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 $(0,0)$

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.

Problem 3

Consider we want to minimize $\sum_{i=1}^{m}(r_i(\boldsymbol{x}))^2$, where $r_i : \mathbb{R}^n \to \mathbb{R},\ i = 1,\ldots,m$ 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 $t_1,\ldots,t_m$ denote the measurements times and $y_1,\ldots,y_m$ 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 $r(\boldsymbol{x}) = [r_1(\boldsymbol{x}),\ldots,r_m(\boldsymbol{x})]^\top$. 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.

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.)