实验报告

报告人：万珂嘉 学号：19020006

一、实验目的

a. 了解线性可分，设计线性可分的数据集与非线性可分的数据集。
b. 采用线性可分数据集，对比 PLA 算法与 Pocket PLA 算法的性能。
c. 采用非线性可分数据集，验证 Pocket PLA 算法的性能。

二、实验步骤

2.1 数据集的设计

2.1.1 线性可分数据集的设计

随机生成一组权值 w = ($w_0, w_1, ..., w_n$)，随机生成设定个数 d 的输入 $x = (x_1, x_2, ..., x_n)$，根据公式 $H(x) = \sum_{i=0}^{n} w_i x_i$ 计算输出，组合成包含 d 个样本的线性可分数据集，如图 2-1 为正负样本数都为 20 时的样本分布。

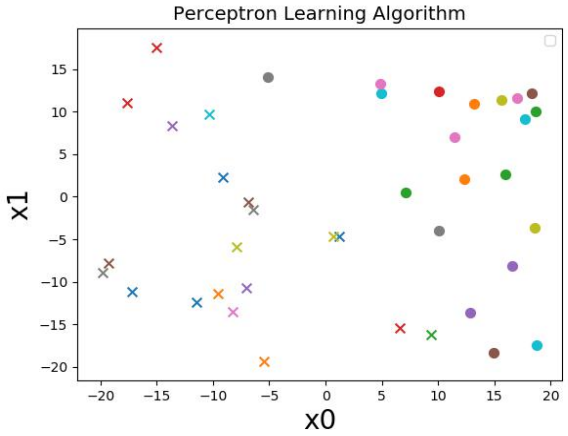

图 2-1 线性可分数据集

2.1.2 非线性可分数据集的设计

采用噪音的方式，将输入的线性可分数据集中的一定比例的样本的标签修改为相反标签，如图 2-2 为噪音比为 10%时的样本分布。

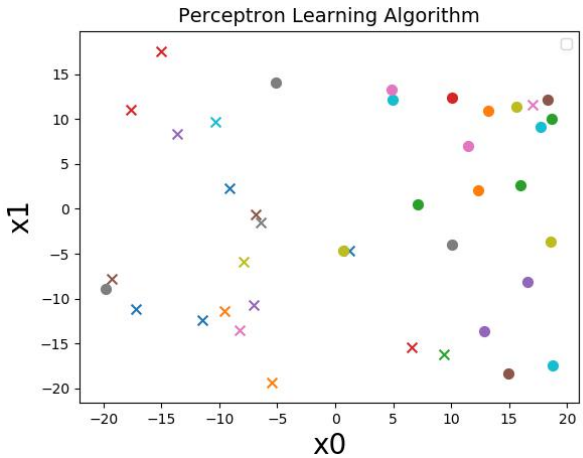

图 2-2 噪音后的非线性可分数据集

2.2 线性可分数据集上，不同条件下 PLA 与 Pocket PLA 的性能比较

2.2.1 单次实验

假设正负样本数目均为 100，特征维度为 2，Pocket 的最大迭代为一百次，进行实验。实验代码见图 4-1。

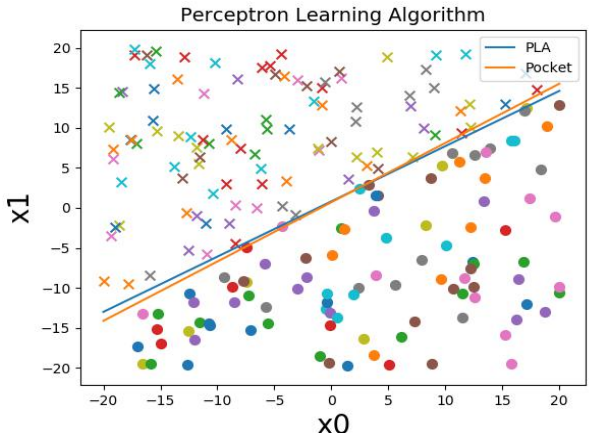

图 2-3 单次实验结果图

图 2-3 为样本分布与 PLA 算法和 PocketPLA 算法所求得的线性分类器，图 2-4 为本次实验的性能报告，从迭代次数、耗时的开销来看，PocketPLA 算法大约是 PLA 算法的 5 倍，PLA 算法的准确率为 100%，而 PocketPLA 的准确率为 99.5%。

```
C:\Users\万珂嘉\AppData\Local\Programs\Pytl
PLA esipode:  19
PLA time cost:  0.015989065170288086 s
PLA Hit:  1.0
Poket esipode:  99
Poket time cost:  0.05298924446105957 s
Poket Hit:  0.995
```

图 2-4 单次实验性能比较

2.2.2 重复实验

为了减小偶然性，在同样的设置下，进行 100 次的重复实验。从实验结果中选取了运行时间和准确率作为比较标准，PLA 平均运行时间为 0.02s，命中率为 100%，PocketPLA 平均运行时间为 0.05s，命中率为 99%，比较结果折线图如图 2-5，图 2-6 所示。

在重复实验中，发现 PLA 比 PocketPLA 的平均消耗要小，准确率要高。但是由于 PocketPLA 的最大迭代次数以及数据集的分布情况等因素的影响，在单次实验中，PLA 的消耗不一定比 PocketPLA 低，但是准确率一定大于等于 PocketPLA 的准确率。

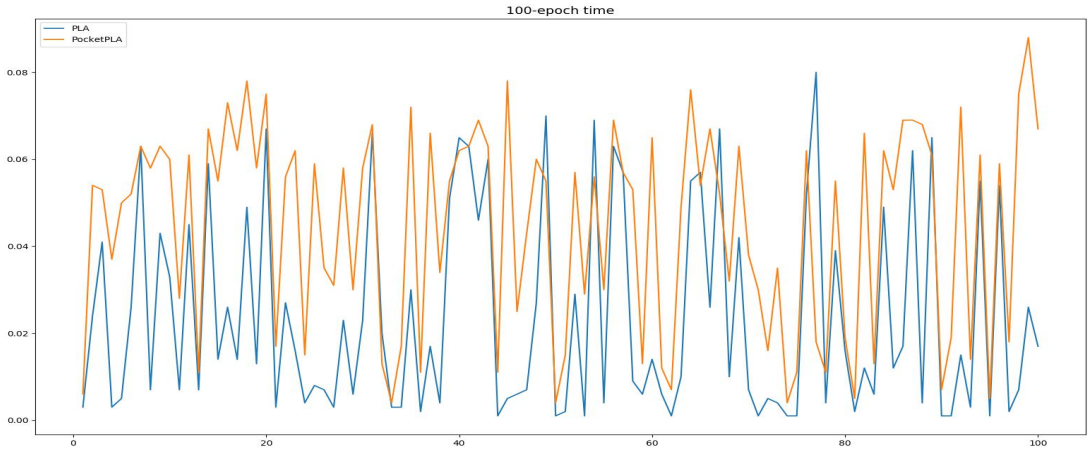

图 2-5 100 次重复实验运行时间比较图

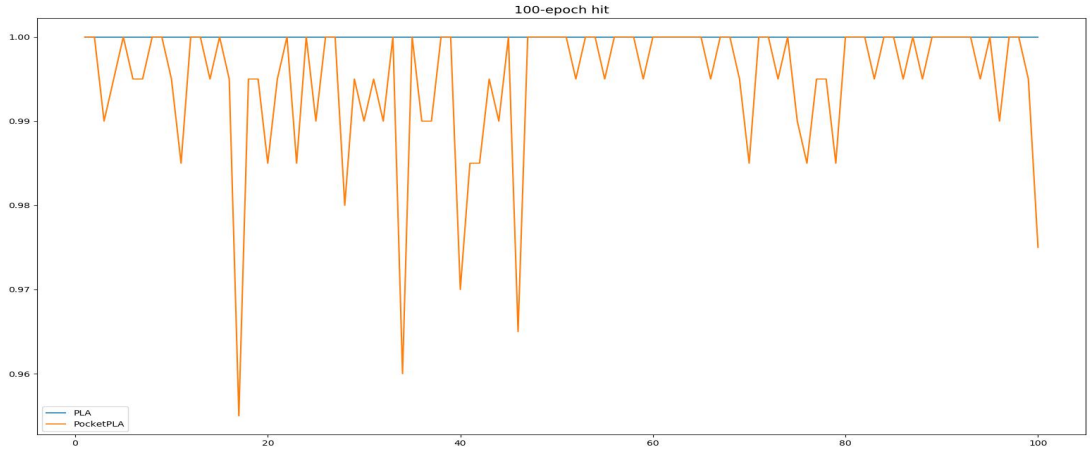

图 2-6 100 次重复实验命中率比较图

2.3 非线性可分数据集上，不同条件下 pocket PLA 的性能分析

2.3.1 单次实验

假设正负样本数目均为 100，特征维度为 2，噪声比为 10%，Pocket 的最大迭代为 100 次，进行实验。每次迭代随机选择一个错误点进行修正，修正后的分类线错误率与之前的分类线比较，若错误率较低，则选择修正后的分类线。继续进行下一次迭代。迭代完毕后，得到更新后的权重系数 w ，实验代码见图 4-2。

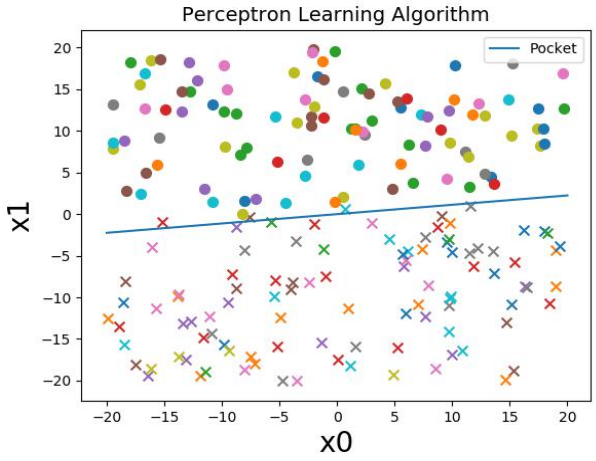

图 2-7 单次实验结果图

图 2-7 为样本分布与 PocketPLA 算法所求得的线性分类器，图 2-8 为本次实验的性能报告，从迭代次数、耗时的开销来看，PocketPLA 的准确率为 84.5%，耗时为 0.06s。

```
C:\Users\万珂嘉\AppData\Local\Programs\P
Poket iteration:  99
Poket time cost:  0.06496334075927734 s
Poket Hit:  0.845
```

图 2-8 单次实验性能报告

2.3.2 重复实验

为了减小偶然性，在同样的设置下，进行 100 次的重复实验。

从实验结果中选取了运行时间和准确率作为比较标准，PocketPLA 平均运行时间为 0.065s，命中率为 89%，比较结果折线图如图 2-9，图 2-10 所示。

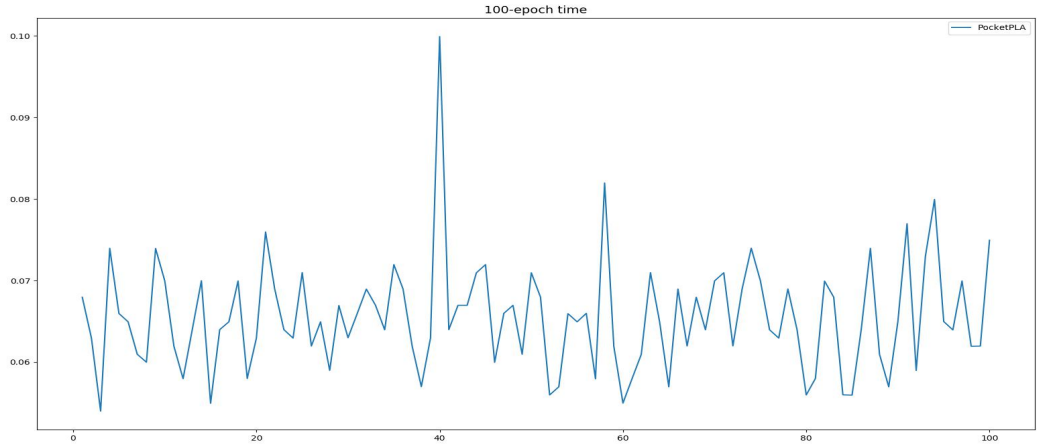

图 2-9 100 次重复实验运行时间图

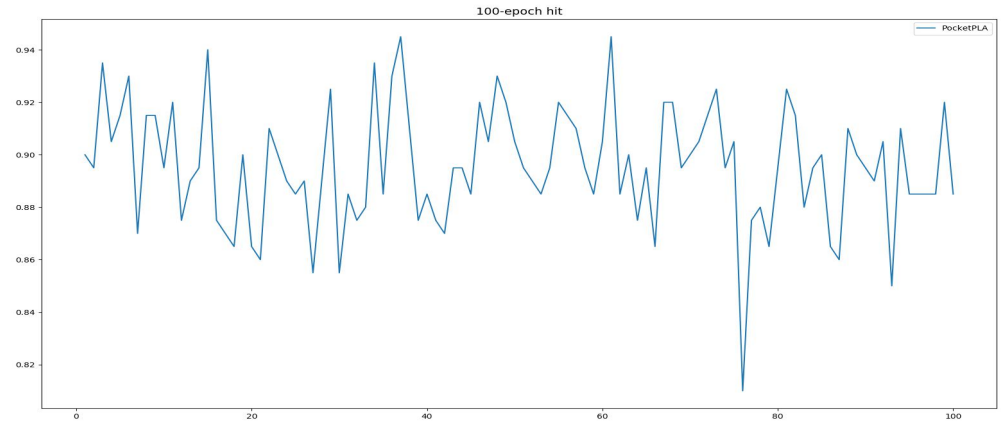

图 2-10 100 次重复实验命中率图

三、实验结论

PLA 的算法局限性比较大，一定要针对线性可分的数据，且只能用来做二元分类，如果是多元分类，还要进一步学习其他算法。

Pocket 算法可以解决 PLA 线性可分的未知性，但是自身缺点也比较大，如果设定的迭代字数太少，可能找到的解不够好，太多又可能给计算带来太大开销。假设数据一开始就是线性可分，那么这个算法找出来的未必是最好解，且时间花费也可能比较大。

四、附录

4.1 伪代码

```
initialize w(0) = 0.
for t = 0,1,2,....
        for i = 1,2,...,n
                w(t+1) = w(t) + y(i) * x(i);
        end
```

图 4-1 PLA 伪代码

```
for t = 1,2,3...  (number of iteration)
        w(t) = random of w ;
        for i = 1,2,3...n
                w(t+1) = w(t) + y(i) * x(i);
        end
        if(cal_error(w(t+1)) < cal_error(w_best)))
                w_best = w(t+1);

end
```

图 4-2 PocketPLA 伪代码

4.2 项目完整代码

https://github.com/dragonvanken/PLA.git