```
姓名: 李宇豪
学号: 21305412
周数:5
成绩:
程序:
import time
import numpy as np
from matplotlib import pyplot as plt
from sklearn.datasets import load breast cancer
from sklearn.model selection import train test split
from sklearn.linear model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.model selection import ShuffleSplit
from sklearn.model_selection import learning curve
from sklearn.preprocessing import PolynomialFeatures
def polynomial model(d, c=0.05, **kwarg):
    polynomial features = PolynomialFeatures(degree=d)
    logistic regression = LogisticRegression(**kwarg, C=c,
                                             solver='liblinear',
max iter=300)
    pipeline = Pipeline([("polynomial features",
polynomial features),
                         ("logistic regression",
logistic regression)])
    return pipeline
def plot learning curve (estimator, title, X, y, ylim=None,
                    cv=None, train sizes=np.linspace(.1, 1.0,5)):
    plt.title(title)
    if ylim is not None:
        plt.ylim(*ylim)
    plt.xlabel("Training example")
    plt.ylabel("Score")
    train sizes, train scores, test scores = learning curve(
        estimator, X, y, cv=cv, train sizes=train sizes)
    train scores mean = np.mean(train scores, axis=1)
    test scores mean = np.mean(test scores, axis=1)
    plt.grid(ls='--')
    plt.plot(train sizes, train scores mean, 'o-', color="r",
             label="Training score")
```

```
plt.plot(train_sizes, test_scores mean, 'o-', color="g",
             label="Cross-validation score")
    plt.legend(loc="best")
    return plt
cancer = load breast cancer()
X = cancer.data
y = cancer.target
X train, X test, y train, y test = \
    train test split(X, y, test size=0.20, random state=0)
time all = time.process time()
for i in [1, 2]:
    for j in ["11", "12"]:
        model = polynomial model(i, penalty=j)
        start = time.process time()
        model.fit(X train, y_train)
        train score = model.score(X train, y train)
        test score = model.score(X test, y test)
        print(i, "order polynomial fitting with", j, "norm:")
        print("elapse: {0:.4f}; train score: {1:.4f}; test score:
             {2:.6f}".format(time.process time() - start,
             train score, test score))
        cv = ShuffleSplit(n splits=10, test size=0.2,
                         random state=0)
        title = 'Learning Curve (degree={0}, penalty={1})'
                 .format(i, j)
        plt.figure(figsize=(6, 4), dpi=200)
        plot learning curve(polynomial model(i, penalty=j),
                            title, X, y, cv=cv)
        plt.show()
        plt.clf()
        if i == 2 and j == "11":
            R train = []
            R \text{ test} = []
            for k in [0.01, 0.1, 1, 10, 100]:
                model = polynomial model(i, k, penalty=j)
                start = time.process time()
                model.fit(X train, y train)
                train score = model.score(X train, y train)
                test score = model.score(X test, y test)
                R train.append(train score)
                R test.append(test score)
                print ("lambda={0}时, 训练集 R 方评分为{1}, 测试集 R 方评
                      分为{2}".format(str(1/k), train score,
                      test score))
            p1, = plt.plot([1, 2, 3, 4, 5], R train, color="red")
            p2, = plt.plot([1, 2, 3, 4, 5], R test, color="blue")
```

print("程序运行总时间: ", time.process time() - time all)

输出:

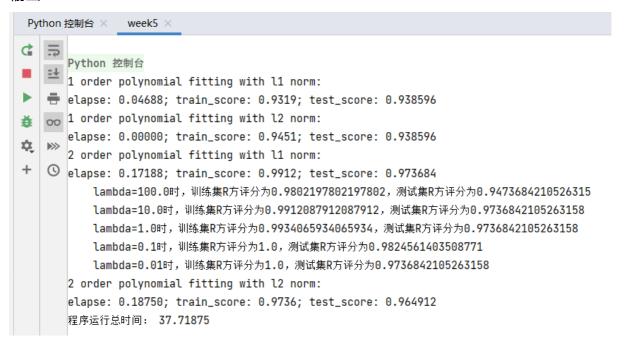


图:

