

内容

1. 線形回帰モデル.....	3
1. 1 線形回帰モデル 1	3
1. 2 線形回帰モデル 2	3
1. 3 線形回帰モデル 3	4
1. 4 線形回帰モデル 4	4
1. 5 データ分割 1	5
1. 6 データ分割 2	5
1. 7 データ分割 3	6
2. 非線形回帰モデル.....	7
2. 1 非線形回帰モデル 1	7
2. 2 非線形回帰モデル 2	7
2. 3 非線形回帰モデル 3	9
2. 4 正規化法 1	9
2. 5 正規化法 2	10
2. 6 正規化法 3	10
2. 7 モデル選択.....	11
3. ロジスティック回帰モデル.....	13
3. 1 ロジスティック回帰モデル 1	13
3. 2 ロジスティック回帰モデル 2	13
3. 3 ロジスティック回帰モデル 3	14
3. 4 ロジスティック回帰モデル 4	14
3. 5 最尤推定 1	15
3. 6 最尤推定 2	15
3. 7 最尤推定 3	15
3. 8 最尤推定 4	16
3. 9 最尤推定 5	16
3. 10 最尤推定 6.....	17
3. 11 勾配降下法.....	17
3. 12 勾配降下法.....	18
3. 13 モデルの評価.....	18
4. 主成分分析.....	20
5. アルゴリズム	22
5. 1 k 近傍法.....	22
5. 2 K-means.....	22
6. サポートベクターマシン.....	23

1. 線形回帰モデル

1. 1 線形回帰モデル 1

1) 回帰問題

ある入力（離散あるいは連続値）から出力を予測する問題である。

- ・線形で予測する場合は線形回帰
- ・曲線で予測する場合は非線形回帰

2) 回帰で扱うデータ

- ①. 入力（各要素を説明変数、または特徴量と呼ぶ）
 - ・ m 次元のベクトル（ $m = 1$ のときスカラー）
- ②. 出力（目的変数）
 - ・ スカラー値

3) 記述方法

- ①. 説明変数

$$\mathbf{x} = (x_1, x_2, \dots, x_m)^T \in \mathbb{R}^m$$

- ②. 目的変数

$$y \in \mathbb{R}^1$$

1. 2 線形回帰モデル 2

1) 線形回帰モデルとは

- ①. 回帰問題を解くための機械学習モデルの 1 つ
- ②. 教師あり学習（教師データから学習）
説明変数とパラメータのペア
- ③. 入力と m 次元パラメータの線形結合を出力するモデル
 - ・ 慣例として予測値にはハットを付ける

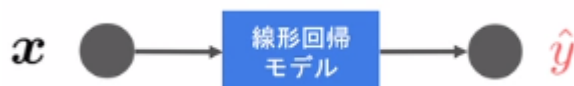
2) パラメータ

$$\mathbf{w} = (w_1, w_2, \dots, w_m)^T \in \mathbb{R}^m$$

3) 線形結合

$$\hat{y} = \underset{\cdot}{\mathbf{w}}^T \mathbf{x} + w_0 = \sum_{j=1}^m w_j x_j + w_0$$

入力 \mathbf{x} との線形結合



y と記述すると学習データと混同するため、予測値はハットを付ける。

1. 3 線形回帰モデル3

1) 線形結合 (入力とパラメータの内積)

- ①. 入力ベクトルと未知のパラメータの各要素を掛け合わせたもの。
- ②. 入力ベクトルの線形結合に加え、切片も足し合わせる。(y 軸との交点)
- ③. (入力のベクトルが多次元でも) 出力は1次元となる。

2) モデルのパラメータ

- ①. モデルに含まれる推定すべき未知のパラメータ。
- ②. 特徴量が予測値に対してどのように影響を与えるかを決定する重みの集合
 - ・ 正の (負の) 重みを付ける場合、その特徴量の値を増加させると、予測の値が増加 (減少)
 - ・ 重みが大きければ (0 であれば)、その特徴量は予測に大きな影響力を持つ (全く影響しない)
- ③. 切片
 - ・ y 軸との交点

3) 計算式

$$\hat{y} = \underline{w}^T \underline{x} + \underline{w}_0 = \sum_{j=1}^m \underline{w}_j x_j + \underline{w}_0$$

パラメータは未知なため、最小二乗法で推定する。

1. 4 線形回帰モデル4

1) 説明変数が1次元の場合

- ・ 単回帰モデル
- ・ 多次元の場合、重回帰モデル

2) データの変換

- ・ データは回帰直線に誤差が加わり観測されていると仮定する。

3) モデル数式

$$y = w_0 + w_1 * x_1 + \varepsilon$$

y : 目的変数

w_0 : 切片 (未知)

w_1 : 回帰係数 (未知)

x_1 : 説明変数

ε : 誤差

4) 連立方程式

$$y_1 = w_0 + w_1 * x_1 + \varepsilon_1$$

$$y_2 = w_0 + w_2 * x_2 + \varepsilon_2$$

:

$$y_n = w_0 + w_n * x_n + \varepsilon_n$$

5) 行列表現

$$y = Xw + \varepsilon$$

T

$$\begin{aligned}
X &= (x_1, x_2, \dots, x_n) \\
y &= (y_1, y_2, \dots, y_n)^T \\
x_i &= (1, x_i)^T \\
\varepsilon &= (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^T \\
w &= (w_0, w_1)^T
\end{aligned}$$

6) 説明変数が多次元の場合 ($m \geq 1$)

- ・線形重回帰モデル
- ・単回帰は直線、重回帰は曲線

7) データへの仮定

- ・データは回帰局面に誤差が加わり観測されていると仮定する。

8) 数式表現

$$y = w_0 + w_1 * x_1 + w_2 * x_2 + \varepsilon$$

y : 目的変数

w_0 : 切片 (未知)

w_1 、 w_2 : 回帰係数 (未知)

x_1 、 x_2 : 説明変数

ε : 誤差

1. 5 データ分割 1

1) データの分割

- ①. 学習用データ : 機械学習モデルの学習に利用するデータ
- ②. 検証用データ : 学習済みモデルの精度を検証するためのデータ

2) なぜ分割するか

- ①. モデルの汎化性能 (Generalization) を測るため
- ②. データへの当てはまりの良さではなく、**未知のデータに対してどれくらい精度が高いか**を測る

train : 訓練データ

test : テスト (検証) データ

1. 6 データ分割 2

線形回帰のパラメータは最小二乗法で推定する。

1) 平均二乗誤差 (残差平方和)

- ①. データとモデル出力の二乗誤差の和
 - ・小さいほど直線とデータの距離が近い
- ②. パラメータのみに依存する関数
 - ・データは既知の値でパラメータのみ未知

$$MSE_{\text{train}} = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} (\hat{y}_i^{(\text{train})} - y_i^{(\text{train})})^2$$

2) 最小二乗法

- ①. 学習データの平均二乗誤差を最小とするパラメータを検索する。

②. 学習データの平均二乗誤差の最小化は、その勾配（微分）が0になる点を求める。

1. 7 データ分割3

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w} \in \mathbb{R}^{m+1}} \text{MSE}_{\text{train}} \quad \frac{\partial}{\partial \mathbf{w}} \text{MSE}_{\text{train}} = 0$$

MSE を最小にするような \mathbf{w} (m次元) MSE を \mathbf{w} に関して微分したものが0になる \mathbf{w} を求める。

$$\begin{aligned} \Rightarrow \frac{\partial}{\partial \mathbf{w}} \left\{ \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} (\hat{y}_i^{(\text{train})} - y_i^{(\text{train})})^2 \right\} &= 0 && \text{平均二乗誤差(残差平方和)を微分} \\ \Rightarrow \frac{1}{n_{\text{train}}} \frac{\partial}{\partial \mathbf{w}} \left\{ (X^{(\text{train})} \mathbf{w} - \mathbf{y}^{(\text{train})})^T (X^{(\text{train})} \mathbf{w} - \mathbf{y}^{(\text{train})}) \right\} &&& \text{行列表現} \\ \Rightarrow \frac{1}{n_{\text{train}}} \frac{\partial}{\partial \mathbf{w}} \left\{ \mathbf{w}^T X^{(\text{train})T} X^{(\text{train})} \mathbf{w} - 2\mathbf{w}^T X^{(\text{train})T} \mathbf{y}^{(\text{train})} + \mathbf{y}^{(\text{train})T} \mathbf{y}^{(\text{train})} \right\} &&& \{ \} \text{展開} \\ \Rightarrow 2X^{(\text{train})T} X^{(\text{train})} \mathbf{w} - 2X^{(\text{train})T} \mathbf{y}^{(\text{train})} &= 0 && \text{行列微分計算} \\ \Rightarrow \hat{\mathbf{w}} = (X^{(\text{train})T} X^{(\text{train})})^{-1} X^{(\text{train})T} \mathbf{y}^{(\text{train})} &&& \text{回帰係数} \end{aligned}$$

回帰係数 $\hat{\mathbf{w}}$ は、説明変数と目的変数だけがあれば一意に決まる。

回帰係数

$$\hat{\mathbf{w}} = (X^{(\text{train})T} X^{(\text{train})})^{-1} X^{(\text{train})T} \mathbf{y}^{(\text{train})}$$

予測値

$$\hat{\mathbf{y}} = X \left(X^{(\text{train})T} X^{(\text{train})} \right)^{-1} X^{(\text{train})T} \mathbf{y}^{(\text{train})}$$

$n_{\text{new}} \times m+1$

2) 最尤法による回帰係数の推定

- ①. 誤差を正規分布に従う確率変数を仮定し、尤度関数の最大かを利用した推定も可能
- ②. 回帰の場合には、最尤法による解は最小二乗法の解と一致する。

2. 非線形回帰モデル

2. 1 非線形回帰モデル 1

1) 複雑な非線形構造を内在する構造に対して、非線形回帰モデリングを実施する

- ①. データの構造を線形で捉えられる場合は限られる。
- ②. 非線形な構造を捉えられる仕組みが必要。

2) 基底展開法

- ①. 回帰関数として、基底関数と呼ばれる既知の非線形関数とパラメータベクトルの線形結合を使用する。
- ②. 未知パラメータは線形回帰モデルと同様に最小2乗法や尤度法により推定する。

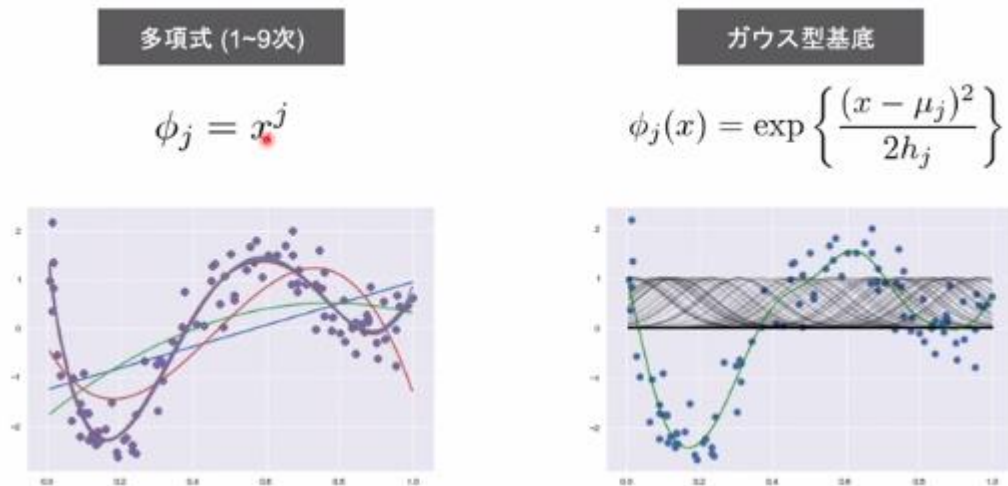
$$y_i = f(\mathbf{x}_i) + \varepsilon_i \qquad y_i = w_0 + \sum_{j=1}^m w_j \phi_j(\mathbf{x}_i) + \varepsilon_i$$

3) 良く使われる基底関数

- ①. 多項式関数
- ②. ガウス型基底関数
- ③. スプライン関数/Bスプライン関数

2. 2 非線形回帰モデル 2

1) 1次元の基底関数に基づく非線形回帰



正規分布は、全てを足して1になる。

既定関数 $\phi(x)$ に多項式関数 $\phi_j = x^j$ 乗

$$\begin{aligned} \hat{y} &= w_0 + w_1 \phi_1(x_i) + w_2 \phi_2(x_i) \\ &= w_0 + w_1 x_i + w_2 x_i^2 \end{aligned}$$

求めるべき w については線形である

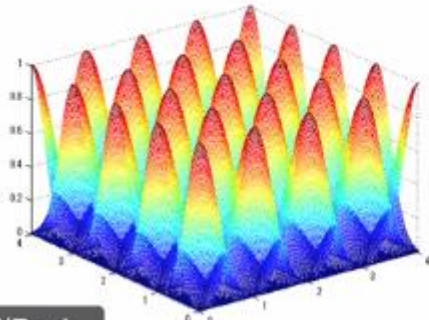
G a u s s 型基底関数

$$\phi_j(x) = \exp \left\{ -\frac{(x - \mu_j)^2}{2h_j} \right\}$$

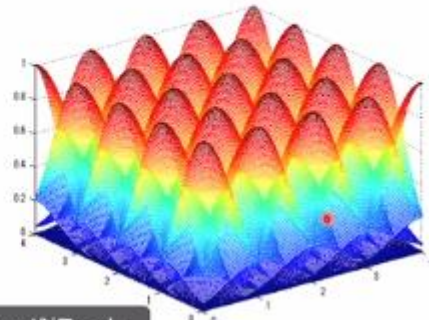
2) 2次元の基底関数に基づく非線形回帰

2次元ガウス型基底関数

$$\phi_j(\mathbf{x}) = \exp \left\{ -\frac{(\mathbf{x} - \boldsymbol{\mu}_j)^T (\mathbf{x} - \boldsymbol{\mu}_j)}{2h_j} \right\}$$



バンド幅：小



バンド幅：大

説明変数

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{im}) \in \mathbb{R}^m$$

説明変数の数

非線形関数ベクトル

$$\boldsymbol{\phi}(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \phi_2(\mathbf{x}_i), \dots, \phi_k(\mathbf{x}_i))^T \in \mathbb{R}^k$$

基底関数の数

非線形関数の計画行列

$$\Phi^{(train)} = (\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \dots, \boldsymbol{\phi}(\mathbf{x}_n))^T \in \mathbb{R}^{n \times k}$$

最尤法による予測値

$$\hat{\mathbf{y}} = \Phi (\Phi^{(train)T} \Phi^{(train)})^{-1} \Phi^{(train)T} \mathbf{y}^{(train)}$$

基底展開法も線形回帰と同じ枠組みで推定可能

非線形関数の計画行列

$$\Phi^{train} = \boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2)$$

尤度法 (最小2乗法)

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

逆行列

$$= (\boldsymbol{\phi}^T \boldsymbol{\phi})^{-1} \boldsymbol{\phi}^T \mathbf{y}$$

2. 3 非線形回帰モデル3

1) 未学習 (underfitting) と過学習 (overfitting)

- ①. 学習データに対して、十分小さな誤差が得られないモデル——>未学習
 - ・(対策) モデルの表現力が低いため、表現力の高いモデルを使用する。
 - ②. 小さな誤差は得られたけど、テスト集合誤差との差が大きいモデル——>過学習
 - ・(対策1) 学習データの数を増やす。
 - ・(対策2) 不要な基底関数 (変数) を削除して表現力を抑止
 - ・(対策3) 正則化法を利用して表現力を抑止
- } モデルの複雑さを調整する2つの方法

2. 4 正則化法1

1) 不要な基底関数を削除する

- ①. 基底関数の数、位置やバンド幅によりモデルの複雑さが変化
- ②. 解きたい問題に対して多くの基底関数を用意してしまうと、過学習の問題が起こるため適切な基底関数を用意する。(CVなどで選択)

2) 正則化法 (罰則化法)

- ①. 「モデルの複雑さに伴って、その値が大きくなる正則化項 (罰則項) を課した関数」を最小化
- ②. 正則化項 (罰則項)
 - ・形状によっていくつもの種類があり、それぞれ推定量の性質が異なる
- ③. 正則化 (平滑化) パラメータ
 - ・モデルの曲線の滑らかさを調節・適切に決める必要がある。

$$S_\gamma = (\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w}) + \gamma R(\mathbf{w}) \quad \gamma (> 0)$$

10C5・・・組合せ爆発

正則化法

線形回帰 $\hat{\mathbf{y}} = \mathbf{w}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

1 2 4

$\mathbf{X} = \begin{pmatrix} 1 & 3 & 5.9 \\ 1 & 4 & 8.1 \end{pmatrix} \Rightarrow (\mathbf{X}^T \mathbf{X})^{-1}$ の要素はメチャメチャ大きくなる

1 4 8.1

ほぼ平行

$$E(\mathbf{w}) = J(\mathbf{w}) + \lambda \mathbf{W}^T \mathbf{W}$$

MSE 罰則項 (L2ノルム)

Wを抑えつつ、MSEを小さくする

$$S_\gamma = (\mathbf{y} - \Phi \mathbf{w})^T (\mathbf{y} - \Phi \mathbf{w}) + \gamma R(\mathbf{w})$$

ハイパーパラメータ

解きたいのはMSEの最小化 s.t. $R(\mathbf{w}) \leq r$

(最適化) KKT条件より

$$\min \text{MSE} + \lambda R(\mathbf{w})$$

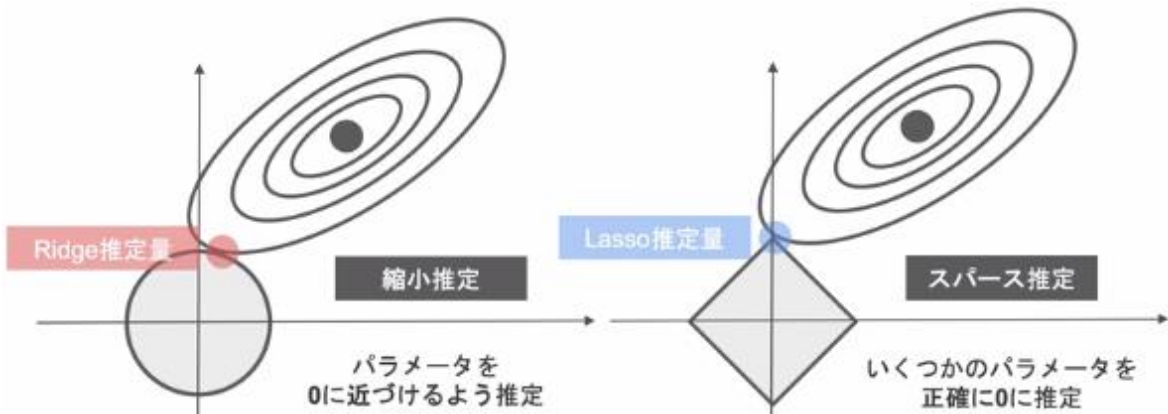
おまけを付けると不等式条件を解除

2. 5 正規化法 2

1) 正則化項（罰則項）の役割

- ①. 無い、最小 2 乗推定量
- ②. L 2 ノルムを利用→リッジ推定量
- ③. L 1 ノルムを利用→ラッソ推定量

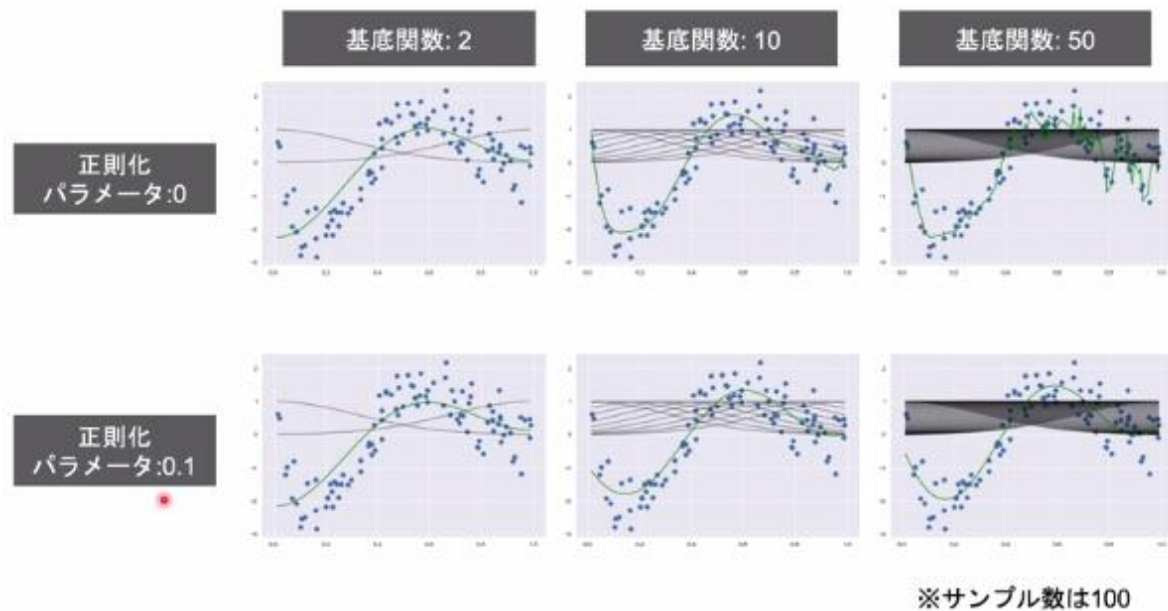
2) 正則化パラメータの役割



横軸 w_0 、縦軸 w_1

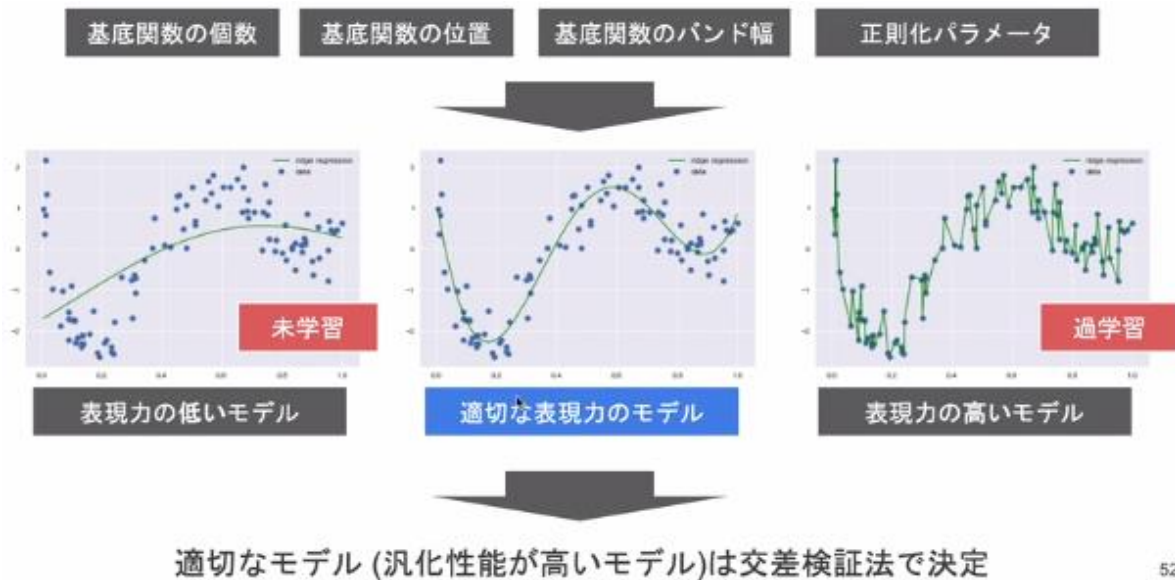
縮小推定・スパース推定

2. 6 正規化法 3



交差検収法

2. 7 モデル選択



52

交差検収法

1) 汎化性能

- ①. 学習に使用した入力だけでなく、これまで見たことのない新たな入力に対する予測性能
 - ・ (学習誤差ではなく) 汎化誤差 (テスト誤差) が小さいものが良い性能を持ったモデル。
 - ・ 新しい入力に対する誤差の期待値で定義される。
- ②. 汎化誤差は通常、学習データとは別に収集された検証データでの性能を測ることで推定

$$\text{MSE}_{\text{train}} = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} (\hat{y}_i^{(\text{train})} - y_i^{(\text{train})})^2 \quad \text{MSE}_{\text{test}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (\hat{y}_i^{(\text{test})} - y_i^{(\text{test})})^2$$

訓練誤差：モデルの学習に使用
(学習データにどれくらいフィットするかの指標、しすぎはだめ)



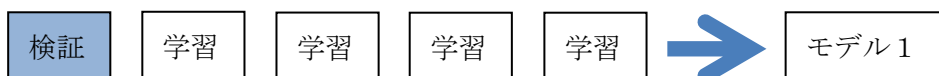
テスト誤差：モデルの性能の指標
(未来のデータに対する性能)

学習誤差

検証誤差

2) ホールドアウト法

- ①. 有限のデータを学習用とテスト用の2つに分割し、「予測精度」や「誤り率」を推定するために使用
 - ・ 学習用を多くすればテスト用が減り、学習制度はよくなるが、性能評価の精度は悪くなる。
 - ・ 逆にテスト用を多くすれば学習用が減少するので、学習そのものの精度が悪くなることになる。
 - ・ 手元にデータが大量にある場合を除いて、よい性能評価を与えないという欠点がある。
- ②. 基底展開法に基づく非線形回帰モデルでは、基底関数の数、位置、バンド幅の値とチューニングパラメータをホールドアウト値を小さくするモデルで決定する。



3) クロスバリデーション（交差検証法）

データを学習用と評価用に分割（5分割の例）



制度の平均値を CV という（モデル 1 の汎化性能）

3. ロジスティック回帰モデル

3. 1 ロジスティック回帰モデル 1

1) 分類問題 (クラス分類)

- ①. ある入力 (数値) 値からクラスに分類する問題

2) 分類で扱うデータ

- ①. 入力 (各要素を説明変数または特徴量と呼ぶ)
 - ・ m次元のベクトル (m = 1 の場合はスカラー)
- ②. 出力 (目的変数)
 - ・ 0 又は 1 の値
- ③. タイタニックデータ、I R I Sデータなど

説明変数 $\mathbf{x} = (x_1, x_2, \dots, x_m)^T \in \mathbb{R}^m$

目的変数 $y \in \{0, 1\}$ 0か1

Using ML to Predict Parking Difficulty[Google AI Blog:2017 Feb]

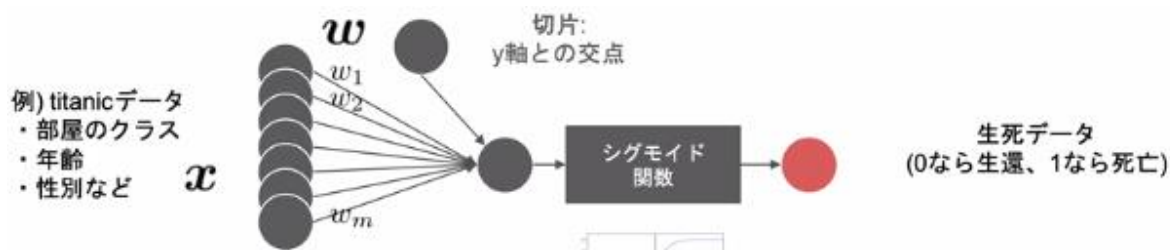
3. 2 ロジスティック回帰モデル 2

1) ロジスティック線形回帰モデル

- ①. 分類問題を解くための教師あり機械学習モデル (教師データから学習)
 - ・ 入力と m次元パラメータの線形結合をシグモイド関数に入力
 - ・ 出力は $y = 1$ となる確率の値になる

パラメータ $\mathbf{w} = (w_1, w_2, \dots, w_m)^T \in \mathbb{R}^m$

線形結合 $\hat{y} = \mathbf{w}^T \mathbf{x} + w_0 = \sum_{j=1}^m w_j x_j + w_0$



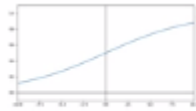
2) シグモイド関数

- ①. 入力の実数・出力は 0 か 1
- ②. (クラス 1 に分類される) 確率を表現する
- ③. 単調増加関数

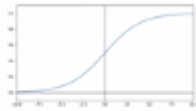
3) パラメータが変わるとシグモイド関数の形が変わる

- ①. a を増加させると、 $x=0$ 付近での曲線の勾配が増加
- ②. a を極めて大きくすると、単位ステップ関数 ($x < 0$ で $f(x)=0, x > 0$ で $f(x)=1$ となるような関数) に近づく
- ③. バイアス変化は段差の位置

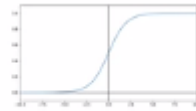
$$\sigma(\mathbf{x}) = 1 / (1 + \exp(-a\mathbf{x}))$$



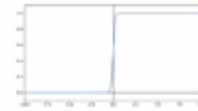
a=0.2



a=0.5



a=1



a= 10

3. 3 ロジスティック回帰モデル 3

1) シグモイド関数の性質

- ①. シグモイド関数の微分は、シグモイド関数自身で表現することが可能
- ②. 尤度関数の微分を行う際にこの事実を利用すると計算が容易

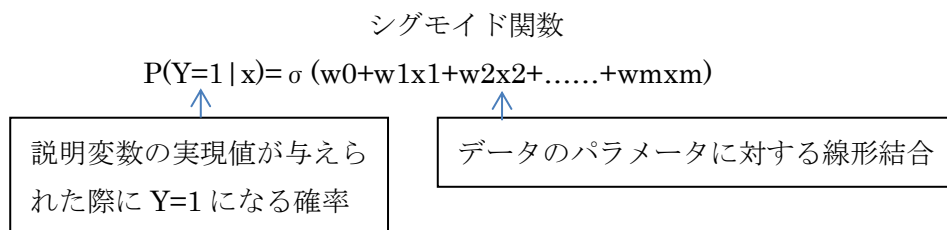
$$\begin{aligned} \frac{\delta \sigma(x)}{\delta x} &= \frac{\delta}{\delta x} \left(\frac{1}{1+\exp(-ax)} \right) \\ &= (-1) \cdot \{1+\exp(-ax)\}^{-2} \cdot \exp(-ax) \cdot (-a) && \text{連鎖律} \\ &= \frac{a \exp(-ax)}{(1+\exp(-ax))^2} = \frac{a}{1+\exp(-ax)} \cdot \frac{1+\exp(-ax)-1}{1+\exp(-ax)} \\ &= a \sigma(x)(1-\sigma(x)) \end{aligned}$$

3. 4 ロジスティック回帰モデル 4

●ロジスティック回帰の考え方

1) シグモイド関数の出力を Y=1 になる確率に対応させる

- ①. データの線形結合を計算
- ②. シグモイド関数に入力、出力が確立に対応する
- ③. [表記]i 番目データを与えた時のシグモイド関数の出力を i 番目のデータが Y=1 になる確率とする
求めたい値



表記

$$P_i = \sigma(w_0 + w_1x_{i1} + w_2x_{i2} + \dots + w_mx_{im})$$

$$P(Y=1 | x) = \sigma(w_0 + w_1x_1)$$

$P(Y=1 | x)$: データが与えられた時に Y=1 になる確率

w_0 : 切片 (未知 : 学習で決める)

w_1 : 回帰係数 (未知 : 学習で決める)

x_1 : 説明変数 (既知 : 入力データ)

データ Y は確立が 0.5 以上ならば 1・未満なら 0 と予測

3. 5 最尤推定 1

1) 世の中には様々な確率分布が存在する

- ①. 正規分布、t 分布、ガンマ分布、一様分布、ディレクレ分布・・・
- ②. ロジスティック回帰ではベルヌーイ分布を利用する。

2) ベルヌーイ分布

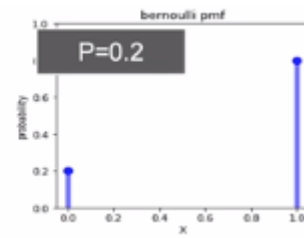
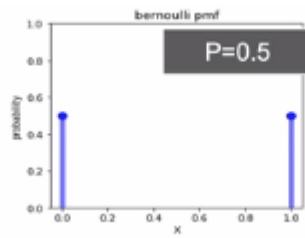
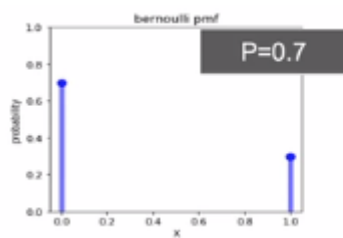
- ①. 数学において、確率 p で 1、確率 $1 - p$ で 0 をとる離散確率分布（例：コイン投げ）
- ②. 「生成されるデータ」は分布のパラメータによって異なる（この場合は確立 p ）

ベルヌーイ分布に従う確率
変数 Y

$$Y \sim \text{Be}(p)$$

$Y=0$ と $Y=1$ になる確率を
まとめて表現

$$P(y) = p^y(1-p)^{1-y}$$



3. 6 最尤推定 2

1) ベルヌーイ分布のパラメータの推定

- ①. ある分布を考えた時、そのパラメータ（既知）によって、生成されるデータは変化する
 - ・ 0.3 と 0.8
- ②. データからそのデータを生成したであろう尤もらしい分布（パラメータ）を推定したい
 - ・ 尤度推定

例 1)

表がでる確率 0.6 のベルヌーイ分布を仮定

100 回投げたら表が 58 回、裏が 42 回のデータを取得

例 2)

データを集めたところ表が 55 回、裏が 45 回だった

これらのデータは、どんな分布から生成されたか

3. 7 最尤推定 3

1) 同時確立

- ①. あるデータが得られた時、それが同時に得られる確率
- ②. 確率変数は独立であることを仮定すると、それぞれの掛け算となる。

2) 尤度関数とは

- ①. データは固定し、パラメータを変化させる
- ②. 尤度関数を最大化するようなパラメータを選ぶ推定方法を最尤推定という

3. 8 最尤推定 4

1 回の試行で $y=y_1$ になる確率

$$P(y)=p^y(1-p)^{1-y}$$

n 回の試行で $y_1 \sim y_n$ が同時に起こる確率

$$P(y_1, y_2, \dots, y_n; p) = \prod p^{y_i} (1-p)^{1-y_i}$$

$y_1 \sim y_n$ のデータが得られた際の尤度関数

$$P(y_1, y_2, \dots, y_n; p) = \prod_{i=1}^n p^{y_i} (1-p)^{1-y_i}$$

3. 9 最尤推定 5

1) ロジスティック回帰モデルの最尤推定

- ①. 確率 p はシグモイド関数となるため、推定するパラメータは重みパラメータとなる。
- ②. $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ を生成するに至った尤もらしいパラメータを探す

P はシグモイド関数で書き換えられる

$$P(Y=y_1 | x_1) = p_1^{y_1} (1-p_1)^{1-y_1} = \sigma(w^T x_1)^{y_1} (1-\sigma(w^T x_1))^{1-y_1}$$

$$P(Y=y_2 | x_2) = p_2^{y_2} (1-p_2)^{1-y_2} = \sigma(w^T x_2)^{y_2} (1-\sigma(w^T x_2))^{1-y_2}$$

:

$$P(Y=y_n | x_n) = p_n^{y_n} (1-p_n)^{1-y_n} = \sigma(w^T x_n)^{y_n} (1-\sigma(w^T x_n))^{1-y_n}$$

$y_1 \sim y_n$ のデータが得られた際の尤度関数

$$P(y_1, y_2, \dots, y_n | w_0, w_1, \dots, w_m) = \prod_{i=1}^n p_i^{y_i} (1-p_i)^{1-y_i}$$

$$= \prod_{i=1}^n \sigma(w^T x_i)^{y_i} (1-\sigma(w^T x_i))^{1-y_i}$$

$$= L(w)$$

尤度関数はパラメータのみに依存する関数

尤度関数 L を最大とするパラメータを探索

3. 1 0 最尤推定 6

1) 尤度関数を最大とするパラメータを探索する

- ①. 対数をとると微分の計算が簡単
 - ・ 同時確立の積が和に変換可能
 - ・ 指数が積の演算に変換可能
- ②. 対数尤度関数が最大になる点と尤度関数が最大となる点は同じ
 - ・ 対数関数は単調増加 (ある尤度の値が $x_1 < x_2$ の時、必ず $\log(x_1) < \log(x_2)$ となる)
- ③. 「尤度関数に－をかけたものを最小化」し、「最小 2 乗法の最小化」と合わせる。

$$\begin{aligned} E(w_0, w_1, \dots, w_m) &= -\log L(w_0, w_1, \dots, w_m) \\ &= -\sum_{i=1}^n \{y_i \log p_i + (1-y_i) \log (1-p_i)\} \end{aligned}$$

3. 1 1 勾配降下法

1) 勾配降下法

- ①. 反復学習によりパラメータを逐一敵に更新するアプローチの一つ
- ②. η は学習率と呼ばれるハイパーパラメータでモデルのパラメータの収束しやすさを調整する。

2) なぜ必要か

- ①. 「線形回帰モデル (最小 2 乗法)」→MSE のパラメータに関する微分が 0 になる値を解析に求めることが可能
- ②. 「ロジスティック回帰モデル (最尤法)」→対数尤度関数をパラメータで微分して 0 になる値を求める必要があるのだが、解析的にこの値を求めることが困難である。

$$w(k+1) = w - \eta \frac{\delta E(w)}{\delta w}$$

3) 対数尤度関数を係数とバイアスに関して微分

$$\begin{aligned} \frac{\delta E(w)}{\delta w} &= \sum \frac{\delta E_i(w)}{\delta p_i} \frac{\delta p_i}{\delta w} && \text{連鎖律} \\ &= \sum \left(\frac{y_i}{p_i} - \frac{1-y_i}{1-p_i} \right) \frac{\delta p_i}{\delta w} && \text{対数尤度関数の } p \text{ に関する微分} \\ &= \sum \left(\frac{y_i}{p_i} - \frac{1-y_i}{1-p_i} \right) p_i(1-p_i)x_i && \text{シグモイド関数の微分} \\ &= -\sum (y_i(1-p_i) - p_i(1-y_i))x_i && \text{式を整理} \\ &= -\sum (y_i - p_i)x_i \end{aligned}$$

4) パラメータが更新されなくなった場合、それは勾配が 0 になったということ。少なくとも反復学習で探索した範囲では最適な解が求められたことになる。

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \eta \sum_{i=1}^n (y_i - p_i) \mathbf{x}_i$$

5) 勾配降下法では、パラメータを更新するのにn個全てのデータに対する和を求める必要がある。

①. nが巨大になった時にデータをオンメモリーに載せる容量が足りない。計算時間が莫大になるなどの問題がある。

②. 確率的勾配降下法を利用して解決する。

3. 1 2 勾配降下法

1) 確率的勾配降下法 (SGD)

①. データを一つずつランダムに（「確率的」に）選んでパラメータを更新する。

②. 勾配降下法でパラメータを1回更新するのと同じ計算量でパラメータをn回更新できるので効率よく最適な解を探索可能。

$$\mathbf{w}(k+1) = \mathbf{w}^k + \eta(y_i - p_i)\mathbf{x}_i$$

3. 1 3 モデルの評価

学習済みの「ロジスティック回帰モデル」の性能を測る指標についてみる。

1) 混同行列 (confusion matrix)

①. 各検証データに対する予測の結果を4つの観点（表）で分類し、それぞれに当てはまる予測結果の個数をまとめた表

		検証データの結果	
		Positive	Negative
モデルの予測結果	Positive	真陽性 (TP)	偽陰性 (FP)
	Negative	偽陽性 (FN)	真陰性 (TN)

2) 分類の評価方法

①. 正解率が良く使われる

②. 正解した数／予測対象となった全データ数

・メールのスパム分類

スパム数が80件・普通のメールが20件あった場合

全てをスパムとする分類器は80%となる

③. どのような問題があるか

・分類したいクラスにはそれぞれ偏りがあることが多い

・この場合、単純な正解率はあまり意味がない。

TP+TN

TP+FP+FN+TN

3) 分類の評価方法

①. 表情から怪しい人物を検知する動画分析ソリューション

- ・ 正解率が適当でない例→リコールやプレジジョンを使う
- ・ 異常値検出のタスクっぽいのが、説明の簡易化のため分類で解く
- ・ False Positive
正常な人を間違えて異常としてしまう。
- ・ False Negative
異常な人を間違えて正常としてしまう。

4) 再現率 (Recall)

①. 「本当に Positive なもの」の中から Positive と予測できる割合 (Negative なものを Positive としてしまうことは考えていない)

②. 「誤り (Falses Positive) が多少多くても抜け、漏れが少ない」予測をしたい際に利用

③. 使用例) 病気の検診で「陽性である物を陰性と誤診 (False Negative)」としてしまうのを避けたい。
「陰性を陽性であると誤診 (False Positive)」とするものが少し増えたとしても再検査すればよい。

FP

TP+FN

5) 適合率 (Precision)

①. モデルが「Positive と予測」したものの中で本当に Positive である割合 (本当に Positive なものを Negative としてしまうことについては考えていない)

②. 「見逃し (False Negative) が多くてもより正確な」予測をしたい際に利用

③. 「重要なメールをスパムメールと誤判別」されるより、「スパムと予測したものが確実にスパム」である方が便利。スパムメールを検出できなくても (False Negative) 自分でやればよい。

TP

TP+FP

6) F 値

①. 理想的にはどちらも高いモデルがいいモデルだが、両者はトレードオフの関係にあり、どちらかを小さくすると、もう片方の値が大きくなってしまいます。

②. Precision と Recall の調和平均

Recall と Precision のバランスを示している

$2 * Precision * Recall$

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

4. 主成分分析

1) 多変量データの持つ構造をより少数個の指標に圧縮する

- ①. 変数の個数を減らすことに伴う、情報の損失はなるべく小さくしたい。
- ②. 少数変数を利用した分析や可視化（2・3次元の場合）が実現可能

学習データ $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{im}) \in \mathbb{R}^m$

平均（ベクトル） $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$

データ行列 $\bar{X} = (\mathbf{x}_1 - \bar{\mathbf{x}}, \dots, \mathbf{x}_n - \bar{\mathbf{x}})^T$

分散・共分散行列 $\Sigma = \text{Var}(\bar{X}) = \frac{1}{n} \bar{X}^T \bar{X}$

線形変換後のベクトル $\mathbf{s}_j = (s_{1j}, \dots, s_{nj})^T = \bar{X} \mathbf{a}_j \quad \mathbf{a}_j \in \mathbb{R}^m$
 $n \times m \quad m \times 1$
 j は射影軸のインデックス

2) 係数ベクトルが変われば線形変換後の値が変化する

- ①. 情報の量を分散の大きさととらえる
- ②. 線形変換後の変数の分散が最大となる射影軸を探索

$$\mathbf{s}_j = (s_{1j}, \dots, s_{nj})^T = \bar{X} \mathbf{a}_j \quad \mathbf{a}_j \in \mathbb{R}^m$$

線形変換後の分散

$$\text{Var}(\mathbf{s}_j) = \frac{1}{n} \mathbf{s}_j^T \mathbf{s}_j = \frac{1}{n} (\bar{X} \mathbf{a}_j)^T (\bar{X} \mathbf{a}_j) = \frac{1}{n} \mathbf{a}_j^T \bar{X}^T \bar{X} \mathbf{a}_j = \mathbf{a}_j^T \text{Var}(\bar{X}) \mathbf{a}_j$$

3) 以下の制約付き最適化問題を解く

- ①. ノルムが1となる制約を入れる（制約を入れないと無限に解がある）

目的関数 $\arg \max_{\mathbf{a} \in \mathbb{R}^m} \mathbf{a}_j^T \text{Var}(\bar{X}) \mathbf{a}_j$

制約条件 $\mathbf{a}_j^T \mathbf{a}_j = 1$

4) 制約付き最適化問題の解き方

- ①. ラグランジュ関数を最大にする計数ベクトルを探索（微分して0になる点）

ラグランジュ乗数

ラグランジュ関数 $E(\mathbf{a}_j) = \mathbf{a}_j^T \text{Var}(\bar{X}) \mathbf{a}_j - \lambda(\mathbf{a}_j^T \mathbf{a}_j - 1)$

目的関数

制約条件

5) ラグランジュ関数を微分して最適解を求める

①. 元のデータの分散共分散行列の固有値と固有ベクトルが、上記の制約付き最適化問題の解となる。

$$\text{微分} \quad \frac{\partial E(\mathbf{a}_j)}{\partial \mathbf{a}_j} = 2\text{Var}(\bar{X})\mathbf{a}_j - 2\lambda\mathbf{a}_j = 0 \quad \rightarrow \text{解} \quad \text{Var}(\bar{X})\mathbf{a}_j = \lambda\mathbf{a}_j$$

②. 射影先の分散は固有値と一致する。

$$\text{Var}(\mathbf{s}_1) = \mathbf{a}_1^T \text{Var}(\bar{X}) \mathbf{a}_1 = \lambda_1 \mathbf{a}_1^T \mathbf{a}_1 = \lambda_1$$

6) 分散共分散行列は正定値対象行列 → 固有値は必ず 0 以上・固有ベクトルは直行

- ・分散共分散行列を計算する
- ・固有値問題を解く
- ・(最大) m個の固有値と固有ベクトルのペアが出現する。
- ・k 番目の固有値を昇順に並べ、対応する固有ベクトルを第 k 主成分と呼ぶ

7) 寄与率

①. 第一～元次元分の主成分の分散は、元のデータの分散と一致する。

- ・2次元のデータを2次元の主成分で表示した時、固有値の和と元のデータの分散が一致
- ・第 k 主成分の分散は主成分に対応する固有値

$$V_{total} = \sum_{i=1}^m \lambda_i$$

元データの総分散 主成分の総分散

②. 寄与率：第 k 主成分の分散の全分散に対する割合（第 k 主成分が持つ情報量の割合）

③. 累積寄与率：第 1 - k 主成分まで圧縮した際の情報損失量の割合

$$c_k = \frac{\lambda_k}{\sum_{i=1}^m \lambda_i} \quad \begin{array}{l} \text{第 k 主成分の分散} \\ \text{主成分の総分散} \end{array}$$

$$r_k = \frac{\sum_{j=1}^k \lambda_j}{\sum_{i=1}^m \lambda_i} \quad \begin{array}{l} \text{第 1 - k 主成分の分散} \\ \text{主成分の総分散} \end{array}$$

5. アルゴリズム

5. 1 k 近傍法

分類問題のための機械学習手法

- 1) 最近傍のデータを k 個取って来てそれらが最も多く所属するクラスに識別する。
- 2) k を変化させると結果も変化する
- 3) $k = 1$ のとき最近傍法という
- 4) k が大きくなると境界が滑らかになる。

5. 2 K-means

- 1) 教師なし学習
- 2) クラスタリング手法
- 3) 与えられたデータを k 個のクラスタに分類する。
- 4) k 平均法のアルゴリズム
 - ①. 各クラスタ中心の初期値を設定する
 - ②. 各データ点に対して、各クラスタ中心との距離を計算し、最も距離が近いクラスタを割り当てる。
 - ③. 各クラスタの平均ベクトル（中心）を計算する。
 - ④. 収束するまで、②、③の処理を繰り返す。
- 5) 中心の初期値を変えるとクラスタリング結果も変わりうる。
 - ①. 初期値が離れる → うまくクラスタリングできる
 - ②. 初期値が近い → うまくクラスタリングできない

6. サポートベクターマシン

データを何らかの特徴に基づき 2 分類する際、双方のデータ群を等しい距離（マージン）で区切る平面を機械学習により発見する手法。

- ・未学習データに対しても高い識別性がある。
- ・仮説の設定や特徴の選択が必要。

サポートベクターを利用して予測を行う教師あり学習のモデルで、カーネル法により非線形分離を可能としている。

回帰と分類

判別境界に最も近い観測はサポートベクトルと呼ばれる。

特徴量から分類する。

Vanpink

スラック数：一部のサンプルの誤分類に寛容になる

カーネル法：決定境界を非線形にする。

カーネルトリック：計算量を現実的に抑えつつ、非線形分離を実現する。

異常検出