哈爾濱工業大學

人工智能数学基础实验报告

题　　目	基于 PCA 和 RPCA 的数据集分类
学　　院	计算机科学与技术
专　　业	人工智能
学　　号	2022113416
学　　生	刘子康
任 课 教 师	刘绍辉

哈尔滨工业大学计算机科学与技术学院

2024.3

实验二: 利用 PCA 和 RPCA 对 MINIST 数据集分类

注意：请按照大家阅读文献的格式进行撰写，确保文档格式的规范性！不能拷贝粘贴，尤其是图和公式，不允许拷贝粘贴，需要自己编辑！或者用 latex 编辑！

一、 实验内容或者文献情况介绍

理解主成分分析（PCA）和鲁棒主成分分析（RPCA）的基本原理，并利用 PCA 和 RPCA 对 MINIST 手写数字数据集进行降维，并利用逻辑回归和 KNN 等分类算法对降维后的数据集进行分类。

二、 算法简介及其实现细节

2.1 PCA 降维

2.1.1 算法简介：

PCA 是一种将数据的多指标转化为少数几个综合指标，从而降低数据集维数和简化数据集的主成分提取技术。PCA 的主要思想是将 n 维特征通过投影映射到 k 维空间上，k 维的正交特征即为数据集的主成分。

在实现过程中，寻找原始数据空间的多组正交坐标轴，使得原始数据投影到第一个坐标轴时数据方差最大（即投影误差最小，保留信息最多），第二个坐标轴应在与第一个坐标轴正交的平面中，并且使得投影方差最大，以此类推。可以发现，前 k 个主成分几乎涵盖绝大部分的信息，而后面的坐标系对应的方差几乎为 0，那么就可以保留数据的前 k 个主成分，从而实现降维。

2.1.2 实现细节：

首先计算数据集矩阵的协方差矩阵，得到协方差矩阵的特征值和特征向量，然后将特征值降序排序，选取最大的前 k 个特征值对应的特征向量组成的矩阵，最后将原始数据通过该矩阵投影到新的数据空间，即可实现降维。

由于直接计算数据集矩阵（设为矩阵 A）的协方差矩阵比较困难，故对矩阵 A 进行奇异值分解获取对应协方差矩阵的特征值和特征向量，该过程可以通过 Numpy 库函数计算 A^TA 矩阵的特征值和特征向量，或调用 Scipy 库的 svd 函数实现。

2.2 RPCA 降维

2.2.1 算法简介：

RPCA 用于处理存在缺失、损坏或受到噪声污染的数据矩阵，使其恢复为有效数据，主要思想是将数据矩阵（观测矩阵）分解为一个低秩矩阵 L（即有效特征）与一个稀疏矩阵 N（即噪声）的和，矩阵 L 即为所求矩阵。

该方法将原问题形式化为 $\min_{L,N} rank(L) + \lambda\|N\|_0, s.t. \quad M = L + N$，通常情

况下 $\lambda = \frac{1}{\sqrt{max(m,n)}}$，其中 m 和 n 分别是观测矩阵的行和列数。由于矩阵 L 的秩和矩阵 N 的 l_0 范数是非凸的，不易于求解。而 l_1 范数是 l_0 范数的最佳凸松弛，矩阵核范数是 rank(·)函数的最佳凸松弛，因此原问题可转换为求解$\min_{L,N} ||L||_* + \lambda||N||_1, s.t.\ M = L + N$。矩阵核范数指矩阵奇异值的和，核范数越小可近似认为秩越低；l_1 范数用矩阵所有元素绝对值的和表示，当其很小时可近似认为矩阵是稀疏的。

2.2.2 实现细节：

通过增广拉格朗日乘子法（ALM）和交替方向乘子法（ADMM）求解，构造拉格朗日函数$L(A,E,Y,\mu) = \|A\|_* + \lambda\|E\|_1 + < Y, D - A - E > + \frac{\mu}{2}\|D - A - E\|_F^2$，其中 Y 为拉格朗日乘子。在每次迭代中，矩阵 A 和 E 通过软阈值函数（Soft Thresholding）更新，该函数主要用于求解形如 $T_\varepsilon(M) = \arg\min_X \varepsilon\|X\|_1 + \frac{1}{2}\|X - M\|_F^2$ 的优化问题。

2.3 数据集分类

逻辑回归和 KNN 算法是十分常用的分类算法，前者通过一个逻辑函数将特征和目标的线性回归结果转换为 0~1 的概率，并通过设定的阈值不断二分类；后者将每个数据的特征的 n 维向量对应于特征空间的一个点，对于测试数据，计算距离其最近的 k 个点，将该数据归为 k 个点中所属最多的类别。

使用时调用 sklearn 库的 LogisticRegression 函数和 KNeighborsClassifier 函数即可。

三、 实验设置及结果分析（包括实验数据集）

首先导入 MNIST 数据集，将数据集下载到本地并通过 mnist.py 的 load()函数调用，内容分别是 50000×784 的训练数据矩阵、50000 个元素的对应标签数组、10000×784 的测试数据矩阵和 10000 个元素的对应标签数组。

3.1 PCA 降维

通过奇异值分解求得数据矩阵的协方差矩阵的特征值和特征向量，设置降维后的维数 n 为大于 20 的特征值个数，降序排序后选取前 n 大的特征值对应的特征向量组成投影矩阵，然后通过矩阵乘法将原数据矩阵投影到新的特征空间，最后使用逻辑回归和 KNN 算法进行分类并绘图。

```
PCA降维：784维->349维
逻辑回归分类准确率：0.9218
KNN分类准确率：0.9671
```

图 1：PCA 降维结果和分类准确率

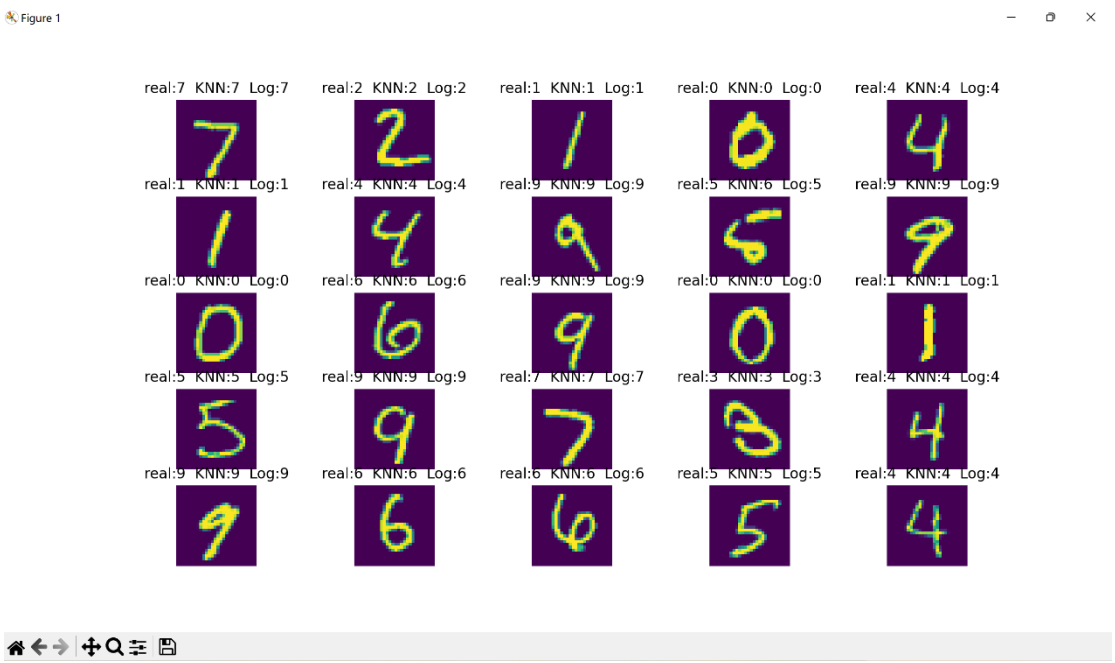

图 2：PCA 降维后分类部分结果

3.2 RPCA 降维

首先初始化矩阵 L、S 和 Y，分别用奇异值收缩算法 SVT 和软阈值函数更新矩阵 L 和 S，并更新 Y，迭代 100 次，利用得到的低秩矩阵 L 进行数据集分类的训练，并以此进行测试集的分类。

```
iter: 0000  err: 0.000494   rank(L): 705    card(S): 5796709
iter: 0010  err: 0.000016   rank(L): 649    card(S): 5917611
iter: 0020  err: 0.000023   rank(L): 615    card(S): 6094186
iter: 0030  err: 0.000028   rank(L): 593    card(S): 6334057
iter: 0040  err: 0.000027   rank(L): 572    card(S): 6590702
iter: 0050  err: 0.000030   rank(L): 557    card(S): 6889462
iter: 0060  err: 0.000027   rank(L): 543    card(S): 7162657
iter: 0070  err: 0.000028   rank(L): 531    card(S): 7483870
iter: 0080  err: 0.000028   rank(L): 521    card(S): 7817185
iter: 0090  err: 0.000029   rank(L): 514    card(S): 8170153
RPCA降维: 784维->510维
逻辑回归分类准确率：0.9192
KNN分类准确率：0.9705
```

图 3：RPCA 降维结果和分类准确率（迭代 100 次）

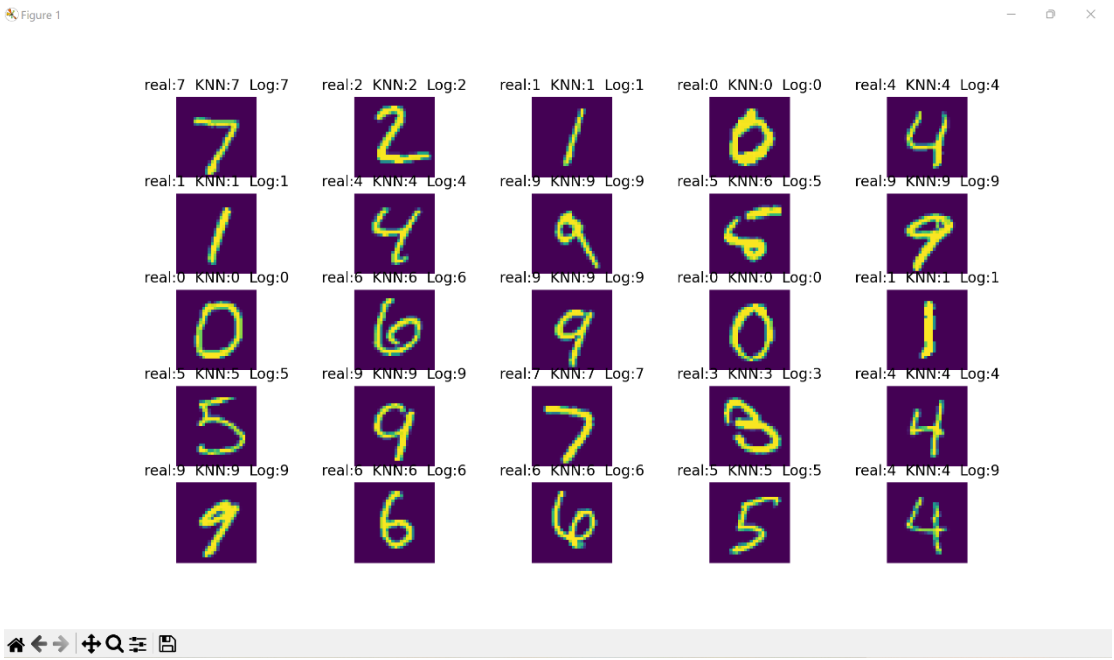

图 4：RPCA 降维后分类部分结果（迭代 100 次）

四、 结论

Robust PCA 与经典 PCA 一样，本质上也是寻找数据在低维空间上的最佳投影问题。由于 MNIST 数据集并不大，PCA 和 RPCA 方法均可以较好地将 MNIST 数据集简化，降维后数据集分类的准确率均较高。

PCA 假设数据集的噪声为高斯噪声，降维效果容易受到噪声较大或离群点较严重情况的影响，相比之下 RPCA 假设数据集的噪声是稀疏的，可以恢复出低秩的有效数据，更适用于受噪声污染较严重的数据集；但同时 RPCA 的计算更为复杂，需要迭代一定次数才能达到较好效果，而 PCA 相对计算速度更快。

五、 参考文献

[1] lys_828.【机器学习】PCA 主成分项目实战：MNIST 手写数据集分类[EB/OL].CSDN 博客,2022-01-23. https://blog.csdn.net/lys_828/article/details/122651759

[2] bwqiang.全面理解主成分分析（PCA）和 MNIST 数据集的 Python 降维实现[EB/OL].CSDN 博客,2021-01-11. https://blog.csdn.net/bwqiang/article/details/110407382

[3] mk12306.主成分分析（PCA）原理和鲁棒主成分分析（RPCA）详解[EB/OL].CSDN 博客,2019-10-21. https://blog.csdn.net/qq_20199965/article/details/102657192

[4] masonwang_513.拉格朗日乘子解 Robust PCA 以及 Python 实现[EB/OL].CSDN 博客,2018-03-13. https://blog.csdn.net/reform513/article/details/79539511

[5] sui_qiang_kaixin_.RPCA 的算法推导-增广拉格朗日乘子法- PPT 讲解[EB/OL].CSDN 博客,2019-12-01. https://blog.csdn.net/weixin_45670912/article/details/103339238