《机器学习》课程小作业（三）

报　　告

学 院　深圳国际研究生院

班 级　深数据硕 212 班

姓 名　陈　　文　　硕

学 号　2 0 2 1 2 1 4 4 8 0

日 期　2021 年 10 月 25 日

预处理

标准化

将训练集与测试集分别存放至 project/train 与 project/test 文件夹下

将数据进行标准化至[0,1]区间

$$x^j = \frac{x^j - \min(x^j)}{std(v^j)}$$

主成分分析

在最初的实验中发现，原训练集的样本矩阵X不可逆，说明部分数据线性相关。为了让X为非奇异矩阵，可以使用 PCA 的方法进行降维。

我们选取 scikit-learn 中的 PCA 工具包（由于作业中并没有考察相关知识，故直接调包），设置主成分个数为‘mle’，即程序会自动根据主成分的方差选取主成分的个数。最终发现，降至 96 维是合理的。将 PCA 的降维与之前手动选取特征（选取前 37 维特征）对比，发现前者在实验中约有 2%的准确率提升。由于 scikit-learn 的 PCA 库中会自动对数据集进行标准化，故对于标准化，无需再单独编写函数。

归一化

实验最初遇到如下提示

ConvergenceWarning: Solver terminated early (max_iter=1000). Consider pre-processing your data with StandardScaler or MinMaxScaler.

warnings.warn('Solver terminated early (max_iter=%i).'

经过归一化处理后，收敛成功。

实验六：SVM

1. sklearn 中的参数

参数：

（1） C L2 正则化的参数 C，必须为正数；

C 越大，相当于惩罚松弛变量，希望松弛变量接近 0，即对误分类的惩罚增大，趋向于对训练集全分对的情况，这样会出现训练集测试时准确率很高，但泛化能力弱。

C 值小，对误分类的惩罚减小，容错能力增强，泛化能力较强。

（2） kernel 可选用{‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’}, default=’rbf’；

（3） degree 多项式的维度，对其他的核函数无效；

（4） gamma ‘rbf’, ‘poly’ 和‘sigmoid’的参数，分为 scale 和 auto 两种选择

（5） coef 核函数的常数项，默认为 0，仅对 poly 和 sigmoid 有用

（6） shrinking 预测哪些变量对应支持向量，加快训练速度，对结果没有影响。默认为 True。

（7） max_iter 最大迭代次数，默认为无穷大

（8） decision_function_shape ovo ovr 默认为 ovr

2. 用 linear, gaussian, polyomial 和三种参数进行训练。计算训练误差和验证误差。

设置交叉验证的 n_splits=5，test_size=0.1，统一参数对每轮交叉验证的结果取平均。

kernel	shrinking	C	decision_function_shape	train_error	cross validation error	training time (ave)
linear	True	0.1	ovo	19.81%	22.44%	0.80
linear	True	0.1	ovr	19.81%	22.44%	0.79
linear	True	1	ovo	19.81%	22.24%	2.57
linear	True	1	ovr	19.81%	22.24%	2.59
linear	True	10	ovo	19.81%	22.28%	19.68
linear	True	10	ovr	19.81%	22.28%	19.57
linear	False	0.1	ovo	19.81%	22.40%	1.65
linear	False	0.1	ovr	19.81%	22.40%	1.64
linear	False	1	ovo	19.83%	22.28%	9.42
linear	False	1	ovr	19.83%	22.28%	9.44
linear	False	10	ovo	19.81%	22.32%	86.95
linear	False	10	ovr	19.81%	22.32%	89.06
rbf	True	0.1	ovo	21.96%	25.64%	1.20
rbf	True	0.1	ovr	21.96%	25.64%	1.19
rbf	True	1	ovo	9.19%	22.28%	1.04
rbf	True	1	ovr	9.19%	22.28%	1.03
rbf	True	10	ovo	0.12%	23.76%	1.53
rbf	True	10	ovr	0.12%	23.76%	1.56
rbf	False	0.1	ovo	21.96%	25.64%	1.18
rbf	False	0.1	ovr	21.96%	25.64%	1.19
rbf	False	1	ovo	9.19%	22.28%	1.03
rbf	False	1	ovr	9.19%	22.28%	1.02
rbf	False	10	ovo	0.12%	23.76%	1.26
rbf	False	10	ovr	0.12%	23.76%	1.24
poly	True	0.1	ovo	41.61%	47.80%	1.06
poly	True	0.1	ovr	41.61%	47.80%	1.06
poly	True	1	ovo	9.23%	28.36%	1.06
poly	True	1	ovr	9.23%	28.36%	1.06
poly	True	10	ovo	0.36%	26.92%	1.35
poly	True	10	ovr	0.36%	26.92%	1.35
poly	False	0.1	ovo	41.61%	47.80%	1.02
poly	False	0.1	ovr	41.61%	47.80%	1.02
poly	False	1	ovo	9.23%	28.36%	1.01
poly	False	1	ovr	9.23%	28.36%	1.02
poly	False	10	ovo	0.36%	26.92%	1.08
poly	False	10	ovr	0.36%	26.92%	1.04

3. 在 TestSet-1 上做测试

根据实验 6.2 中交叉验证的结果，我们选取参数 C=1, kernel='linear'。 训练模型后，在 TestSet-1 上的正确率为 79.3%。（结果保存在 submissions.csv 中）

4. 实验小结

主要依据实验 6-2 的观察结果有以下结论：

（1） 更复杂的模型取得的效果不一定比简单模型好，如我们采用 linear 核函数的效果就比高斯、多项式核函数效果好；

（2） shrinking 操作能智能选取局部向量进行分类，大大加快训练操作；

（3） 在本次实验中，并没有观察到惩罚系数对测试准确率的影响。

（4） 本实验还对比了 decision_function_shape 的参数。观察结果之前很惊讶 ovo 和 ovr 的训练时长是一样的。是由于本问题是二分类问题，所以没有差别。通常 ovo 在多分类问题中会比 ovr 慢。