模式识别：作业 #3

201928014629008
牛李金梁

2019 年 11 月 7 日

第一部分：计算与证明

Question 1

首先，对样本进行规范化增广。根据设定，样本 $\boldsymbol{x}$ 增加齐次坐标后 $\boldsymbol{y}=(\boldsymbol{x}^T,1)^T$，之后将第二类样本 $\boldsymbol{y}$ 变为 $-\boldsymbol{y}$，便得到 4 个规范化增广样本：

第一类样本为 $(1,4,1)^T$ 和 $(2,3,1)^T$，第二类样本为 $(-4,-1,-1)^T$ 和 $(-3,-2,-1)^T$。

初始权向量为 $\boldsymbol{a}=(0,1,0)^T$，梯度更新步长 $\eta_k=1$，利用 Batch Perceptron 算法迭代：

1) $\boldsymbol{a}=(0,1,0)^T$，对于样本 $(-4,-1,-1)^T$ 和 $(-3,-2,-1)^T$，有 $\boldsymbol{a}^T\boldsymbol{y}<0$。对错分样本求和，$\sum_{y\in Y}\boldsymbol{y}=(-7,-3,-2)^T$。由更新准则 $\boldsymbol{a}_2=\boldsymbol{a}_1+\eta_k\sum_{y\in Y}\boldsymbol{y}=(-7,-2,-2)^T$。

2) $\boldsymbol{a}=(-7,-2,-2)^T$，对于样本 $(1,4,1)^T$ 和 $(2,3,1)^T$，有 $\boldsymbol{a}^T\boldsymbol{y}<0$。对错分样本求和，$\sum_{y\in Y}\boldsymbol{y}=(3,7,2)^T$。由更新准则 $\boldsymbol{a}_3=\boldsymbol{a}_2+\eta_k\sum_{y\in Y}\boldsymbol{y}=(-4,5,0)^T$。

3) $\boldsymbol{a}=(-4,5,0)^T$，对于所有样本有 $\boldsymbol{a}^T\boldsymbol{y}>0$。无错分样本。

所以线性判别函数 $g(\boldsymbol{y})=\boldsymbol{a}^T\boldsymbol{y}$ 的权向量为 $\boldsymbol{a}=(-4,5,0)^T$。

Question 2

根据 one-vs-all 的规则，本问题需设计一个线性机器，将样本空间分为 3 个决策区域 R_1，R_2，R_3，分别对应决策为 3 个类别 ω_1，ω_2，ω_3。

* R_1 和 R_2 的分界面 H_{12} 为 $g_1(\boldsymbol{x})-g_2(\boldsymbol{x})=-2x_1+1=0$。
* R_1 和 R_3 的分界面 H_{13} 为 $g_1(\boldsymbol{x})-g_3(\boldsymbol{x})=-x_1+2x_2=0$。
* R_2 和 R_3 的分界面 H_{23} 为 $g_2(\boldsymbol{x})-g_3(\boldsymbol{x})=x_1+2x_2-1=0$。

画出 H_{12}、H_{13}、H_{23}3 个分界面，并根据决策规则确定 R_1、R_2、R_3 的位置，如图：

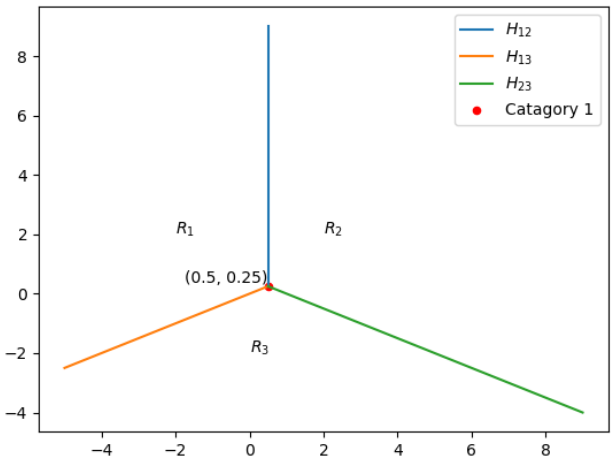

图 1：决策区域示意图

可见，决策区域的分界面相交于 $(0.5, 0.25)$，该问题不存在不确定区域。同样地，使用该决策规则构造线性机器，实际上是将样本分类到判别函数最大的那个类别中，因此除边界上的点外不会存在不确定区域。

第二部分：计算机编程

环境为 python3.7
依赖库：numpy、matplotlib
运行方式：直接在 cmd 中执行相应 py 文件

Question 1

使用 Batch perception 算法训练判别函数，首先要对样本进行规范化增广，然后利用这些样本训练线性判别函数。

具体的训练过程需要将每一步错分的点相加并乘以步长来修正权向量 $\boldsymbol{a}$，如此迭代直至完全分开或修正值很小（以应对线性不可分的情况）。

实验中，取初始权向量 $\boldsymbol{a} = \mathbf{0}$，$\eta = 1$，$\theta = 0.01$，结果如下图：

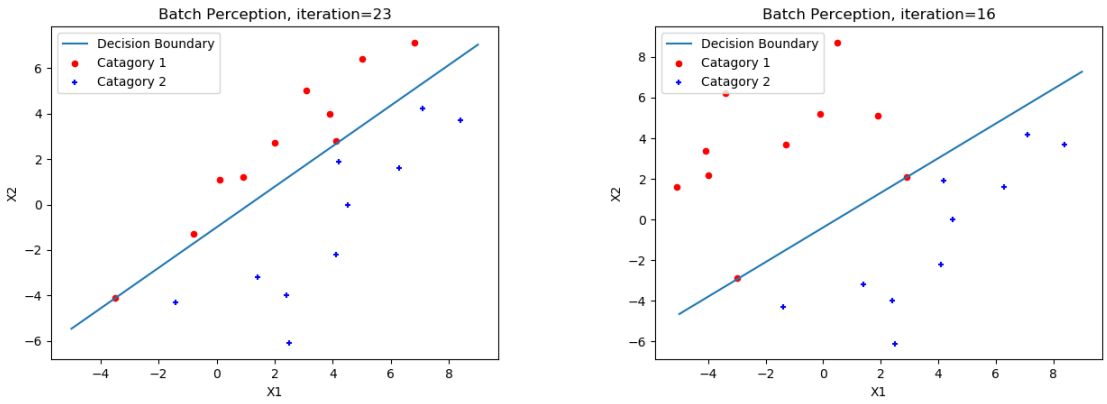

图 2: Batch perception 算法结果

问题 (a) 中分类 ω_1 和 ω_2 的结果如左图所示，判别函数左上判别为 ω_1，右下判别为 ω_2，收敛步数为 23。

问题 (b) 中分类 ω_3 和 ω_2 的结果如右图所示，判别函数左上判别为 ω_3，右下判别为 ω_2，收敛步数为 16。

Question 2

Ho-Kashyap 算法通过最小化误差来优化 b，并通过 $a = Y^{+}b$ 计算对应的 a，进而得到一组 a, b 使 $Ya = b > 0$。

搜索到最优解是比较困难的，需要设定最大迭代次数和收敛误差。本程序设定步长为 0.8，最大迭代次数为 2000，收敛误差为 0.01，若所有样本的误差小于 b_{min}（设定为 0.001）或迭代 2000 次，则停止迭代。

ω_1 和 ω_3 的分类结果如下图中左图所示，决策面左上方为 ω_3，右下方为 ω_1。可以看到有 2 个分类错误点，并且输出了 No solution found，说明该问题是线性不可分的。

ω_2 和 ω_4 的分类结果如下图中右图所示，决策面左下方为 ω_4，右上方为 ω_2。可以看出该问题是线性可分的，迭代次数为 899。

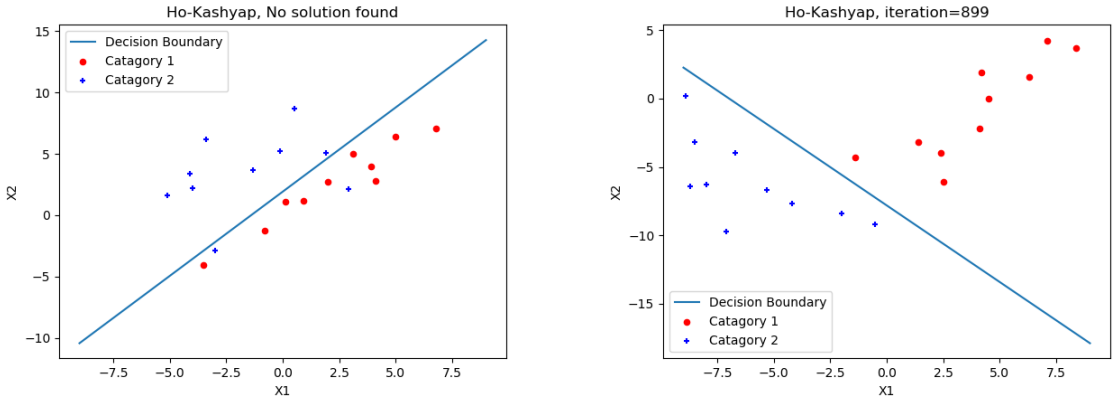

图 3: Ho-Kashyap 算法结果

Question 3

使用 MSE 多类扩展方法对 4 个类别进行分类。

具体的算法实现中，主要是要构造 $\boldsymbol{Y}$，并利用 $\hat{W} = (\hat{X}\hat{X}^T + \lambda I)^{-1}\hat{X}Y^T$ 来计算 $\hat{W}$ 确定分类区域，程序中设定 $\lambda = 0.01$。

根据决策规则 $x \in \omega_i, g_i(x) = \max g_j(x), j = 1, \ldots, 4$ 可以绘制出决策区域（本程序直接利用点采样确定了决策区域，而非计算线性判别函数），

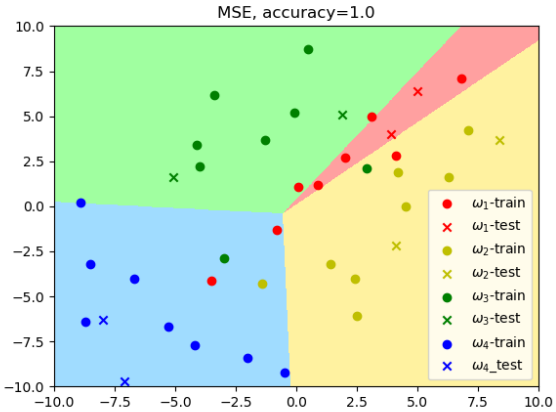

图 4: 决策区域示意图

图中红色为 ω_1 决策区域，黄色为 ω_2 决策区域，绿色为 ω_3 决策区域，蓝色为 ω_4 决策区域。并以实心圆标记训练样本，x 标记测试样本，通过统计可以得到测试样本的分类正确率为 100%。