機会学習 要点のまとめ

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。

①線形で予測する場合は線形回帰

②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）

②出力（目的変数）

・スカラー値

（３）記述方法

①説明変数

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

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

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

（３）線形結合

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

入力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

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

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

①単回帰モデル

②多次元の場合、重回帰モデル

（２）データの変換

①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＝ｗ０＋ｗ１＊ｘ１＋ε

ｙ：目的変数

ｗ０：切片（未知）

ｗ１：回帰係数（未知）

ｘ１：説明変数

ε：誤差

（４）連立方程式

ｙ１＝ｗ０＋ｗ１＊ｘ１＋ε１

ｙ２＝ｗ０＋ｗ２＊ｘ２＋ε２

：

ｙｎ＝ｗ０＋ｗｎ＊ｘｎ＋εｎ

（５）行列表現

$y = Xw + \varepsilon$

$X=(x1,x2,.....xn)^T$

$y=(y1,y2,.....yn)^T$

$xi=(1,x1)^T$

$\varepsilon=(\varepsilon 1, \varepsilon 2,..... \varepsilon n)^T$

$w=(w0,w1)^T$

（６）説明変数が多次元の場合（ｍ≧１）

①線形重回帰モデル

②単回帰は直線、重回帰は曲線

（７）データへの仮定

①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

$y = w0 + w1 * x1 + w2 * x2 + \varepsilon$

ｙ：目的変数

ｗ０：切片（未知）

ｗ１、ｗ２：回帰係数（未知）

ｘ１、ｘ２：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）分割の必要性

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

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

（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

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^{m+1}} \mathrm{MSE}_{\mathrm{train}} \qquad \frac{\partial}{\partial \boldsymbol{w}}\mathrm{MSE}_{\mathrm{train}} = 0$$

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

$$\Rightarrow \frac{\partial}{\partial \boldsymbol{w}}\left\{\frac{1}{n_{\mathrm{train}}}\sum_{i=1}^{n_{\mathrm{train}}}(\hat{y}_i^{(\mathrm{train})} - y_i^{(\mathrm{train})})^2\right\} = 0$$

平均二乗誤差(残差平方和)を微分

$$\Rightarrow \frac{1}{n_{\mathrm{train}}}\frac{\partial}{\partial \boldsymbol{w}}\left\{(X^{(\mathrm{train})}\boldsymbol{w} - \boldsymbol{y}^{(\mathrm{train})})^T(X^{(\mathrm{train})}\boldsymbol{w} - \boldsymbol{y}^{(\mathrm{train})})\right\}$$

行列表現

$$\Rightarrow \frac{1}{n_{\mathrm{train}}}\frac{\partial}{\partial \boldsymbol{w}}\left\{\boldsymbol{w}^T X^{(\mathrm{train})T} X^{(\mathrm{train})}\boldsymbol{w} - 2\boldsymbol{w}^T X^{(\mathrm{train})T}\boldsymbol{y}^{(\mathrm{train})} + \boldsymbol{y}^{(\mathrm{train})T}\boldsymbol{y}^{(\mathrm{train})}\right\}$$

{ } 展開

$$\Rightarrow 2X^{(\mathrm{train})T}X^{(\mathrm{train})}\boldsymbol{w} - 2X^{(\mathrm{train})T}\boldsymbol{y}^{(\mathrm{train})} = 0$$

行列微分計算

$$\Rightarrow \hat{\boldsymbol{w}} = (X^{(\mathrm{train})T}X^{(\mathrm{train})})^{-1}X^{(\mathrm{train})T}\boldsymbol{y}^{(\mathrm{train})}$$

回帰係数

回帰係数ŵ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

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

予測値

$$\hat{\boldsymbol{y}} = \underset{n_{new}\times m+1}{X}(X^{(train)T}X^{(train)})^{-1}X^{(train)T}\boldsymbol{y}^{(train)}$$

（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。

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

（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

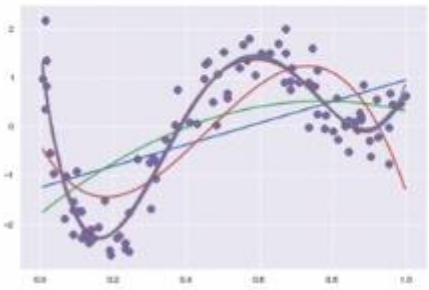
③スプライン関数／B スプライン関数

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

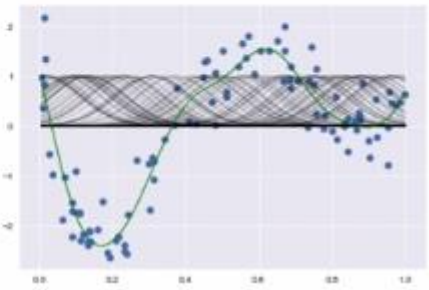
多項式 (1~9次)

$$\phi_j = x^j$$



ガウス型基底

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



正規分布は、全てを足して１になる。

既定関数ϕ（ｘ）に多項式関数ϕｊ＝ｘｊ乗

$\hat{y}$＝ｗ０＋ｗ１ϕ１（ｘｉ）＋ｗ２ϕ２（ｘｉ）

＝ｗ０＋ｗ１ｘｉ＋ｗ２ｘｉ^2

求めるべきｗについては線形である

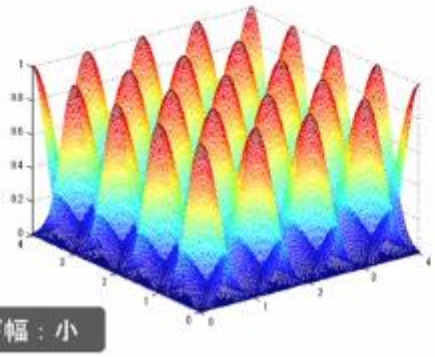
Ｇａｕｓｓ型基底関数

ϕｊ（ｘ）＝ｅｘｐ｛－（ｘ－μｊ）^{2}／２ｈｊ｝（－（ｘ－μｊ）^{2}／σ^2）

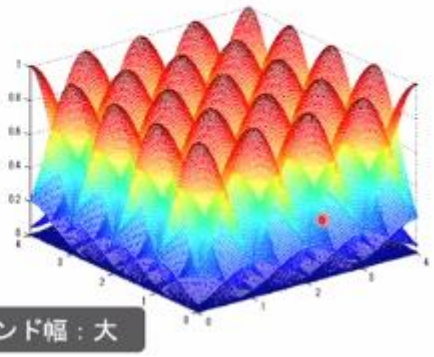
（２） ２次元の基底関数に基づく非線形回帰

2次元ガウス型基底関数

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



バンド幅：小



バンド幅：大

説明変数の数

説明変数 $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{im}) \in \mathbb{R}^m$

基底関数の数

非線形関数ベクトル $\boldsymbol{\phi}(\boldsymbol{x}_i) = (\phi_1(\boldsymbol{x}_i), \phi_2(\boldsymbol{x}_i), \cdots, \phi_k(\boldsymbol{x}_i))^T \in \mathbb{R}^k$

非線形関数の計画行列 $\Phi^{(train)} = (\boldsymbol{\phi}(\boldsymbol{x}_1), \boldsymbol{\phi}(\boldsymbol{x}_2), \cdots, \boldsymbol{\phi}(\boldsymbol{x}_n))^T \in \mathbb{R}^{n\times k}$

最尤法による予測値 $\hat{\boldsymbol{y}} = \Phi(\Phi^{(train)T}\Phi^{(train)})^{-1}\Phi^{(train)T}\boldsymbol{y}^{(train)}$

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

ｐｈｉｔｒａｉｎ＝ｐｈｉ（ｘ１）、ｐｈｉ（ｘ２）

尤度法（最小２乗法）

$\hat{w}$＝（ＸＴＸ）－１ＸＴｙ

逆行列

＝（ｐｈｉＴｐｈｉ）－１ｐｈｉＴｙ

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　正規化法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。

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

１０Ｃ５・・・・組合せ爆発

正則化法

線形回帰 $\hat{y}$ ＝ｗ（ＸＴＸ）－１ＸＴｙ

$\hat{w}$ ＝（ＸＴＸ）－１ＸＴｙ

$$X = \begin{matrix} 1 & 2 & 4 \\ 1 & 3 & 5.9 \\ 1 & 4 & 8.1 \end{matrix}$$

＝＞（ＸＴＸ）－１の要素はメチャメチャ大きくなる

ほぼ平行

Ｅ（ｗ）＝Ｊ（ｗ）＋λＷＴＷ

ＭＳＥ　　罰則項（Ｌ２ノルム）

Ｗを抑えつつ、ＭＳＥを小さくする

Ｓγ＝（ｙ－φｗ）Ｔ（ｙ－φｗ）＋γＲ（ｗ）

ハイパーパラメータ

解きたいのはＭＳＥの最小化　ｓ．ｔ．　Ｒ（ｗ）　≦　ｒ

（最適化）ＫＫＴ条件より

ｍｉｎ　ＭＥＳ＋λＲ（ｗ）

おまけを付けると不等式条件を解除

２．５　正規化法２

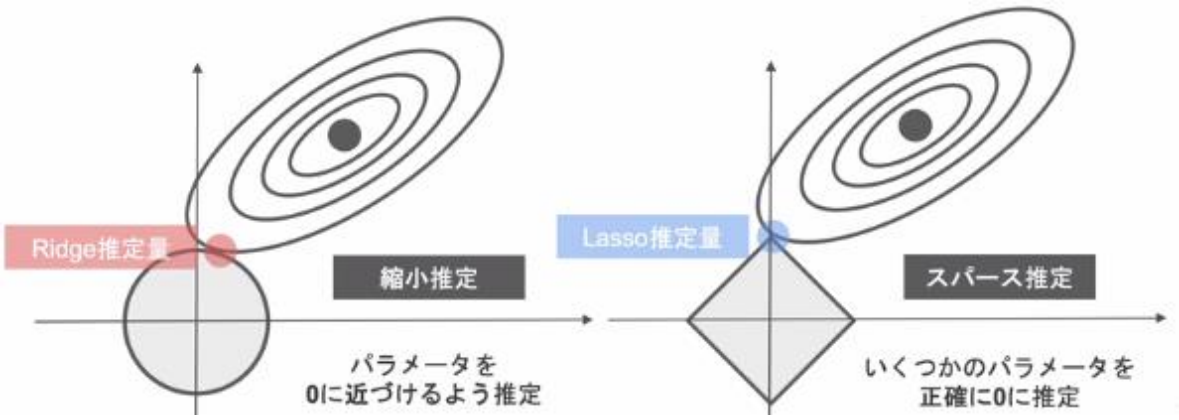
（１）正則化項（罰則項）の役割

①無い、最小２乗推定量

②Ｌ２ノルムを利用―＞リッジ推定量

③Ｌ１ノルムを利用―＞ラッソ推定量

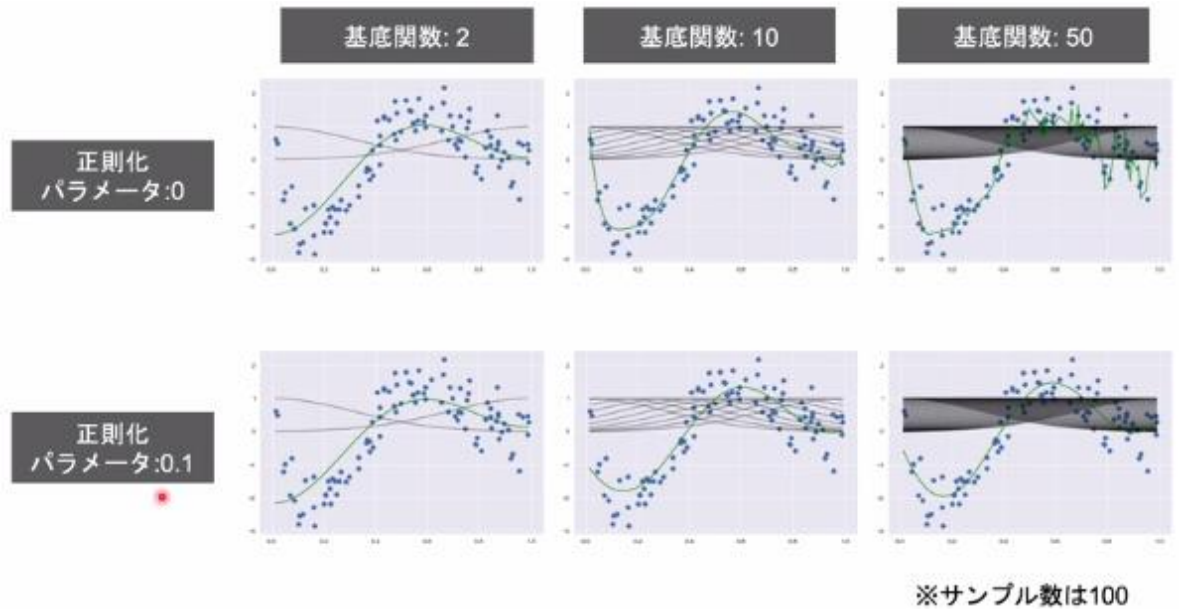
（２）正則化パラメータの役割



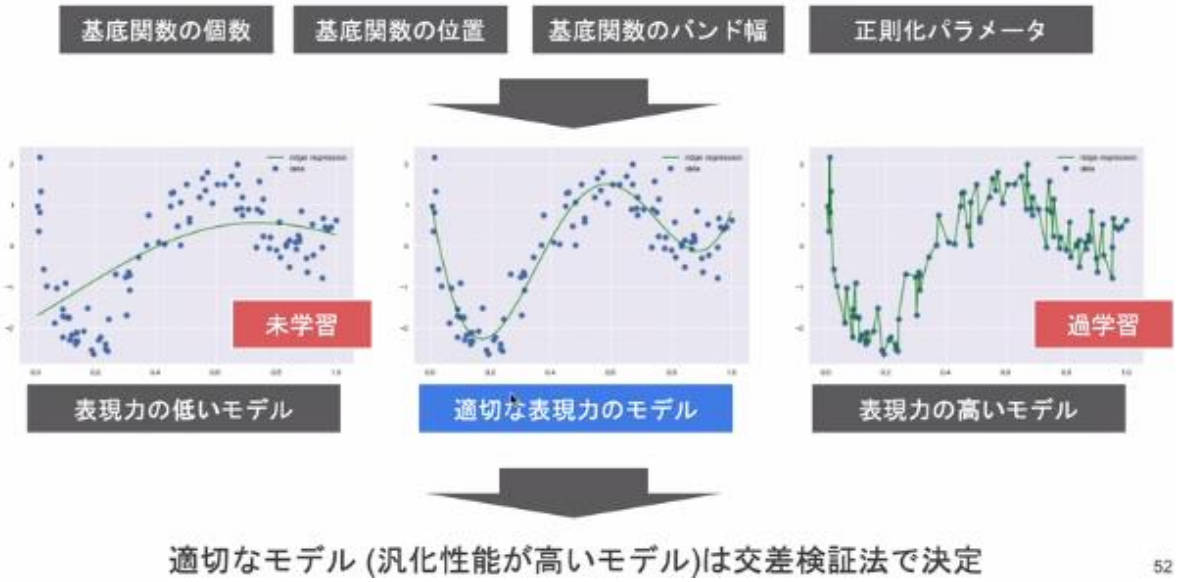
横軸ｗ０、縦軸ｗ１

縮小推定・スパース推定

２．６　正規化法３



２．７　モデル選択



（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

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

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で小さくなるモデルで決定する。

検証　学習　学習　学習　学習　→　モデル１

（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）

②出力（目的変数）

・０又は１の値

③タイタニックデータ、ＩＲＩＳデータなど

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

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

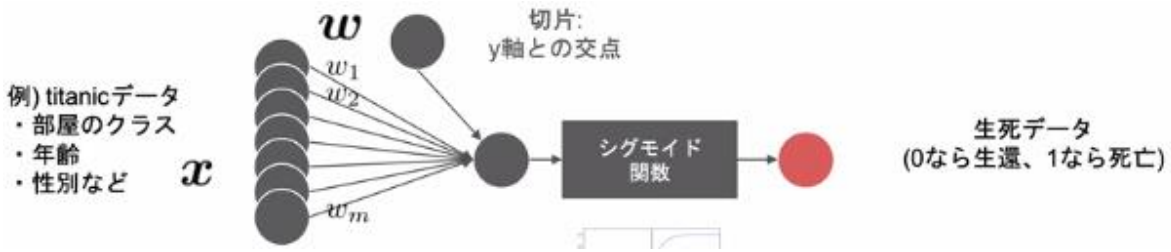
①分類問題を解くための教師あり機械学習モデル（教師データから学習）

・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　$\boldsymbol{w} = (w_1, w_2, \cdots, w_m)^T \in \mathbb{R}^m$

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



（２）シグモイド関数

①入力は実数・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

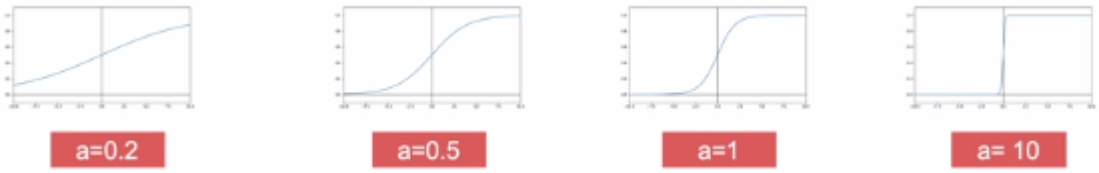
（３）パラメータが変わるとシグモイド関数の形が変わる

①a を増加させると、x=0 付近での曲線の勾配が増加

②a 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

$\sigma(x)=1/(1+\exp(-ax))$



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

$$\frac{\delta\ \sigma(x)}{\delta x} = \frac{\delta}{\delta x}\left(\frac{1}{1+\exp(-ax)}\right)$$

$$=(-1)\cdot\{1+\exp(-ax)\}^{2}\cdot\exp(-ax)\cdot(-a)$$ 連鎖律

$$=\frac{a\exp(-ax)}{(1+\exp(-ax))^{2}} = \frac{a}{1+\exp(-ax)}\cdot\frac{1+\exp(-ax)-1}{1+\exp(-ax)}$$

$$=a\,\sigma(x)(1-\sigma(x))$$

３．４　ロジスティック回帰モデル４

（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

シグモイド関数

$$P(Y=1\mid x)=\sigma(w0+w1x1+w2x2+\ldots\ldots+wmxm)$$

説明変数の実現値が与えられた際に Y=1 になる確率

データのパラメータに対する線形結合

表記

$$Pi=\sigma(w0+w1xi1+w2xi2+\ldots\ldots+wmxim)$$

$$P(Y=1\mid x)=\sigma(w0+w1x1)$$

P(Y=1 | x)：データが与えられた時に Y=1 になる確率

w0：切片（未知：学習で決める）

w1：回帰係数（未知：学習で決める）

x1：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・・・・

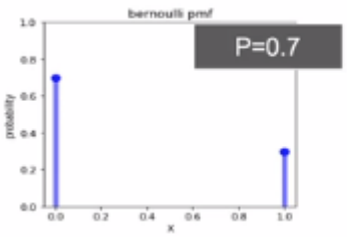
②ロジスティック回帰ではベルヌーイ分布を利用する。

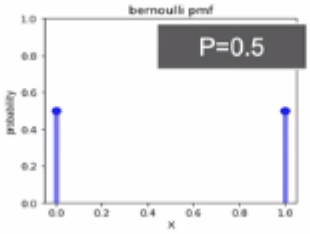
（２）ベルヌーイ分布

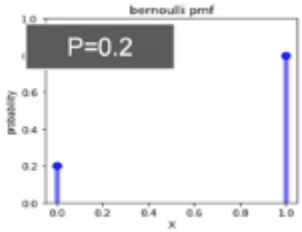
①数学において、確率ｐで１、確率１－ｐで０をとる離散確率分布（例：コイン投げ）

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	$Y \sim Be(p)$
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 y1～yn が同時に起こる確率	$P(y1,y2,\ldots.yn;p)=\Pi p^{yi}\ (1-p)^{1-yi}$
y1～yn のデータが得られた際の尤度関数	$P(y1,y2,\ldots.yn;p)=\prod_{i=1}^{n} p^{yi}\ (1-p)^{1-yi}$

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(x1,y1),(x2,y2),.....(xn,yn)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

$$P(Y=y1\mid x1)=p1^{y1}(1-p1)^{1-y1}=\sigma(w^{T}\ x1)^{y1}(1-\sigma(w^{T}\ x1))^{1-y1}$$

$$P(Y=y2\mid x2)=p2^{y2}(1-p2)^{1-y2}=\sigma(w^{T}\ x2)^{y2}(1-\sigma(w^{T}\ x2))^{1-y2}$$

:

$$P(Y=yn\mid xn)=pn^{yn}(1-pn)^{1-yn}=\sigma(w^{T}\ xn)^{yn}(1-\sigma(w^{T}\ xn))^{1-yn}$$

y1～yn のデータが得られた際の尤度関数

$$P(y1,y2,\ldots,yn\mid w0,w1,\ldots.,wm)=\prod_{i=1}^{n} pi^{yi}\ (1-pi)^{1-yi}$$

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

$$=L(w)$$　　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

３．１０　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ｘ１＜ｘ２の時、必ずlog(x1)<(log(x2)となる）

③「尤度関数にー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。

$$E(w0,w1,\ldots,wm)=-\log L(w0,w1,\ldots,wm)$$
$$=\sum_{i=1}^{n}\{yi\log pi+(1-yi)\log(1-pi)\}$$

３．１１　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

②η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

$$\frac{\delta E(w)}{\delta w}=\sum\frac{\delta Ei(w)}{\delta pi}\frac{\delta pi}{\delta w}$$ 連鎖律

$$=\sum\left(\frac{yi}{pi}-\frac{1-yi}{1-pi}\right)\frac{\delta pi}{\delta w}$$ 対数尤度関数のpに関する微分

$$=\sum\left(\frac{yi}{pi}-\frac{1-yi}{1-pi}\right)Pi(1-pi)xi$$ シグモイド関数の微分

$$=-\sum(yi(1-pi)-pi(1-yi))xi$$ 式を整理

$$=-\sum(yi-pi)xi$$

（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。

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

（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。

①ｎ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る。

②確率的勾配降下法を利用して解決する。

３．１２　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。

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

３．１３　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を以下に記述します。

		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。

$$\frac{TP+TN}{TP+FP+FN+TN}$$

（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―＞リコールやプレシジョンを使う

・異常値検出のタスクっぽいが、説明の簡易化のため分類で解く

・Folse 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（Fales 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。

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。

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う。

②Precision と Recall の調和平均

Recall と Precision のバランスを示している

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

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変量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

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

平均（ベクトル）

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$$

データ行列

$$\bar{X} = (\boldsymbol{x}_1 - \bar{\boldsymbol{x}}, \cdots, \boldsymbol{x}_n - \bar{\boldsymbol{x}})^T$$

分散・共分散行列

$$\Sigma = Var(\bar{X}) = \frac{1}{n}\bar{X}^T\bar{X}$$

線形変換後のベクトル

$$\boldsymbol{s}_j = (s_{1j}, \cdots, s_{nj})^T = \underset{n \times m}{\bar{X}}\underset{m \times 1}{\boldsymbol{a}_j} \qquad \boldsymbol{a}_j \in \mathbb{R}^m$$

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

$$\boldsymbol{s}_j = (s_{1j}, \cdots, s_{nj})^T = \bar{X}\boldsymbol{a}_j \qquad \boldsymbol{a}_j \in \mathbb{R}^m$$

線形変換後の分散

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

（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

$$\arg\max_{\boldsymbol{a}\in\mathbb{R}^m} \boldsymbol{a}_j^T Var(\bar{X})\boldsymbol{a}_j$$

制約条件

$$\boldsymbol{a}_j^T\boldsymbol{a}_j = 1$$

（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）

ラグランジュ関数

$$E(\boldsymbol{a}_j) = \underset{\text{目的関数}}{\boldsymbol{a}_j^T Var(\bar{X})\boldsymbol{a}_j} - \underset{\text{ラグランジュ乗数}}{\lambda}\underset{\text{制約条件}}{(\boldsymbol{a}_j^T\boldsymbol{a}_j - 1)}$$

（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。

微分 $\frac{\partial E(\boldsymbol{a}_j)}{\partial \boldsymbol{a}_j} = 2Var(\bar{X})\boldsymbol{a}_j - 2\lambda \boldsymbol{a}_j = 0$ →解 $Var(\bar{X})\boldsymbol{a}_j = \lambda \boldsymbol{a}_j$

②射影先の分散は固有値と一致する。

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

（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

①分散共分散行列を計算する

②固有値問題を解く

③（最大）m個の固有値と固有ベクトルのペアが出現する。

④k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

$c_k = \frac{\lambda_k}{\sum_{i=1}^{m} \lambda_i}$　第k主成分の分散 / 主成分の総分散

$r_k = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{i=1}^{m} \lambda_i}$　第１－k主成分の分散 / 主成分の総分散

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。

①未学習データに対しても高い識別性がある。

②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。