

微分、偏微分、数値微分等に関して

前川眞一

20210603,10, 2021. 6. 10

1 微分、偏微分、数値微分

1.1 定義等

スカラー値を持つスカラー量 x の関数

$$y = f(x) \quad (1)$$

の x に関する微分 (derivative) を差分商 (difference quotient)

$$q(x) = \frac{f(x+h) - f(x)}{h} \quad (2)$$

の極限

$$f'(x) = \frac{dy}{dx} = \frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (3)$$

と定義される。関数 $f(x)$ が x で微分可能とは、 h が上方ならびに下方から 0 に近づく極限が存在し、それらが一致する場合である。

$f(\mathbf{x})$ が $p \times 1$ のベクトル \mathbf{x} の関数である場合、 $f(\mathbf{x})$ の x_j に関する偏微分 (partial derivative) を

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_{j-1}, x_j + h, x_{j+1}, \dots, x_p) - f(\mathbf{x})}{h} \quad (4)$$

で定義する。また、これをまとめて $p \times 1$ のベクトルの形に書いたものを

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_j} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_p} \end{pmatrix} \quad (5)$$

と記す。

1.2 Gradient, Hessian, Jacobian

$r(\boldsymbol{\xi})$ を $p \times 1$ のパラメタベクトル $\boldsymbol{\xi}$

$$\boldsymbol{\xi} = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_j \\ \vdots \\ \xi_p \end{pmatrix} \quad (6)$$

を持つスカラー値の関数とする。 r の $\boldsymbol{\xi}$ の各要素 ξ_j に関する一次微分を $\boldsymbol{\xi}$ と同じ $p \times 1$ の形に並べたものを r の gradient vector (勾配ベクトル) と呼び、 $\nabla r(\boldsymbol{\xi})$ または $\frac{\partial r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}$ と記す。

$$\nabla r(\boldsymbol{\xi}) = \frac{\partial r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \begin{pmatrix} \frac{\partial r(\boldsymbol{\xi})}{\partial \xi_1} \\ \frac{\partial r(\boldsymbol{\xi})}{\partial \xi_2} \\ \vdots \\ \frac{\partial r(\boldsymbol{\xi})}{\partial \xi_j} \\ \vdots \\ \frac{\partial r(\boldsymbol{\xi})}{\partial \xi_p} \end{pmatrix} \quad (7)$$

また、 $\frac{\partial r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}$ の第 j 番目の要素 $\frac{\partial r(\boldsymbol{\xi})}{\partial \xi_j}$ を ξ_i で微分したものを ij 要素として持つ $p \times p$ の行列を Hessian 行列と呼び $H(\boldsymbol{\xi})$ または $\frac{\partial^2 r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}'}$ と記す。

$$H(\boldsymbol{\xi}) = \frac{\partial^2 r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}'} = \left\{ \frac{\partial^2 r(\boldsymbol{\xi})}{\partial \xi_i \partial \xi_j} \right\} \quad (8)$$

この表記は $\frac{\partial^2 r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi} \partial \boldsymbol{\xi}'}$ が、 $p \times 1$ のベクトル $\frac{\partial r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}$ の転置である $1 \times p$ のベクトルを $\frac{\partial r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}'}$ と記し、その第 j 要素、すなわち第 j 列である $\frac{\partial r(\boldsymbol{\xi})}{\partial \xi_j}$ を $\boldsymbol{\xi}$ で微分

した結果である $p \times 1$ のベクトル $\frac{\partial}{\partial \boldsymbol{\xi}} \frac{\partial r(\boldsymbol{\xi})}{\partial \xi_j}$ を横に並べたものであることを示している。なお、 $H(\boldsymbol{\xi})$ は対称行列となり、その第 j 番目の対角要素には $\frac{\partial^2 r(\boldsymbol{\xi})}{\partial \xi_j^2}$ が入る。

一般的に、 n 次元のベクトル値を持つ関数 $r(\boldsymbol{\xi})$

$$\boldsymbol{r}(\boldsymbol{\xi}) = \begin{pmatrix} r_1(\boldsymbol{\xi}) \\ r_2(\boldsymbol{\xi}) \\ \vdots \\ r_i(\boldsymbol{\xi}) \\ \vdots \\ r_n(\boldsymbol{\xi}) \end{pmatrix} \quad (9)$$

をそのパラメタである $p \times 1$ のベクトル $\boldsymbol{\xi}$ の第 j 要素 ξ_j で微分したものを $\frac{\partial r_i(\boldsymbol{\xi})}{\partial \xi_j}$ を $n \times p$ の行列の形に並べたものを $\nabla \mathbf{r}$ もしくは $\frac{\partial \mathbf{r}}{\partial \boldsymbol{\xi}}$ と記しヤコビアン行列 (Jacobian matrix) と呼ぶ。

$$\nabla \mathbf{r} = \frac{\partial \mathbf{r}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \left\{ \frac{\partial r_i(\boldsymbol{\xi})}{\partial \xi_j} \right\} = \begin{pmatrix} \frac{\partial r_1(\boldsymbol{\xi})}{\partial \xi_1} \\ \frac{\partial r_2(\boldsymbol{\xi})}{\partial \xi_1} \\ \vdots \\ \frac{\partial r_i(\boldsymbol{\xi})}{\partial \xi_1} \\ \vdots \\ \frac{\partial r_n(\boldsymbol{\xi})}{\partial \xi_1} \end{pmatrix} \quad (10)$$

伝統的に、Jacobian の場合はその列にパラメタ $\boldsymbol{\xi}$ を配置するため、Jacobian の第 i 行には $r_i(\boldsymbol{\xi})$ の gradient の転置が入っていることになる。すなわち、スカラー値を返す関数 $r(\boldsymbol{\xi})$ の Jacobian 行列は gradient の転置である。同様に、 $r(\boldsymbol{\xi})$ の Hessian は Jacobian の表記を使えば

$$H(\boldsymbol{\xi}) = \left(\nabla \frac{\partial r(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right)' \quad (11)$$

となる。

1.3 連鎖律 (合成関数の微分) Chain Rule for Composite Function

$q \times 1$ のベクトル \mathbf{y} が実は $p \times 1$ のベクトル \mathbf{x} の関数として

$$\mathbf{y} = \mathbf{y}(\mathbf{x}) = \begin{pmatrix} y_1(\mathbf{x}) \\ y_2(\mathbf{x}) \\ \vdots \\ y_q(\mathbf{x}) \end{pmatrix} \quad (12)$$

として表現できる場合、スカラー値を取る \mathbf{y} の関数 $r(\mathbf{y})$ を x_j で偏微分したものは

$$\frac{\partial r(\mathbf{y})}{\partial x_j} = \sum_{i=1}^q \frac{\partial r(\mathbf{y})}{\partial y_i} \frac{\partial y_i}{\partial x_j} = \sum_{i=1}^q \frac{\partial r(\mathbf{y})}{\partial y_i} \frac{\partial y_i(\mathbf{x})}{\partial x_j} \quad (13)$$

である。したがって、これをまとめて $p \times 1$ の gradient の形で表現すれば

$$\frac{\partial r(\mathbf{y})}{\partial \mathbf{x}} = \left(\left(\frac{\partial r(\mathbf{y})}{\partial \mathbf{y}} \right)' \frac{\partial \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}} \right)' \quad (14)$$

$$= \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)' \frac{\partial r(\mathbf{y})}{\partial \mathbf{y}} \quad (15)$$

となる。ただし、 $\frac{\partial \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}}$ は $q \times p$ の Jacobian 行列、 $\frac{\partial r(\mathbf{y})}{\partial \mathbf{y}}$ は $q \times 1$ の $r(\mathbf{y})$ の gradient である。

また、 $r(\mathbf{y})$ を n 次元の関数とすればその \mathbf{x} に関する $n \times p$ の Jacobian 行列は

$$\frac{\partial \mathbf{r}(\mathbf{y})}{\partial \mathbf{x}} = \frac{\partial \mathbf{r}(\mathbf{y})}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \quad (16)$$

と書ける。ただし、 $\frac{\partial \mathbf{r}(\mathbf{y})}{\partial \mathbf{y}}$ は $n \times q$ の Jacobian 行列である。

1.4 数値微分

$f(x)$ の x に関する微分のを $x = a$ で評価したものは、たとえば、ごく小さな h の値を用いて

$$\left. \frac{df(x)}{dx} \right|_{x=a} = f'(a) \approx \frac{f(a+h) - f(a-h)}{2h} \quad (17)$$

で近似的に求めることが出来るが、これを数値微分 (numerical differentiation) と呼ぶ。同様に、 $f(\mathbf{x})$ の x_j に関する偏微分の値を $\mathbf{x} = \mathbf{a}$ で評価したものは

$$\left. \frac{\partial f(\mathbf{x})}{\partial x_j} \right|_{\mathbf{x}=\mathbf{a}} \approx \frac{f(a_1, a_2, \dots, a_{j-1}, a_j + h, a_{j+1}, \dots, a_p) - f(a_1, a_2, \dots, a_{j-1}, a_j - h, a_{j+1}, \dots, a_p)}{2h} \quad (18)$$

である。

一般的に、 n 次元のベクトルを返す関数 $f(\mathbf{x})$ の Jacobian 行列を $\mathbf{x} = \mathbf{a}$ で数値的に評価したものを返す関数があると便利であるが以下にその例を示す。関数の引数は、Eq.(17) や Eq.(18) に示したものである。

```
JacobianMat <- function( a, f, h=1e-06, ... ){
# Function to calculate the difference-quotient approx Jacobian matrix
# of f(x) at x=a
#
# Args:
#   a      parameter vector of size p x 1
#   f      the function which returns a vector of size n x 1
#   h      eps for difference quotient
#   ...    additional parameters to f
#
# Values:
#   J      n x p Jacobian matrix consisting of
#           part d f / part d x
#
p=length(a)
n=length(f(a, ...))

epsmat=diag(p)*h
J=matrix(0, n, p)
for(j in 1:p)
  J[, j]=( f(a+epsmat[, j], ...) - f(a-epsmat[, j], ...) )/(2*h)

return( J )
} # end of JacobianMat
```

この関数を使えば、 $r(\xi)$ を p 次元のパラメタベクトル ξ を持つスカラー値を返す関数 (目的関数) とすれば、その gradient や Hessian は以下のようにして計算することが出来る。

```

# objective function
r <- function( x, additional_param ){
  # calculation of the function value at x
  return( function_value )
}

# gradient vector of r at x
grad <- function( x, additional_param ){
  g <- JacobianMat( x, r, additional_param as ... )
  return( g )
}
# Hessian matrix of r at x
Hess <- function( x, additional_param ){
  H <- JacobianMat( x, grad, additional_param as ... )
  return( H )
}

# gradient at x=a
a <- the value of x
gradient <- g( a, additional_param )
Hessian <- Hess( a, additional_param )

```

という形で計算することが出来る。なお、関数 grad の中に r が、また関数 Hess の中に grad が書かれているが、これらは grad や Hess が定義された環境の中に存在するものとして認識されている。

以下にロジスティック回帰の例を示す。

```

# probability (logistic function with slope and intercept, no 1.7)
P <- function( X, param ){
  a=param[1]; b=param[2]
  z=a*X+b
  P=1/(1+exp(-z))
  return( P )
} # end of P

# minus log likelihood
mllh <- function( param, data ){
  x=data$x; f=data$f; r=data$r
  P=P(x,param)
  llh=sum( r*log(P)+(f-r)*log(1-P) )
  return( -llh )
} # end of mllh

# numeric first derivative of mllh
dmlh <- function( param, data ){
  # mllh comes from the parent environment
  res=c( JacobianMat( param, mllh, data=data ) )
  cat( ".dn." )
  return( res )
} # end of dmlh

# second derivative matrix using numeric first derivative of mllh
d2mllh <- function( param, data ){
  # dmlh and mllh come from the parent environment
  res=JacobianMat( param, dmlh, data=data )
  cat( ".d2nn." )
  return( res )
} # end of d2mllh

```