贾成君 2014011552 自46

1. **解：**
2. 线性核，B二次多项式核，C三次多项式核，D径向基()，E径向基()，F径向基()。分别对应的图形为：

A-c,B-b,C-f;D-d,E-a,F-e。

因为径向基核的支持向量应该都距离分界面较近，而且越大，支持向量越多，且具有一定的包含关系（较大的对应的径向基核函数对应的支持向量包含较小的对应的支持向量）。线性核对应的分界面应该是一条直线；多项式核函数对应的支持向量可能出现在较远处。

1. 对应该数据集，应该使用不太大的径向基核函数较合理。因为明显两组数据之间有比较清晰明显的分界——没有混杂的情况；同时为了让分界面比较连续，应该使用不太大的径向基核函数。
2. **解：**

根据之前的统计，可知’4’和’9’是最容易进行混淆的两个数字。使用Matlab R2016b分类工具箱，进行支持向量机分类（把4的结果视为’-1’,9的判断结果视为’1’），可得，各种支持向量机的训练结果如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SVM** | **Accuracy** | **Training Time/s** | **"-1->-1"** | **"-1->1"** | **"1->-1"** | **"1->1"** |
| linear | 96.2% | 157.23 | 1125 | 43 | 46 | 1144 |
| Quadratic | 97.6% | 137.86 | 1142 | 26 | 31 | 1159 |
| Cubic | 98.0% | 239.04 | 1146 | 22 | 26 | 1164 |
| Fine Gaussian | 51.9% | 582.91 | 34 | 1134 | 0 | 1190 |
| Medium Gaussian | 97.0% | 372.46 | 1144 | 24 | 47 | 1143 |
| Coarse Gaussian | 95.7% | 534.87 | 1117 | 51 | 50 | 1140 |

其中’-1->1’的意思为：把真实为’-1’类型的样本分为了’-1’类型。

可以看出，除了Fine Gaussian支持向量机的分类结果不理想（假阳性非常严重），其余支持向量机的分类结果还是很好地，而且正确率相差也不大。

使用测试样本进行测试，得到结果为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SVM | Test Worry Num | RightRate | FalsePos Num | FalseNeg Num |
| linear | 50 | 97.5% | 28 | 22 |
| Quadratic | 35 | 98.2% | 18 | 17 |
| Cubic | 30 | 98.5% | 17 | 13 |
| Fine Gaussian | 903 | 54.6% | 903 | 0 |
| Medium Gaussian | 46 | 97.7% | 15 | 31 |
| Coarse Gaussian | 64 | 96.8% | 33 | 31 |

对样本的测试结果和训练结果相近——Fine Gaussian假阳性严重，其余正确率较高。

整理各种判别模型如下（**神经网络使用100个隐藏节点，针对4和9两种类型重新训练**）：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ‘4’误判‘9’ | ‘9’误判‘4’ | 总错误率 | 训练用时 |
| 神经网络 | 24，2.4% | 19，1.9% | 43，2.2% | 21s |
| Logistics迭代上限100次 | 35，3.56% | 35，3.47% | 70，3.52% | 17s |
| Logistics迭代上限500次 | 23，2.34% | 39，3.87% | 62，3.11% | 94s |
| Logistics迭代上限1000次 | 33，3.36% | 29，2.87% | 62，3.11% | 214s |
| Fisher判据[[1]](#footnote-1) | 42，4.28% | 31，3.07% | 73，3.67% | 61s |
| linear SVM | 28,2.8% | 22,2.1% | 50,2.5% | 157s |
| Quadratic SVM | 18,1.8% | 17,1.6% | **35,1.7%** | 138s |
| Cubic SVM | 17,1.7% | 13,1.2% | **30,1.5%** | 239s |
| Fine Gaussian SVM | 903,91.% | 0,0% | 903,45.% | 583s |
| Medium Gaussian SVM | 15,1.5% | 31,3.0% | 46,2.3% | 372s |
| Coarse Gaussian SVM | 33,3.3% | 31,3.0% | 64,3.2% | 535s |

可以看出，除了Fine Gaussian SVM的错误率较高为，其他分类器的错误率都比较小，最小的是SVM类型的分类器和神经网络（这也看出了使用网络进行函数逼近的优越性），但是相差不大。

就训练时间来看，SVM的训练时间较高（**这与我进行SVM训练时使用的是多个模型同时并行训练的因素有关，对Linear SVM进行并行单独训练，用时76.248s，对Quadratic SVM进行单独训练[不并行]，用时36.582s,，可见训练用时其实并不太高，但准确率了上升一些** ），简单的Logistic、 Fisher判据和神经网络构成的分类器都比较快些；复杂的Logistics判据方法需要时间太长，不是很好的判断模型。

1. 先验概率之比取1——认为4和9的出现概率等价 [↑](#footnote-ref-1)