支持向量机实验报告

PB20010429 侯相龙

2022 年 10 月 30 日

1 实验内容

完成类 SVM1 和 SVM2 ，并且使用不同的算法去寻找支持向量机的解

2 实验设备和环境

- **实验设备：**

设备名称：LAPTOP-9J92NDCJ
处理器：Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
RAM：16.0 GB

- **实验环境：**

PyCharm 2021.3.2

3 实验方法与步骤

主要介绍一下在 SMO 算法的理论和在实际处理的过程中遇到的问题及解决办法

3.1 SMO 算法（软间隔）

（部分参考《统计学习方法（李航著）》》

方法理论

在将原问题转化为对偶问题求解之后，若结果不满足 KKT 条件，我们选择两个变量，固定其他变量，对这两个变量进行优化通过解析方法进行二次规划。故 SMO 算法包括两部分。其一，求解两个变量的二次规划的解析方法，实际上这并不复杂，通过消元，转化为单变量在指定定义域上的二次函数，这是很好求解的，只需要注意定义域的截断即可；其二，选择变量的启发式方法，第一个变量 x_i 我们选择违反 KKT 条件最严重的样本点，第二个变量 x_j 我们通常选择 $|E_i - E_j|$ 最大者，其中 $E_i = g(x_i) - y_i$ 为当前预测值与真实值得偏差。

问题及解决办法

主要问题都出现在变量选取的过程中。

i) 按照上面介绍的方法进行变量选取的过程中，可能会出现以下情况：选取变量，更新阈值使得新的两个变量都满足了 KKT 条件。但是函数值并没有优化。我们有如下解决办法：假设能优化，得到优化后的值进行判断：如果计算得到的 $\alpha_i^{(new)}, \alpha_j^{(new)}$ 和优化前的 $\alpha_i^{(old)}$ $\alpha_j^{(old)}$ 相同，则不执行优化过程而重新选择变量。

ii) 在重新寻找新的第二个优化变量时，可采取下面的启发式规则：遍历边界上的支持向量点依次作为第二个变量试用，若仍然没有足够下降选择其他值；若仍找不到，则修改第一个选择的变量，重新寻找。

iii) 完全按照书上的方法寻找变量，迭代会出现循环某两个变量，耗时长且最后的结果也不是很好。我略微做了一些改动，对于已经找到的第一个变量 α_i, 在后面的选择中不再重复选择，而去寻找其他的变量。

3.2 梯度下降法

（参考网页链接）

该算法基于一个很简单的想法：为了优化我们的目标函数 $\frac{1}{2}||w||^2$，使用梯度下降法即可。但是，与无约束点优化不同的是，我们针对满足约束条件的点和违反约束条件的点分别优化。对于满足约束条件的点，使用梯度下降即可；对于不满足约束的点，我们使其向满足约束条件的方向进行优化。

4 实验结果

本实验采用留出法用上述两种方法针对不同维度进行了多次测试，下面列举三组实验的结果。

4.1 dim=20,num=8000 分类结果:

```
错标率： 0.038
accuracy: 0.9254166666666667
Time: 266.4710738658905 s
```

图 1: Test1

```
错标率： 0.038
accuracy: 0.94125
Time: 21.725547313690186 s
```

图 2: Test2

4.2 dim=30,num=6000 分类结果:

```
错标率： 0.039142857142857146
accuracy: 0.9414285714285714
Time: 167.44898009300232 s
```

图 3: Test1

```
错标率： 0.039142857142857146
accuracy: 0.9504761904761905
Time: 33.757142543792725 s
```

图 4: Test2

4.3 dim=2,num=8000:

对于二维的分类，我们采取了可视化处理。

- 分类结果：

```
错标率： 0.039
accuracy: 0.9475
Time: 128.10960292816162 s
```

图 5: Test1

```
错标率： 0.039
accuracy: 0.95625
Time: 63.025753021240234 s
```

图 6: Test2

- 训练点及分类超平面图：

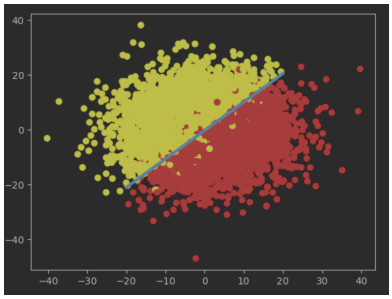

图 7: SMO 算法

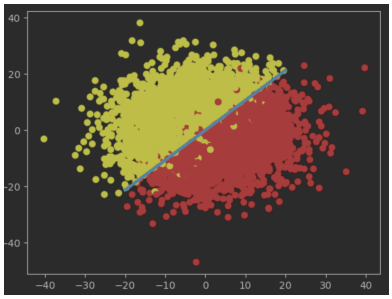

图 8: 梯度下降法

4.4 结果比较

就实验结果来看，朴素的梯度下降法比 SMO 方法的正确率更高且消耗时间更少。一方面可能的原因是问题本身是简单线性可分的，没有涉及到核技巧，简单的方法在时间成本较低、迭代次数较少时亦可以处理；另一方面，SMO 方法可能因迭代次数不够导致正确率略低于梯度下降法；在具体编写 SMO 方法时，所遇到的问题可能还有更好的解决办法，可以降低时间复杂度提高准确率。