山东大学 计算机科学与技术 学院

机器学习与模式识别 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202300130183 | 姓名： 宋浩宇 | | 班级：23级人工智能班 |
| 实验题目：Logistic Regression and Newton’s Method | | | |
| 实验学时：2 | | 实验日期：2025/3/18 | |
| 实验环境：  软件环境：  系统：Windows 11 家庭中文版23H2 22631.4317  计算软件：MATLAB 版本: 9.8.0.1323502 (R2020a)  Java 版本: Java 1.8.0\_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode  硬件环境：  CPU：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz  内存：32.0 GB (31.6 GB 可用)  磁盘驱动器：NVMe WD\_BLACKSN850X2000GB  显示适配器：NVIDIA GeForce RTX 4080 Laptop GPU | | | |
| 1. 实验内容   In this exercise, you will use Newton’s Method to implement logistic regression  on a classification problem.   1. 实验步骤 2. 获取实验需要的数据      1. 构造模型      1. 构造代价函数      1. 构造牛顿法逻辑回归的计算方法      1. 用matlab完成计算 2. 完成结果可视化 3. 计算要求的测试题目 4. 测试结果   完成计算后的分类如下图：    数据为：    经过5次迭代后收敛，θ的值为[-16.3787,0.1483,0.1589]^T    计算出的录取率为0.3320，未被录取率为0.6680   1. 附录：实现源代码  |  | | --- | | %% 清空工作区  clc; clear; close all;  %% 加载数据  x = load('ex4Data/ex4x.dat')  y = load('ex4Data/ex4y.dat')  % disp(x)  % disp(y)  x = [ones(size(x,1),1) x];  pos = find(y==1);  neg = find(y==0);  figure;  plot(x(pos,2),x(pos,3),'+');hold on;  plot(x(neg,2),x(neg,3),'o');  %% 全局变量  global theta;% 参数  %% 计算  theta = zeros(size(x,2),1);  gradientDescent(x,y,15);  disp(theta)  x\_l = linspace(0,100,100);  y\_l = (-theta(1)-theta(2)\*x\_l)/theta(3);  plot(x\_l,y\_l);hold on;  test = [1 20 80];  disp("通过率为")  disp(hx(test))  %% 函数  %% hx  function result = hx(x)      global theta;      result = sigmoid(x \* theta);  end  %% cost  function J = cost(x,y)      global theta;      m = length(y);      h = hx(x);      J = (1/m) \* (-y' \* log(h) - (1 - y)' \* log(1 - h));  end  %% gradient decent  function [theta,J\_history] = gradientDescent(x,y,num\_iters)      global theta;      J\_history = cost(x,y);      for i = 1:num\_iters          H = HMatrix(x,y);          disp(H)          grad = deltaJ(x,y);          H = inv(H);          theta = theta - H\*grad;          J\_history = [J\_history cost(x,y)]          if abs(J\_history(end) - J\_history(end-1)) < 1e-6              disp("迭代次数");              disp(i);              break;          end      end  end  % HesMatrix  % 海森矩阵  function H = HMatrix(x, y)      global theta;      m = length(y);      h = sigmoid(x \* theta);      H = zeros(size(theta));      for i = 1:m          h\_i = h(i);          x\_i = x(i, :);          H = H + h\_i \* (1 - h\_i) \* (x\_i' \* x\_i);      end      H = H / m;  end  % ΔJ  function result = deltaJ(x,y)      global theta;      m = length(y);      h = hx(x);      result = (1/m) \* (x' \* (h - y));  end  % sigmoid  function result = sigmoid(z)      result = 1 ./ (1 + exp(-z));  end | | | | |