機械学習 課題1

高林秀
三宅研究室 博士前期課程１年
V-CampusID：23vr008n

June 11, 2023

Abstract

本稿は本年度必修授業の機械学習の第１回レポートの答案用紙である。
本稿は、第１回授業〜第４回授業までの範囲を対象とし、各回で課された課題に対する解答を記載する。
答案の問題番号は各章のタイトルに記載している。
各問に対する解答は本稿に、コードなどの実行結果は別途 GoogleColaboratory のノートブックに記載した巻末の付録から参照できる。

1 第１回授業： 4/11 宿題１

1.1 問題文

二つのデータ点の近さを測る定量的指標の一つとして距離がある。いま、次の三つのデータ点があるとする

$$\text{データ点 } 1 : (5, 2, 5.8) \tag{1}$$
$$\text{データ点 } 2 : (7, 10, 1, 12) \tag{2}$$
$$\text{データ点 } 3 : (3, 2, 6, 3) \tag{3}$$

上の各データ点は４種類の計測値で与えられている（例えば「緯度、経度、水深、温度」のような感じ）。このデータ点同士の間の近さをユークリッド距離で計算し、互いの距離が近いペアの順番を答えよ。データが近いほど距離が小さいことに注意。

1.2 解答

データ点の近さは以下の順番である。

1. データ点 1 とデータ点 3、距離：5.4
2. データ点 1 とデータ点 2、距離：10.0
3. データ点 2 とデータ点 3、距離：13.6

ユークリッド距離は以下の式で計算できる。

$$d(x,y): データ点 x とデータ点 y の距離として、$$

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

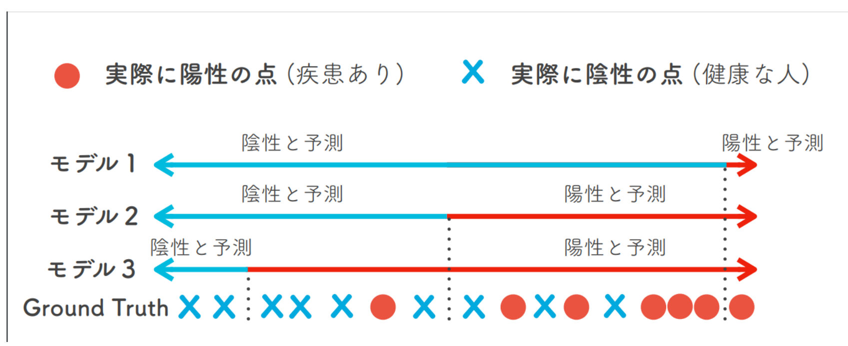
※ x_i, y_iはそれぞれデータ点 x, y の i 番目の要素を表す

各点間の距離は以下のように計算できる

$$d(1,3) = \sqrt{(5-3)^2 + (2-2)^2 + (5.8-6)^2 + (0-3)^2} = 5.4$$
$$d(1,2) = \sqrt{(5-7)^2 + (2-10)^2 + (5.8-1)^2 + (0-12)^2} = 10.0$$
$$d(2,3) = \sqrt{(7-3)^2 + (10-2)^2 + (1-6)^2 + (12-3)^2} = 13.6$$

2　第 2 回授業： 4/18 宿題 2

2.1　問題文



1. 上の三つの分類モデルそれぞれに対し、混同行列、Precision、Recall を計算しなさい。

2. モデル 1 →モデル 2 →モデル 3 の順に、Precision は増加や減少の傾向にあるでしょうか？その傾向はなぜ生じるでしょう？また、Recall についても傾向はどうなっていますか？Precision の傾向と対比して議論して見てください。

2.2　解答: 問 1

まず、各モデルの混同行列と Precision、Recall を計算した結果は以下のようになる。

2.2.1 モデル１

		予測	
		陽性	陰性
実データ	陽性	1	6
	陰性	0	9

Table 1: モデル 1 の混同行列

- Precision： $\frac{TP}{TP+FP} = \frac{1}{1+0} = 1.0$
- Recall： $\frac{TP}{TP+FN} = \frac{1}{1+6} = \frac{1}{7} \approx 0.14$

2.2.2 モデル 2

		予測	
		陽性	陰性
実データ	陽性	6	1
	陰性	3	6

Table 2: モデル 2 の混同行列

- Precision： $\frac{6}{6+3} = \frac{2}{3} \approx 0.67$
- Recall： $\frac{6}{6+1} = \frac{6}{7} \approx 0.86$

2.2.3 モデル 3

		予測	
		陽性	陰性
実データ	陽性	7	0
	陰性	7	2

Table 3: モデル 3 の混同行列

- Precision： $\frac{7}{7+7} = \frac{1}{2} = 0.5$
- Recall： $\frac{7}{7+0} = 1.0$

2.3　解答: 問 2

計算結果から、Precision（適合率）は減少の傾向にあるといえる。Precision は、陽性と予測したデータ数のうち、実際に陽性だったデータ数の割合を示す。計算式で示すと以下の通り。

$$Precision = \frac{TP}{TP + FP}$$

※ TP は真陽性、FP は偽陽性を表す

この式より、Precision は分母の、モデルが陽性と予測した数が少なければ値が大きくなる傾向にあることがわかる。モデル 1 では陽性と予測されたデータ数は 1 で、モデル 2 では 9、モデル 3 では 14 と徐々に陽性予測のデータ数が増えており、Precision の式の分母が大きくなっているので、モデル 1〜モデル 3 にかけて Precision は減少していると考えられる。

また、Recall（再現率）は増加の傾向にあるといえる。Recall は、実際に陽性だったデータのうち、モデルが正しく陽性と予測したデータ数の割合を示す。計算式で示すと以下の通り。

$$Recall = \frac{TP}{TP + FN}$$

※ TP は真陽性、FN は偽陰性を表す

この式より、Recall は分母の、実際は陽性だが予測では陰性だった割合が少ないほど Recall は大きくなる傾向にあることがわかる。実際は陽性だったデータのうち、モデルが陰性と予測したデータの割合がモデル 1〜モデル 3 にかけて減少したことから、Recall は増加していると考えられる。

3　第 3 回授業： 4/25 宿題 1

3.1　問題文

X の数値スケールが大きく (例えば 0〜1000000)、Y のスケールが小さい (例えば 0〜1) データに対して、スケーラを使わないで回帰をしたとする。

1. 勾配降下法で学習率を大きくしても小さくしても問題が起こることを説明せよ。
2. パラメータ a, b ごとに学習率を変えて勾配降下法をすれば、スケーラを使わなくても 1. の問題は抑制できるか？具体的な誤差関数の形を使って議論せよ。

3.2　解答: 問 1

特徴量の間のスケールの差が大きい場合、勾配降下法実行時に行われるパラメータの更新幅が、パラメータによって偏りが生じるという現象が生じる。つまり、パラメータのスケールが異なる場合、学習率の大小に関係なく、勾配の進み幅（パラメータの更新幅）がバラバラになる。その結果、最適解に収束しないことがあ

るので、スケーラーを使わない場合よりも、学習を完了するまで時間が長くなったり、モデルの精度が悪くなるという問題がある。
この問題はなぜ生じるのか、それはパラメータの更新幅が、特徴量の大きさに依存するからである。勾配降下法では、パラメータの更新幅は以下の式で表される。

$$\theta_{new} = \theta_{old} - \eta \frac{\partial E(\theta)}{\partial \theta}$$

※ $E(\theta)$ は誤差関数、ηは学習率を表す

この式より、パラメータの更新幅は、学習率と誤差関数の勾配の積に比例することがわかる。ゆえに、誤差関数の勾配は、特徴量の大きさに依存し、特徴量の大きさ・スケールが異なる場合、誤差関数の勾配も異なる。そのため、特徴量のスケールが異なる場合、パラメータの更新幅も異なるため、最適解に収束しないことがある。

3.3 解答: 問２

結論を先に考えれば、スケーラーを使わなくても、パラメータ毎に学習率を設定できればこの問題は抑止できるはずである。具体的な誤差関数として平均 2 乗誤差を用いた場合、

$$\hat{y}_i = ax_i + b$$

$$MSE(a, b) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

これを等高線プロットした図を以下の URL から引用する。

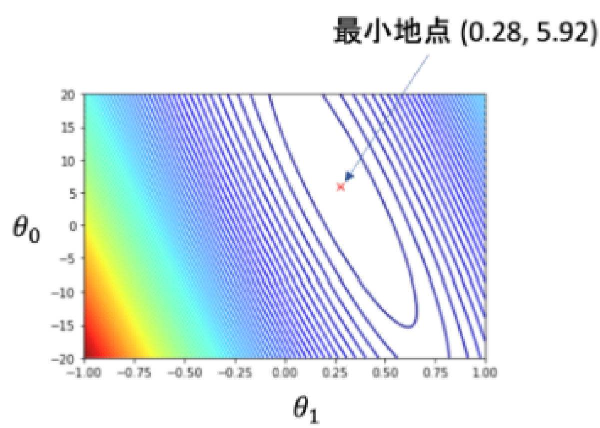


Figure 1: MSE の等高線プロット, $\theta_0 = b, \theta_1 = a$ と読み替えてください.
【引用元】: 米国データサイエンティストのブログ〜損失関数の可視化 2022.02.16
https://datawokagaku.com/cost_function/

この図からわかるように、パラメータ a と b の方向に対し、それぞれ異なる勾配を持っていることが分かる。学習率とは、勾配をどれだけの幅で更新するかを決めるハイパーパラメータである。ゆえに、パラメータ毎に学習率を設定することができれば、パラメータ毎に適切な勾配の進み幅、すなわちパラメータの更新幅を制御できるはずなので、スケーラーを用いずともこの問題は抑止できると考えられる。

現にそのような観点から最適化を図る手法として、Adam などの適応学習率と呼ばれる、勾配の大きさに従って適切な学習率を設定できる手法も提案されている。

4 第 4 回授業：5/9 宿題 1

4.1 問題文

R^2 による score 計算を Numpy を用いて実装して見ましょう（sklearn の出来合いの関数を使ってはいけません）。R2 の定義式は講義のスライド参照。

4.2 解答

授業資料のノートブックに以下の関数を作成し、テストコードを実行した。そのコードと結果を以下に示す。

```
from numpy import ndarray
import numpy as np

def r_square(y_hat: ndarray, y: ndarray):

  avg_y = (1/y.size) * y.sum()
  error_sq = np.square(y_hat - y)

  residual = error_sq.sum()
  totalSquares = (np.square(avg_y - y)).sum()

  return 1 - (residual / totalSquares)


# 実際の値と予測値の生成
y = np.array([1, 2, 3, 4, 5])
y_hat = np.array([1.1, 1.9, 3.2, 3.7, 5.1])

# $R^2スコアの計算$
r2_score = r_square(y_hat, y)

print(f"R^2 Score: {r2_score}")
#R^2 Score: 0.984
```

5 付録

各課題のコード (notebook 形式) は、下記リンクの GoogleDrive フォルダに格納している。ノートブックのファイル名は **23vr008n-Report1.ipynb**。

- 提出用 GoogleDrive フォルダ
  - `https://drive.google.com/drive/folders/1DZd7YFLM37_FVvvhxdTS5-6y9YXI4ulB?usp=drive_link`