知能システム パーセプトロン

大枝真一

2018 年 1 月 12 日

1　はじめに

大量に蓄積されたデータから有益な知識を抽出しようとするビッグデータ解析に注目が集まっている．データから規則性や外れ値などの知識を抽出する手法はデータマイニングと呼ばれ古くから研究されてきた．それが，技術の進歩とともにさまざまなデータを自動取得することが可能となり，真の価値を抽出しようとする試みがビジネスや医療など様々な分野で行われている．データマイニングを行うための手法として，計算機を使用してデータから自動的に規則性や外れ値を抽出する手法として機械学習がある．大量のデータを計算機に投入すると，機械が自分自身で学習してルールを抽出するということから，機械学習に強い関心を持つ人が多い．しかし，機械学習を習得しようとするといくつかの障壁にぶつかる．

機械学習の初学者が一番最初に驚くことは，複雑な数学の知識を必要とすることである．華やかな「ビッグデータ解析」という言葉とは裏腹に，実際は黙々と数式を読み解かなければならない．次に，立ちはだかる壁が，理解した数式をどうやってプログラムを作成するかということである．プログラミングの知識のある人はゲームを作ったり，簡単なデータ処理プログラムを作ったことがあったとしても，数式をプログラムに落とし込むのは難しく感じる人も多いようである．

つまり，機械学習を理解して，実際にデータマイニングツールとして使いこなせるようになるには，数式によって表現されたアルゴリズムとプログラミングのスキルが要求されるが，さっそくその 2 つにつまづいてしまう．しかし，最初の導入さえしっかり行い，機械学習を学ぶための方法を習得すれば，後は自学自習が可能となる．

そこで，本文書は機械学習の初学者に向けた内容となっている．また，機械学習はデータ処理のための手法である．したがって，機械学習を学ぼうとする読者は，実際にどうやって機械学習をデータに適用するのか関心がある人も多いと思われる．そこで，Python を用いて，プログラミングを作成して，実行するまでの過程も説明する．本文書は，初学者が一通りの機械学習の基礎を学び，データ解析ができるように，基本的な内容に絞ってある．機械学習について書かれた書籍はたくさん出版されている．より深く高度な機械学習技術について学びたい読者は，ぜひステップアップしていただきたい．そのときに本文書の内容が役にたてば幸いである．

2　パターン認識とは

次のような例を考えてみよう．魚を自動的に仕分ける機械を作成することになったとしよう．魚は A と B の 2 種類である．漁師によって魚 A と B の仕分けが既に済んでいる魚 A，B がそれぞれ 100 匹おり，魚の特

徴として体長と体重を計測している．これを表 1 に示す．

表 1　魚のデータ

データ番号	体長 (cm)	体重 (g)	魚の種類
1	85.0	2745	A
2	119.8	2680	B
3	96.1	2712	B
4	85.0	2745	A
5	108.2	2786	A
...	...	...	...

図 1 に横軸に体長，縦軸に体重としてプロットしたものを示す．これを見ると，おおよそ 2 種類の魚の分類はできそうである．一方，魚 A,B の境界領域も存在し，いかにして識別するかが問題となる．ここで重要なことは，これらの 200 匹のデータは既に漁師が釣った魚のものであり，これから釣る魚のデータはわからないという点である．これから釣る魚の分類がうまくできるように，既にあるデータから識別を行う機械を作ることが目的となる．このように既にあるデータをトレーニングデータ，未知のデータをテストデータという．また，テストデータに対する識別精度を汎化能力という．また，トレーニングデータに対しての当てはまりが強すぎて，ノイズまで学習してしまう状態を過学習という．我々の目的は過学習を抑えつつ，かつ汎化能力が高い機械を作ることである．

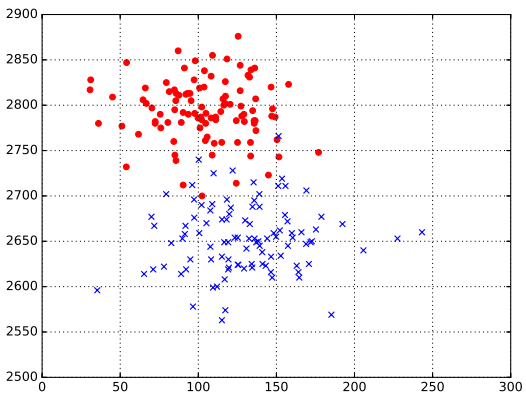

図 1　魚 A と B の分布

3　必要な Python の知識

機械学習を実装するためのプログラミング言語として，これから Python を用いる．機械学習には様々な手法が提案されている．これらを実際に試すには Python が最適である．Python はインタプリタ型の高水準言語であり，多種多様なことを素早く検証するには特に適している．何らかのプログラミング言語を知っている人ならば，Python の習得は簡単である．以下に，Python を初めて使う人のための資料があるので参考にしてみると良い．

https://github.com/OedaLab/BeginnerForPython
https://github.com/crotsu/LearningPython/blob/master/numpy.ipynb

ここでは，データ識別を行う機械学習プログラムを作成せするために必要と思われる Python の知識を最短で習得するための解説を行う．

3.1 データ読み込み

まず，データを読み込んでみよう．データとプログラムは
/home/class/j5/IntelligentSystem/2015/MachineLearning
にある．

```
# 2 次元プロットデータ（2 クラス）
# 表示

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# データ読み込み
df = pd.read_csv('fishData.csv')

# 散布図をプロットする
for i in range(len(df)):
    if df.cls[i]==1:
        plt.scatter(df.x1[i],df.x2[i], color='r',marker='o', s=30)
    else:
        plt.scatter(df.x1[i],df.x2[i], color='b',marker='x', s=30)

# グリッド表示
plt.grid(True)

# 表示
plt.show()
```

3.2 データの平均と標準偏差

データには魚 A,B にラベルが割り当てられている．魚 A には 1 が，魚 B には-1 が割り当てられている．次に，魚の平均と標準偏差を調べてみよう．

Pandas は非常に強力なデータ解析ツールであるが，慣れるまでに多少時間がかかる．素朴にデータにアクセスするには次のようにする．

```
a_x1 = df[df['cls']==1].x1
a_x2 = df[df['cls']==1].x2
b_x1 = df[df['cls']==-1].x1
b_x2 = df[df['cls']==-1].x2
```

課題 1 魚 **A** と魚 **B** の平均と標準偏差をそれぞれ計算するプログラムを作成せよ．

3.3 データの正規化

データは魚の特徴量となっている．特徴量は体長 (cm) と体重 (g) であり，これらは単位が異なる．そこで，データの正規化を行う．データの正規化は次の式によって変換する．

$$z = \frac{x - \mu}{\sigma}$$

変換された後の値を z 値と呼ぶ．

課題 2 データを正規化して，プロットするプログラムを作成せよ．

4 データ生成モデル

これまで，データは自然に取得できると考えていたはずである．しかし，我々はデータを観察しながら，「魚 A はだいたいこういう特徴をもっている」などの経験則を用いて分類するはずである．ここからはデータの生成モデルを考える．すなわち，観測データは何らかの確率モデルから生成されたと考えるのである．データには個体差が生じるが，これは正規分布として考えることが自然である．

そこで，実際にデータを生成することを通して，データ生成モデルを理解しよう．

課題 3 魚 **A,B** のデータ生成プログラムを作成せよ．魚 **A** の体長の平均は **110cm** であり標準偏差は **30**，体重の平均は **2800g** であり標準偏差は **35** とする．また，魚 **B** の体長の平均は **130cm** であり標準偏差は **40**，体重の平均は **2650g** であり標準偏差は **45** とする．

5 パーセプトロン（線形識別器）

2 クラスの識別問題を考える．d 次元の特徴ベクトルを $\boldsymbol{x} = (x_1, \cdots, x_d)^T$ とし，それらに 2 クラス C_1, C_2 のラベルが割り当てられているとする．2 クラスを線形分離する線形識別関数は，

$$f(\boldsymbol{x}) = \sum_{i=1}^{d} w_i x_i + w_0 \tag{1}$$

と表される．このとき，$w_0, w_1, \cdots, w_d$ は重み係数と呼ばれる．とくに，w_0 はバイアス項と呼ぶ．重み係数ベクトルを $\boldsymbol{w} = (w_1, \cdots, w_d)^T$ と表すと，

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 \tag{2}$$

とベクトルで表すことができる．さらに，

$$\mathbf{x} = (1, \boldsymbol{x}^T)^T = (1, x_1, x_2, \cdots, x_d)^T \tag{3}$$

$$\mathbf{w} = (w_0, w_1, w_2, \cdots, w_d)^T \tag{4}$$

とおけば，ベクトルの内積を使って，

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 = \mathbf{w^T x} \tag{5}$$

と記述することができる．

識別境界を $f(\boldsymbol{x}) = 0$ とすれば，識別規則は

$$\begin{cases} f(\boldsymbol{x}) \geq 0 \Rightarrow C_1 \\ f(\boldsymbol{x}) < 0 \Rightarrow C_2 \end{cases} \tag{6}$$

となる．

クラス c_i の線形識別関数を

$$g_i(\boldsymbol{x}) = \mathbf{w_i^T x} \tag{7}$$

とすれば，$\boldsymbol{x}$ を判別する演算は，この式にある線形和の計算と i に関する最大化によって簡単に構成することができる．一般に，このような線形演算と最大化によるパターン識別系をパーセプトロン (perceptron) と呼ぶ．ただし，パーセプトロンは線形分離可能なデータ集合に対する学習メカニズムを有したモデルであり，非線形問題には対応してないことに注意する必要がある．

学習データが 2 つのクラス C_1, C_2 から得られるとき，$g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$ とすれば，$g(\boldsymbol{x}) \geq 0$ なら C_1，$g(\boldsymbol{x}) < 0$ なら C_2 と判別すればよい．この場合の学習は，最初の重み係数を適当な値に初期化した上で，学習データ $(\boldsymbol{x}_p, y_p)(p = 1, 2, \cdots, n)$ に対し，識別関数 $g(\boldsymbol{x}_p) = \mathbf{w^T x_p}$ によって識別を行う．

誤識別が起きたときは，

$$\begin{cases} \mathbf{w}' = \mathbf{w} + \epsilon \mathbf{x}, (\mathbf{C_1} \text{を } \mathbf{C_2} \text{と誤った場合}) \\ \mathbf{w}' = \mathbf{w} - \epsilon \mathbf{x}, (\mathbf{C_2} \text{を } \mathbf{C_1} \text{と誤った場合}) \end{cases} \tag{8}$$

と更新すればよい．ただし，ϵ を正の定数とする．この更新式を用いて，すべてのデータについて正しく識別できるようになるまで繰り返す．

課題 4 線形分離可能な魚 **A** と魚 **B** を分類するパーセプトロンを作成せよ．

課題 5 作成したパーセプトロンを非線形問題である魚データに適用し，うまくいかないことを確認せよ．