

模式识别实验二

实验人：叶平

实验内容：

用课本中第五章算法 3~12（除算法 8 外）在 Iris 数据集上进行分类。

(a)、对 class1 与 class3 采用算法 3~12（除算法 8 外），从每类中随机选取 25 个样本用于训练分类器，其余 25 个样本用于测试分类器，重复该过程多遍，估计分类精度的均值及方差；

(b)、对 class2 与 class3 采用相应的算法，从每类中随机选取 25 个样本用于训练分类器，其余 25 个样本用于测试分类器，重复该过程多遍（100 遍），估计分类精度的均值及方差；

实验结果：

运行试验二文件夹下 Res.m 文件，可以得到实验结果，每个算法循环 100 次，用时大概 5 分钟左右。

(a)、对 class1 与 class3 采用算法 3~12（除算法 8 外）的运行结果如下：

```
对class1与class3采用算法3~12的运行结果！
批处理感知器算法的均值为：0.999000    方差为0.000019
固定增量单样本感知器算法的均值为：1.000000 方差为0.000000
带裕量的变增量感知器算法的均值为：1.000000 方差为0.000000
批处理变增量感知器算法的均值为：0.999000    方差为0.000019
平衡winnow算法的均值为：1.000000    方差为0.000000
单样本裕量松弛算法的均值为：0.964600    方差为0.001890
LMS算法的均值为：0.955600    方差为0.019156
HK算法的均值为：1.000000    方差为0.000000
MHK算法的均值为：1.000000    方差为0.000000
```

由截图可以看出结果很好。除了 LMS 算法和单样本裕量松弛算法效果稍差，其它算法的分类效果基本达 100%。

(b)、对 class2 与 class3 采用相应的算法的运行结果如下：

对class2与class3采用算法3~12的运行结果！

批处理感知器算法的均值为：	0.947600	方差为0.001121
固定增量单样本感知器算法的均值为：	0.917400	方差为0.004975
带裕量的变增量感知器算法的均值为：	0.919800	方差为0.005063
批处理变增量感知器算法的均值为：	0.947800	方差为0.001114
平衡winnow算法的均值为：	1.000000	方差为0.000000
单样本裕量松弛算法的均值为：	0.795800	方差为0.001756
LMS算法的均值为：	0.660000	方差为0.025341
HK算法的均值为：	0.944600	方差为0.000855
MHK算法的均值为：	0.942800	方差为0.000849

由截图可以看出，还是 LMS 算法和单样本裕量松弛算法效果稍差，LMS 算法可能和选择的列向量 **b** 有关。算法中采用的是全一向量。单样本裕量松弛算法可能与选择的阈值有关。运行程序的时候可以发现，很多算法都达到了最大迭代次数，表明算法并没有收敛，这表示 class2 与 class3 应该是属于比较线性难分的。

此外，在实现算法过程中发现算法 11 有些问题，在书的 205 页，第 5 行，该行代码，即下图的第五行代码：

Algorithm 11 (Ho-Kashyap)

```

1 begin initialize  $\mathbf{a}, \mathbf{b}, \eta(\cdot) < 1$ , criteria  $b_{min}, k_{max}$ 
2   do  $k \leftarrow k + 1$ 
3      $\mathbf{e} \leftarrow \mathbf{Y}\mathbf{a} - \mathbf{b}$ 
4      $\mathbf{e}^+ \leftarrow 1/2(\mathbf{e} + \text{Abs}[\mathbf{e}])$ 
5      $\mathbf{b} \leftarrow \mathbf{a} + 2\eta(k)\mathbf{e}^+$ 
6      $\mathbf{a} \leftarrow \mathbf{Y}^+\mathbf{b}$ 
7     if  $\text{Abs}[\mathbf{e}] \leq b_{min}$  then return  $\mathbf{a}, \mathbf{b}$  and exit
8   until  $k = k_{max}$ 
9   Print NO SOLUTION FOUND
10 end

```

由于 **b** 和 **a** 的维数不一样，它们并不能相加，所以在我实现的算法中，我把 **a** 换成了 **b** 来实现的。