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

（2017-2018 学年秋季学期）

课程名称：**Artificial Intelligence**

教学班级	1501 班	专业（方向）	互联网
学号	15352010	姓名	蔡烨

一、 实验题目

感知机学习 PLA

二、 实验内容

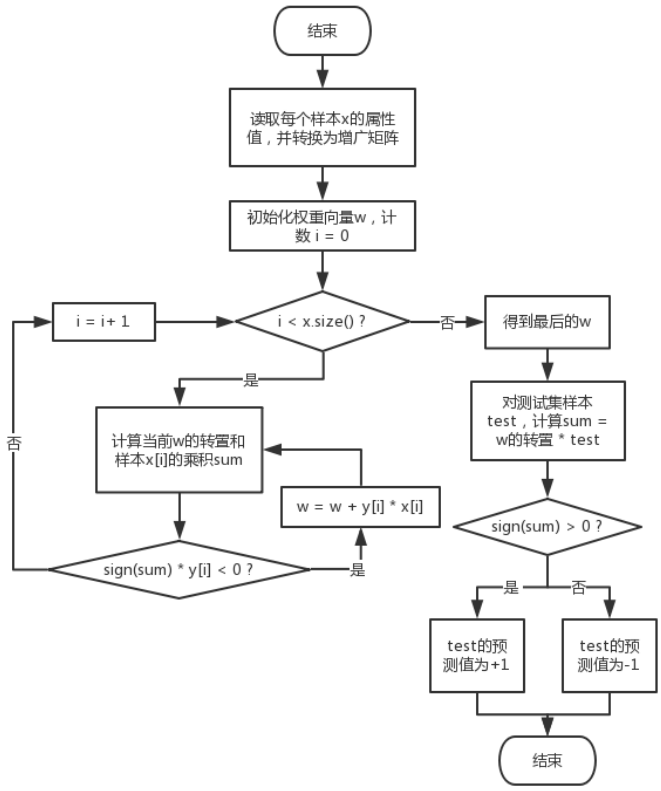
1. 算法原理

PLA 适用于二维的线性可划分问题。对于向量 x={x1，x2，…，xn}，每一个 xi 代表了一个属性值，y 则是该向量的标签，只有+1 和-1 之分。而权重向量 w={w1，w2，…，wn}代表每个属性的重要程度，wi 的变化会产生不同的数据，用 w 的增广向量 W={w0，w1，…，wn}的转置乘以}x 的增广矩阵 X={1，x1，x2，…，xn}，得到的值记为 sum。如果 sum 大于 0，则记为+1，否则记为-1.如果得到的结果和 y 相同，则说明此时的 W 适用，否则将 W+y·x 赋给 W，再循环上面的步骤，直到得到适用的 W。

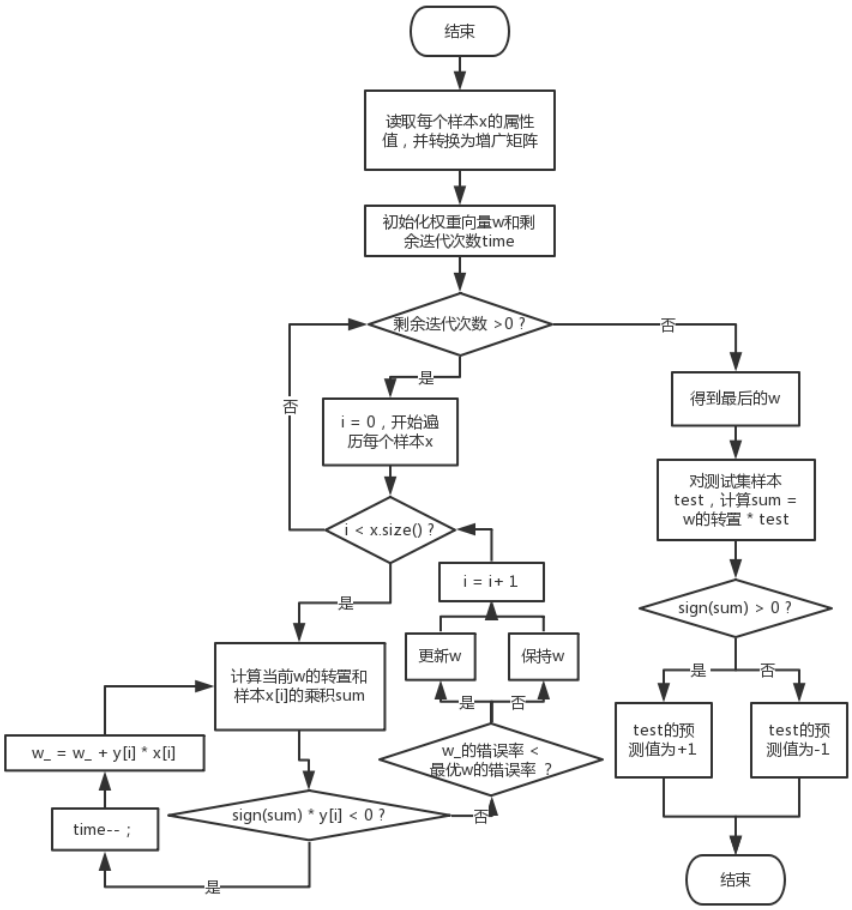
然而很多情况下并不是纯粹的线性可划分问题。此时找不到一个适用于所有 X 的 W，于是，利用口袋算法，即如果新得到的 W 的错误率更低，才将原先的 W 值换掉，否则保留原来的值，迭代一定次数后，得到一个错误率最低的 W，将其作为答案。

2. 伪代码

原始算法：


结束
读取每个样本x的属性值，并转换为增广矩阵
初始化权重向量w，计数 i = 0
i = i+ 1
i < x.size() ?
否
得到最后的w
是
对测试集样本test，计算sum = w的转置 * test
计算当前w的转置和样本x[i]的乘积sum
w = w + y[i] * x[i]
否
sign(sum) * y[i] < 0 ?
是
sign(sum) > 0 ?
是
否
test的预测值为+1
test的预测值为-1
结束

口袋算法：


结束
读取每个样本x的属性值，并转换为增广矩阵
初始化权重向量w和剩余迭代次数time
剩余迭代次数 >0 ?
否
是
得到最后的w
否
i = 0，开始遍历每个样本x
对测试集样本test，计算sum = w的转置 * test
i < x.size() ?
i = i+ 1
sign(sum) > 0 ?
是
更新w
保持w
否
是
否
计算当前w的转置和样本x[i]的乘积sum
w_ = w_ + y[i] * x[i]
w_的错误率 < 最优w的错误率 ?
test的预测值为+1
test的预测值为-1
time-- ;
sign(sum) * y[i] < 0 ?
否
是
结束

3. 关键代码截图（带注释）

原始算法：

```
for(int i=0; i<x[0].size(); i++)//w初始化
    w.push_back(1);
//遍历样本，计算w
for(int i=0; i<x.size(); i++){
    double sum = 0;
    for(int j=0; j<x[i].size(); j++)//计算w的转置和x的乘积
        sum += x[i][j] * w[j];

    while(sum*y[i] <= 0)  { //得到的结果和y不一样
        for(int j=0; j<w.size(); j++)
            w[j] = w[j] + y[i]*x[i][j];

        for(int j=0; j<x[i].size(); j++) //计算w的转置和x的乘积
            sum += x[i][j] * w[j];
    }
}
```

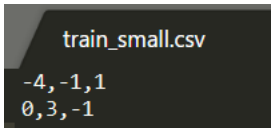
口袋算法：

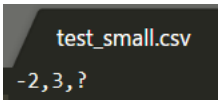
```
//w和w_初始化
vector<int> w_;
for(int i=0; i<x[0].size(); i++){
    w.push_back(1);//最优的w，口袋里的w
    w_.push_back(1);
}
int time = 300000;//迭代次数
while(time>0){
    for(int i=0; i<x.size(); i++){
        double sum = 0;
        for(int j=0; j<x[i].size(); j++)//计算w_的转置和x[i]的乘积
            sum += x[i][j] * w_[j];

        while(sum*y[i] <= 0){//更新w_，直到w_适用于样本x[i]
            for(int j=0; j<w.size(); j++)
                w_[j] = w_[j] + y[i]*x[i][j];
            time--;

            for(int j=0; j<x[i].size(); j++) //计算w_的转置和x[i]的乘积
                sum += x[i][j] * w_[j];
        }

        if(error(w_) <= error(w)){//如果w_的错误率低，则更新w
            for(int j=0; j<w.size(); j++)
                w[j] = w_[j];
        }
        if(time<0) break;
```

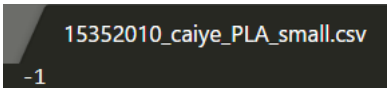
三、 实验结果及分析

1. 实验结果展示示例（可图可表可文字，尽量可视化）

使用 PPT 里的小数据集，使用原始算法或口袋算法结果都如下：

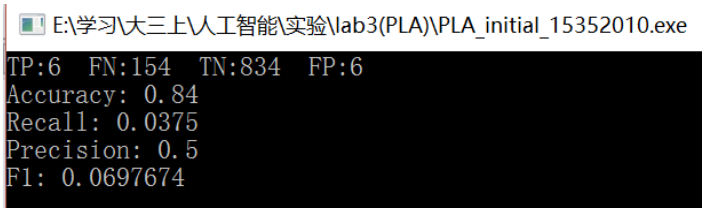






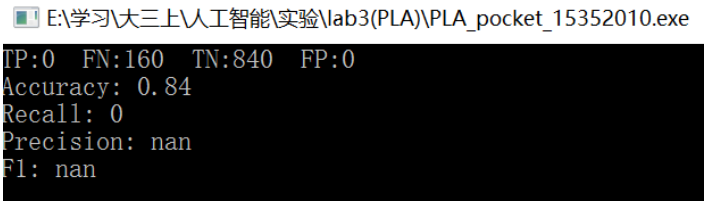
2. 评测指标展示即分析（如果实验题目有特殊要求，否则使用准确率）

原始算法：

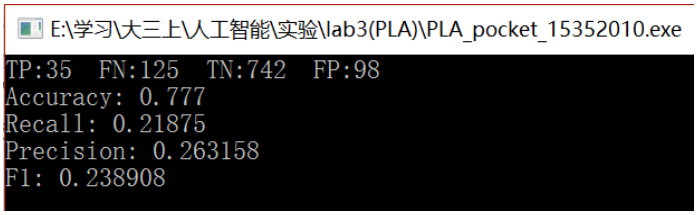


口袋算法：

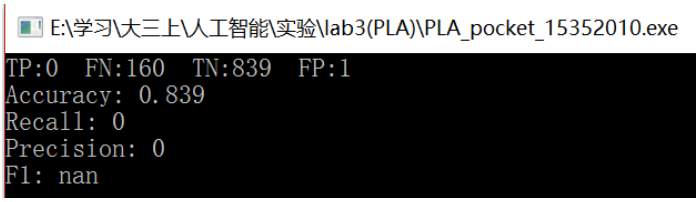
迭代 10000 次：



迭代 30000 次：



迭代 100000 次：



3. 思考题

（1）有什么其他的手段可以解决数据集非线性可分的问题？

答：①用多个 PLA 同时跑，这些 PLA 拥有初始化不同的权重向量。将得到的结果加权，算众数。（神经网络方法的雏形）

②换一种更新 w 的方法，而不采用 w+y • x，而这种方法要使得错误点对结果的影响最小。

③拟定一个容忍点，支持错误集。

④改变特征向量，转换为多维空间，而不止二维。

（2）为什么要用这四种评测指标：准确率、精确率、召回率、F 值？

答：①准确率：对于给定的数据集，分类器正确分类的样本数与总样本数之比。准确率越高，说明对数据的预测正确的概率更大。

②精确率：当+1 代表相关，而-1 代表不相关时，精确率就是被找到（被预测到）的相关的数据集/所有相关的数据集数。

③召回率：被找到的（被预测为）相关的数据集数/所有被找到（被预测）的数

据集。

④F 值：精确率和召回率的调和均值。当数据集对精确率和召回率的要求都高时，可以用 F 值来衡量。

这四个指标从不同维度反应了预测的结果，因为不清楚被预测的数据集是什么类型的，预测的目标是什么，只是简单地做了二分，当有了明确的目标时，就会有一个或多个明确的指标。

|---------- 如有优化，重复 1，2 步的展示，分析优化后结果 ----------|

PS：可以自己设计报告模板，但是内容必须包括上述的几个部分，不需要写实验感想