深 圳 大 学 实 验 报 告

课程名称 机器学习

项目名称 实验三：线性回归

学　　院 计算机与软件学院

专　　业 软件工程（腾班）

指导教师 赖志辉

报 告 人 黄亮铭 学号 2022155028

实验时间 2024 年 3 月 25 日至 2024 年 4 月 7 日

实验报告提交时间 2024 年 4 月 7 日

教务处制

一、实验目的与要求

1. 熟悉线性回归算法，要求理解线性回归的基本原理；
2. 使用实际数据进行线性回归分析；
3. 使用回归模型进行预测和评估；
4. 分析回归模型的拟合程度与预测效果。

二、实验内容与方法

1. 简要介绍并实现基本的线性回归算法及其它论文中的 2 个回归学习方法（模型与优化都要有）。

2. 自行设计一个全新的线性回归算法（不是别人论文里的！而是自己创造的！展开你的想象力来设计），包括建模与优化，收敛性证明等（如果有），要求：至少在 2~3 个数据库中与 1 中的方法进行实验比较。 全方位比较你的方法与你复现的方法在不同维数的识别率。

三、实验步骤与过程

后面所有回归算法均以 AR 数据集为例进行截图，其他数据集与 AR 数据集的区别只有导入的区别，因此不展示。

1. 线性回归（Simple Linear Regression）是一种用于建立和分析变量之间关系的基本统计方法。它适用于一种被称为“因变量”（dependent variable）与另一种被称为“自变量”（independent variable）之间的线性关系。这种方法的目标是找到一条直线，即回归线，最大程度地拟合数据点。回归线的表达式通常为：$y = mx + b$，其中，y 是因变量，x是自变量，m是回归线的斜率，b是截距。

线性回归的原理如下：

预测值： $\hat{y_i} = mx_i + b$

损失函数： $L = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$

对m*求偏导并令偏导数为*0：$\frac{\partial L}{\partial m} = -2\sum_{i=1}^{n} x_i\big(y_i - (mx_i + b)\big) = 0$

对b*求偏导并令偏导数为*0：$\frac{\partial L}{\partial b} = -2\sum_{i=1}^{n}\big(y_i - (mx_i + b)\big) = 0$

得到结果如下： $m = \frac{n(\sum_{i=1}^{n} x_i\, y_i) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n\big(\sum_{i=1}^{n} x_i^2\big) - \big(\sum_{i=1}^{n} x_i\big)^2}$

$$b = \frac{\sum_{i=1}^{n} y_i - m\sum_{i=1}^{n} x_i}{n}$$

2. 简单线性回归的核心是最小化预测值与实际值之间的误差，通常使用最小二乘法来实现。最小二乘法的目标是找到一条直线，使得所有数据点到该直线的距离之和最小。

简单线性回归的步骤如下：

1. 数据收集：收集自变量和因变量的数据。

```
% 导入人脸数据集
reshaped_faces=[];
% 数据库名
database_name = "AR";
%AR5040
if (database_name == "AR")
    for i=1:40
      for j=1:10
        if(i<10)
           pic=imread(strcat('C:\Users\黄亮铭\Desktop\大学课程\机器学习\实验1PCA\AR
        else
           pic=imread(strcat('C:\Users\黄亮铭\Desktop\大学课程\机器学习\实验1PCA\AR
        end
        reshaped_pic = reshape(pic,2000,1);
        reshaped_pic=double(reshaped_pic);
        reshaped_faces=[reshaped_faces, reshaped_pic];
      end
    end
end
row = 50;
```

文件 编辑 查看

黄亮铭 软件工程（腾班）
2022155028 4.5

图 1：数据集导入

2. 数据预处理：对数据进行清洗和预处理，包括处理缺失值、异常值等。

```
row = 50;
col = 40;
people_num = 40;
each_pic_num = 10;
each_train_num = 4;
each_test_num = 6;
test_sum = each_test_num * people_num;
```

文件 编辑 查看

黄亮铭 软件工程（腾班
2022155028 4.5

图 2：划分训练集和测试集

3. 建立模型：选择适当的回归模型，对数据进行拟合。

   b 融入 w 中得到回归模型：$\hat{Y} = XW$。

4. 模型拟合：使用最小二乘法等技术拟合回归模型，得到最优的斜率和截距。

```
% 1.线性回归
w = inv(train_data' * train_data) * train_data' * correct;
correct_hat = train_data * w; % 计算预测图片
```

黄亮铭 软件工程（腾班）
2022155028 3.23

3：根据 3 中的公式拟合

5. 评估模型：评估模型的性能，通常使用指标如平均误差、均方误差等。

```
%输出结果
recognition_rate = count_right / test_sum;
persentage_sign = "%";

mean_var = all_var / all_num;
fprintf("均方误差为：%f\n", mean_var);
fprintf("正确率为：%d%s\n",  recognition_rate * 100, persentage_sign);
```

文件 编辑 查看

黄亮铭 软件工程（腾班）
2022155028 3.23

图 4：输出模型性能指标

模型结果：使用测试集对模型进行测试，得到正确率。

```
正确率为：90%
>>
```

黄亮铭 软件工程（腾班）
2022155028 3.23

图 5：模型正确率

3. 岭回归算法

岭回归算法数学原理：

当面对线性回归问题时，岭回归是一种常用的正则化技术，用于处理数据集中存在共线性（即特征之间存在高度相关性）的情况。岭回归通过引入一个正则化项来控制模型的复杂度，从而改善模型的稳定性和泛化能力。

考虑普通的线性回归问题：

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon$$

其中， $\mathbf{y}$是目标变量向量，$\mathbf{X}$ 是特征矩阵，$\mathbf{w}$是回归系数向量，ϵ 是误差项向量。

岭回归的目标是找到一个最优的回归系数向量$\mathbf{w}$，使得拟合数据和正则化项的和最小。这可以通过最小化岭回归的损失函数来实现：

岭回归的数学表达表示如下：

$$L(\mathbf{w}) = || \mathbf{y} - \mathbf{X}\mathbf{w} ||_2^2 + \alpha || \mathbf{w}||_2^2$$

其中， $\mathbf{y}$是目标变量向量，$\mathbf{X}$是特征矩阵，$\mathbf{w}$ 是回归系数向量，α是正则化参数。

其中，第一项是普通的最小二乘损失，第二项是正则化项， α 是正则化参数，控制了正则化项的影响程度。通过调整α，可以平衡拟合数据和控制模型复杂度之间的权衡关系。

通过求解上述损失函数的最小化问题，我们可以得到岭回归的系数解析解：

$$\hat{\mathbf{w}} = (\mathbf{X}^{\mathbf{TX}} + \alpha\mathbf{I})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y}$$

其中，$\mathbf{I}$ 是单位矩阵。

岭回归通过引入正则化项，限制了回归系数的增长，从而降低了模型的复杂度，提高了模型的泛化能力，特别是在面对高度共线性数据时。

步骤和基础线性回归算法类似，但是在其中加入了 L2 正则化。通过正则化扰动，有效避免过拟合。

优化后的代码如下：

```
% 2.岭回归
rr_data = (train_data' * train_data) + eye(each_train_num)*10^-6;
w = inv(rr_data) * train_data' * correct;
correct_hat = train_data * w; % 计算预测图片
```

文件 编辑 查看

黄亮铭 软件工程（腾班）
2022155028 3.23

图 6：优化代码

模型性能指标及测试数据集在该模型下测得的正确率如下：

```
>> LR
均方误差为：3098.609763
正确率为：91.402000%
>>
```

黄亮铭 软件工程（腾班）
2022155028 3.23

图 7：模型性能指标

4. lasso 回归算法

Lasso 回归算法的数学原理：

假设我们有一个包含 n 个样本的数据集，每个样本包含 m 个特征。我们用 $\mathbf{X}$ 表示一个 $n \times m$ 的特征矩阵，每行代表一个样本，每列代表一个特征。目标变量 $\mathbf{y}$ 是一个长度为n的向量，表示每个样本的目标值。

我们的目标是找到一个线性模型，使得预测$\hat{\mathbf{y}}$尽可能接近真实的目标$\mathbf{y}$。线性模型的预测值可以表示为：

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$$

其中$\mathbf{w}$ 是一个长度为m 的系数向量，表示每个特征对目标变量的影响程度。

为了最小化预测值与真实值之间的误差，我们定义损失函数为预测值与真实值之间的平方误差，即均方误差（Mean Squared Error，MSE）：

$$\text{MSE}(\mathbf{w}) = \frac{1}{2n}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2$$

其中$||\cdot||_2$ 表示向量的二范数.

为了防止模型过拟合，我们引入正则化项。Lasso 回归使用 L1 正则化，其正则化项定义为系数向$\mathbf{w}$的 L1 范数：

$$\text{L1 正则化项}(\mathbf{w}) = \alpha||\mathbf{w}||_1$$

其中 α 是正则化参数，控制正则化项的权重。将正则化项加到损失函数中得到 Lasso 回归的目标函数：

$$\begin{aligned}\text{目标函数}(\mathbf{w}) &= \text{MSE}(\mathbf{w}) + \text{L1 正则化项}(\mathbf{w}) \\ &= \frac{1}{2n}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \alpha||\mathbf{w}||_1\end{aligned}$$

现在，我们的目标是最小化目标函数。由于目标函数是关于系数向$\mathbf{w}$的函数，我们可以通过求解目标函数的梯度为零来找到最优的系数向$\mathbf{w}$。

然而，L1 正则化项的非光滑性质使得目标函数不再是光滑的，导致无法直接通过求梯度为零的方法得到解析解。因此，常用的方法是使用梯度下降算法或坐标下降算法等迭代优化方法来求解。

步骤和基础线性回归算法类似，但是在其中加入了 L1 正则化。高维的数据其线性关系一般是稀疏的，lasso 回归算法在这种情况下一般比较优异的。

优化后的代码如下：

```
% 3.lasso回归
[B,FitInfo] = lasso(train_data , correct);
correct_hat = train_data * B + FitInfo.Intercept;
```

黄亮铭 软件工程（腾班）
2022155028 3.23

图 8：优化代码

模型性能指标及测试数据集在该模型下测得的正确率如下：

```
>> LR
均方误差为：27014.151830
正确率为：94.583333%
>>
```

黄亮铭 软件工程（腾班）
2022155028 3.23

图 9：模型性能指标

5. 自行设计的全新的线性回归算法：PCA 降维线性回归算法。

PCA 的数学原理：

主成分分析（Principal Component Analysis，PCA）是一种常用的数据降维技术，它通过线性变换将高维数据映射到低维空间中，以保留尽可能多的数据信息。PCA 的主要思想是找到数据中的主要方差方向，并将数据投影到这些方向上，从而实现降维。

以下是 PCA 的数学推导过程：

假设我们有一个数据集 X，其中每一行代表一个样本，每一列代表一个特征。我们的目标是找到一个 d 维的子空间（d 远小于原始数据的维度），使得数据在该子空间中的投影能够最大程度地保留原始数据的方差。

1. 数据中心化：

首先，我们对数据集 X 进行中心化处理，即将每个特征的均值减去相应的列均值，以确保数据的均值为零。中心化后的数据集表示为 $\tilde{X}$。

2. 计算协方差矩阵：

接下来，我们计算中心化后的数据集 $\tilde{X}$ 的协方差矩阵 C。协方差矩阵 C 的元素 c_{ij} 表示 $\tilde{X}$ 的第 i 和第 j 个特征之间的协方差。协方差矩阵 C 是对称矩阵。

协方差矩阵C 的计算公式如下：

$$C = \frac{1}{m}\widetilde{X^T}\tilde{X}$$

其中，m 是样本数量。

3. 计算特征值和特征向量：

对协方差矩阵 C 进行特征值分解，得到特征值和对应的特征向量。假设特征值为 $\lambda_1, \lambda_2, \dots, \lambda_n$，对应的特征向量为 $\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_n}$。

4. 选择主成分：

根据特征值的大小顺序，选择前 d 个特征值对应的特征向量作为主成分。这些特征向量构成了主成分空间，也就是我们要将数据投影到的低维子空间。

5. 数据投影：

将中心化后的数据 $\tilde{X}$ 投影到选择的主成分上，得到降维后的数据集。

$$Z = \tilde{X}\mathbf{V_d}$$

其中，Z 是降维后的数据集，$\mathbf{V_d}$ 是选择的前 d 个特征向量组成的矩阵。

这就是 PCA 的数学推导过程。通过对数据的协方差矩阵进行特征值分解，并选择前 d 个特征值对应的特征向量，我们可以找到一个能够最大程度地保留原始数据方差的 d 维子空间，从而实现数据的降维。

图片的像素一般比较高，拉成向量后维度很高，导致计算时间复杂度大大增加。为此，我们可以在使用线性回归算法之前，使用 PCA 算法对人脸数据集进行降维，从而大大降低时间复杂度。

PCA 降维部分的代码：

```
% 求平均脸
mean_face = mean(reshaped_faces,2);
% 中心化
centered_face = (reshaped_faces - mean_face);
% 协方差矩阵
cov_matrix = centered_face * centered_face';
[eigen_vectors, dianogol_matrix] = eig(cov_matrix);
% 从对角矩阵获取特征值
eigen_values = diag(dianogol_matrix);
% 对特征值按索引进行从大到小排序
[sorted_eigen_values, index] = sort(eigen_values, 'descend');
% 获取排序后的征值对应的特征向量
sorted_eigen_vectors = eigen_vectors(:, index);

 % 取出相应数量特征脸(降到n维)
   n = 200;
   eigen_faces = sorted_eigen_vectors(:,1:n);
    % 测试、训练数据降维
    projected_data = eigen_faces' * (reshaped_faces - mean_face);
    % 使用PCA降维
     reshaped_faces = projected_data;
```


图 10：PCA 降维

线性回归部分的代码：

```
% 5.提前PCA降维线性回归
rr_data = (train_data' * train_data) + eye(each_train_num)*10^-6;
w = inv(rr_data) * train_data' * test_data;
correct_hat = train_data * w; % 计算预测图片
```


图 11：线性回归

模型性能指标及测试数据集在该模型下测得的正确率如下：

```
>> LR
均方误差为：7222.341686
正确率为：90.985333%
>>
```


图 12：模型性能指标

6. 多种回归算法在人脸识别领域对比

	ORL	AR	FERET
线性回归	84.50%	90%	86.25%
岭回归	84.50%	91.40%	86.43%
lasso回归	90.13%	94.58%	90.07%
PCA回归	79.86%	90.99%	78.93%

图 13：不同回归算法在不同数据集中的对比

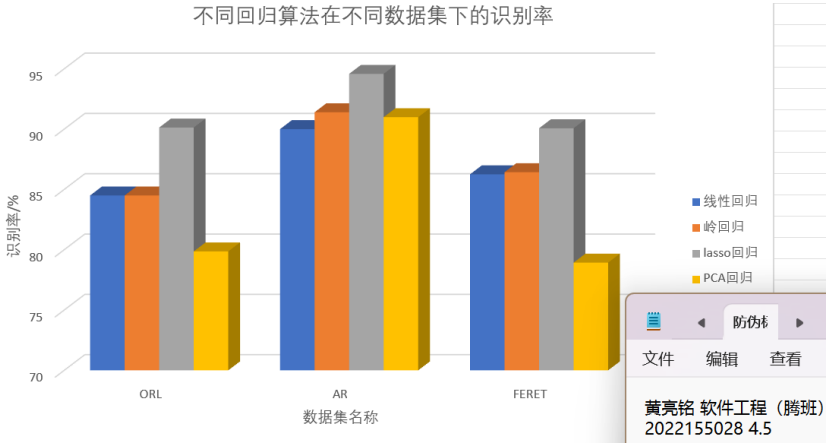

图 14：可视化对比

由上面的表格和可视化图像可以看出：

1） 在三个数据集中，lasso 回归算法的识别率均为最高。

2） 简单线性回归和岭回归的识别率几乎相同，因为我们处理人脸数据时将人脸拉成向量，维度较小，不存在奇异矩阵，矩阵可逆。稍微不同可能是因为在跑代码时某个方法使用了伪逆近似。

3） 自行设计的 PCA 回归在 AR 数据集中表现较好，但是在 ORL 和 FERET 数据集中效果较差，识别率低于简单线性回归。

四、实验结论或体会

1. 线性回归常用于预测因变量的值，或者理解自变量与因变量之间的关系。然而，它也有一些局限性，例如它假设自变量和因变量之间的关系是线性的，并且对异常值和离群点比较敏感。

2. Linear Regression 作基础的回归分析方法，对于线性数据的拟合与预测过程简单，效果优秀，Linear Regression Classification 及其多种优化模型在数据分类的任务中也表现优异。且作为一种有监督学习方法，Linear Regression 能得到较好的保留数据信息用于分类。但 Linear Regression 仍存在对异常值很敏感、容易造就过拟合模型、不好刻画非线性问题等等，从而影响实验结果，Linear Regression 的一些优化模型解决了部分不足。

3. 通过实验，我对线性回归算法有了更深入的理解。我发现在不同的数据集和维度下，不同的算法表现出不同的性能。自创算法在某些情况下可能表现优于传统的最小二乘法。

4. 在实验中发现，不同算法的表现受数据特征影响较大。在特征相关性较高的情况下，简单线性回归算法和 PCA 回归算法算法表现良好，而在特征相关性较低的情况下，lasso 回归算法可能更有效。

指导教师批阅意见： 成绩评定： 指导教师签字： 年　　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。