レポート：ロジスティック回帰のプログラムソースを開発せよ。（C,C++,JAVAなど) （締切６月２４日）

- $y_i = \frac{1}{1+\exp(-ax_i-b)} + \varepsilon,$
- $\varepsilon \sim N(0,\sigma^2)$
- 入力 (x,y)データファイル の読み込み
- パラメータ a, b の推定値
- ニュートン ラフソン法と最急降下法のプログラムをひとつづつ作成せよ。
- 提出物
- プログラムの ソース
- 次ページのレポート

乱数データ発生プログラムと解析結果

- $x_i \sim N(0, 1^2)$

でデータ数 20. 50, 100個 それぞれ発生する。
それぞれのデータx_iについて $a = 0.8, b = -0.3$と設定する。

- $y_i = \frac{1}{1+\exp(-ax_i-b)} + \varepsilon,$
- $\varepsilon \sim N(0, 0.1^2)$
- により、データ数 20. 50, 100個 それぞれ発生する。
- 結果、(x, y)の組み合わせが20個、50個、100個の３種類のデータがそろう。
- 20個、50個、100個の３種類のデータに対して
- ニュートンラフソン法のプログラム、最急降下法のプログラムで

(a, b)の推定値と真値の誤差、フィッシャー情報量を用いた漸近誤差、推定平均時間、を求め、 レポートで解析せよ。

次のロジスティック回帰の尤度方程式を求めよ.

- $y_i = \frac{1}{1+\exp(-ax_i-b)} + \varepsilon, \varepsilon \sim N(0, \sigma^2)$
- 尤度関数Lは, $L(a, b, \sigma^2|D) = \prod_{i=1}^{n} f(y_i|a, b, \sigma^2)$

$$L(a,b,\sigma^2|D) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)^2}{2\sigma^2}\right]$$

- 対数尤度lは, $l(a, b, \sigma^2|D) = \ln\left(L(a, b, \sigma^2|D)\right)$

$$l(a,b,\sigma^2|D) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)^2$$

- $l(a, b, \sigma^2|D)$をa,bそれぞれについて偏微分すると,
- $\frac{\partial l}{\partial a} = \frac{\partial}{\partial a}\left[\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)^2\right]$

$$= \frac{1}{2\sigma^2}\sum_{i=1}^{n}\frac{\partial}{\partial a}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)^2$$

$$= \frac{1}{2\sigma^2}\sum_{i=1}^{n} 2\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)\frac{\partial}{\partial a}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)$$

$$= \sum_{i=1}^{n}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)\left(\frac{x_i\exp(-ax_i-b)}{(1+\exp(-ax_i-b))^2}\right).$$

- $\frac{\partial l}{\partial \mathrm{b}} = \frac{\partial}{\partial \mathrm{b}}\left[\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)^2\right]$

$$= \frac{1}{2\sigma^2}\sum_{i=1}^{n}\frac{\partial}{\partial \mathrm{b}}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)^2$$

$$= \frac{1}{2\sigma^2}\sum_{i=1}^{n} 2\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)\frac{\partial}{\partial \mathrm{b}}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)$$

$$= \sum_{i=1}^{n}\left(y_i - \frac{1}{1+\exp(-ax_i-b)}\right)\left(\frac{\exp(-ax_i-b)}{(1+\exp(-ax_i-b))^2}\right).$$

- a,bの最尤推定値を求めるために, ニュートン法を用いる.

ニュートン法（多次元）

- $x = [x_1, x_2, \cdots, x_n]^T$ のとき, ヘッセ行列Hを用いて, 以下のように表せる.

$$x_{n+1} = x_n - H^{-1}\nabla f(x_n)$$

ここで, $H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}, \nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$

- 2変数の時は以下のように表せる.

$$H^{-1} = \frac{1}{H_{11}H_{22} - H_{12}H_{21}}\begin{pmatrix} H_{22} & -H_{12} \\ -H_{21} & H_{11} \end{pmatrix}$$

$$H_{11} = \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_1}, H_{12} = \frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2},$$
$$H_{11} = \frac{\partial^2 f(x_1, x_2)}{\partial x_2 \partial x_1}, H_{12} = \frac{\partial^2 f(x_1, x_2)}{\partial x_2 \partial x_2}$$

ニュートン法（1次元）

- 関数f（x）がf（x）＝0となるxを近似的に求める方法である.
- 例として, 図のような関数f(x)を考え, f（x）＝0となるxを求める. 点x=aの周りでテーラー展開すると,

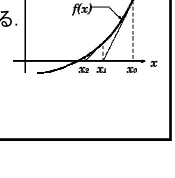
$$f(x) = f(x_0) + f^{(1)}(x_0)(x - x_0) + \frac{f^{(2)}(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$$

- 1次の項までを用いて, f(x)は以下のように近似できる.

$$f(x) = f(x_0) + f^{(1)}(x_0)(x - x_0)$$

- f（x）＝0とすると, 以下のようにxを求めることが出来る.

$$x_{n+1} \approx x_n - \frac{f(x_n)}{f^{(1)}(x_n)}$$



最急降下法

- ニュートン法の特殊解として最急降下法があり下記のように表せる.

$$x_{n+1} = x_n - \alpha^{-1} \nabla f(x_n) \qquad (\alpha > 0)$$

  - （注）ニュートン法はヘッセ行列をHとして以下のように表せた.

$$x_{n+1} = x_n - H^{-1} \nabla f(x_n)$$

- αと初期値を設定し，繰り返し計算することにより極小値を求める.

最急降下法を用いたロジスティック回帰の最尤推定

- 今回の問題では，以下のように表せる.

$$\begin{pmatrix} a_{k+1} \\ b_{k+1} \end{pmatrix} = \begin{pmatrix} a_k \\ b_k \end{pmatrix} - \alpha \begin{pmatrix} \dfrac{\partial l}{\partial a_k} \\ \dfrac{\partial l}{\partial b_k} \end{pmatrix}$$

$$\frac{\partial l}{\partial a} = \sum_{i=1}^{n} \left(y_i - \frac{1}{1 + \exp(-ax_i - b)} \right) \left(\frac{x_i \exp(-ax_i - b)}{(1 + \exp(-ax_i - b))^2} \right)$$

$$\frac{\partial l}{\partial b} = \sum_{i=1}^{n} \left(y_i - \frac{1}{1 + \exp(-ax_i - b)} \right) \left(\frac{\exp(-ax_i - b)}{(1 + \exp(-ax_i - b))^2} \right)$$

- αを任意に設定し，a,bの更新値を交互に上の式にいれて、それらの値が収束するまで繰り返し計算する。
- 収束条件は, $|a_{k+1} - a_k| \le \varepsilon_1, |b_{k+1} - b_k| \le \varepsilon_2$ などが使われる. 10^{-3}や10^{-5}が多いが、収束しない場合は条件を緩くする。
- a,bの初期値は，ロジスティック回帰などの凸関数の場合，(a,b)=(0,0)とすることが多い（精度への影響はない）.