机器学习第三章作业

任齐轩
重庆大学-计卓 2 班-20204154

日期： September 19, 2022

1 第一题

1. 当数据集做过归一化时，不必考虑 b
2. 当预测结果 y 只与 x 的线性变换结果有关时，不必考虑 b

2 第三题

通过编程实现对率回归 **代码如下：**

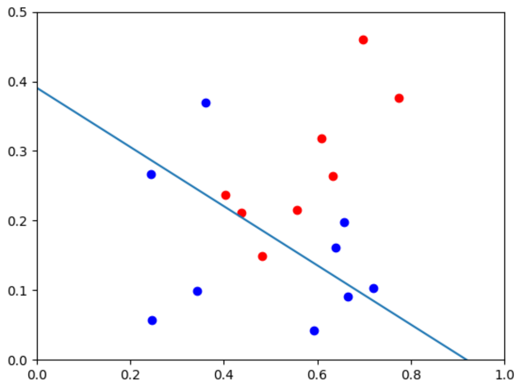

图 1: 实现结果

```
import numpy as np
from matplotlib import pyplot as plt


def load_dataset():
    file = open(r'melon3_dataset.txt')
    dataMat = []
    labelMat = []
    for line in file.readlines():
        lineArr = line.strip().split()
        dataMat.append([1.0, float(lineArr[0]), float(lineArr[1])])
        labelMat.append(int(lineArr[2]))
```

```python
    return dataMat, labelMat


def sigmoid(inX):
    return 1.0 / (1 + np.exp(-inX))


def mat_plot(dataMatrix, labelMatrix, weights):
    fig = plt.figure()
    ax = fig.add_subplot(111)
    n = np.shape(dataMatrix)[0]
    for i in range(n):
        if labelMatrix[i] == 1:
            ax.scatter(dataMatrix[i, 1], dataMatrix[i, 2], c='r')
        else:
            ax.scatter(dataMatrix[i, 1], dataMatrix[i, 2], c='b')
    x = np.arange(0, 1, 0.01)
    y = (-weights[0] - weights[1] * x) / weights[2]
    ax.plot(x, y)
    plt.xlim(0, 1)
    plt.ylim(0, 0.5)
    plt.show()


def model():
    dataMat, labelMat = load_dataset()
    dataMatrix = np.mat(dataMat)
    labelMatrix = np.mat(labelMat).transpose()
    m, n = np.shape(dataMatrix)
    alpha = 0.001
    maxCycles = 500
    weights = np.ones((n, 1))
    for i in range(maxCycles):
        h = sigmoid(dataMatrix * weights)
        error = (labelMatrix - h)
        gradient = dataMatrix.transpose() * error
        weights += alpha * gradient
    mat_plot(dataMatrix, labelMatrix, weights)


if __name__ == '__main__':
    model()
```

3 第五题

通过编程实现线性判别分析 **代码如下：**

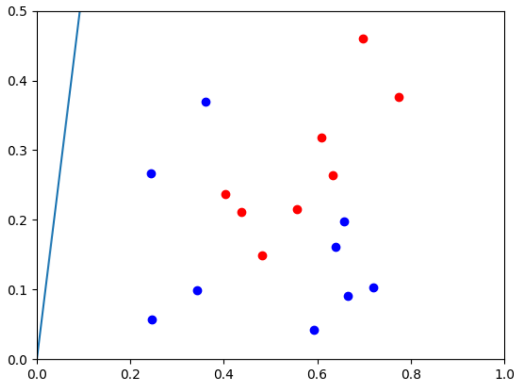

图 2: 实现结果

```
import numpy as np
from matplotlib import pyplot as plt


def load_dataset():
    file = open(r'melon3_dataset.txt')
    dataMat = []
    labelMat = []
    for line in file.readlines():
        lineArr = line.strip().split()
        dataMat.append([float(lineArr[0]), float(lineArr[1])])
        labelMat.append(int(lineArr[2]))
    return dataMat, labelMat


def mat_plot(w, data):
    fig = plt.figure()
    ax = fig.add_subplot(111)
    n = np.shape(data)[0]
    for i in range(n):
        if i < 8:
            ax.scatter(data[i, 0], data[i, 1], c='r')
        else:
            ax.scatter(data[i, 0], data[i, 1], c='b')
    x = np.arange(0, 1, 0.01)
    y = (w[0, 0] / w[1, 0]) * x
    ax.plot(x, y)
    plt.xlim(0, 1)
    plt.ylim(0, 0.5)
    plt.show()
```

```python
def model():
    dataMat, labelMat = load_dataset()
    data = np.array(dataMat)
    X_0 = np.array(data[:8])
    X_1 = np.array(data[8:])
    miu_0 = np.mean(X_0, axis=0).reshape((-1, 1))
    miu_1 = np.mean(X_1, axis=0).reshape((-1, 1))
    cov_0 = np.cov(X_0, rowvar=False)
    cov_1 = np.cov(X_1, rowvar=False)
    S_w = np.mat(cov_0 + cov_1)
    Omega = S_w.I * (miu_0 - miu_1)
    mat_plot(Omega, data)


if __name__ == '__main__':
    model()
```

4 第七题

理论最优的 EOOC 编码为:

$$
\begin{aligned}
C_1 &= +1,+1,+1,+1,+1,+1,+1,+1,+1 \\
C_2 &= +1,+1,+1,-1,-1,-1,-1,-1,-1 \\
C_3 &= -1,-1,-1,+1,+1,+1,-1,-1,-1 \\
C_4 &= -1,-1,-1,-1,-1,-1,+1,+1,+1
\end{aligned}
$$

此时最大海明距离为 6

证明. 假设存在海明距离为 7 的编码，则此时 EOOC 编码之间 C_i 不同的位置至少有 7 个，则有:

$$
\begin{gathered}
dif_{ij} = \{x|C_{ix} \neq C_{jx}\} \geq 7 \\
dif_{jk} = dif_{ij} \cup dif_{ik} - dif_{ij} \cap dif_{ik} \\
dif_{jk} = |dif_{ij}| + |dif_{ik}| - 2 * |dif_{ij} \cap dif_{ik}|
\end{gathered}
$$

设 $dif_{xj} = 7, dif_{ix} = 7$，则有 $dif_{ij} = 6$

$$
dif_{ix} + duf_{ij} - 2 * |dif_{ix} \cap dif_{ij}| \leq 5 < 7
$$

因此，显然不成立，得证 □

5 第九题

因为对 OvR、MvM 来说，由于对每个类进行了相同的处理，其拆解出的二分类任务中类别不平衡的影响会相互抵消，因此通常不需要进行专门处理