機械学習レポート

2024/6/1

1　機械学習の分類

機械学習は、以下の 2 つに分類される。

1. 教師あり学習
   入力データとそれに対応する出力データを、機械学習アルゴリズムに与えて学習する。例えば、イヌ・ネコのラベルがある画像を学習させて画像認識に活用する、などがある。

2. 教師なし学習
   入力データのみを、機械学習アルゴリズムに与えて学習する。例えば、市場調査の購買分析など、正解がないデータについて調査するときに用いる。

次節より、代表的なアルゴリズムの概要をまとめる。

1.1　教師あり学習

1.1.1　線形回帰モデル

ある m 次元の入力 $\boldsymbol{x}$（離散あるいは連続値）から出力 y（連続値）を予測する、回帰問題を解くモデルの一つ。出力が、式 1 のように、入力と m 次元パラメータ $\boldsymbol{w}$ の線形結合で表せるととらえる。ただし、式 1 で求まる出力値は実際の正解データとは異なる予測値であるため、ハットをつけている。

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

$\boldsymbol{w}$ が求まれば出力の予測値も求まるため、学習データと最も誤差が小さい $\boldsymbol{w} = (w_0, w_1, \cdots, w_m)$ を推定することが目的となる。
$\boldsymbol{w}$ の推定には、式 2 で表される、学習データと予測値の平均二乗誤差の最小値を探索する最小 2 乗法を用いる。

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

学習データの平均二乗誤差の最小化は、その勾配が 0 になる点を求めればよいため、MSE を $\boldsymbol{w}$ に関して微分したもの（式 3）が 0 となる $\boldsymbol{w}$ を求めればよい。

$$\frac{\partial}{\partial \boldsymbol{w}}\mathrm{MSE}_{\mathsf{train}} = \frac{\partial}{\partial \boldsymbol{w}}\{\frac{1}{n_{\mathsf{train}}}\sum_{i=1}^{n_{\mathsf{train}}}(\hat{y}_i^{(\mathsf{train})} - y_i^{(\mathsf{train})})^2\} = 0 \tag{3}$$

式 3 を $\boldsymbol{w}$ について求めると、$\boldsymbol{w}$ の予測値 $\hat{\boldsymbol{w}}$ は式 4 と求まる（ただし、$X^{(\mathsf{train})\mathsf{T}}X^{(\mathsf{train})}$ の逆行列が存在すると仮定する）。ここで、x_{ij} は、i 番目の学習データで j 次元目の値である。

$$\hat{\boldsymbol{w}} = (X^{(\mathsf{train})\mathsf{T}}X^{(\mathsf{train})})^{-1}X^{(\mathsf{train})\mathsf{T}}\boldsymbol{y}^{(\mathsf{train})} \tag{4}$$

where,

$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1m} \\ 1 & x_{21} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n_{\mathsf{train}}1} & \cdots & x_{n_{\mathsf{train}}m} \end{pmatrix} \tag{5}$$

1.1.2 非線形回帰モデル

回帰問題を解くモデルの一つ。出力が、式 6 のように、入力と基底関数と呼ばれる既知の非線形関数 ϕ の線形結合で表せるととらえる。

$$\hat{y} = \boldsymbol{w}^{\mathsf{T}}\phi(\boldsymbol{x}) + w_0 = \sum_{j=1}^{m} w_j\phi_j(\boldsymbol{x}) + w_0 \tag{6}$$

$\boldsymbol{w}$ が求まれば出力の予測値も求まるため、学習データと最も誤差が小さい $\boldsymbol{w} = (w_0, w_1, \cdots, w_m)$ を推定することが目的となる。
$\boldsymbol{w}$ の推定には、線形結合同様に最小二乗法が用いられる。さらに、式 1 と 6 より、非線形結合は線型結合において $\boldsymbol{x}$ が $\phi(\boldsymbol{x})$ に置き換わったものである。そのため、式 4 より、基底関数の数を k とすると $\hat{\boldsymbol{w}}$ は式 7 のように求まる（ただし、$\Phi^{(\mathsf{train})\mathsf{T}}\Phi^{(\mathsf{train})}$ の逆行列が存在すると仮定する）。

$$\hat{\boldsymbol{w}} = (\Phi^{(\mathsf{train})\mathsf{T}}\Phi^{(\mathsf{train})})^{-1}\Phi^{(\mathsf{train})\mathsf{T}}\boldsymbol{y}^{(\mathsf{train})} \tag{7}$$

where,

$$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{im}) \tag{8}$$

$$\Phi^{(\mathsf{train})} = \begin{pmatrix} 1 & \phi_1(\boldsymbol{x}_1) & \cdots & \phi_k(\boldsymbol{x}_1) \\ 1 & \phi_1(\boldsymbol{x}_2) & \cdots & \phi_k(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_1(\boldsymbol{x}_{n_{\mathsf{train}}}) & \cdots & \phi_k(\boldsymbol{x}_{n_{\mathsf{train}}}) \end{pmatrix} \tag{9}$$

1.1.2.1 実装演習

図 1 は、データ点に対して基底関数が $x^j(j = 1, 2, \cdots, 9)$ の非線形回帰モデルで学習した結果である。4 次元以降は全て似たようなモデルとなっているため、5 次元以上のモデルを選択しても 4 次元と比べて大幅に性能が向上しないといえる。また、高次元モデルを選択することで表現力が増えたことで過学習が起きる可能性がある。

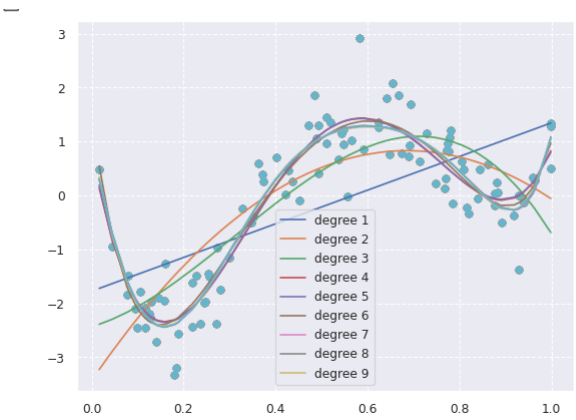

図 1: 基底関数が多項式（1〜9 次）の非線形回帰モデル

1.1.3 ロジスティック回帰モデル

ある m 次元の入力 $\boldsymbol{x}$（離散あるいは連続値）からクラス $Y \in \{0, 1\}$ に分類する、分類問題を解くモデルの一つ。出力が、式 10 のように、線型結合モデルを入力とするシグモイド関数 σ で表せるととらえる。シグモイド関数は、入力に対して 0 から 1 の値をとる関数である。よって、$\hat{y}$ は出力が 1 となる確率 $P(Y = 1|\boldsymbol{x})$ の値ともいえる。ただし、実際には確率が 0.5 以上ならば 1、未満なら 0 と予測する。

$$\hat{y} = P(Y = 1|\boldsymbol{x}) = \sigma(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + w_0) = \sigma(\sum_{j=1}^{m} w_j x_j + w_0) \tag{10}$$

where,

$$\sigma(x) = \frac{1}{1 + \exp(-ax)} \tag{11}$$

推定するべきパラメータは、n_{train} 個の学習データに対して、最も当てはまりのよい $\boldsymbol{w}$ である。言い換えれば、$\boldsymbol{y} = (y_1, y_2, \cdots, y_{n_{\mathsf{train}}})$ の値が同時にとれる確率が最大となるような $\boldsymbol{w}$ を求めればよい。
ロジスティック回帰モデルでは、同時確率の計算にベルヌーイ分布を用いる。つまり、値が 1 となる確率 p のもとで、n_{train} 回の試行で $\boldsymbol{y} = (y_1, y_2, \cdots, y_{n_{\mathsf{train}}})$ の値が同時にとれる確率 $P(y_1, y_2, \cdots, y_{n_{\mathsf{train}}}; p)$ は式 12 のように求まる。

$$P(y_1, y_2, \cdots, y_{n_{\mathsf{train}}}; p) = \prod_{i=1}^{n} p^{y_i}(1 - p)^{1-y_i} \tag{12}$$

さらに、p は式 10 で表されるシグモイド関数であるため、式 13 のように変形できる。尤度関数 L を最大とする $\boldsymbol{w}$ を求めればよい。

$$\begin{aligned} &P(y_1, y_2, \cdots, y_{n_{\mathsf{train}}}; w_0, w_1, \cdots, w_{n_{\mathsf{train}}}) = L(\boldsymbol{w}) \\ &= \prod_{i=1}^{n} \sigma(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_i + w_0)^{y_i}(1 - \sigma(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_i + w_0))^{1-y_i} \end{aligned} \tag{13}$$

ここで、式 13 は積の形であるため、微分しやすいように対数をとり、さらにマイナスをかけて最小化問題に帰着させた $E(\boldsymbol{w})$ は、式 14 となる。

$$\begin{aligned} E(\boldsymbol{w}) &= -\log L(\boldsymbol{w}) \\ &= -\sum_{i=1}^{n_{\text{train}}} \{y_i \log \sigma(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_i + w_0) + (1-y_i)\log\big(1-\sigma(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_i + w_0)\big)\} \end{aligned} \tag{14}$$

式 14 は推定したい $\boldsymbol{w}$ にシグモイド関数がかかっているため、線形回帰モデルのように解析的に最小化となる $\boldsymbol{w}$ を求めることが難しい。そのため、反復学習によりパラメータを逐次的に更新するアプローチの一つである勾配降下法で、$\boldsymbol{w}$ を推定する。具体的には、学習率 η として式、15 に従って $\boldsymbol{w}$ を更新していく。

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^k - \eta\frac{\partial E(\boldsymbol{w})}{\partial \boldsymbol{w}} = \boldsymbol{w}^k + \eta\sum_{i=1}^{n_{\text{train}}}(y_i - \sigma(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_i + w_0))\boldsymbol{x}_i \tag{15}$$

式 15 をみると、$\boldsymbol{w}$ を更新するのに n_{train} 個の総和を求める必要があり、学習データ量が大きいとその分計算コストも高くなる。そのため、更新の際に n_{train} 個全て計算するのではなく、一つ（あるいは少量）のデータしか計算しない、確率的勾配降下法という方法も存在する。

1.1.3.1　実装演習　図 2 は、タイタニックの乗客データ（titanic_train.csv）について、null 値を中央値で補完した年齢（AgeFill）と 1〜3 等の旅客クラス（Pclass）を入力 $\boldsymbol{x}$ として、その客が生存したかどうかをロジスティック回帰モデルで学習した結果である。結果から、旅客クラスが上であるほど生存している割合が高いことと、旅客クラスに関わらず年齢が高いほど生存者が少ないことがわかる。

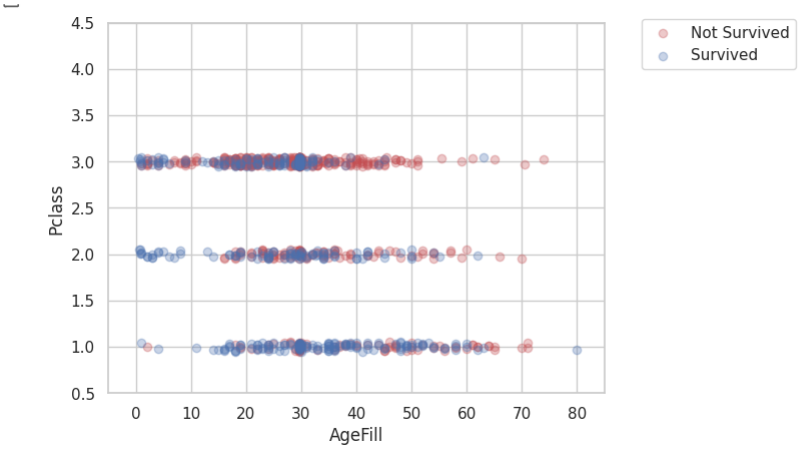

図 2: 年齢と旅客クラスを入力としたロジスティック回帰モデル

1.1.4 k 近傍法

分類問題を解く方法の一つである。最近傍のデータを k 個とってきて、それらがもっとも多く所属するクラスに識別する方法である。k を変化させると結果も変わるため、適切な k を探索するようなアルゴリズムも存在する。

1.1.4.1 実装演習 図 3 は、黄色と青色の 2 クラスに分けられた 2 次元データである。これを k 近似法 (k=3) で学習すると、図 4 となる。k=3 で、比較的周囲の点しか探索しないため、学習データのうち周囲に黄色が存在する一部の青色の点が黄色のクラスと判別される結果となっている。

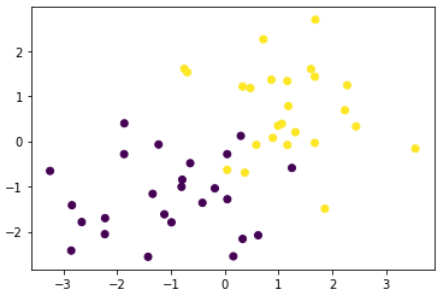

図 3: 学習に使う 2 次元データ

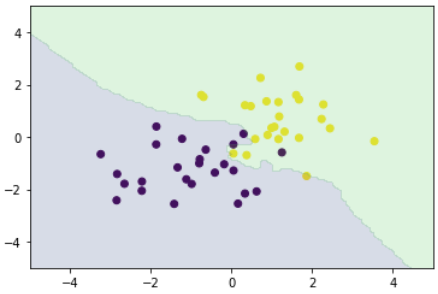

図 4: k 近似法 (k=3)

1.2 教師なし学習

1.2.1 主成分分析

多変量のデータをより少数個の指標に圧縮させる、次元圧縮の一つである。多変量データに対して、少数変数を利用した分析や可視化（2,3 次元の場合）が実現可能となる。ベースとなる考え方は、情報の量を分散の大きさととらえ、どの指標に切り取れば最も分散が大きくなるか（＝情報の損失が少なくなるか）を考える。

1.2.2 k-means

データを k 個のクラスタ（特徴の似ているもの同士のグループ）に分類する、クラスタリング手法の一つである。下記の手順に従って各データをクラスタに割り当てる。

1. 各クラスタ中心の初期値をランダムに設定する
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

1.2.2.1 実装演習 図 5 は、ランダムに生成した 2 次元データである。これを k-means(k=3) で学習すると、図 6 となる。なお、×がクラスタ中心である。一方で図 7 のように、k=4 といったデータの分布とは異なるクラス数を指定すると、クラスの分布も大きく異なってしまう。そのため、k 近似法同様、k の決定には注意が必要である。

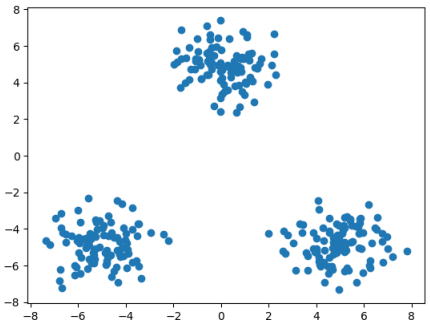

図 5: 学習に使う 2 次元データ

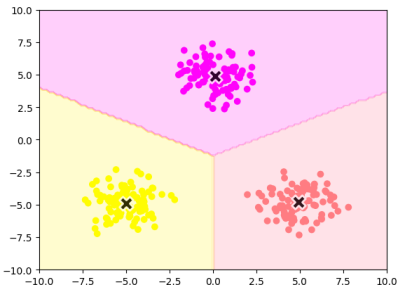

図 6: k-means(k=3)

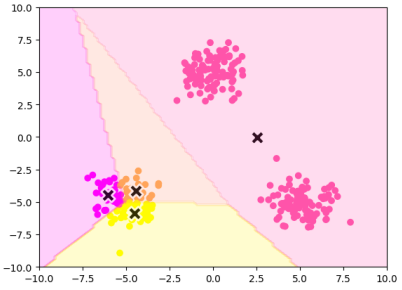

図 7: k-means(k=4)

2　機械学習の課題

2.1　訓練誤差と汎化誤差

1 章の機械学習モデルの性能を評価するため、全データを学習と検証データに分け、学習データで学習させたモデルを検証データで評価する。その際、学習データに対するモデルの誤差を訓練誤差、検証データに対するモデルの誤差を汎化誤差とすると、以下のことがいえる。

- 訓練誤差も汎化誤差も小さい場合、そのモデルはよく汎化している可能性がある
- 訓練誤差は小さいが汎化誤差が大きい場合、そのモデルは学習データに対して過学習している可能性がある
- 訓練誤差も汎化誤差も大きい場合、そのモデルは未学習の可能性がある

未学習に対する対策として、より表現力のあるモデルを利用するなどがある。一方、過学習について、非線形関数の基底関数を増やすと表現力が増え、学習データに対して過学習することが知られている。図 8 は、青色のデータ点に対して基底関数が 50 個のガウス型基底で学習した非線形関数モデルである。このように、学習データの分布に対して滑らかなモデルを作ることができず、未知のデータに対して良い識別をしにくくなる。このことから、過学習に対する対策として以下

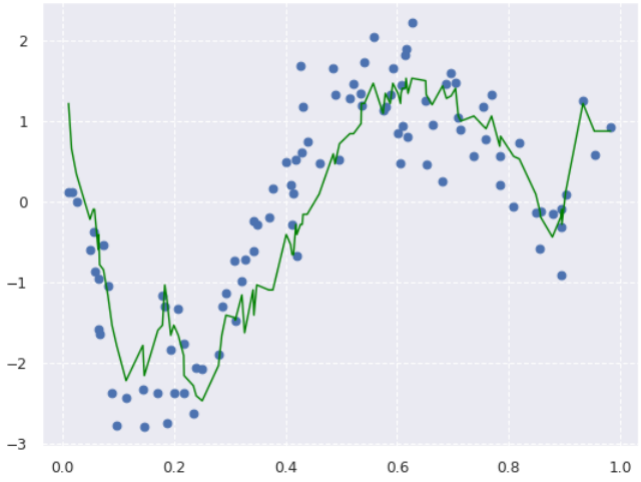

図 8: ガウス型基底 (規定関数の数 50) の非線形関数モデル

の 3 点が挙げられる。

1. 学習データの数を増やす
2. 不要な変数（もしくは基底関数）を削除してモデルの表現力を抑止する
3. 正規化法を利用して表現力を抑止する

次節では、3 点目の正規化法についてまとめる。

2.2　正規化法

正規化法とは、非線形回帰モデルに対して、「モデルの複雑さに伴って、その値が大きくなる正規化項（罰則項）を課した関数」を最小化する条件のもとで、モデルの最適化を行う方法である。具体的には、正規化項を $\gamma R(\boldsymbol{w})$（γ はハイパーパラメータ）とすると、式 16 の最小化の条件のもとで、モデルの最適化を行う。

$$S_\gamma = (\boldsymbol{y} - \Phi\boldsymbol{w})^\mathsf{T}(\boldsymbol{y} - \Phi\boldsymbol{w}) + \gamma R(\boldsymbol{w}) \tag{16}$$

$R(\boldsymbol{w})$ には、L1 ノルムを利用した Ridge 推定量や L2 ノルムを利用した Lasso 推定量がある。図 9 は、Ridge 推定量を用いた、基底関数が 50 個のガウス型基底で学習した非線形回帰モデルである。推定量を用いない図 8 に比べ、学習データに対して滑らかな曲線となっており、未知のデータに対しても良い識別を与えやすくなっている。

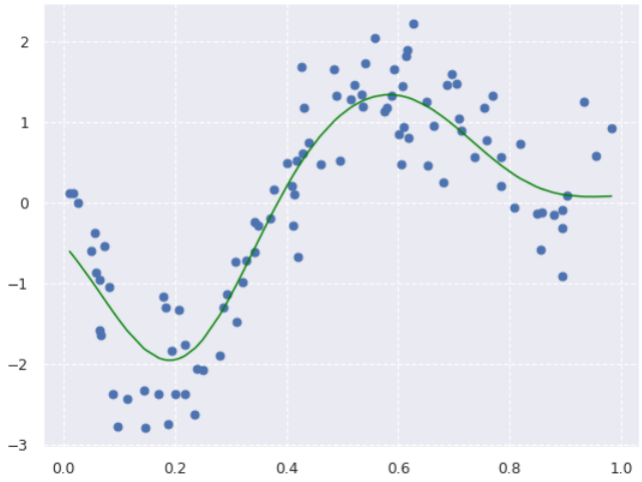

図 9: Ridge 推定量ガウス型基底 (基底関数の数 50) の非線形回帰モデル

3　検証集合

この章では、機械学習モデルの性能を評価するための方法について、以下の 2 つをまとめる。

1. ホールドアウト法
2. 交差検証法

3.1　ホールドアウト法

データを学習用と検証用の 2 つに分類して評価する、最も基本的な方法である。具体的には、学習用でモデルを学習させ、検証用でモデルの性能を評価する。学

習用と検証用データ数の配分によって、学習精度と性能評価の精度に差が生じる。また、決まったデータを学習・検証に用いるため、データ数が少ないとバイアスなどによって良い性能評価を与えにくいという欠点がある。

3.2 交差検証法

ホールドアウト法同様に、データを学習用と検証用の 2 つに分類して評価するが、データの分け方を変えることで 1 つのモデルに対して複数回学習と検証を行う点が異なる。最終的なモデルの性能評価としては、複数回の検証結果の精度の平均（CV 値）がよく用いられている。

4 性能指標

ここでは、学習済みのロジスティック回帰モデルの性能を評価する指標についてまとめる。以下の 3 つがある。

1. 正解率
2. 再現率
3. 適合率
4. F 値

4.1 正解率

モデルの予測が正解した数を検証用データの数で割った値である。

4.2 再現率

正解が 1（あるいは positive）であるデータのうち、モデルが 1（あるいは positive）と予測できた割合である。病気の検診など、陽性であるものを陰性と誤診するのをなるべく避けたい状況におけるモデルの評価で用いられる。

4.3 適合率

モデルが 1（あるいは positive）と予測したデータのうち、本当に 1（あるいは positive）であった割合である。スパムメールの検出など、陰性（スパム）と予測したものが確実にスパムであるほうが嬉しい状況におけるモデルの評価で用いられる。

4.4 F 値

再現率と適合率の調和平均である。再現率と適合率がトレードオフの関係にあることに着目し、再現率と適合率がバランスよく高いモデルを評価するために用いられる指標である。