山东大学 SHANDONG UNIVERSITY

山东大学机器学习课程

实验报告

——实验五：线性分类器的设计与实现

姓名：刘梦源

学院：计算机科学与技术学院

班级：计算机 14.4

学号：201400301007

一、实验目的：

（1）了解线性分类器的原理和思想。
（2）用梯度下降和牛顿法实现线性分类器估计器中准则函数极小值的求解
（3）比较两种方法的异同，评价其优缺点
（4）体会学习率对学习效果与学习速度的影响，求出不同阈值下无法收敛的最小学习率

二、实验环境：

（1）硬件环境：
英特尔® 酷睿™ i7-7500U 处理器
512 GB PCIe® NVMe™ M.2 SSD
8 GB LPDDR3-1866 SDRAM
（2）软件环境：
Windows10 家庭版 64 位操作系统
Matlab R2016b

三、实验内容

实验数据如下

样本	W1		W3	
1	0.1	1.1	-3.0	-2.9
2	6.8	7.1	0.5	8.7
3	-3.5	-4.1	2.9	2.1
4	2.0	2.7	-0.1	5.2
5	4.1	2.8	-4.0	2.2
6	3.1	5.0	-1.3	3.7
7	-0.8	-1.3	-3.4	6.2
8	0.9	1.2	-4.1	3.4
9	5.0	6.4	-5.1	1.6
10	3.9	4.0	1.9	5.1

（a） 用基本梯度下降算法和牛顿法对二维数据给出 w1 和 w3 的判别。对梯度下降法取 $\eta(k)=0.1$。画出准则函数关于迭代次数的变化曲线。

（b） 估计基本梯度下降法和牛顿法的数学运算量。

（c） 画出收敛时间-学习率曲线，求出无法收敛的最小学习率。

首先设计出合适的准则函数，本题设计的准则函数为

$$J(a)=\sum_{y\in\gamma}\frac{1}{2}(ay-b)^{2} \tag{1}$$

其中，a 为权重向量与偏置的增广向量，为$[w_0,w_1,w_2]$，y 为特征值的增广项链，为$[1,x_1,x_2]$，b 为期望，w1 类样本期望设为 1，w3 类样本期望设为-1，γ 为被错分的样本集。

然后设计判别函数，本题的判别函数设为

$$F(a,y)=a^{T}y \tag{2}$$

当$a^{T}y$ >0 时，判别为第一类，当$a^{T}y$ <0 时，判别为第三类。

接下来对准则函数进行求导，

$$\nabla J(a)=(ay-b)y \tag{3}$$

对基本梯度下降法来说，先后两次迭代的待求权向量 a(k)，a(k+1)满足下面的关系：

$$a(k+1)=a(k)-\eta(k)\nabla J(a(k)) \tag{4}$$

联合式(1)-(4),初始化权向量之后，不断计算准则函数和判别函数，通过准则函数下降的方向向量来更新权向量，通过调整阈值得到取得近似准则函数极小值下的权向量，完成学习。

对于牛顿法来说，只需将学习率替换成赫森矩阵的逆即可，对(3)再次求导，得

$$H=\nabla J(a)\frac{dJ}{da}=y.y^{T} \tag{5}$$

(4)式也被替代为

$$a(k+1)=a(k)-H^{-1}\nabla J \tag{6}$$

四、实验结果

（a）设计梯度下降算法后，带入$\eta(k)$=0.1，设定阈值为J(a)<1时跳出循环，发现无法跳出。初步怀疑学习率为0.1时准则函数不收敛。进一步研究，规定迭代次数，迭代次数达到150后强行终止程序。做出准则函数随迭代次数的变化曲线。发现效果十分差。图像如下。

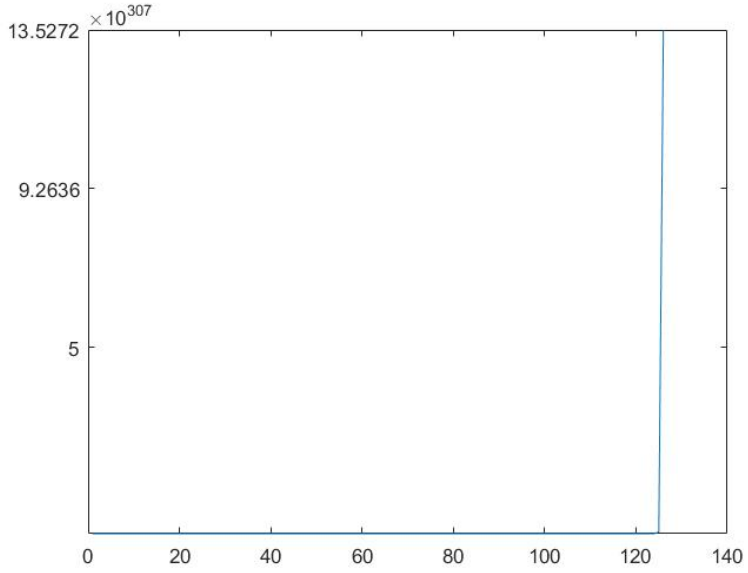


图1. $\eta(k)$=0.1时准则函数随迭代次数的变化曲线

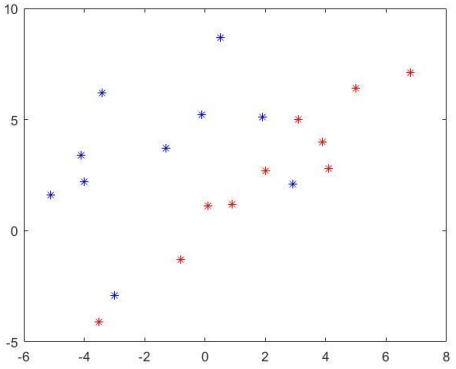


图2. $\eta(k)$=0.1时分类前

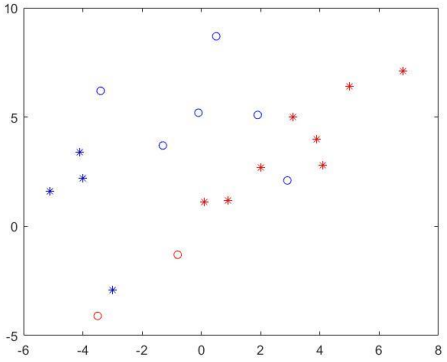


图3. $\eta(k)$=0.1时分类后

通过图1-3可以看出，不仅损失函数没有收敛，反而呈现一个越来越大的趋势，而且样本的分类结果也很差（蓝色为w1，红色为w2，*为分类错误（分类前认为全部分类错误），o表示分类正确）。

我们认为，0.1这个学习率显然是太大的，达不到分类的要求，损失函数也不收敛。

打印出损失函数，如图4，损失函数不仅没有变小而且是急速增长的，基本上呈现指数增长。这印证了我们的猜测，0.1是一个过大的学习率，在这个学习率下，准则函数并不收敛。

	1
1	206.0550
2	5.8095e+04
3	1.5476e+07
4	4.5360e+09
5	1.2096e+12
6	3.5457e+14
7	9.4550e+16
8	2.7716e+19
9	7.3907e+21
10	2.1665e+24
11	5.7771e+26
12	1.6935e+29
13	4.5158e+31
14	1.3237e+34
15	3.5299e+36
16	1.0347e+39
17	2.7592e+41

图 4. 准则（损失）函数随迭代次数的变化

更改学习率，让其变成一个合适的数值。作图如下。（图5、6、7跳出循环是因为设定步数100步，而不是因为收敛到阈值之内，图8、图9在阈值范围内完成收敛）

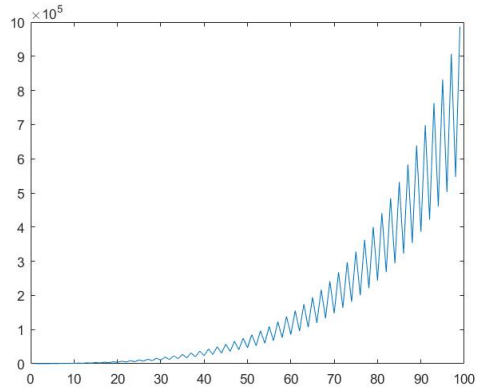


图 5 学习率为 0.013

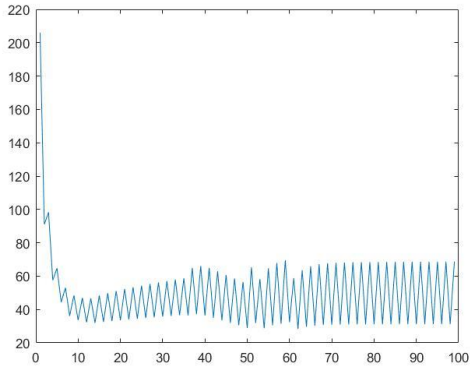


图 6 学习率为 0.01

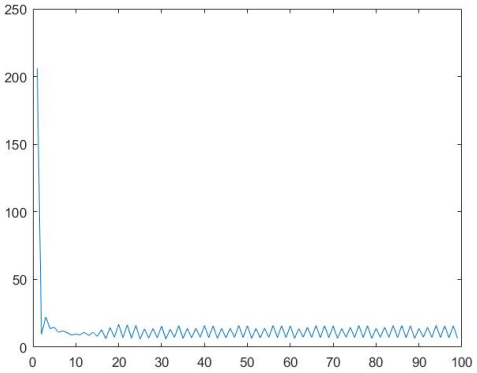


图7 学习率为0.007

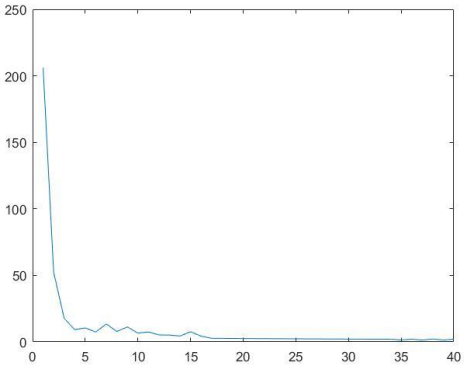


图8 学习率为0.004

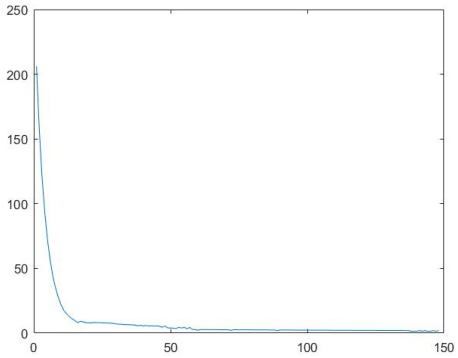
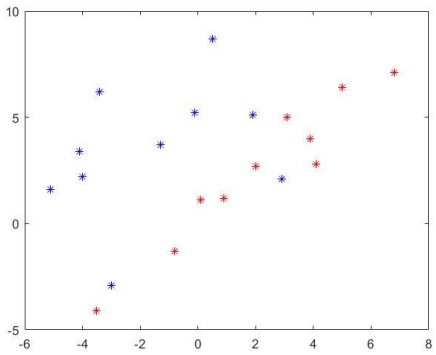


图9 学习率为0.001

这几组中，我们认为0.004（图8）就是比较好的学习率：图5-7来说因为学习率过大而不收敛，图9来说因为学习率过小而导致收敛过慢。

观察学习率为0.004时的分类情况。



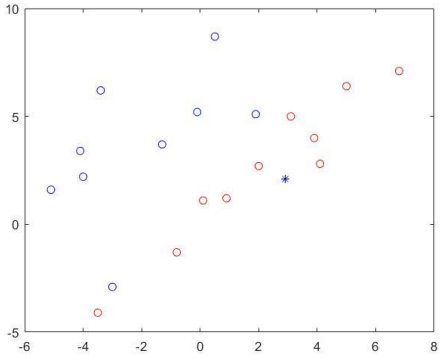


图10. $\eta(k)$ =0.004时分类前

图11. $\eta(k)$ =0.004时分类后

发现此时基本梯度法分类情况极好，只有w3类中的第三个点无法分类，正确率高达95%。

对于牛顿法来说，比较基本梯度学习法改变的也仅仅是学习率，作图如下所示。只用了20步就完成了阈值为2的收敛。

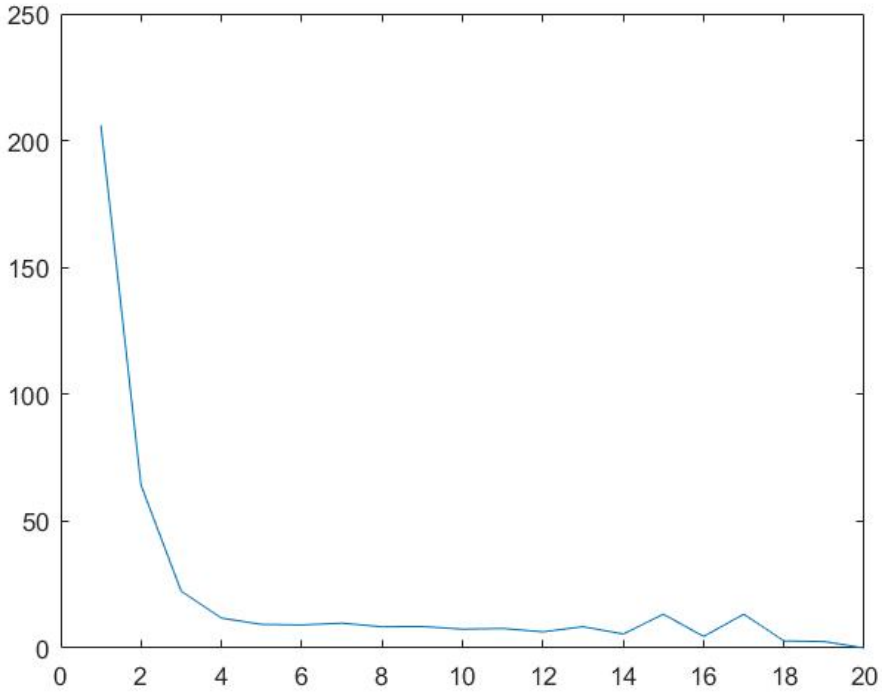


图 12 牛顿法准则函数随收敛次数变化

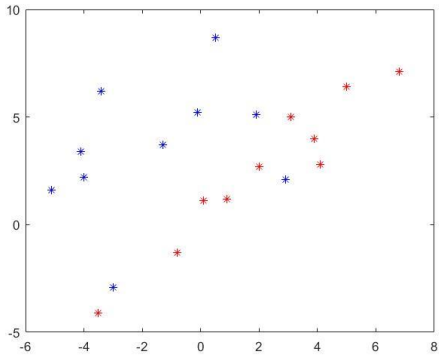


图13 牛顿法分类前

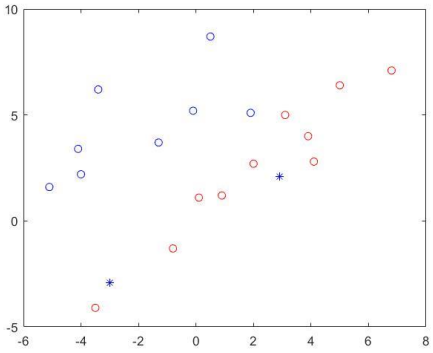


图14 牛顿法分类后

（b）

表面上来看，牛顿法提供了更好的学习率，而不用我们认为干预造成的不恰当的学习率。但牛顿法就是完美的吗？当然不是。

按照牛顿法，在这个实验中，如果设置阈值为2，实验结果就如图12-图14所示，虽然20步迭代呈现出一个较好的运算量，但是正确率却只有90%，达不到我们用最佳的基本梯度法达到的实验效果（学习率为0.004时，阈值为1，实验结果如图8，10，11所示）；我怀疑是因为阈值为2过大导致，于是改小了阈值，发现当阈值改小时，却造成了另一种危机：无法收敛到满足阈值的解，导致程序无法跳出循环。我的实验证明，一旦精度很小的时候，准则函数泰勒展开的极小项就不能忽略，这种情况下，牛顿法无法满足我们对解的高精度要求。

其次，这个牛顿法的程序造成了多余的系统开销，那就是我们计算赫森矩阵的时候，正如课本P226所说，每次递归时计算H逆矩阵所需的时间可轻易地讲牛顿法带来的好处抵消。并且，如果赫森矩阵为奇异矩阵时也就不能用牛顿法。

综上所述，我认为，牛顿法迭代次数虽然通常较少，但数学运算量不一定比基本迭代法更少（尤其是当基本迭代法的学习率设置较为恰当时），而且还有他自己的局限性。所以这两个方法的使用还需具体问题具体分析。

（c）本小题用收敛次数代替收敛时间进行实验分析

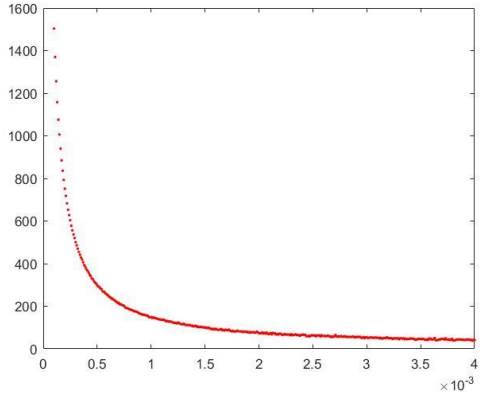


图15 阈值为1时收敛次数-学习率曲线

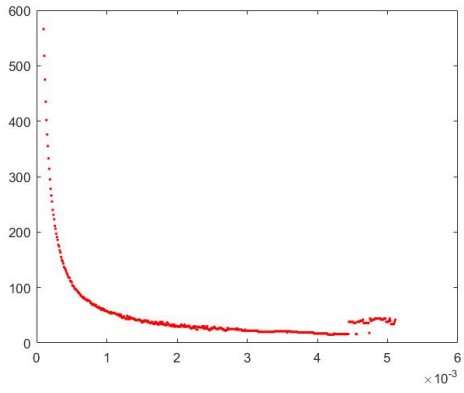


图16 阈值为3时收敛次数-学习率曲线

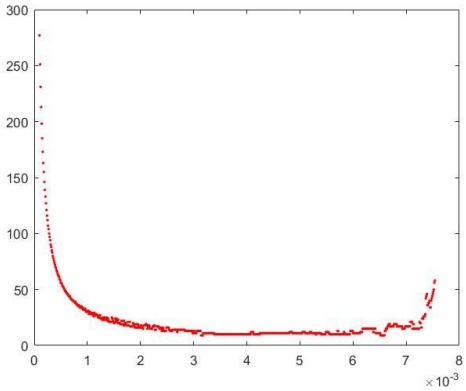


图17 阈值为7时收敛次数-学习率曲线

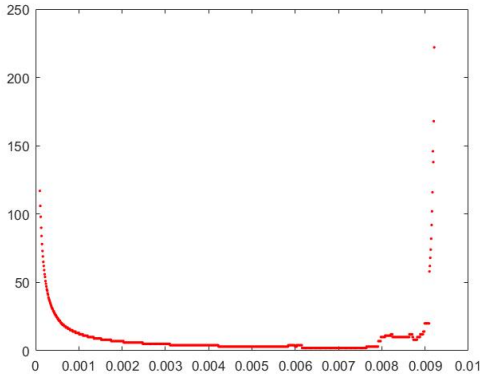


图18 阈值为10时收敛次数-学习率曲线

实验结果十分直观，图15中，我们采取了较小阈值，学习率越大，迭代次数越小；而当我们增大阈值时，出现了另外的现象，学习率增大的时候，迭代次数反而更大，图17，图18最为明显。究其原因，课本P253的**图6-16**很好的解释这种情况，这里不再赘述。

不同的阈值下，最小无法收敛学习率是不同的，以阈值为3时为例，最小无法收敛学习率在0.005附近。

至此，实验较为完美地得到了理想的结果，并分析出了原因。