機械学習入門

第6回～10回　教師あり学習/ロジス
ティック回帰

ロジスティック回帰1

- 回帰とは言っているが分類に使われる。
- ある確率(閾値)以上を1、ある確率以下を0として2値分類に適用(多クラス分類はポワソン回帰というものがある)
- Ex)テストのデータから合格か不合格かを分類する、腫瘍データから悪性か良性かを分類する。

ロジスティック回帰2

線形回帰式：y=ax+bを用いて確率を予測する.
しかし、ここで注意！この式はすべての値を取りうる！
一方確率は、0~1までの間の値．どうする．．．
ちょいとここで確率について考えてみる.

オッズについて

確率を次のように考えてみよう.

$$\mathrm{p} \leftrightarrow \frac{p}{1-p}$$

これが一対一の変換であることはわかるだろうか?
つまり、確率としての情報を保持したままである.
しかし右のように表すと何が嬉しいかというと、値域が0~∞に拡張できる！

この $\frac{p}{1-p}$ を**オッズ**という．確率と思ってくれてほぼ間違いはない.

ロジット変換について

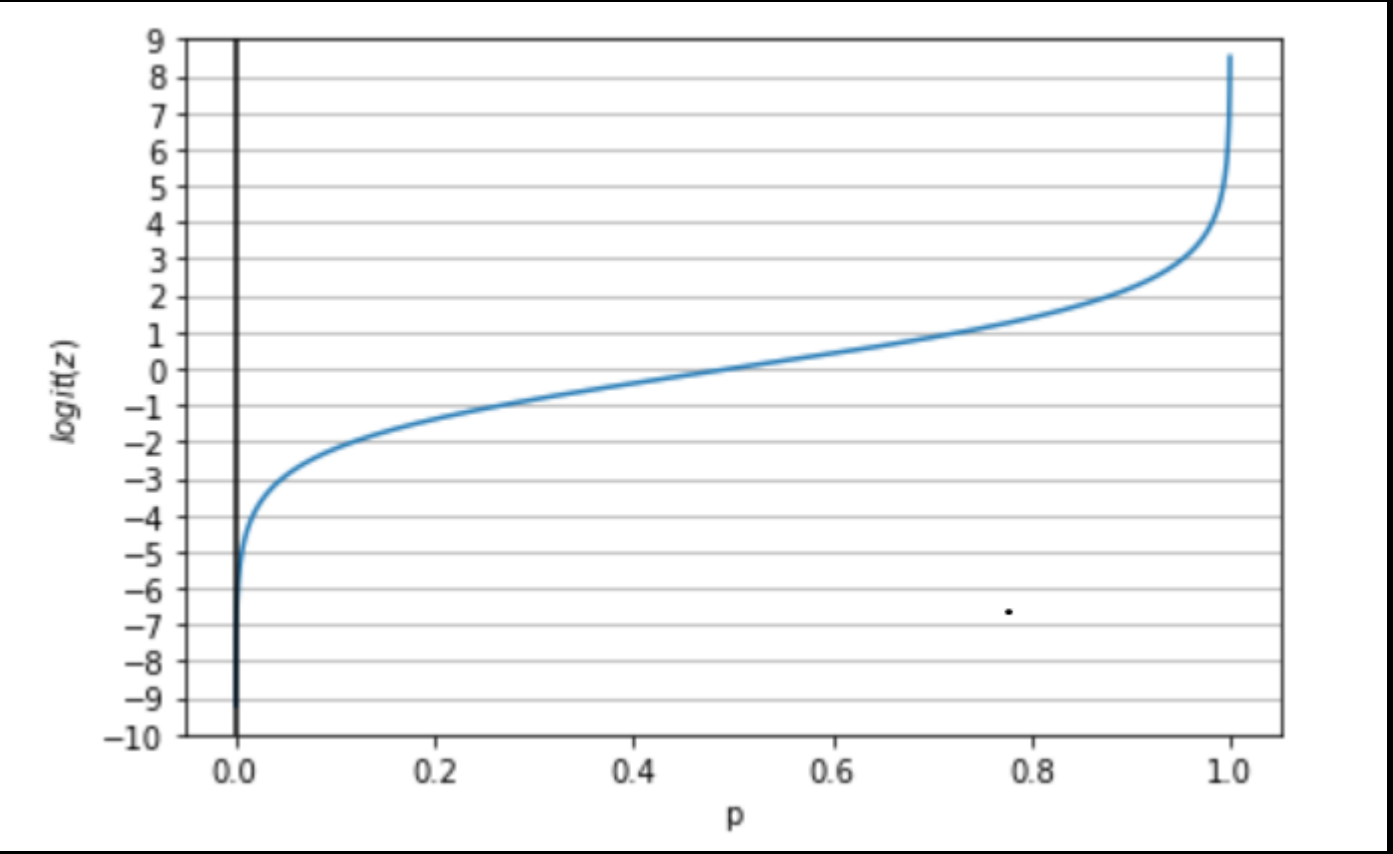
ここからは少しテクニカル．．．

先のオッズ $\frac{p}{1-p}$に対して、対数を取る．

$$z = \frac{p}{1-p} \Leftrightarrow \ln z = \ln\left(\frac{p}{1-p}\right)$$

となって、これは対数のグラフを考えて貰えばわかるが、$-\infty$~∞までの値を取る．確率pに対する右辺の変換をロジット変換という．

ロジット関数のグラフ



逆ロジット変換

$\ln\left(\frac{p}{1-p}\right)$とY＝aX+bとの対応を考えて，

$$\ln\left(\frac{p}{1-p}\right) = aX + b$$

と，ロジットを予測するようなものを考えれば実質的に確率を予測できるのではないか？と考えるのがロジスティック回帰の考え方.

学習によって，a,bを求め、a,bを使って任意のXを入力として与えれば，確率が出力として返ってくる，という話.

Sub 0

$$\ln\left(\frac{p}{1-p}\right) = aX + b$$

この式をp=なんとか、の式に書き直してみよう.

ans.

$$\ln\left(\frac{p}{1-p}\right) = aX + b$$

$$\frac{p}{1-p} = e^{aX+b}$$

$$\frac{1}{p} - 1 = \frac{1}{e^{aX+b}}$$

$$\frac{1}{p} = \frac{1 + e^{aX+b}}{e^{aX+b}}$$

$$p = \frac{e^{aX+b}}{1 + e^{aX+b}}$$

$$p = \frac{1}{1 + e^{-(aX+b)}}$$

ちなみに。

- 目的変数が0or1なので最小二乗法は使えない。
- 使うのは最尤法という方法。尤度関数を最急降下法で最大化する
- 簡単に言うと「尤もらしさ」(=確率)が一番大きくなるようなパラメタを決めていく。
- 詳しくは後期に乞うご期待.

ロジスティック回帰

要するに。

普通に確率を予測する問題に対して、

重回帰モデルで予測した出力値を逆ロジット変換をして

確率を得る。

＝**本質的にやっていることは重線形回帰モデルを作ることと一緒**！

ただし、結果の解釈が少し異なる(後述)

オッズ比

最尤法を用いて，それぞれの係数を求めた.

$$\frac{p_0}{1-p_0} = e^{a_1x_1+a_2x_2+\cdots+a_nx_n+b}$$

このとき、各説明変数が 1 単位増えたときオッズはどう変わるだろうか？

オッズ比

先の式の説明変数を 1 単位増やしてみる

$$\frac{p_1}{1-p_1} = e^{a_1(x_1+1)+a_2x_2+\cdots+a_nx_n+b}$$

このままではわかりずらいので次のように比を取る.

$$Odds_{ratio} = \frac{\frac{p_1}{1-p_1}}{\frac{p_0}{1-p_0}}$$

Sub0-2

$$Odds_{ratio} = \frac{\frac{p_1}{1-p_1}}{\frac{p_0}{1-p_0}} = \frac{e^{a_1(x_1+1)+a_2x_2+\cdots+a_nx_n+b}}{e^{a_1x_1+a_2x_2+\cdots+a_nx_n+b}}$$

を簡単な式に直せ.

ans.

$$Odds_{ratio} = \frac{\frac{p_1}{1-p_1}}{\frac{p_0}{1-p_0}} = \frac{e^{a_1(x_1+1)+a_2x_2+\cdots+a_nx_n+b}}{e^{a_1x_1+a_2x_2+\cdots+a_nx_n+b}}$$

$$= e^{a_1}$$

オッズ比

$$Odds_{ratio} = \frac{\frac{p_1}{1-p_1}}{\frac{p_0}{1-p_0}} = \frac{e^{a_1(x_1+1)+a_2x_2+\cdots+a_nx_n+b}}{e^{a_1x_1+a_2x_2+\cdots+a_nx_n+b}}$$

$$= e^{a_1}$$

この式の意味は、変化しなかったとき(分母)と変化したとき(分子)の比である.

すなわち、変化しなかったときと比べて変化したときはどれくらいの差がありますか？という値を表している.

今回の例だと、 x_1という説明変数が 1 単位増えるとオッズが e^{a_1}だけ増えるということを意味している

ロジスティック回帰について調べてみる

・疑問・質問・分かったこと

実装してみる

実装

使うデータ: load_cancerより腫瘍のデータセット

使用モデル: logistic回帰モデル

で、ロジスティック回帰.ipynbを参考に実装せよ

正則化LogisticRegressionを実装する

- SklearnではデフォルトでRidge正則化が行われている.
- Lassoもできる、Lasso＋Ridgeもできる.
- そのパラメータ(罰則項を決める値)はCという値.
- このCをいじって結果の精度がどうなるかを見てみる.

C値

- Cは先のαの逆数(λの逆数)に対応する.
- Cを大きくすると正則化の影響が小さくなる．つまり、訓練データにより適合したモデルが得られる.
- Cを小さくすると係数を0に近づけるように働く。つまり、モデルをより単純化しようとする.

数値を眺める

- C=100

訓練：0.9812206572769953

テスト：0.965034965034965

Default(C=1):

訓練： 0.9530516431924883

テスト：0.958041958041958

数値を眺める

- C=0.01

訓練：0.9530516431924883

テスト：0.951048951048951

Default(C=1):

訓練： 0.9530516431924883

テスト：0.958041958041958

Subject2

- Titanicのデータを使ってやってみる
- 欠損値があるデータなので、df.fillna()などで埋める
- カテゴリ変数を含むカラムがあるので、df.map()などで数値に直す

上記の操作が必要になるので、ググりながらやってみること。

HINT

```
drop_df = ["sibsp","parch","fare","embarked","class","who","adult_male","deck","embark_town","alive","alone"]

titanic_data = titanic_data.drop(columns = drop_df, axis = 1)

titanic_data["age"] = titanic_data["age"].fillna(titanic_data["age"].mean())

titanic_data["sex"] = titanic_data["sex"].map({"male": 1, "female":0})
```

回帰まとめ

- 回帰はアルゴリズム的にも最も基本的なものとなっている。
- その結果の解釈性からデータ分析においては非常によく使われる。