信号处理的智能化方法及应用 第七次作业

1. **SVM设计**

手机屏幕截图

描述已自动生成

1. 实现环境：Scikit-learn Version 0.23.1 on Python 3.7
2. 实现方法：使用sklearn包自带的支持向量机使用不同的核函数进行训练，训练后使用网格方式生成测试点进行测试，同时画出决策边界。
3. 程序代码

训练SVM和绘制决策边界的代码如下，作图部分代码参考了彭成磊老师课程作业的代码：

1. **def** plotDecisionBoundary(clf, data):  # 绘制超平面函数
2. x\_min, x\_max = np.array(data['X1']).min(), np.array(data['X1']).max()
3. y\_min, y\_max = np.array(data['X2']).min(), np.array(data['X2']).max()
4. xx, yy = np.meshgrid(np.linspace(x\_min, x\_max, 500), np.linspace(y\_min, y\_max, 500))
5. Z = clf.predict(np.c\_[xx.ravel(), yy.ravel()])
6. Z = Z.reshape(xx.shape)
7. plt.contour(xx, yy, Z, colors = 'red')

10. **def** train\_test\_plot(svc, data, title):
11. clfs = svc.fit(data[['X1', 'X2']], data['y'])
12. **print**('Score of ' + title + ' SVM: ' + str(svc.score(data[['X1', 'X2']], data['y'])))
14. data[title + ' SVM Confidence'] = svc.decision\_function(data[['X1', 'X2']])
16. fig, ax = plt.subplots(figsize=(6,4))
17. ax.scatter(data['X1'], data['X2'], s=50, c=data[title + ' SVM Confidence'], cmap='seismic')
18. plotDecisionBoundary(clfs, data)
19. ax.set\_title(title + ' SVM Decision Confidence')
20. plt.show()
21. 运行结果

程序选用铰接损失作为损失函数，正则化参数对于线性核函数为1，对于其他核函数为100，分别训练了线性、二次多项式、三次多项式和高斯函数作为核函数的SVM。

对于第一问，其输入数据如下：

手机屏幕截图

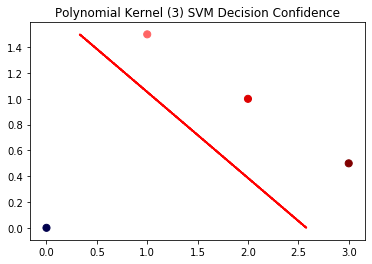
描述已自动生成

显然，该数据可以线性分割，对于全部四种SVM，分割效果都很好。

手机屏幕截图

描述已自动生成手机屏幕截图

描述已自动生成

手机屏幕截图

描述已自动生成

|  | **X1** | **X2** | **y** | **Linear SVM Confidence** | **Polynomial Kernel (2) SVM Confidence** | **Polynomial Kernel (3) SVM Confidence** | **Gaussian Kernel SVM Confidence** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | 0.0 | 0 | -0.529412 | -1.000000 | -1.000000 | -1.0 |
| **1** | 1 | 1.5 | 1 | 1.000000 | 1.000000 | 1.000000 | 1.0 |
| **2** | 2 | 1.0 | 1 | 1.117647 | 1.319527 | 1.497952 | 1.0 |
| **3** | 3 | 0.5 | 1 | 1.235294 | 1.662722 | 2.072371 | 1.0 |

可以看出训练集中的各个点的预测值置信度得分都较高，且预测结果与实际值一致。

对于第二问，输入数据如下：

手机屏幕截图

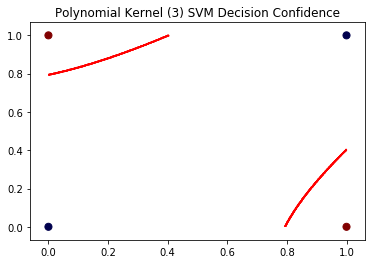
描述已自动生成

显然，该数据无法使用线性分割进行分类，因此对于线性SVM，训练出的结果置信度得分很低，无法进行预测，对于其他核函数，可以正常预测。

手机屏幕截图

描述已自动生成地图的截图

描述已自动生成

手机屏幕截图

描述已自动生成

|  | **X1** | **X2** | **y** | **Linear SVM Confidence** | **Polynomial Kernel (2) SVM Confidence** | **Polynomial Kernel (3) SVM Confidence** | **Gaussian Kernel SVM Confidence** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | 1 | 1 | 0.0 | 1.000203 | 1.000238 | 1.000000 |
| **1** | 1 | 0 | 1 | 0.0 | 0.999390 | 0.999286 | 0.999955 |
| **2** | 0 | 0 | 0 | 0.0 | -0.999797 | -0.999762 | -1.000000 |
| **3** | 1 | 1 | 0 | 0.0 | -0.999797 | -0.999762 | -0.999955 |

1. **小结**

相比简单的神经网络分类器，SVM的鲁棒性更强，这是因为SVM的最终用于预测的函数仅与少数的支持向量有关，从而对输入数据的鲁棒性更强，同时支持向量的机制也可以简化计算复杂度。不过SVM对核函数的选取有一定的要求，选取不当的核函数可能会使SVM无法工作，另外SVM本身仅适用于二分类问题，如果要进行多分类问题，需要使用一对多的方式转化为二分类问题，或者采用决策树的方式完成分类。

完整代码源文件请查看：

<https://git.nju.edu.cn/Minaduki/Intelligent_Signal_Processing>