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

- 回帰とは言っているが分類に使われる。
- 確率の予測を行う場合に用いられる。
- 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。

確率の予測？？

- 目的変数が0~1の間の数。
- 重回帰モデルの式(目的変数：確率)

$$Y = \beta_0 x_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

右辺は-∞から+∞まで取りうる。

＝確率(0以上1以下の値)を予測できない。

オッズ

- 確率Yが起こる/確率Yが起こらない

$$odds = \frac{p}{1-p}$$

例えば、雨が降る確率＝70%、降らない確率30%とかだと

オッズは70/30=2.33となる。

このオッズを用いると確率がのとうなものが0～+∞までの等

ロジット変換

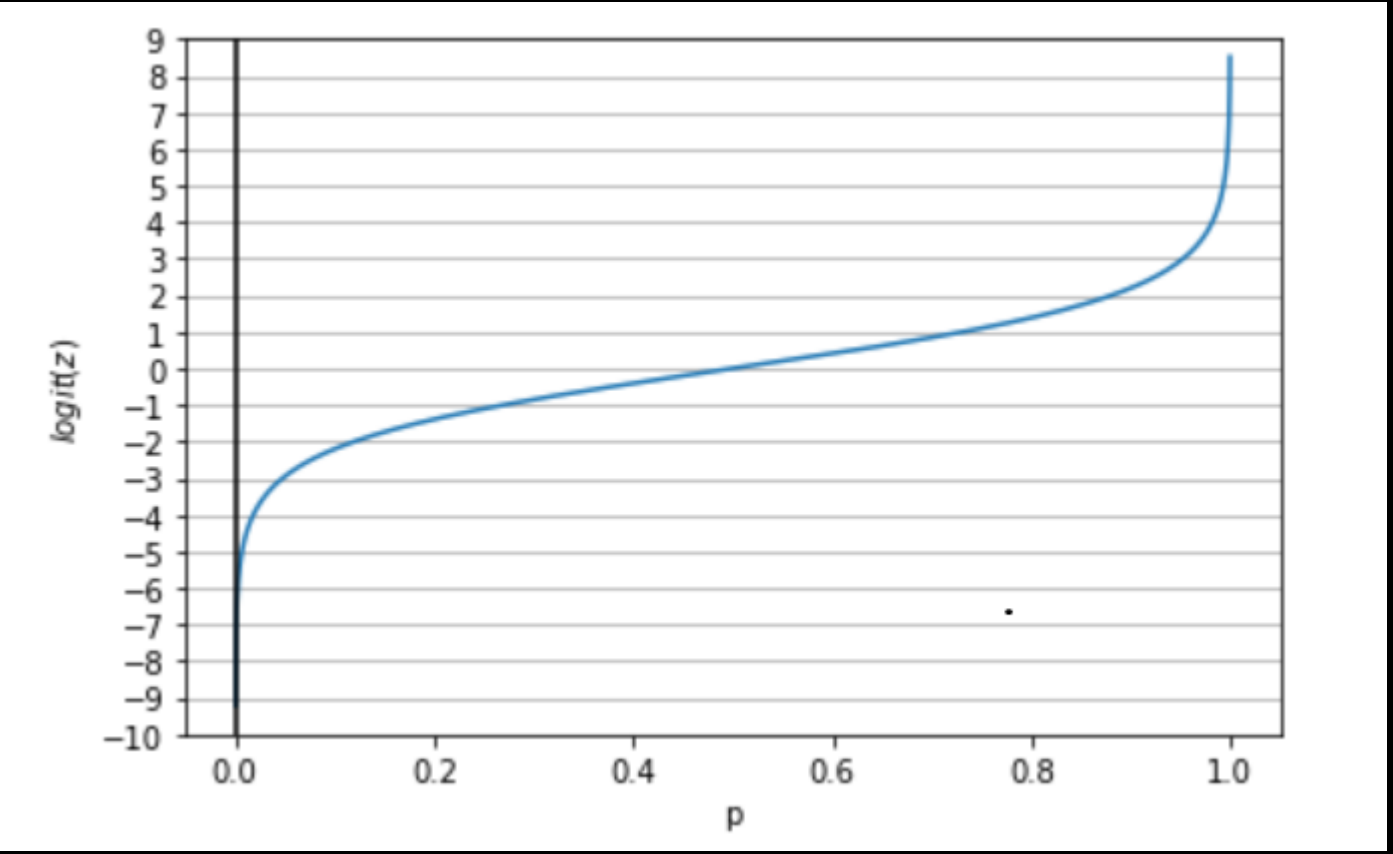
- ０～１の間の目的変数を右辺の変域と一致させる。

オッズを以下の式によって

$$\text{logit} = \ln\left(\frac{p}{1-p}\right)$$

ロジット変換をしてやれば、-∞～+∞までの範囲に拡張され

ロジット関数のグラフ



逆ロジット変換

得られた式

重回帰モデルの式
↓

$$\text{logit} = \ln\left(\frac{p}{1-p}\right) = Y = \beta_0 x_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

最後に得られたロジット値を式変形し、確率を得ることができる。

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」が一番大きくなるようなパラメタを決めていく。
- 詳しく知りたい方は「二項分布」or「ベルヌーイ分布」とかを調べてみると良いかも。

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)

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

過学習(overfitting)と適合不足(underfitting)

- 過学習とは、学習データから得られたモデルが複雑すぎて学習データに過度な一致をしてしまう状態。
- 適合不足とは、学習データから得られたモデルが単純すぎて訓練データにもテストデータにも上手くfitしない状態。

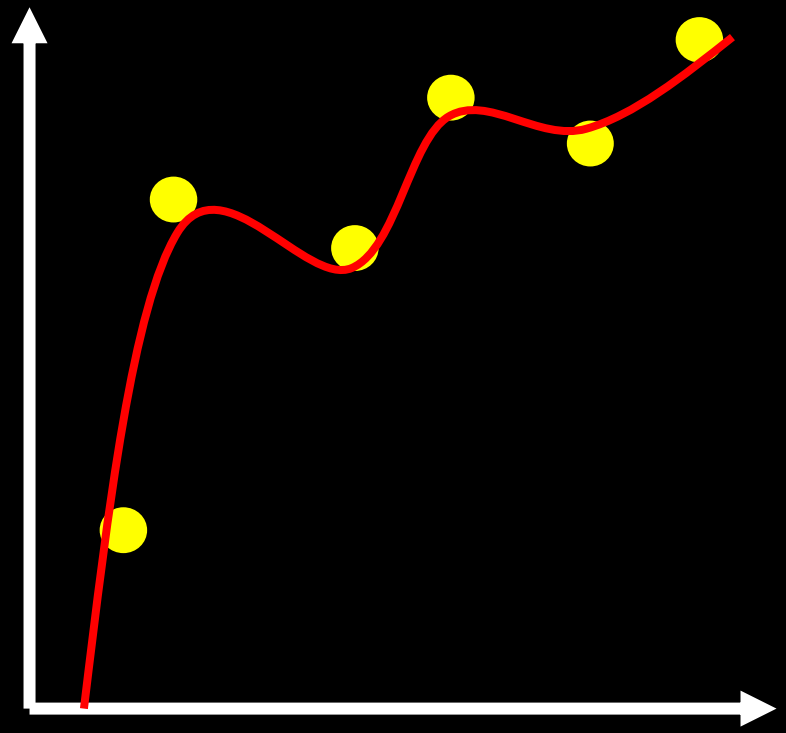
→**防ぐには、学習データの過度なfittingを抑えながらデータ量を増やしたり、モデルを複雑化していく。**

正則化

- 過学習を防ぐ手法の一つ。
- モデルに条件を加えることで、モデルが複雑になりすぎるのを防ぐ。≒モデルを単純化する。
- 線形(単純)で仮定した場合過学習は起こりにくい(適合不足の可能性はある)が、非線形(複雑)の場合、過学習が起こる可能性がある。
- 授業では扱ってないが、非線形回帰モデルのようなもの(回帰モデル式

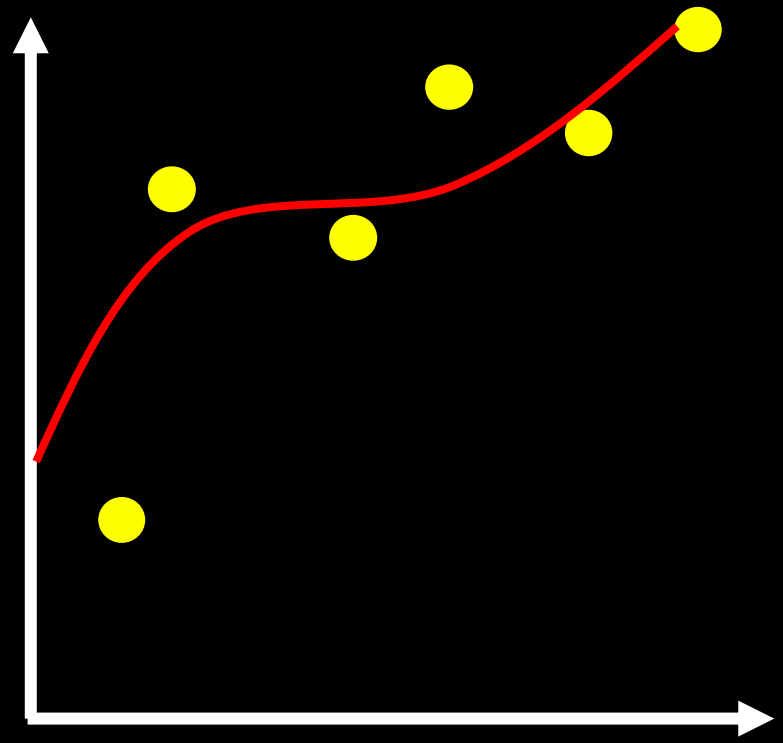
正則化

過学習！



正則化

ちょっとマシ。



正則化

- L1正則化
- L2正則化

の2種類、どちらも最適化したい関数に何らかの罰則項を追加したもの。

≒過剰に最適になりすぎない

→個々の説明変数が、過度に影響しないようにその係数が0に近くなるように調整する。

L1正則化

- いくつかの係数を 0 に出来る≒次元圧縮的な効果。
- 自動的に特徴量を選択している、とも言える。
- 特徴量が減るので、結果の解釈が容易になる。

＝どの特徴量が重要かが分かりやすくなる。

＊罰則項が絶対値であったりと、解析的(微分するなど)に最適化できない

L2正則化

- 係数が0にならない場合がある。
- L1では次元圧縮を行っているため、L2の方が精度が高い傾向がある。

正則化

- どちらが良いか、は目的による。
- 特徴量のうち重要な物がわずかしかない、解釈しやすいモデルがほしいなら、L1正則化を行う。
- 基本的にはL2で試してみるのが良い。
- L1とL2の２つの正則化を用いても良い(罰則項を選択するのにコストがかかる)

正則化LogisticRegressionを実装する

- Sklearnではデフォルトで L2正則化が行われている。
- そのパラメータ(罰則項を決める物)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：0.9553990610328639

テスト：0.9440559440559441

Default(C=1):

訓練： 0.9530516431924883

テスト：0.958041958041958

数値を眺める

- C=0.01

訓練：0.9342723004694836

テスト：0.9300699300699301

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

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})
```