清华大学电子工程系

媒体与认知 课堂 2

2021-2022 学年春季学期

作业 4

郭中贺 2022 年 5 月 1 日

理论部分

1 单选题（15 分）

1.1 **D**

1.2 **C**

1.3 **B**

1.4 **A**

1.5 **D**

2 计算题（15 分）

2.1 **假设邮件粗略分为垃圾邮件和正常邮件，且存在一种垃圾邮件的检测方法，其中垃圾邮件被正确检测的概率为 a，正常邮件被误判为垃圾邮件的概率为 b。针对某一邮箱，所有邮件中垃圾邮件占的比例为 c，如果某封邮件被判定为垃圾邮件，根据贝叶斯定理，这封邮件是垃圾邮件的概率是多少？(提示：全概率公式 $P(Y)=\sum_{i=1}^{N}P(Y|X_i)P(X_i)$)**

2.1. 设事件A：这封邮件是垃圾邮件 c
B：是正常邮件 (1-c)
C：垃圾邮件被正确检测 a
D：正常邮件被误判 b
Y：被判定为垃圾邮件

$\therefore P(Y)=P(Y|A)P(A)+P(Y|B)\cdot P(B)$
$=P(C)\cdot P(A)+P(D)\cdot P(B)$
$=ac+b(1-c)$
$=ac-bc+b$

$P(A|Y)=\frac{P(AY)}{P(Y)}=\frac{P(Y|A)\cdot P(A)}{P(Y)}=\frac{ac}{ac-bc+b}$

2.2 给定样本集，其均值为 $\mu = [1, 2]^T$，样本协方差矩阵为 C，且已知 $CU = U\lambda$。

其中 $U = \begin{bmatrix} 0.5 & -0.4 \\ 0.5 & 0.4 \end{bmatrix}$，$\lambda = \begin{bmatrix} 10.7 & 0 \\ 0 & 0.4 \end{bmatrix}$。

试用主成分分析 PCA 将样本 $x = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ 变换至一维。

(提示: 样本数据应减去均值；特征向量应归一化)

2.2. $\because U = \begin{bmatrix} 0.5 & -0.4 \\ 0.5 & 0.4 \end{bmatrix}$ $CU = U\lambda$ $\therefore C = U\lambda U^{-1}$

对 C 进行特征值分解 $\lambda_1 = 10.7$ 特征向量 $W_1 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$ 归一化为 $\frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\lambda_2 = 0.4$ W_2 归一化为 $\frac{\sqrt{2}}{2}\begin{bmatrix} -1 \\ 1 \end{bmatrix}$

投影向量 $W = W_1 = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\therefore x = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ 变换到一维是 $W^T x = 2\sqrt{2} = 2.828$

2.3 设有两类正态分布的样本集，第一类均值为 $\mu_1 = [1, 0]^T$，第二类均值为 $\mu_2 = [0, -1]^T$。两类样本集的协方差矩阵和出现的先验概率都相等：$\Sigma_1 = \Sigma_2 = \Sigma = \begin{bmatrix} 0.7 & 0.2 \\ 0.2 & 1.2 \end{bmatrix}$，$p(\omega_1) = p(\omega_2)$。试计算分类界面，并对特征向量 $x = [0.2, 0.5]^T$ 分类。

2.3. $\because p(w_1) = p(w_2)$ $\Sigma_1 = \Sigma_2 = \Sigma$

$\therefore$ 判决函数 $g_{LDF1}(x) = (\Sigma^{-1}\mu_1)^T x - \frac{1}{2}\mu_1^T \Sigma^{-1}\mu_1 = [1.5 \ 0.25]x + 0.75$

$g_{LDF2}(x) = (\Sigma^{-1}\mu_2)^T x - \frac{1}{2}\mu_2^T \Sigma^{-1}\mu_2 = \begin{bmatrix} 0.25 \\ -2.875 \end{bmatrix}^T x + 2.4375$

分界面 $g_{LDF1}(x) - g_{LDF2}(x) = 0 \Rightarrow 1.25x_1 + 2.625x_2 + 3.3125 = 0$

$g_{LDF1}(x) = 1.175$ $g_{LDF2}(x) = 2.05$

$\therefore [0.2, 0.5]^T$ 分类为 类别 1

编程部分

3 编程作业报告

3.1 实现 hinge loss 模拟支持向量机并运行自动评判程序

经过代码补全之后，运行命令：python check.py，发现测试成功，相关结果截图见图 1 所示：

```
(meiren) D:\media-and-cognition\hw4>python check.py
Linear successully tested!
Hinge successfully tested!
SVM_HINGE successfully tested!
```

图 1

3.2 Hinge loss 模拟 SVM 的训练及验证

使用 hinge loss 模拟 SVM，按缺省参数训练和验证，执行命令：python classify_hw.py –mode hinge 训练 200 轮后，loss=58.385，同时在验证集上的准确率为 92.8%。下图 2 左右分别是这种训练情况下的 loss 变化曲线以及训练集上的特征点分布图，图 3 为验证集上的特征点分布图。

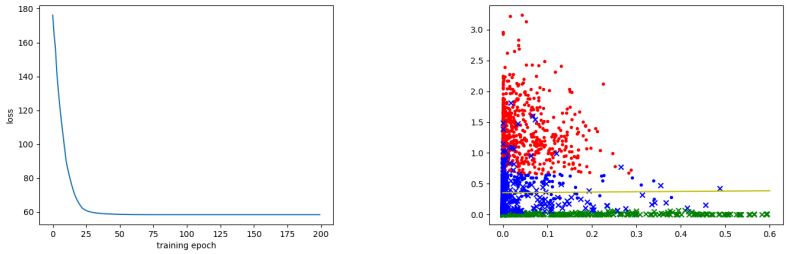


图 2

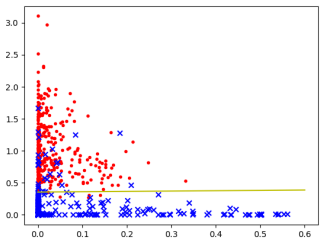


图 3

可以发现，在验证集上分类也并不是完全正确，但是总体来讲，分类边界能够将大多数的两类点分开。

3.3 使用 libsvm 库对数据集进行分类

运行使用 libsvm 库实现分类的命令: python classify_hw.py –mode baseline，图 4 左右分别是这种训练情况下训练集和验证集上的特征点分布图。图 5 为这种训练情况下的 cmd 输出结果

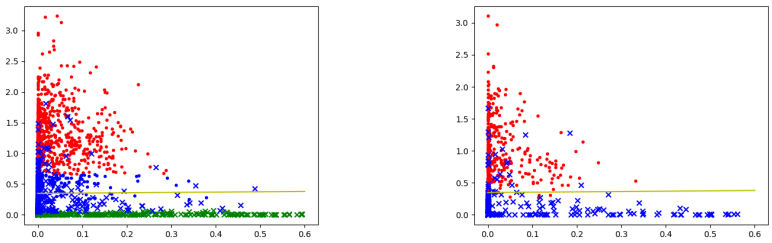


图 4

```
(meiren) D:\media-and-cognition\hw4>python classify_hw.py --mode baseline
*
optimization finished, #iter = 380
nu = 0.266243
obj = -58.385376, rho = 1.178906
nSV = 641, nBSV = 638
Total nSV = 641
Accuracy = 92.75% (742/800) (classification)
```

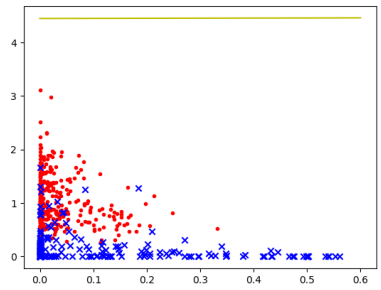
图 5

比较 Hinge loss 模拟 SVM 和 libsvm 库函数的训练结果，发现二者在验证集上的准确率几乎相同，都为 92.8%，通过比较二者的特征点分布图，也很难看出差别，说明 Hinge loss 模拟很成功。

3.4 调整正则化系数 C，体会不同的 C 对分类效果的影响

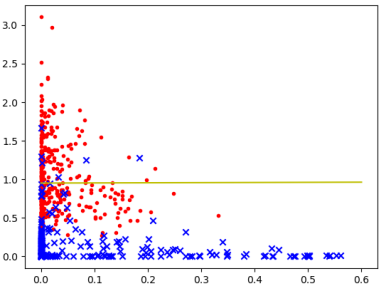
3.4.1 C=0.0001

运行命令 python classify_hw.py –mode hinge –C 0.0001，根据命令行输出结果，得知最后一轮训练的 loss 为 50.234，在验证集上的准确率 50.0%，特征点分布图如下:



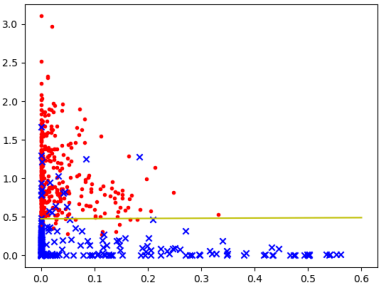
3.4.2 C=0.001

运行命令 python classify_hw.py –mode hinge –C 0.001，根据命令行输出结果，得知最后一轮训练的 loss 为 1.799，在验证集上的准确率 68.6%，特征点分布图如下：



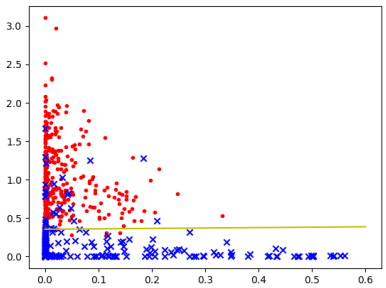
3.4.3 C=0.01

运行命令 python classify_hw.py –mode hinge –C 0.01，根据命令行输出结果，得知最后一轮训练的 loss 为 8.787，在验证集上的准确率 91.4%，特征点分布图如下：



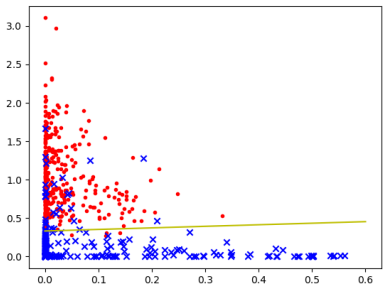
3.4.4 C=0.01

运行命令 python classify_hw.py –mode hinge –C 0.1，根据命令行输出结果，得知最后一轮训练的 loss 为 58.385，在验证集上的准确率 92.8%，特征点分布图如下：



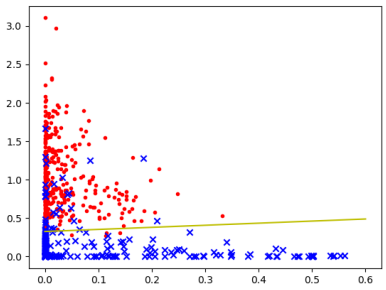
3.4.5 C=1.0

运行命令 python classify_hw.py –mode hinge –C 1.0，根据命令行输出结果，得知最后一轮训练的 loss 为 525.903，在验证集上的准确率 92.4%，特征点分布图如下：



3.4.6 C=10.0

运行命令 python classify_hw.py –mode hinge –C 10.0，根据命令行输出结果，得知最后一轮训练的 loss 为 5183.176，在验证集上的准确率 92.4%，特征点分布图如下：



3.4.7 比较结果

由于利用库函数进行训练的结果和使用 linge loss 模拟的结果基本一致，因此此处不再放出相应的结果，只进行 C 的取值对训练结果的分析。可以明显看出，C 越大，训练的 loss 就越大，并且和 loss 与 C 成正比。由 loss 函数的计算公式，确实可以近似认为 loss 和 C 成正比。另一方面，当 C 过小时，准确率大幅度下降，比如当 C=0.0001 时，准确率只有一半，特征点分布图中分类界面也显然不正确，说明 C 过小时，该训练模式无法执行分类任务。当 C 增大时，准确率呈现增大的趋势，但是当 C 过大时，准确率有小幅度下降。

3.5 本次作业遇到的问题及解决方法

无。

3.6 对本次作业的意见及建议

本次作业也较为简单，通过代码注释中的一步一步教程，我们能很快了解下一步该做什么，该如何实现。