实验报告

降维技术的实验性应用：PCA和流形学习算法（MDS）

实验目的

- 理解和实现PCA（主成分分析）和MDS（多维缩放）算法。
- 应用这些算法对MNIST数据集进行降维，并分析重构误差和降维效果。
- 通过可视化手段，探索降维后数据的特性。

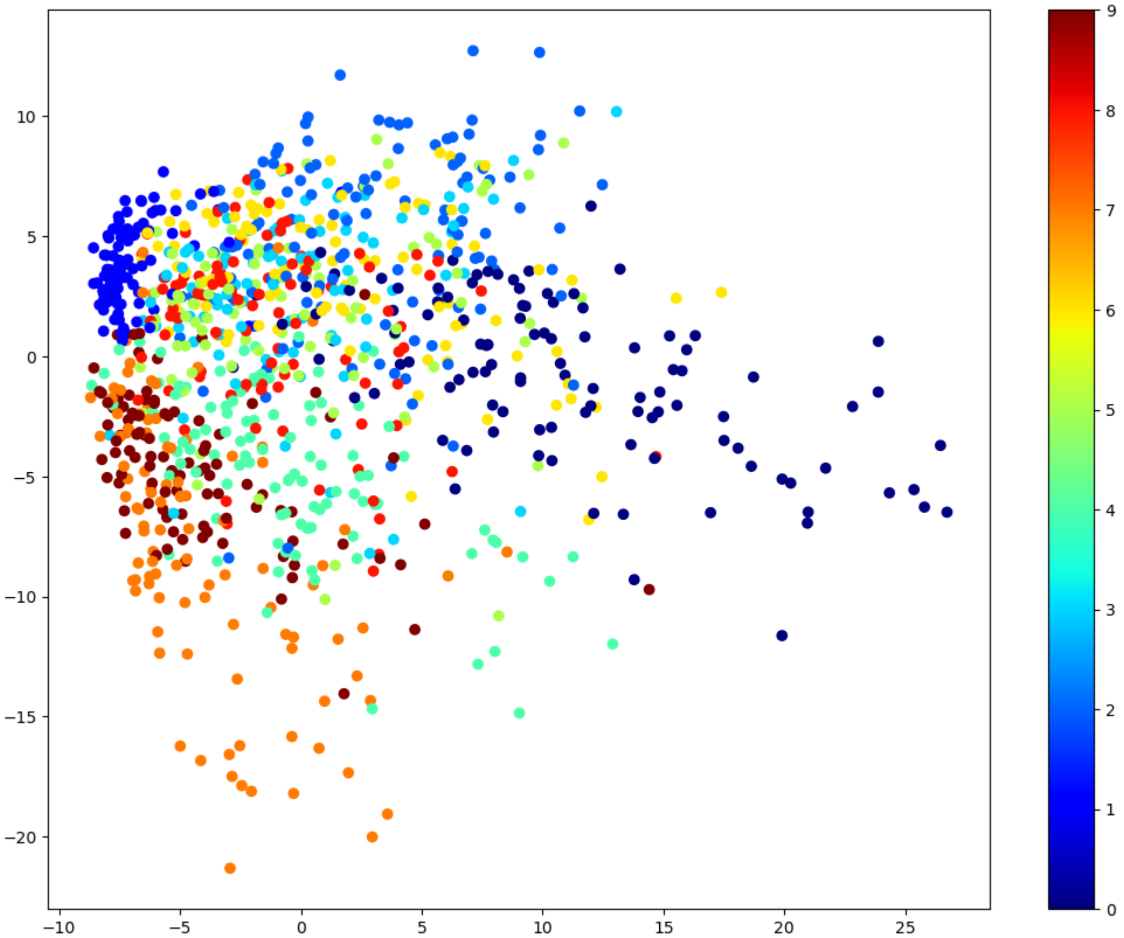
理论回顾

1. **PCA（主成分分析）**
    - PCA旨在从高维数据中提取主要特征，通过正交变换将原始数据转换为一组线性不相关的变量，称为主成分。
    - PCA的数学基础是协方差矩阵的特征值分解或奇异值分解。
2. **MDS（多维缩放）**
    - MDS是一种流形学习算法，用于将高维空间中的数据点映射到低维空间，同时尽可能保持原始数据中点对之间的距离。
    - MDS通过最小化高维空间和低维空间中距离之间的差异来实现降维。

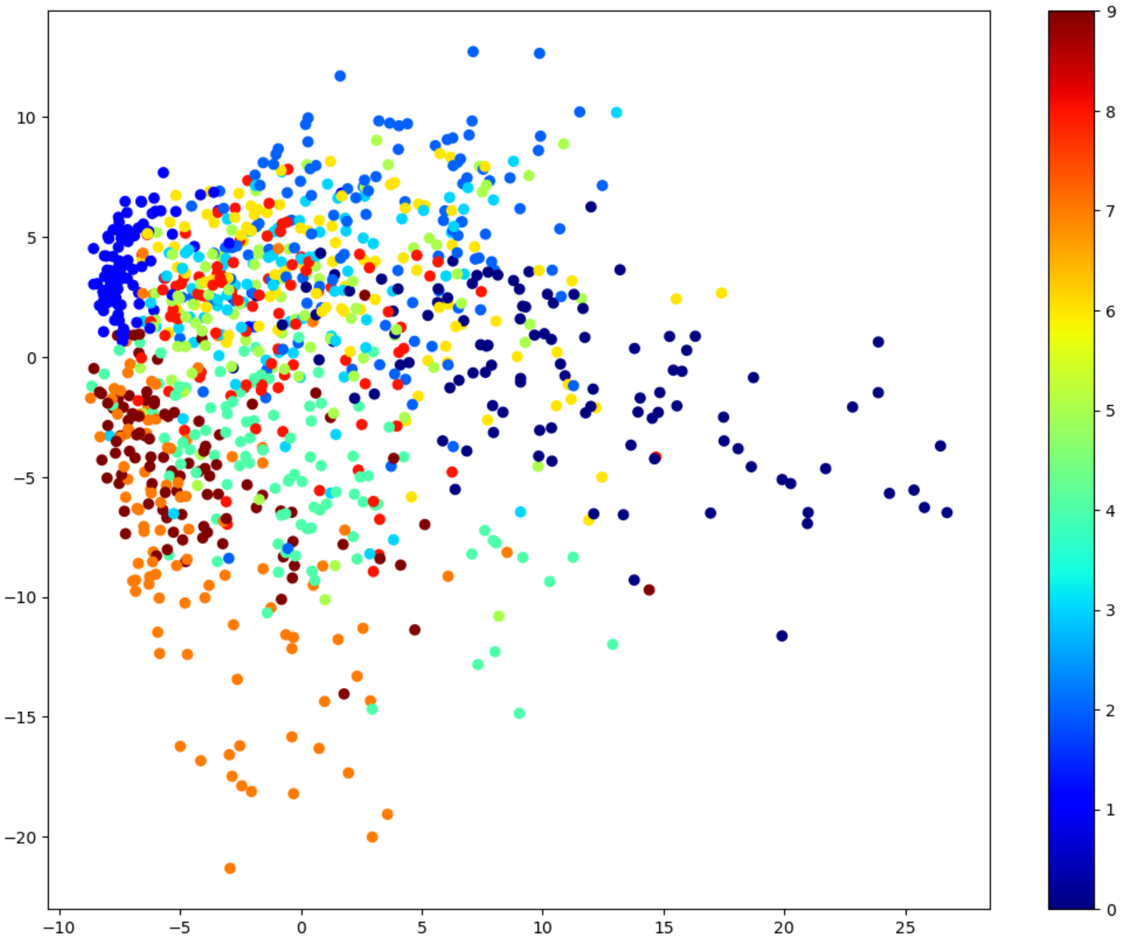
实验步骤和结果

1. **PCA的实现与应用**
    - 实现了PCA算法，并在MNIST数据集上进行了测试。使用不同数量的主成分进行数据重构，并计算重构误差。
    - 结果表明，随着主成分数的增加，重构误差减少，但也伴随着计算复杂度的提高。

> 标注库产生效果


10
5
0
−5
−10
−15
−20
−10
−5
0
5
10
15
20
25
9
8
7
6
5
4
3
2
1
0

> 我的pca效果


10
5
0
−5
−10
−15
−20
−10
−5
0
5
10
15
20
25
9
8
7
6
5
4
3
2
1
0

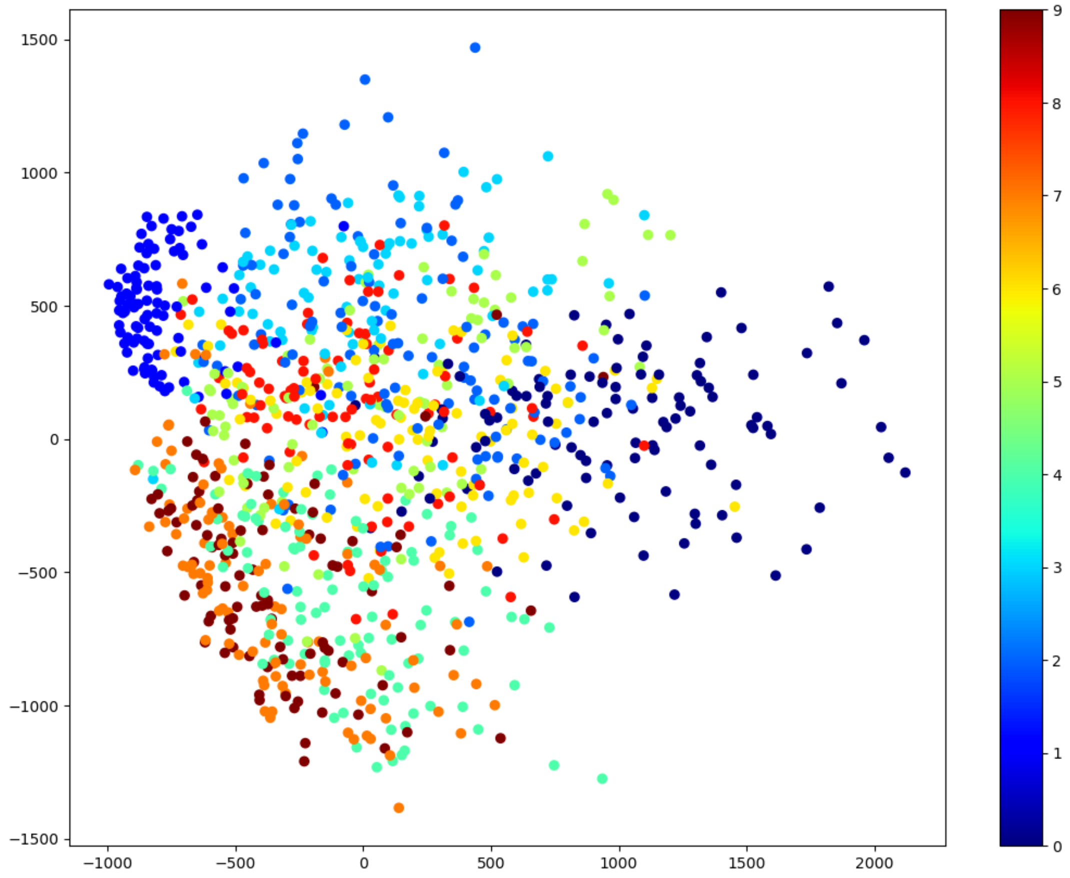
- 误差

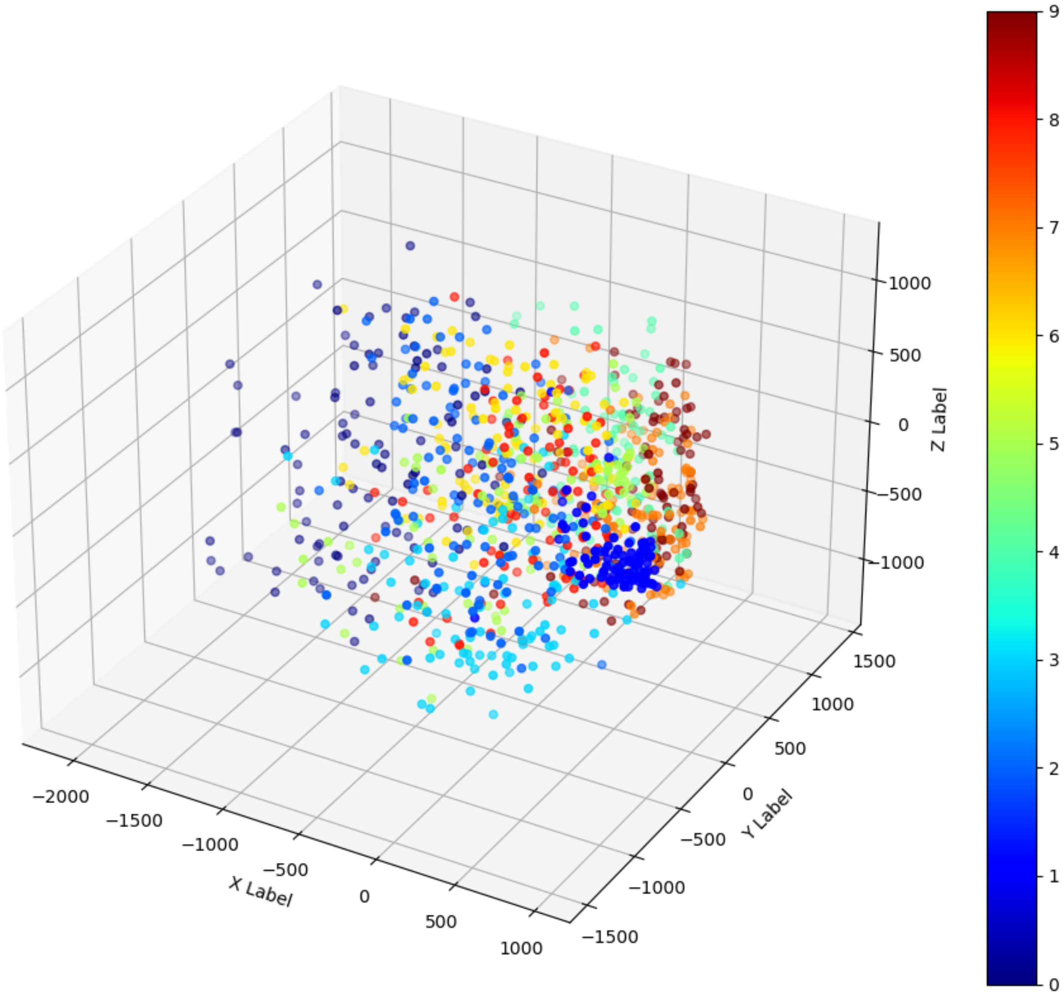
```python
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import numpy as np

# 假设 X 是你的特征矩阵


# 首先标准化数据
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_reduced_my_pca = pca(X, 2)
X_reduced_sklearn_pca = PCA(n_components=2).fit_transform(X_scaled)
# 计算两个降维结果之间的误差，欧式距离除以矩阵范数
#print(X_reduced_my_pca)
#print(X_reduced_sklearn_pca)
# 由于存在正负号的问题，所以需要取绝对值

error = np.linalg.norm(np.abs(X_reduced_my_pca) -np.abs(X_reduced_sklearn_pca) ) / np.linalg.norm(X_reduced_sklearn_pca)
print(f"error = { error}")

```

✓ 0.3s

```
error = 6.201489732936502e-05
```

由此可见误差比较小，推测是不同计算方式导致的浮点精度问题，结果大致相同，可能由于特征向量的符号选择导致某一维符号不同。

2. **MDS的实现与应用**
   - 实现了MDS算法，对MNIST数据集进行降维至二维和三维空间。
   - 



- 可视化结果显示，MDS能够在降维后的空间中有效地区分不同类别的数据点。

结果分析

- PCA是一种有效的降维技术，特别适用于去除数据中的噪声和冗余信息。
- MDS提供了对数据结构的深入洞见，尤其在探索数据的内在维度和模式时。
- 在MNIST数据集上，两种方法都能有效降维，但MDS在保持数据结构方面表现更佳。

结论

通过实验，我们成功地实现和应用了PCA和MDS算法。这两种算法在MNIST数据集的降维任务中表现出了各自的优势和特点。PCA在处理线性关系时效果显著，而MDS更适合探索和可视化数据的非线性结构。