

**人工智能数学基础实验报告**

题 目 PCA+RPCA

学 院 计算机科学与技术

专 业 人工智能

学 号

学 生

任 课 教 师 刘绍辉

哈尔滨工业大学计算机科学与技术学院

2023.5

**实验二:PCA+RPCA**

1. **实验内容**

**主成份分析和鲁棒主成份分析**

**1.1.理解PCA和RPCA基本原理**

(1)PCA的基本原理是将n维特征映射到k维上，这k维是全新的正交特征也被称为主成分，是在原有n维特征的基础上重新构造出来的k维特征。

(2)RPCA的基本原理是将一个数据矩阵分解为一个低秩矩阵和一个稀疏矩阵的和。

**1.2.用PCA和RPCA用来对mnist数据集进行分类（可以用任何数据集的处理）**

(1)PCA 参考来源：

https://github.com/topics/principal-component-analysis-pca

(2)Robust-PCA 参考来源：

<https://github.com/dlaptev/RobustPCA>

Learned Robust PCA: A Scalable Deep Unfolding Approach for High-Dimensional Outlier Detection

1. **算法简介及其实现细节**

**2.1 PCA**

**2.1.1 PCA算法简介**

PCA (Principal Component Analysis) 即主成分分析，是一种无监督的数据降维方法。PCA的主要思想是将n维特征映射到k维上，这k维是全新的正交特征也被称为主成分，是在原有n维特征的基础上重新构造出来的k维特征。

PCA的工作就是从原始的空间中顺序地找一组相互正交的坐标轴，新的坐标轴的选择与数据本身是密切相关的。其中，第一个新坐标轴选择是原始数据中方差最大的方向，第二个新坐标轴选取是与第一个坐标轴正交的平面中使得方差最大的，依次类推。PCA可以用于数据压缩、去噪、可视化等领域。

**2.1.2 PCA算法实现细节**

PCA算法有两种实现方法：基于特征值分解协方差矩阵实现PCA算法、基于SVD分解协方差矩阵实现PCA算法。

基于特征值分解协方差矩阵实现PCA算法的步骤如下：

1. 对数据进行中心化处理，即每个维度减去该维度的均值。

2. 计算中心化后数据的协方差矩阵，即每个维度之间的协方差构成的矩阵。

3. 对协方差矩阵进行特征值分解，得到特征值和特征向量。

4. 选择前k个最大的特征值对应的特征向量作为新的基，形成一个变换矩阵。

5. 将原始数据乘以变换矩阵，得到降维后的数据。

基于SVD分解协方差矩阵实现PCA算法的步骤如下:

1. 对数据进行中心化处理，即每个维度减去该维度的均值。

2. 计算中心化后数据的散度矩阵，即数据矩阵乘以其转置。

3. 对散度矩阵进行SVD分解，得到奇异值和奇异向量。

4. 选择前k个最大的奇异值对应的奇异向量作为新的基，形成一个变换矩阵。

5. 将原始数据乘以变换矩阵，得到降维后的数据。

**2.2 RPCA**

**2.2.1 RPCA算法简介**

RPCA（鲁棒主成分分析）是一种数据降维算法，它可以将一个数据矩阵分解为一个低秩矩阵和一个稀疏矩阵的和，即 ，其中L是低秩矩阵，S是稀疏矩阵。RPCA的目的是从数据中提取出低秩结构和稀疏结构，从而实现数据的压缩、去噪、异常检测等应用。

**2.2.2 RPCA算法实现细节**

**1.定义优化问题**：

，

其中表示L的核范数，即L的奇异值之和，表示S的范数，即S的所有元素的绝对值之和，是一个正则化参数。

**2.使用交替方向乘子法（ADMM）求解优化问题，具体步骤如下**：

**2.1** 初始化，其中是拉格朗日乘子矩阵，是惩罚参数。

**2.2** 迭代更新，直到收敛或达到最大迭代次数。更新规则如下：

**(1)** 固定，更新：

，

其中表示奇异值阈值函数，即对输入矩阵进行奇异值分解，然后将奇异值减去并截断为非负值，再重构输出矩阵。

**(2)**固定，更新：

，

其中表示软阈值函数，即对输入矩阵的每个元素减去并截断为非负值或加上并截断为非正值。

**(3)**固定，更新：

**3.输出最终的作为RPCA的结果。**

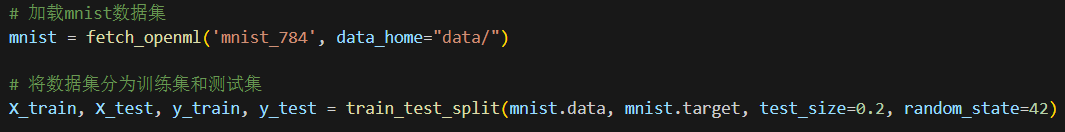
1. **实验设置及结果分析（包括实验数据集）**

**3.1数据集下载**

数据集的下载使用了scikit-learn库。

scikit-learn，又写作sklearn，是一个开源的基于python语言的机器学习工具包。它通过NumPy,SciPy和Matplotlib等python数值计算的库实现高效的算法应用，并且涵盖了几乎所有主流机器学习算法。

使用fetch\_openml()函数从openml下载数据库到本地并进行载入，并使用train\_test\_split()函数将数据集划分为训练集和测试集。

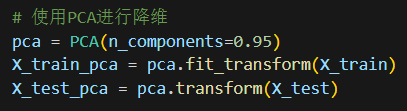


**3.2 PCA**

实验中使用了scikit-learn库中已经实现的PCA算法对数据进行降维。

n\_components设置为0.95，则表示要选择足够的主成分，使得它们解释的累计方差达到总方差的95%。在这种情况下，PCA会自动选择所需的最少主成分数量，以满足累计方差比例的要求。

这种设置方式可以很方便地控制保留的信息量。通过设置累计方差比例，可以确保保留足够的信息来解释大部分的方差，同时降低数据的维度。



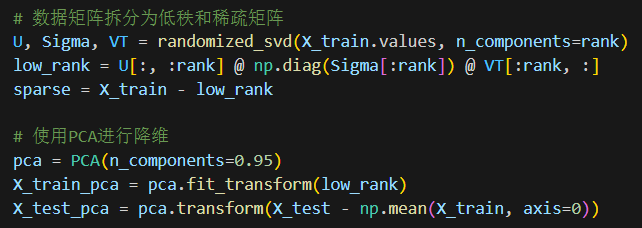
**3.3 RPCA**

在RPCA的实现中，使用了随机SVD算法对训练集进行了降维。

调用了randomized\_svd函数，将训练集X\_train的值作为输入，并指定了降维后的维度rank。函数的返回值包括三个矩阵u、Sigma和VT，它们分别代表了X\_train的左奇异矩阵、奇异值和右奇异矩阵。

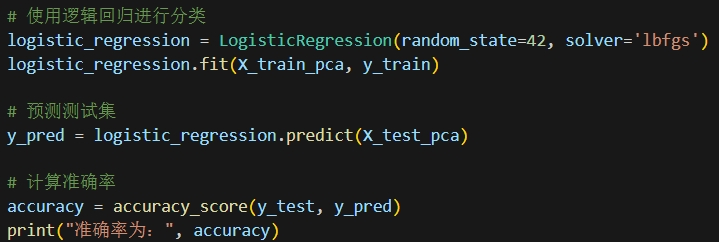
使用这些矩阵计算了一个低秩矩阵low\_rank，它的维度为X\_train的维度，并且只保留了前rank个奇异值。

计算了一个稀疏矩阵sparse，它等于X\_train减去低秩矩阵low\_rank。这个过程可以被看作是对X\_train进行了分解，将其分解为一个低秩矩阵和一个稀疏矩阵的和。之后，通过PCA算法再对提取出的低秩矩阵进行降维操作，具体代码如下：



**3.4数据分类**

为了检测两种降维方法的实验效果，使用逻辑回归分类器对降维后的数据进行分类。



分类结果如下图所示

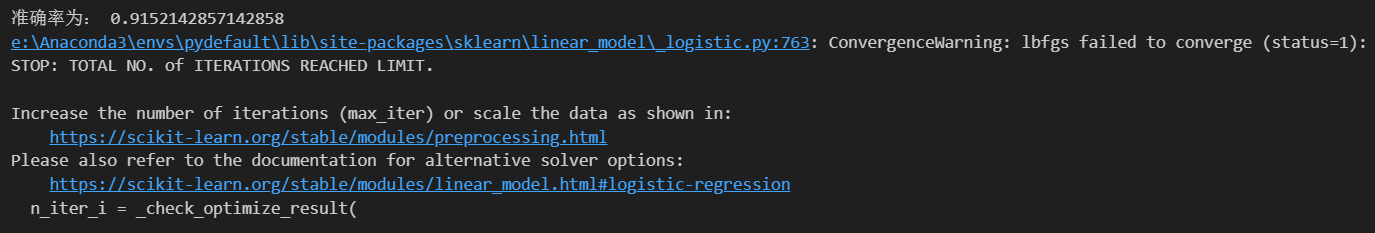


图 1: 使用 PCA 算法的分类结果

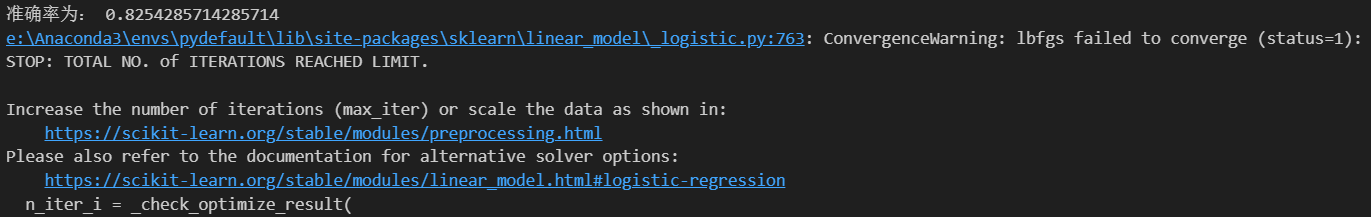


图 2: 使用 RPCA 算法的分类结果

实验结果显示PCA在MNIST数据集上具有较高的分类准确率，能够有效地提取数据中的主要特征。PCA通过降维将高维数据映射到低维空间，保留了大部分的信息，有助于分类任务的准确性。

RPCA在MNIST数据集上的分类准确率相对较低。RPCA可能在处理MNIST数据集的特征提取方面存在一些局限性，无法充分地捕捉手写数字的重要特征。

实验结果表明PCA在MNIST数据集上能够提取出更具有区分度的特征，使得分类器能够更好地区分手写数字。相比之下，RPCA可能在特征提取方面的效果较弱，导致分类器的性能不如PCA。

**四、 结论**

PCA和RPCA是两种不同的降维方法，它们在特征提取和分类性能上可能会有不同的表现。以下是可能导致PCA精确度比RPCA高的一些原因：

PCA保留了更多的信息：PCA通过选择方差最大的主成分来进行特征提取，这意味着它尽可能保留了原始数据中的主要信息。PCA的目标是最大化投影后数据的方差，以保持数据的多样性和可区分性。这使得PCA能够更好地保留数据的结构和关键特征，有助于提高分类性能。

RPCA可能无法准确分离低秩和稀疏部分：RPCA通过将数据分解为低秩和稀疏矩阵来进行特征提取。然而，RPCA可能在准确分离低秩和稀疏部分方面存在一定的挑战。如果RPCA不能准确地分离出低秩部分，那么它可能会丢失一些重要的特征信息，从而降低分类性能。

**五、 参考文献**

[1]https://zhuanlan.zhihu.com/p/33191810

[2]<https://blog.csdn.net/NSSWTT/article/details/107207823>

[3]https://zhuanlan.zhihu.com/p/37777074

[4]https://www.jianshu.com/p/3dcdc5055e4f

[5]https://zhuanlan.zhihu.com/p/144766191

[6]https://www.jianshu.com/p/6d8000d53bd7

[7]<https://blog.csdn.net/qq_20199965/article/details/102657192>