1.线性回归

给定满足函数关系的一组训练样本$\{(x_1, y_1), \ldots, (x_N, y_N)\}, N = 300$,使用线性回归模型拟合函数$y = f(x)$。

说明：

1.模型

线性回归是一种对自变量和因变量之间关系进行建模的回归分析。

自变量为 1 时称为简单回归，自变量数量大于 1 时成为多元回归。

自变量是样本的特征向量，因变量是样本的标签，其中标签为实数或者连续整数。

假设空间是一组参数化的线性函数，这个线性函数即线性模型。

2.学习准则

使用经验风险最小化。

3.优化算法

线性回归通过伪逆矩阵求参数。

4.评价指标

预测值与真实值的标准差。

5.代码

```
#加载 numpy 函数库
#加载 pyplot 函数库
import numpy as np
import matplotlib.pyplot as plt

"""载入数据"""
#定义载入数据函数
#初始化列表
#打开文件
#对文件按行遍历行
#append 在列表末尾添加新的对象
#map 根据提供的函数对指定的列表做映射
#float 将列表中数据类型转换为 float 类型
#s.strip(str)移除字符串首尾指定的字符 str（默认为空字符）
#s.split(str,num)通过指定分隔符 str 对字符串 s 进行切片
#（默认为分隔符 str 空字符，分割次数 num 为-1，即分割所有）
#将元组组成的列表解压
#返回自变量数组 x 和因变量数组 y
def load_data(filename):
    xys = []
    with open(filename, 'r') as f:
        for line in f:
            xys.append(map(float, line.strip().split()))
        xs, ys = zip(*xys)
        return np.asarray(xs), np.asarray(ys)

"""评估模型"""
#定义评价函数
#评价指标为标准差
def evaluate(ys, ys_pred):
    std = np.sqrt(np.mean(np.abs(ys - ys_pred) ** 2))
    return std

"""训练模型，并返回从 x 到 y 的映射"""
#定义模型训练函数
#使用线性回归训练模型，根据训练集计算最优化参数
```

```
#返回与 x_train 数组具有相同形状和数据类型的数组 phi0，且数组中的值为 1。
#phi0 数组维度从(300,)变为(300,1)
#phi1 数组维度从(300)变为(300,1)
#将两个维度为(300,1)的矩阵按列拼接为(300,2)维度的矩阵，即增广特征向量矩阵 phi。
#np.dot(x,y)表示矩阵 x 和矩阵 y 相乘
#np.linalg.pinv(phi)表示求矩阵 phi 的伪逆矩阵，维度为(2,300)
#y_train 维度为(300,)
#w 为增广权重向量矩阵，维度为(2,)
    #定义预测函数 f
    #返回从 x 到 y 的映射函数 y=f(x)
    #注意：函数 f(x)的变量只有 x，参数 w 应作为内部变量
#返回预测函数 f
def main(x_train, y_train):
    phi0 = np.expand_dims(np.ones_like(x_train), axis=1)
    phi1 = np.expand_dims(x_train, axis=1)
    phi = np.concatenate([phi0, phi1], axis=1)
    w = np.dot(np.linalg.pinv(phi), y_train)

    def f(x):
        phi0 = np.expand_dims(np.ones_like(x), axis=1)
        phi1 = np.expand_dims(x, axis=1)
        phi = np.concatenate([phi0, phi1], axis=1)
        y = np.dot(phi, w)
        return y
        pass

    return f

# 程序主入口（建议不要改动以下函数的接口）
if __name__ == '__main__':

    #载入数据
    #训练集特征为 300 行 1 列
    #测试集特征为 200 行 1 列
    train_file = 'train.txt'
    test_file = 'test.txt'
    x_train, y_train = load_data(train_file)
    x_test, y_test = load_data(test_file)

    #使用线性回归训练模型，返回一个函数 f()使得 y = f(x)
    #计算预测的输出值
    f = main(x_train, y_train)
    y_test_pred = f(x_test)

    #使用测试集评估模型
    std = evaluate(y_test, y_test_pred)
    print('预测值与真实值的标准差：{:.1f}'.format(std))

    #显示结果
    #训练集数据图、测试集数据图、测试集预测图
    #图横纵坐标标签、标题、图例
    plt.plot(x_train, y_train, 'ro', markersize=3)
```

```
plt.plot(x_test, y_test, 'k')
plt.plot(x_test, y_test_pred)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Linear Regression')
plt.legend(['train', 'test', 'pred'])
plt.show()
```