L05 ロジスティック回帰

樋口さぶろお https://hig3.net

龍谷大学 先端理工学部 数理・情報科学課程

多変量解析☆演習 L05(2021-10-28 Thu)

最終更新: Time-stamp: "2021-10-27 Wed 14:45 JST hig"

今日の目標

- 回帰と分類の違いを説明できる
- ロジスティック回帰の統計モデルを説明できる
- np.* で関数値を, matplotlib で式からグラフを得られる
- statsmodels.glm でロジスティック回帰を実行できる

ここまで来たよ

予測問題の回帰タイプと分類タイプ

線形回帰の学習データ

	説明変数 x	目的変数 Y
1	10.3	21.1
2	12.1	22.9
$\vdots$		
n	19.8	41.2

機械学習の言葉では

教師あり, 回帰＝目的変数が連続値

Y には小さい (?) 誤差を含む.

ロジスティック回帰の学習データ

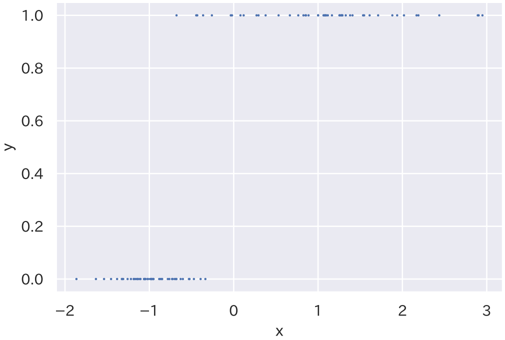
	説明変数 x	目的変数 Y	ダミー変数
1	10.3	A	0
2	12.1	B	1
$\vdots$			
n	19.8	A	0

機械学習の言葉では

教師あり, 分類＝目的変数がカテゴリ変数 $\simeq$ 離散

ロジスティック回帰の学習データ

X 連続値
Y 間隔に意味のない有限個 $\simeq$ 離散型, 今日は A$\rightarrow$0, B$\rightarrow$1 に限定)



- 予測
  - 回帰 連続値変数
    - 線形回帰 単回帰
    - 線形回帰 重回帰
  - 分類 カテゴリ変数
    - **ロジスティック回帰 単回帰**
    - ロジスティック回帰 重回帰

ここまで来たよ

3 線形回帰モデル - 単回帰

5 ロジスティック回帰
- 分類問題
- ロジスティック回帰の統計モデル
- Python で関数値を求める・グラフを描く
- ロジスティック回帰の実行

ロジスティック回帰

ロジスティック回帰

データ (x_i, y_i) $(i = 1, \ldots, n)$ x は連続値, $y =$ A xor B から

$$P(Y = \mathrm{B}) = \frac{\mathrm{e}^{\,{}^{\mathrm{t}}\boldsymbol{\beta}\cdot\boldsymbol{x}}}{1 + \mathrm{e}^{\,{}^{\mathrm{t}}\boldsymbol{\beta}\cdot\boldsymbol{x}}}, \quad P(Y = \mathrm{A}) = 1 - P(Y = \mathrm{B}),$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \end{bmatrix}, \; {}^{\mathrm{t}}\boldsymbol{\beta} \cdot \boldsymbol{x} = \beta_0 + \beta_1 \cdot x_1$$

という説明がもっともになるように, β_0, β_1 を推定すること．それを使って, $\boldsymbol{x}$ が与えられたときの $P(Y = \mathrm{B})$ や Y を推定すること．

回帰っていうけど分類じゃん? by 機械学習の人

$\beta_0 + \beta_1 \cdot x_1$ ってあたりが回帰? 0 と 1 の間の連続値 (確率) だから回帰? by 統計学の人

$\beta_0 + \beta_1 \cdot x_1 + \epsilon$, $\epsilon \sim \mathrm{N}(0, \sigma^2)$ ではない．推定するもの (左辺) がすでに確率

ロジスティック回帰はもっとも 1:1 次関数 対 ロジスティック関数

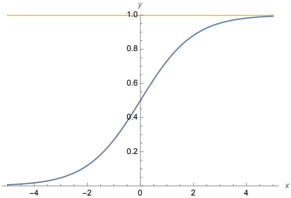
$$p = P(Y=1) = \beta_0 + \beta_1 \cdot x_1$$

じゃだめなの?

$\rightsquigarrow p = \beta_0 + \beta_1 x$ は,
$-\infty < x < +\infty$ のとき $-\infty < y < +\infty$.
確率のとる範囲 $0 \leq p \leq 1$ に収まらない…
$x \in (-\infty, \infty)$ を $y \in (0,1)$ に写す写像ある?

一例 (標準) シグモイド関数. sigmoid function $y = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$

x	$-\infty$		0		$+\infty$
y'	$+$	$+$	$\frac{1}{4}$	$+$	$+$
y	0	$\nearrow$	$\frac{1}{2}$	$\nearrow$	$+1$



標準シグモイド関数を移動拡大縮小したもの ロジスティック関数 (logistic function)

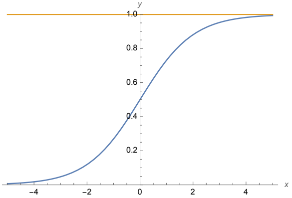
$\beta_1 > 0$ とする.

x_1	$-\infty$		$-\beta_0/\beta_1$		$+\infty$
$\frac{\mathrm{d}p}{\mathrm{d}x_1}$	$+$	$+$	$\frac{\beta_1}{4}$	$+$	$+$
$p = \frac{\mathrm{e}^{{}^t\boldsymbol{\beta}\cdot\boldsymbol{x}}}{1+\mathrm{e}^{{}^t\boldsymbol{\beta}\cdot\boldsymbol{x}}}$	0	$\nearrow$	$\frac{1}{2}$	$\nearrow$	$+1$

$\beta_1(x - (-\frac{\beta_0}{\beta_1}))$

β_1 横拡大縮小

β_0 横移動



ロジスティック回帰はもっとも 2:母平均値 対 母比率

	説明変数なし	説明変数あり
母平均値	$Y \sim \mathrm{N}(\mu, \sigma^2)$ の母平均値 μ の推定 岩薩林 確率・統計 §7.1 $\overline{Y} = \frac{1}{n}(Y_1 + \cdots)$	線形回帰 $Y = \beta_0 + \beta_1 x + \epsilon$ β_j:連立 1 次方程式を解いて
カテゴリ変数の確率	$Y \sim \mathrm{B}(1, p)$ の母比率 p の推定 岩薩林 確率・統計 §7.3 $\hat{p} = \frac{k}{n}$	ロジスティック回帰 $P(Y=1) = \frac{\mathrm{e}^{\,{}^t\boldsymbol{\beta}\cdot\boldsymbol{x}}}{1+\mathrm{e}^{\,{}^t\boldsymbol{\beta}\cdot\boldsymbol{x}}}$ β_j:ニュートン法で

ここまで来たよ

3 線形回帰モデル - 単回帰

5 ロジスティック回帰
- 分類問題
- ロジスティック回帰の統計モデル
- Python で関数値を求める・グラフを描く
- ロジスティック回帰の実行

Python で関数値を求める・グラフを描く

- 数学関数 標準以外に, math, numpy, scipy などに含まれる
  `np.exp(x)`
- リスト: `[-0.5,0,0.5]`
  - https://paiza.jp/works/python3/primer/beginner-python4
- 多くの関数は, リストの各要素に適用される: np.exp([-0.5,0,0.5])
- ユーザによる関数定義

```
def 関数名(引数1,引数2):
    x=1+1
    # 中略
    return 返り値
```

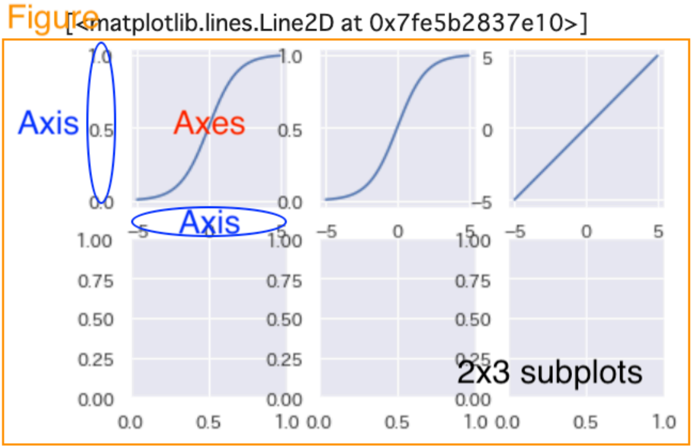
  ブロックは中括弧の代わりにインデントで. 文末に; は不要.
- matplotlib の plot: `plot(横軸の値のリスト, 縦軸の値のリスト)`
  - リストの長さは同じである必要
  - i 番目同士を組にしてデータ点とみなす

Matplotlib でのグラフ描画

[18]

```
import matplotlib.pyplot as plt #低レベルグラフ描画
fig, ax=plt.subplots(2,3)
ax[0][0].plot(x,y1)
ax[0][1].plot(x,y1)
ax[0][2].plot(x,x)
```

[<matplotlib.lines.Line2D at 0x7fe5b2837e10>]



ここまで来たよ

3. 線形回帰モデル - 単回帰

5. ロジスティック回帰
   - 分類問題
   - ロジスティック回帰の統計モデル
   - Python で関数値を求める・グラフを描く
   - ロジスティック回帰の実行

ロジスティック回帰の回帰係数の推定方法

尤度 (ゆうど)(2 変数関数) 確率統計 I(2021)L12

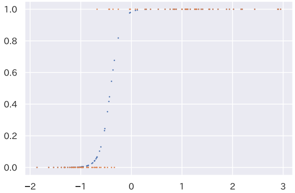
$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} p(y_i | x_i)$$

x_i で測定したときに, 測定値 y_i が得られる同時確率 (を, 回帰係数 β_0, β_1 の関数としてみたもの).

$L(\beta_0, \beta_1)$ を変数 β_0, β_1 について最大化. 微積分 II

$\rightsquigarrow \frac{\partial L}{\partial \beta_0}(\beta_0, \beta_1) = \frac{\partial L}{\partial \beta_1}(\beta_0, \beta_1) = 0$

$\rightsquigarrow$ ニュートン法 (反復法 iteration) で数値的に解く 数値計算法



statsmodels.glm

```
import statsmodels.api.formula as smf
```

glm in statsmodels

```
[ ]
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

```
[121]
result_g=smf.glm(formula='y~x',data=dfb, family=sm.families.Binomial(sm.families.links.logit())).fit()
```

```
[122]
result_g.summary()
```

Generalized Linear Model Regression Results

モデル

Dep. Variable:	y	**No. Observations:**	100
Model:	GLM	**Df Residuals:**	98
Model Family:	Binomial	**Df Model:**	1
Link Function:	logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-8.7065
Date:	Fri, 22 Oct 2021	**Deviance:**	17.413
Time:	08:45:09	**Pearson chi2:**	22.8
No. Iterations:	10		
Covariance Type:	nonrobust		

あてはまり

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	4.0625	1.843	2.204	0.028	0.450	7.675
x	9.9026	3.610	2.743	0.006	2.828	16.977

回帰係数