机器学习课程实验四

2022 年 9 月 29 日　苏博南　202000460020

1　数据打印

利用 Matlab 工具载入并打印数据，得到如下图 1 所示结果

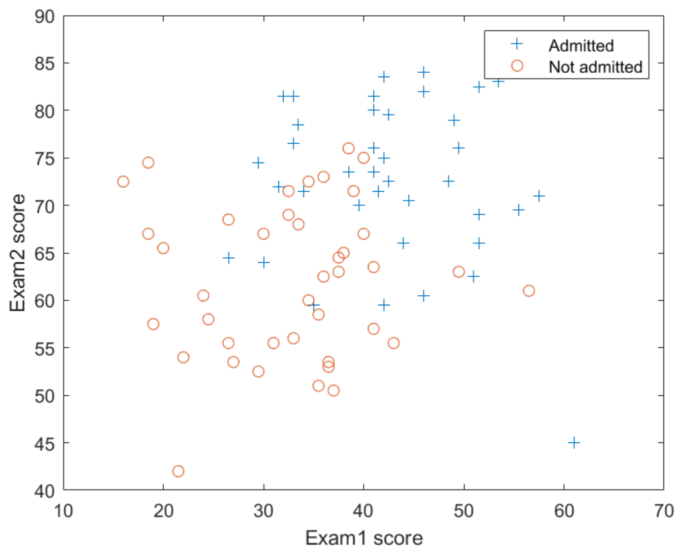

图 1: 数据打印结果

```
X = load('ex4Data/ex4x.dat');
Y = load('ex4Data/ex4y.dat');

pos = find(Y == 1);
neg = find(Y == 0);
[m, n] = size(X);
X = [ones(m, 1), X];
n = n + 1;

plot(X(pos, 2), X(pos, 3), '+');
hold on;
plot(X(neg, 2), X(neg, 3), 'o');
legend('Admitted', 'Not admitted');
```

2　牛顿法迭代收敛 Logistic 回归

考虑 Logistic 二分类问题:

$$P(y = 1 \mid x; \theta) = h_\theta(x) = sigmond(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \in (0, 1) \tag{1}$$

对于数据集 $X \in R^{m\times n}, Y \in R^n$，设置 Cross-Entropy Loss Function 为：

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}ln(h_\theta(x^{(i)})) - (1-y^{(i)})ln(1-h_\theta(x^{(i)}))] \tag{2}$$

然后可以求得 Loss Function 的梯度为：

$$\nabla J(\theta) = \sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]x^{(i)} = X^T(\hat{Y} - Y) \tag{3}$$

根据牛顿法要求，我们可以最小化 Loss Function 的二阶泰勒展开式子。即

$$\min_\theta \ J(\theta) \approx \min_\theta \ J(\theta_t) + \nabla J(\theta_t)^T(\theta-\theta_t) + \frac{1}{2}(\theta-\theta_t)^T\nabla^2 J(\theta_t)(\theta-\theta_t) \tag{4}$$

为使 (4) 式极小，可以要求其导数为 0，即：

$$\frac{dJ(\theta)}{d\theta} = \nabla J(\theta_t) + \nabla^2 J(\theta_t)(\theta-\theta_t) = \mathbf{0} \tag{5}$$

故有迭代过程：

$$\theta = \theta_t - [\nabla^2 J(\theta_t)]^{-1}\nabla J(\theta_t) \tag{6}$$

在 Logistic 回归中，有

$$\mathbf{H}(\theta) = \nabla^2 J(\theta) = \sum_{i=1}^{m}\frac{\nabla h_\theta(x^{(i)})x^{(i)}}{\nabla\theta} = \sum_{i=1}^{m}x^{(i)}h_\theta(x^{(i)})h_\theta(-x^{(i)})(x^{(i)})^T \tag{7}$$

其中 **H** 也被称为 Hessian Matrix。于是得到迭代过程：

$$\theta = \theta_t - \mathbf{H}(\theta_t)^{-1}\nabla J(\theta_t) \tag{8}$$

对应的 matlab 代码如下：

```matlab
X = load('ex4Data/ex4x.dat');
Y = load('ex4Data/ex4y.dat');

pos = find(Y == 1);
neg = find(Y == 0);
[m, n] = size(X);
X = [ones(m, 1), X];
n = n + 1;

plot(X(pos, 2), X(pos, 3), '+');
hold on;
plot(X(neg, 2), X(neg, 3), 'o');

nIt = 0;
delta = 1;
theta = zeros(n, 1);
while (delta > 1e-6)
    H = X' * diag(sigmond(X * theta) .* sigmond(-X * theta), 0) * X;
    grad = X' * (sigmond(X * theta) - Y);
    theta = theta - H \ grad;
```

```
        delta = max(abs(H \ grad));
        nIt = nIt + 1;
    end

    nIt
    xs = 12 : 1 : 63;
    ys = (theta(1) + theta(2) * xs) / (-theta(3));
    plot(xs, ys, 'b-');
    legend('Admitted', 'Not admitted', 'Decision boundary');


    xlabel('Exam1 score');
    ylabel('Exam2 score');
```

最终可以得到，仅通过 7 次迭代就完成了收敛 (θ 变化值小于 10^{-6})。得到的 $\theta^* = (-16.3787, 0.1483, 0.1589)^T$。(作出 $(\theta^*)^T x = 0$ 的图像如下图 2 所示：

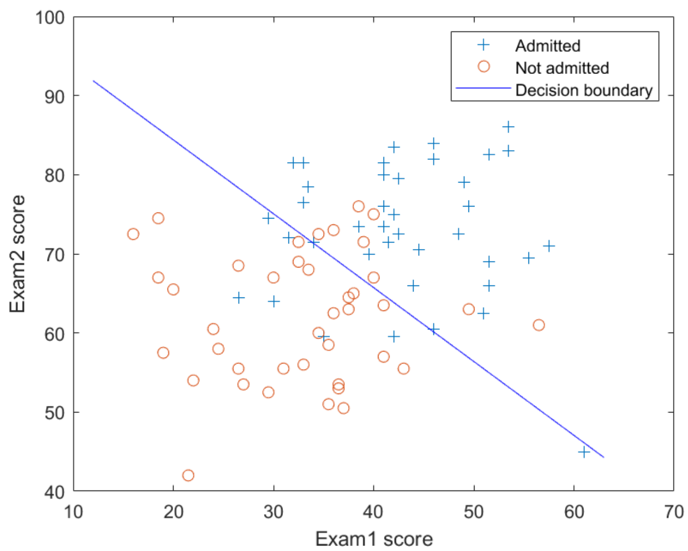

图 2: 分类结果

对于一个 Exam 1 为 20 分，Exam 2 为 80 分的同学，预测被录取概率为：

$$P(+1 \mid (1, 20, 80)^T; \theta^*) = 33.2\% \tag{9}$$