



## 中山大学数据科学与计算机学院

### 移动信息工程专业-人工智能

### 本科生实验报告

(2016 学年秋季学期)

课程名称: Artificial Intelligence

教学班级	M2	专业 (方向)	移动互联网
学号	14353205	姓名	刘万里

## 一、 实验题目

逻辑回归算法实现:

本次数据集为乳腺癌诊断, 标签为 0 , (不患病) 1 (患病)

## 二、 实验内容

### 1. 算法原理

对于二元分类问题, 向量的每一维对应一个权重, 我们尽可能地找出这样的一个权重向量, 使得可以完美地 (至少也是错误率最低的) 划分二元分类, 通过逻辑回归的算法每次更新权重向量, 直到不能再更新为止 (每一维的梯度为 0 的时候就不会继续更新了), 再使用该权重向量去划分测试样本得到测试集的分类。

### 2. 伪代码

```
int main()
{
    readFile();
    FindW(); //找权重系数
    exam(); //测试
    show(TP, TN, FP, FN);
    return 0;
}
```

### 3. 关键代码截图 (带注释)

找到正确划分的  $w$  向量:



```
73 void FindW()  
74 {  
75     for(times = 0; times < MAX_ITER_TIMES; times++)  
76     {  
77         memset(err, 0, sizeof(err));  
78         int flag=0;  
79         memset(pre1, 0, sizeof(pre1));  
80         for(int k = 0; k < VECTOR_NUMS; k++)  
81             w2[k] = w1[k]; //更新系数  
82         for(j = 0; j < row1; j++) //遍历所有训练样本, 计算出每一个样例权重分数  
83         {  
84             for(int k = 0; k < VECTOR_NUMS; k++)  
85             {  
86                 pre1[j] = pre1[j] + w2[k] * trainData[j][k];  
87             }  
88         }  
  
89         for(int j = 0; j < VECTOR_NUMS; j++)  
90         {  
91             for(int k = 0; k < row1; k++) //计算梯度  
92             {  
93                 err[j] += ( 1/(1+exp(-pre1[k])) - trainLabels[k] ) * (trainData[k][j]);  
94             }  
95             w1[j] = w2[j] - n * err[j]; //更新权重  
96             if(err[j] != 0)  
97             {  
98                 flag++;  
99             }  
100         }  
  
101         if(flag)  
102         {  
103             for(int k = 0; k < VECTOR_NUMS; k++)  
104                 w[k] = w1[k];  
105         }  
106         else  
107         {  
108             for(int k=0; k<VECTOR_NUMS; k++)  
109                 w[k] = w2[k];  
110             break; //如果梯度全部为0, 停止迭代, 因为权重不会继续更新的了。  
111         }  
112     }  
113     cout<<"迭代次数是 "<<times<<endl;  
114 }
```

对测试集进行测试:

```
115 void exam()  
116 {  
117     ofstream out("result.txt");  
118     for(i = 0; i < row2; i++) //对test的每个样例进行预测  
119     {  
120         for(int m = 0; m < VECTOR_NUMS; m++)  
121             findLabels[i] = findLabels[i] + w[m] * testData[i][m];  
122         float pre = 1 / ( 1 + exp(-findLabels[i]) );  
123         out<<pre<<endl;  
124         if (pre >= 0.5) findLabels[i] = 1;  
125         else if(pre<0.5) findLabels[i] = 0;  
126     }
```



```
127     for(i = 0; i < row2; i++)
128     {
129         if(testLabels[i]==findLabels[i])
130         {
131             if(testLabels[i]==1)    TP++;
132             else    TN++;
133         }
134         else
135         {
136             if(testLabels[i] == 1 && findLabels[i] == 0)    FN++;
137             else if(testLabels[i] == 0 && findLabels[i] == 1)    FP++;
138         }
139     }
140     cout<<"TP:"<<TP<<" TN:"<<TN<<" FP:"<<FP<<" FN:"<<FN<<endl;
141 }
```

#### 4. 创新点&优化（如果有）

### 三、 实验结果及分析

#### 1. 实验结果展示示例（可图可表可文字，尽量可视化）

(原始算法结果)

\\请截图你们的运行结果，包含四种指标

```
D:\sysu_three\one\Artificial_Intelligence\lab5.逻辑回归\实验五之DT&LR\数据集\LR.exe
迭代次数是 4000
TP:41 TN:48 FP:2 FN:9
accuracy:0.89
recall:0.82
precision:0.953488
F1:0.88172

-----
Process exited after 1.086 seconds with return value 0
请按任意键继续...
```

命名格式为：学号\_姓名拼音.pdf