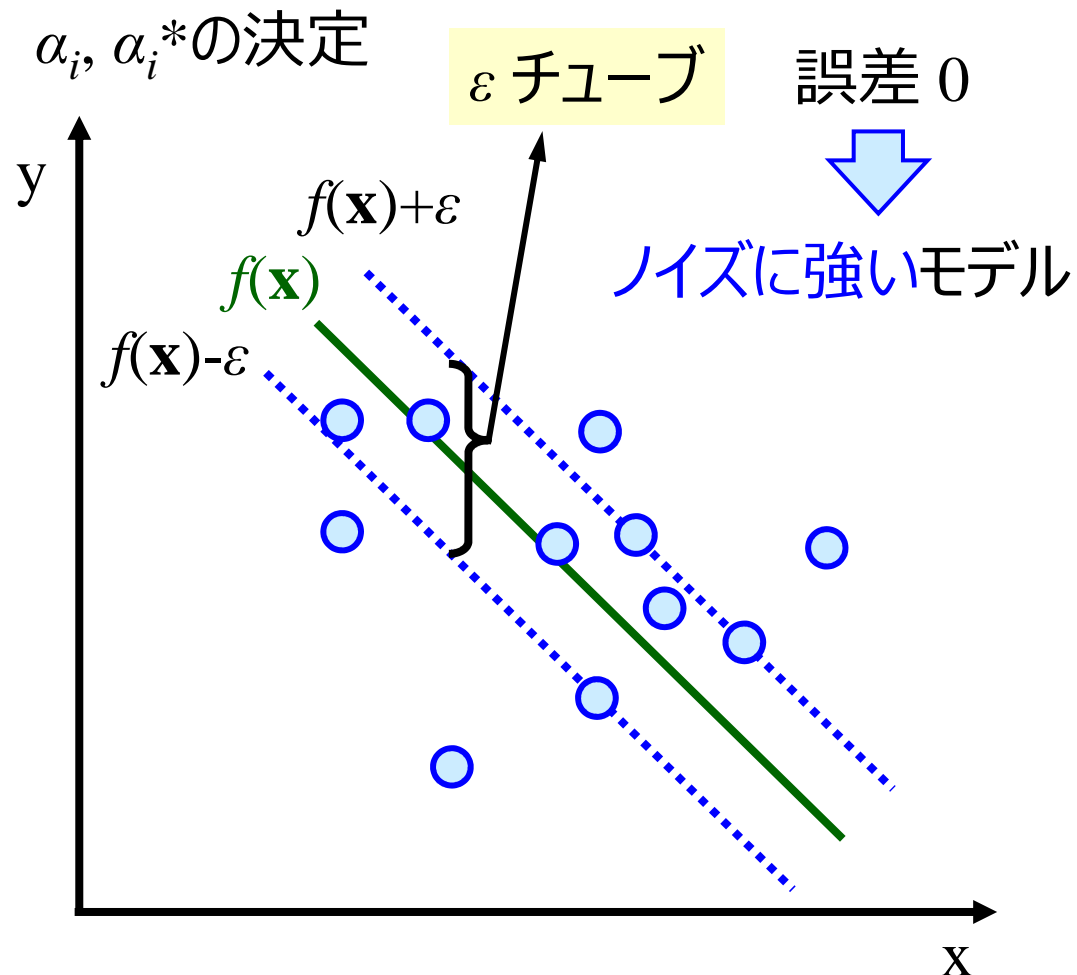


サポートベクター

SVRのまとめ・特徴

100

SVRの回帰式
$$f(\mathbf{x}^{(j)}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}) + c$$



α_i, α_i^* の範囲

$$0 \leq \alpha_i \leq C$$
$$0 \leq \alpha_i^* \leq C$$

モデルの複雑度を調整

カーネル関数 K

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left(-\gamma \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2\right)$$

非線形回帰モデル

C, ε , γ の決め方

✓ グリッドサーチによって、[クロスバリデーション](#)の後の r^2 の値がもっとも高い C , ε , γ (ガウシアンカーネル) の組み合わせとする

- C の候補の例: $2^{-5}, 2^{-4}, \dots, 2^9, 2^{10}$
- ε の候補の例: $2^{-10}, 2^{-9}, \dots, 2^{-1}, 2^0$
- γ の候補の例: $2^{-20}, 2^{-19}, \dots, 2^9, 2^{10}$

✓変数選択

- Stepwise (ステップワイズ) 法
 - LASSO、EN、RFでも変数選択可能

Stepwise (ステップワイズ) 法とは？

- ✓説明変数（入力変数・記述子・特徴量）を選択する手法
- ✓1つずつ説明変数を追加したり、削除したりしながら、最適な説明変数の組合せを探す
- ✓回帰モデルの構築を繰り返す数が増えると時間がかかる
- ✓どんな回帰分析手法とも組み合わせることができる

Stepwise法の種類

- ✓変数増加法 (forward stepwise)
 - 説明変数なしからはじめて、1 つずつ説明変数を増やす
- ✓変数減少法 (backward stepwise)
 - すべての説明変数からはじめて、1 つずつ説明変数を減らす
- ✓変数増減法 (forward-backward stepwise)
 - 説明変数なしからはじめて、1 つずつ説明変数を増やすか減らすかする
- ✓変数減増法 (backward-forward stepwise)
 - すべての説明変数からはじめて、1 つずつ説明変数を増やすか減らすかする

どのように説明変数を増やすか？

- ✓ある説明変数の組み合わせが選ばれているとき
(最初はもちろん説明変数なし)
- ✓選ばれていない説明変数の中から、1つ選んで追加して、
評価指標の値を計算する
- ✓計算し終わったら、追加した説明変数を戻す
- ✓選ばれていない説明変数すべてで評価指標の計算を行い、
評価指標の値が一番もっとも良くなった説明変数を実際に追加する

評価指標（最小二乗法による重回帰分析用） 1/2¹⁰⁶

- ✓ Mallows's C_p
- 小さいほど良い

$$C_p = \frac{SSE}{S^2} - n + 2m$$

$$SSE = \sum_{i=1}^n \left(y^{(i)} - y_{\text{EST}}^{(i)} \right)^2$$

n : サンプル数

m : 回帰モデルを構築した
説明変数の数

S^2 : すべての説明変数を用いて
回帰分析を行ったときの
誤差の二乗の平均

$y^{(i)}$: i 番目のサンプルにおける
目的変数の値

$y_{\text{EST}}^{(i)}$: i 番目のサンプルにおける
目的変数の推定値

評価指標 (最小二乗法による重回帰分析用) 2/2¹⁰⁷

✓ 赤池情報量規準 (Akaike's Information Criterion, AIC)

- 小さいほど良い

$$AIC = m \log \left(\frac{SSE}{m} \right) + 2$$

✓ Bayesian Information Criterion (BIC)

- 小さいほど良い

$$BIC = m \log \left(\frac{SSE}{m} \right) + n \log(m)$$

評価指標（任意の回帰分析手法で使える）¹⁰⁸

- ✓ $RMSE_{CV}$: クロスバリデーション後のRoot Mean Squared Error
 - 小さいほど良い

$$RMSE_{CV} = \sqrt{\frac{\sum_{i=1}^n \left(y^{(i)} - y_{CV\text{EST}}^{(i)} \right)^2}{n}}$$

$y_{CV\text{EST}}^{(i)}$: i 番目のサンプルにおける
クロスバリデーション後の
目的変数の推定値

- ✓ MAE_{CV} : クロスバリデーション後のMean Absolute Error (MAE)
 - 小さいほど良い

$$MAE_{CV} = \sqrt{\frac{\sum_{i=1}^n \left| y^{(i)} - y_{CV\text{EST}}^{(i)} \right|}{n}}$$

どのように説明変数を減らすか？

- ✓ある説明変数の組み合わせが選ばれているとき
(最初はもちろん説明変数なし)
- ✓選ばれている説明変数の中から、1つ選んで削除して、
評価指標の値を計算する
- ✓計算し終わったら、削除した説明変数を戻す
- ✓選ばれている説明変数すべてで評価指標の計算を行い、
評価指標の値が一番もっとも良くなった説明変数を実際に追加する

減らすときだけで使える手法

- ✓ 回帰分析手法の標準回帰係数に基づく変数削除
 - 標準回帰係数：標準化（オートスケーリング）後に計算された回帰係数
 - 選ばれている説明変数を用いて一度線形回帰分析を行い、標準回帰係数の絶対値がもっとも小さい変数を削除
 - 説明変数を 1 つ減らすときに、削除して回帰モデル構築を繰り返さなくてよいため、計算時間が短い

どのように説明変数を増やすか減らすか？¹¹

- ✓ある説明変数の組み合わせが選ばれているとき
(最初はもちろん説明変数なし)
- ✓選ばれていない説明変数の中から、1つ選んで追加して、
評価指標の値を計算する
- ✓計算し終わったら、追加した説明変数を戻す
- ✓選ばれている説明変数の中から、1つ選んで削除して、
評価指標の値を計算する
- ✓計算し終わったら、削除した説明変数を戻す
- ✓全通りで評価指標の計算を行い、評価指標の値が一番もっとも
良くなった説明変数を実際に追加 or 削除する

scikit-learn を使う方へ 1/2

- ✓ OLS, PLS : `sklearn.cross_decomposition.PLSRegression`
 - PLSで最大成分数まで計算するとOLSです
- ✓ LDA : `sklearn.discriminant_analysis.LinearDiscriminantAnalysis`
- ✓ SVM : `sklearn.svm.SVC`
- ✓ SVR : `sklearn.svm.SVR`
 - クロスバリデーション+グリッドサーチ : `sklearn.grid_search.GridSearchCV`
- ✓ DT : `sklearn.tree.DecisionTreeRegressor`
`sklearn.tree.DecisionTreeClassifier`
- ✓ RF : `sklearn.ensemble.RandomForestRegressor`,
`sklearn.ensemble.RandomForestClassifier`

scikit-learn を使う方へ 2/2

- ✓ RR : `sklearn.linear_model.Ridge`
- ✓ LASSO : `sklearn.linear_model.Lasso`, `sklearn.linear_model.LassoCV`
- ✓ EN : `sklearn.linear_model.ElasticNet`,
`sklearn.linear_model.ElasticNetCV`
- ✓ Stepwise : `sklearn.feature_selection.RFE`
`sklearn.feature_selection import RFECV`
- ✓ クロスバリデーション : `sklearn.model_selection.cross_val_predict`

- ✓ 最小二乗法による重回帰分析 [1,2,4]
- ✓ Partial Least Squares (PLS) [1,2,3]
- ✓ 線形判別分析 (Linear Discriminant Analysis, LDA) [4,6]
- ✓ Support Vector Machine (SVM) [2,5]
- ✓ Support Vector Regression (SVR) [5]
- ✓ 決定木 (Decision Tree, DT) [7]
- ✓ ランダムフォレスト (Random Forests, RF) [7,8]
- ✓ リッジ回帰 (Ridge Regression, RR) [9]
- ✓ Least Absolute Shrinkage and Selection Operator(LASSO) [9]
- ✓ Elastic Net (EN) [9]

- [1] 宮下芳勝・佐々木慎一, コンピュータ・ケミストリー シリーズ3 ケモメトリックスー化学パターン認識と多変量解析一, 共立出版 (1995)
- [2] 船津公人・金子弘昌, ソフトセンサー入門～基礎から実用的研究例まで～, コロナ社 (2014)
- [3] S. Wold, et. al., Chemom. Intell. Lab. Syst., 58, 109–130, 2001.
- [4] C.M. ビショップ, パターン認識と機械学習 上, 丸善出版 (2012)
- [5] C.M. ビショップ, パターン認識と機械学習 下, 丸善出版 (2012)
- [6] 金 明哲, 金森 敬文, 竹之内 高志, 村田 昇, Rで学ぶデータサイエンス<5>パターン認識, 共立出版 (2009)
- [7] 金 明哲, Rによるデータサイエンス～データ解析の基礎から最新手法まで～, 森北出版 (2007)
- [8] L. Breiman, “Random Forests”, Machine Learning, 45, 5–32, 2001
- [9] Jared P. Lander, みんなのR -データ分析と統計解析の新しい教科書-, マイナビ (2015)