AI6104 - MATHEMATICS FOR AI

TUTORIAL 7 - TAYLOR SERIES & NEWTON'S METHOD

Problem 1

Calculate the second-degree Taylor polynomial of the following functions

(a) $f(x,y) = e^{x+2y}$ at $(0,0)$

(b) $f(x,y) = x\sqrt{y}$ at $(1,4)$

(c) $f(x,y) = \arctan(x+2y)$ at $(1,0)$

(d) $f(x,y) = x^2y + y^2$ at $(1,3)$

(e) $f(x,y) = \ln(x^2+y^2+1)$ at $(0,0)$

Solution:

(a) $f(x,y) \approx 1 + x + 2y + \frac{1}{2}x^2 + 2xy + 2y^2$

(b) $f(x,y) \approx -1 + 2x + \frac{1}{4}y + \frac{1}{4}(x-1)(y-4) - \frac{1}{64}(y-4)^2$

(c) $f(x,y) \approx \frac{1}{4}\pi - \frac{3}{4} + x + 2y - \frac{x^2}{4} - xy - y^2$

(d) $f(x,y) \approx -15 + 6x + 7y + 3(x-1)^2 + 2(x-1)(y-3) + (y-3)^2$

(e) $f(x,y) \approx x^2 + y^2$

Problem 2

Use Newton's method to minimize the Powell function:

$$f(x_1,x_2,x_3,x_4) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4$$

Use as the starting point $x^{(0)} = [3,-1,0,1]^\top$, perform three iterations.

Solution:

Note that $f(x^{(0)}) = 215$, We have

$$\nabla f(x) = \begin{bmatrix} 2(x_1+10x_2) + 40(x_1-x_4)^3 \\ 20(x_1+10x_2) + 4(x_2-2x_3)^3 \\ 10(x_3-x_4) - 8(x_2-2x_3)^3 \\ -10(x_3-x_4) - 40(x_1-x_4)^3 \end{bmatrix}$$

and the Hessian $H(x)$ is

$$\begin{bmatrix} 2+120(x_1-x_4)^2 & 20 & 0 & -120(x_1-x_4)^2 \\ 20 & 200+12(x_2-2x_3)^2 & -24(x_2-2x_3)^2 & 0 \\ 0 & -24(x_2-2x_3)^2 & 10+48(x_2-2x_3)^2 & -10 \\ -120(x_1-x_4)^2 & 0 & -10 & 10+120(x_1-x_4)^2 \end{bmatrix}$$

Plug in the initial point $x^{(0)}$, we have

$$\nabla f(x^{(0)}) = [306, -144, -2, -310]^\top$$

$$H(x^{(0)}) = \begin{bmatrix} 482 & 20 & 0 & -480 \\ 20 & 212 & -24 & 0 \\ 0 & -24 & 58 & -10 \\ -480 & 0 & -10 & 490 \end{bmatrix}$$

Then use the updating formula

$$x^{(k+1)} = x^{(k)} - H(x^{(k)})^{-1} \nabla f(x^{(k)})$$

we can get

$$\begin{aligned} x^{(1)} &= x^{(0)} - H(x^{(0)})^{-1} \nabla f(x^{(0)}) \\ &= \begin{bmatrix} 3 \\ -1 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 482 & 20 & 0 & -480 \\ 20 & 212 & -24 & 0 \\ 0 & -24 & 58 & -10 \\ -480 & 0 & -10 & 490 \end{bmatrix}^{-1} \times \begin{bmatrix} 306 \\ -144 \\ -2 \\ -310 \end{bmatrix} \\ &= [1.5873, -0.1587, 0.2540, 0.2540]^\top & f(x^{(1)}) = 31.8 \\ x^{(2)} &= [1.0582, -0.1058, 0.1694, 0.1694]^\top & f(x^{(2)}) = 6.28 \\ x^{(3)} &= [0.7037, -0.0704, 0.1121, 0.1111]^\top & f(x^{(3)}) = 1.24 \end{aligned}$$

Problem 3

Consider we want to minimize $\sum_{i=1}^{m} (r_i(\boldsymbol{x}))^2$, where $r_i : \mathbb{R}^n \to \mathbb{R},\ i = 1, \dots, m$ are given functions. This particular problem is called a *nonlinear least-squares problem.* Suppose that we are given m measurements of a process at m points in time. Let $t_1, \dots, t_m$ denote the measurements times and $y_1, \dots, y_m$ the measurements values. For example, we may fit a sinusoid function to the measurement data, where the sinusoid function is

$$y = A \sin(\omega t + \phi)$$

with appropriate choices of the parameters A, ω, ϕ.

(a) In the case that we fit the data with sinusoid function, construct the objective function, in the form of $\sum_{i=1}^{m} (r_i(\boldsymbol{x}))^2$, to represent the sum of squared errors between the measurement values and the function values at the corresponding points in time.

(b) We consider the general case here. Let $\boldsymbol{x}$ represent the vector of decision variables, and $r(\boldsymbol{x}) = [r_1(\boldsymbol{x}), \dots, r_m(\boldsymbol{x})]^\top$. Write the objective function as $f(\boldsymbol{x}) = r(\boldsymbol{x})^\top r(\boldsymbol{x})$. Write down the gradient and the Hessian of f using the Jacobian of r, and the updating formula of x using Newton's method.

Solution:

(a) The objective function is

$$\sum_{i=1}^{m} (y_i - A \sin(\omega t_i + \phi))^2$$

(b) The j-th component of $\nabla f(\boldsymbol{x})$ is

$$(\nabla f(\boldsymbol{x}))_j = \frac{\partial f}{\partial x_j}(\boldsymbol{x}) = 2\sum_{i=1}^{m} r_i(\boldsymbol{x})\frac{\partial r_i}{\partial x_j}(\boldsymbol{x})$$

Note that the Jacobian matrix of r is

$$J(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1}(\boldsymbol{x}) & \cdots & \frac{\partial r_1}{\partial x_n}(\boldsymbol{x}) \\ \vdots & & \\ \frac{\partial r_m}{\partial x_1}(\boldsymbol{x}) & \cdots & \frac{\partial r_m}{\partial x_n}(\boldsymbol{x}) \end{bmatrix}$$

Thus, the gradient of f can be written as

$$\nabla f(\boldsymbol{x}) = 2J(\boldsymbol{x})^\top r(\boldsymbol{x})$$

Now we compute the Hessian matrix. Note that the (k, j)-th component of the Hessian is given by

$$\begin{aligned} \frac{\partial^2 f}{\partial x_k \partial x_j}(\boldsymbol{x}) &= \frac{\partial}{\partial x_k}\left(\frac{\partial f}{\partial x_j}(\boldsymbol{x})\right) \\ &= \frac{\partial}{\partial x_k}\left(2\sum_{i=1}^{m} r_i(\boldsymbol{x})\frac{\partial r_i}{\partial x_j}(\boldsymbol{x})\right) \\ &= 2\sum_{i=1}^{m}\left(\frac{\partial r_i}{\partial x_k}(\boldsymbol{x})\frac{\partial r_i}{\partial x_j}(\boldsymbol{x}) + r_i(\boldsymbol{x})\frac{\partial^2 r_i}{\partial x_k \partial x_j}(\boldsymbol{x})\right) \end{aligned}$$

Letting $S(\boldsymbol{x})$ be the matrix whose (k, j)-th component is

$$\sum_{i=1}^{m} r_i(\boldsymbol{x})\frac{\partial^2 r_i}{\partial x_k \partial x_j}(\boldsymbol{x})$$

We then writhe the Hessian matrix as

$$H(\boldsymbol{x}) = 2(J(\boldsymbol{x})^\top J(\boldsymbol{x}) + S(\boldsymbol{x}))$$

Therefore, the updating formula using Newton's method is given by

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - (J(\boldsymbol{x})^\top J(\boldsymbol{x}) + S(\boldsymbol{x}))^{-1} J(\boldsymbol{x})^\top r(\boldsymbol{x})$$

Problem 4

In this question, we will focus on simple linear regression which takes the following form

$$\hat{Y} = f(X) = X\beta = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & \cdots & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{(N)} & x_1^{(N)} & \cdots & x_n^{(N)} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(N)} \end{bmatrix}$$

where X is called input variable and $\hat{Y}$ is output variable. The coefficient β is also called model parameter. N is the number of samples. We will use a mean squared error (MSE) function defined as follows,

$$\mathcal{L}(\beta) = \frac{1}{2N}(X\beta - Y)^\top (X\beta - Y)$$

where Y is the ground truth. Our goal is to find β^* that minimize this cost. Write down the updating rules for β using gradient descent. (Use learning rate α.)

Solution:

Consider the derivation of cost function to all β can be vectorized as

$$\frac{\partial \mathcal{L}}{\partial \beta} = \frac{1}{N} X^\top (X\beta - Y)$$

Therefore, the updating rules for β using gradient descent is

$$\beta^{(k+1)} = \beta^{(k)} - \alpha \cdot \frac{1}{N} X^\top (X\beta - Y)$$