第八章 支持向量机

1、训练数据集中共有四个数据点，其中（3,3）、（4,3）为正例点，（1,1）为负例点，求线性可分支持向量机。

解：

1. 构造最优化问题：

利用训练数据集构造约束最优化问题得：

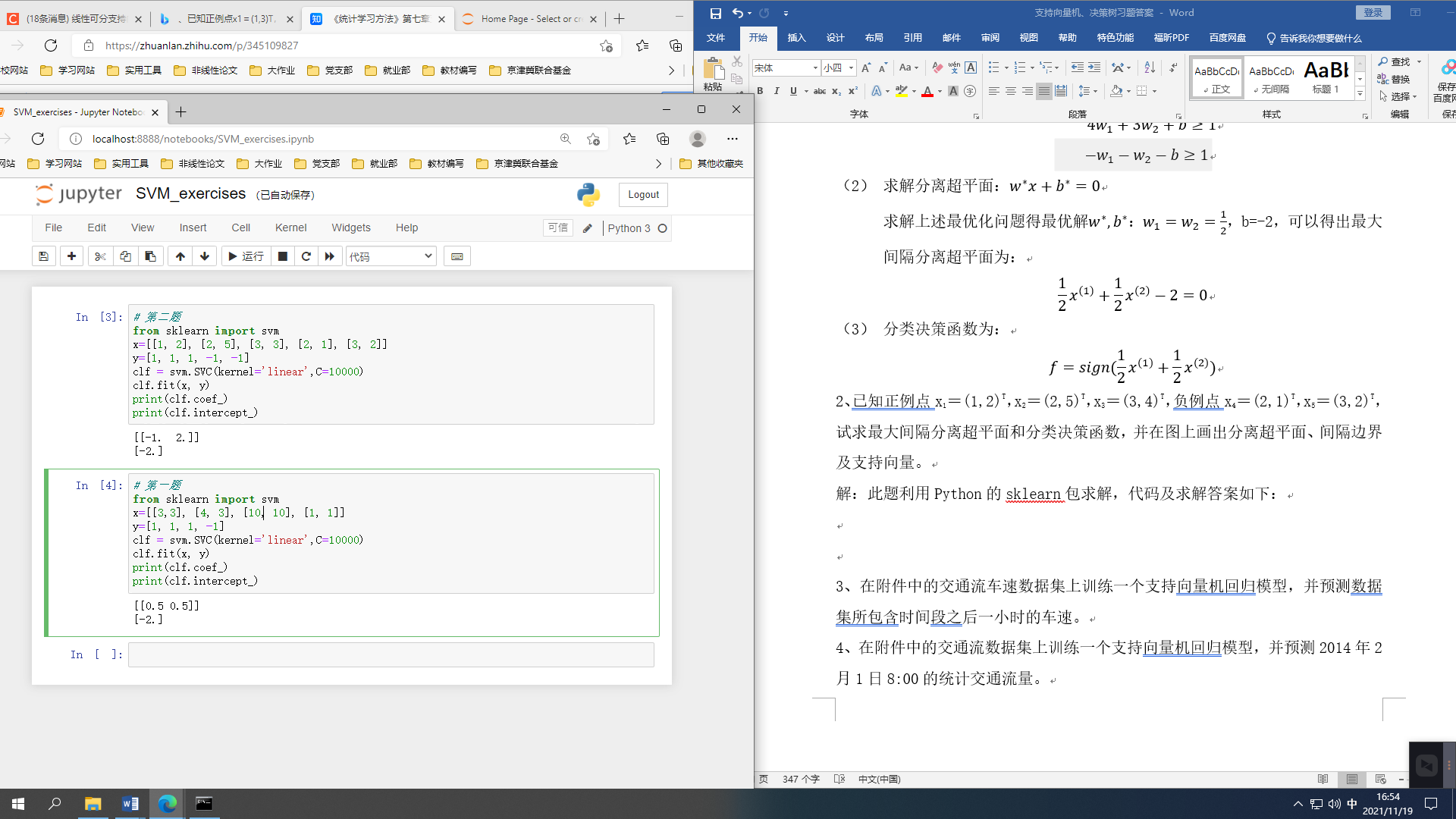
1. 求解分离超平面：

求解上述最优化问题得最优解：，b=-2，可以得出最大间隔分离超平面为：

1. 分类决策函数为：

2、已知正例点x1＝(1,2)T，x2＝(2,5)T，x3＝(3,3)T，负例点x4＝(2,1)T，x5＝(3,2)T，试求最大间隔分离超平面和分类决策函数，并在图上画出分离超平面、间隔边界及支持向量。

解：此题利用Python的sklearn包求解，代码及求解答案如下：



可知，分离超平面为：,为支持向量。分类决策函数为：

作图画出分离超平面、间隔边界及支持向量为：



3、在附件中的交通流车速数据集（Speed\_data.csv）上训练一个支持向量机回归模型，并预测数据集所包含时间段之后一小时的车速。

解：结合案例使用过的Python代码，本题目求解代码为：

1. # 导入需要的包
2. **import** pandas as pd
3. **import** numpy as np
4. **from** sklearn.svm **import** SVR
5. **from** sklearn.model\_selection **import** train\_test\_split
6. **from** sklearn.model\_selection **import** GridSearchCV
7. **from** sklearn.metrics **import** r2\_score
9. # 导入数据
10. original\_data = pd.read\_csv('D:/交通大数据/教材编写/支持向量机/Speed\_data.csv')
11. data = pd.DataFrame(original\_data)
13. # 新增一列索引列
14. data['index'] = data.index
16. # 设置自变量X和因变量y
17. X = data['index']
18. y = data['Speed']
20. # 划分训练集和测试集
21. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,random\_state = 0,train\_size = 0.8)
23. # 转换数据类型
24. X\_list = X\_train.to\_list()
25. X\_array = np.array(X\_list)
26. X\_train = X\_array.reshape(len(X\_train),-1)
28. X\_list = X\_test.to\_list()
29. X\_array = np.array(X\_list)
30. X\_test = X\_array.reshape(len(X\_test),-1)

33. y\_list = y\_train.to\_list()
34. y\_array = np.array(y\_list)
35. y\_train = y\_array.reshape(len(y\_train),-1)
37. y\_list = y\_test.to\_list()
38. y\_array = np.array(y\_list)
39. y\_test = y\_array.reshape(len(y\_test),-1)
41. # 网格搜索最优参数
42. svr = GridSearchCV(SVR(kernel = 'rbf',gamma = 0.1), cv = 5,
43. param\_grid = {'C':[1e0, 1e1, 1e2, 1e3],'gamma':np.logspace(-2, 2, 5)})
45. # 训练svr模型
46. svr.fit(X\_train,y\_train)
47. # 模型预测
48. y\_svr = svr.predict(X\_test)
50. # 计算R2
51. **print**("得分:", r2\_score(y\_test, y\_svr))
53. # 预测之后一小时的车速
54. X\_predict = np.array([2976,2977,2978,2979])
55. X\_predict = X\_predict.reshape(-1, 1)
56. y\_svr = svr.predict(X\_predict)
57. **print**(y\_svr)

输出的最终结果为：[81.11482024 80.75234621 80.64643299 80.38199647]。即2014年4月1日0:00-1：00的以15min为间隔的平均车速预测值分别为[81.11482024 80.75234621 80.64643299 80.38199647]。读者可自行继续调整参数提高模型性能。

4、在附件中的交通流数据集（Flow\_data.csv）上训练一个支持向量机回归模型，并预测2014年4月1日8:00的统计交通流量。

解：与第三题同理，本题目求解代码为：

1. # 导入需要的包
2. **import** pandas as pd
3. **import** numpy as np
4. **from** sklearn.svm **import** SVR
5. **from** sklearn.model\_selection **import** train\_test\_split
6. **from** sklearn.model\_selection **import** GridSearchCV
7. **from** sklearn.metrics **import** r2\_score
8. # 导入数据
9. original\_data = pd.read\_csv('D:/交通大数据/教材编写/支持向量机/ Flow\_data.csv')
10. data = pd.DataFrame(original\_data)
12. # 新增一列索引列
13. data['index'] = data.index
15. # 设置自变量X和因变量y
16. X = data['index']
17. y = data['FLOW']
19. # 划分训练集和测试集
20. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,random\_state = 0,train\_size = 0.8)
22. # 转换数据类型
23. X\_list = X\_train.to\_list()
24. X\_array = np.array(X\_list)
25. X\_train = X\_array.reshape(len(X\_train),-1)
27. X\_list = X\_test.to\_list()
28. X\_array = np.array(X\_list)
29. X\_test = X\_array.reshape(len(X\_test),-1)

32. y\_list = y\_train.to\_list()
33. y\_array = np.array(y\_list)
34. y\_train = y\_array.reshape(len(y\_train),-1)
36. y\_list = y\_test.to\_list()
37. y\_array = np.array(y\_list)
38. y\_test = y\_array.reshape(len(y\_test),-1)
40. # 网格搜索最优参数
41. svr = GridSearchCV(SVR(kernel = 'rbf',gamma = 0.1), cv = 5,
42. param\_grid = {'C':[1e0, 1e1, 1e2, 1e3],'gamma':np.logspace(-2, 2, 5)})
44. # 训练svr模型
45. svr.fit(X\_train,y\_train)
47. # 模型预测
48. y\_svr = svr.predict(X\_test)
50. # 计算R2
51. **print**("得分:", r2\_score(y\_test, y\_svr))
53. # 预测2014/4/1 8：00的车流量数
54. X\_predict = np.array([3008])
55. y\_svr = svr.predict(X\_predict)
56. **print**(y\_svr)

输出的最终结果为： [361.57592194]。即2014年4月1日8:00的统计交通流量预测为361.58。读者可自行继续调整参数提高模型性能。