

# Report

## 实验目标

实现k-means聚类算法对MNIST数据集进行聚类，并在真实数据集上进行评测。

## 实验设计

代码中定义了class `k_means`，其中包含了以下方法：

`__init__` 初始化方法，初始化聚类中心个数、初始聚类中心、初始聚类结果。

`update_label` 根据新的聚类中心更新聚类结果

`iterate` 迭代一次，更新聚类中心和聚类结果

`max_distance` 计算所有点中到聚类中心的最大距离

`calc_label` 根据训练集中标注的结果，根据多数投票得到该聚类的标注

之后，在主程序中，读取数据集，调用 `k_means` 类，进行训练和评测，最后将聚类标注的结果可视化展示。

其中，由于不同标签的个数为 10 故在实验中分别取  $k = 10, 20, 50$  作为聚类的数量。

初始的聚类中心是直接随机

$k$  个数据中的点作为中心。

对于每张图片，直接使用

$256 \times 256$  的矩阵来表示。两张图片之间的距离使用Frobenius矩阵范数来进行计算  $\|A - B\|_F$

k-means 的终止条件设置为迭代之后只有少于

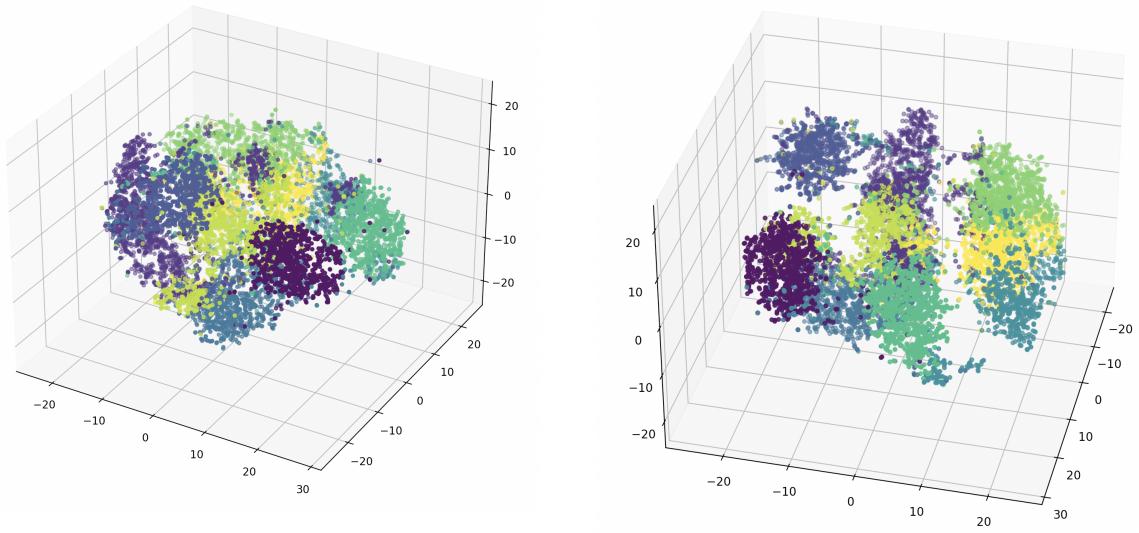
0.1% 的点的聚类结果发生变化。

## 实验结果

以下表格展示分别取  $k = 10, 20, 50$  时，矩阵范数为  $LF$  范数时的正确率：

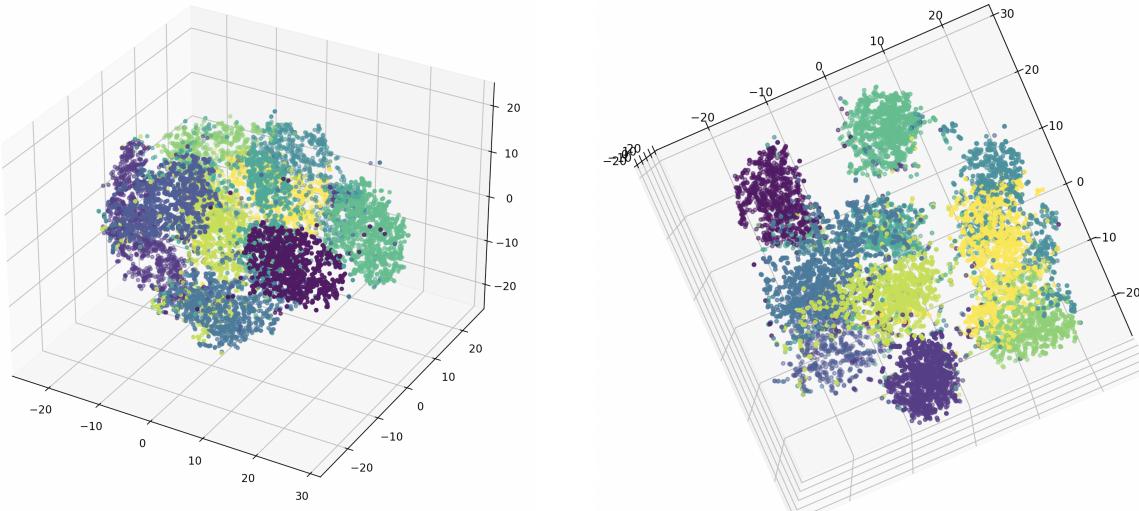
$k =$	10	20	50
Accuracy	0.5778833333333333	0.72385	0.81625

以下图片为聚类结果的可视化展示（取样本中的 10000 个点进行可视化），将所有向量降维至三维后进行展示，每次运行结果展示了两个视角的图片。



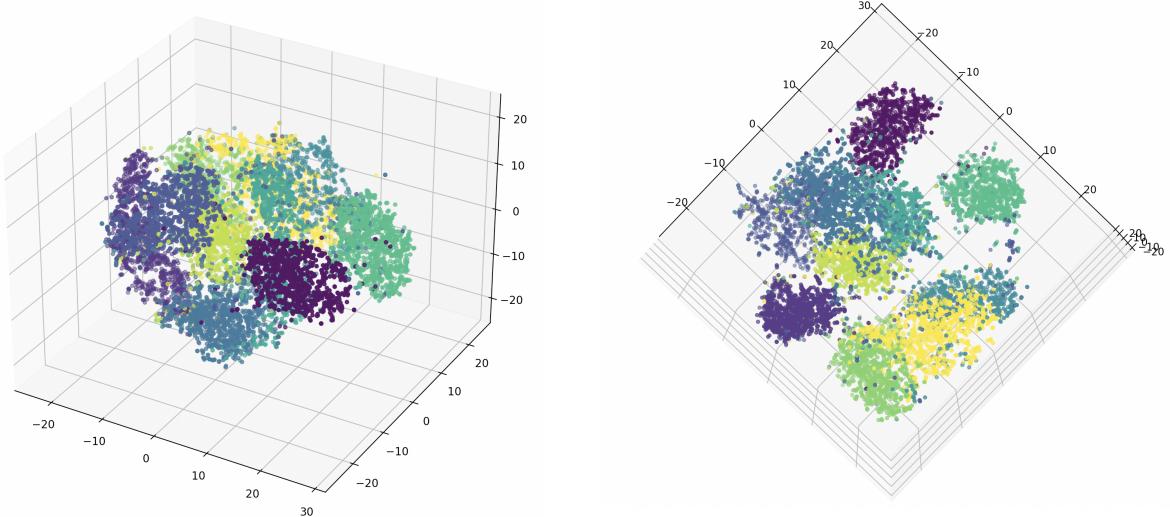
$k = 10$  , 图1

$k = 10$  , 图2



$k = 20$  , 图1

$k = 20$  , 图2



$k = 50$  , 图2

$k = 50$  , 图1

以下表格展示了取  $k = 10$  , 矩阵范数分别取

范数	$L_F$	$L_1$	$L_2$	$L_\infty$
$k = 10$	0.5778833333333333	0.48625	0.6221833333333333	0.4624

## 实验分析

在实验过程中，在一个细节处理上调试了很久：由于直接使用 `datasets` 库读取图片数据后，张量内数据类型为。由于是无符号型整数，故在之后计算矩阵差的范数时会导致负数变为大整数，影响距离判断的结果。

通过分析不同  $k$  取值的运行结果，可以发现在一定范围之内正确率随着  $k$  的增加而变大。推测是由于聚类中心变多，则对于每一个样本归类到正确的标签类别的可能性增加，因此会增加正确率。

观察实验结果中的可视化结果，可以发现k-means算法的确很好的将相似的图片归类在了一起，但还是有部分零散的点没能很好的正确分类，而随着  $k$  值的增加，未正确归类的点有较为明显的减少。

观察实验结果中不同范数的选取对于正确率的影响，可以发现在  $k = 10$  时，使用  $L_2$  范数的效果最好， $L_F$  范数其次，而  $L_1$  和  $L_\infty$  范数的效果则略逊一些。推测是由于  $L_1$  和  $L_\infty$  范数求得的是最大的某行（列）绝对值之和，对于整体的相似程度度量有所偏差。

以下为在  $k = 10$  范数取  $L_F$  时，几个归类错误的图片（以打点形式展示）：

5 (实际为3)

```
.....  
.....  
.....  
.....  
.....  
.....##.  
.....#####.  
.....######.
```

.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....

1 (实际为5)

.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....  
.....#####.....

7 (实际为2)

```
.....#####.....  
.....#########..  
....###....##..  
....##....###..  
....#....##...  
.....##...  
.....###...  
....###...  
....###...  
....##...  
.....#####..  
.....########..  
....########..  
....###....###..  
....##....###..  
....##....###..  
....#####....###..  
....#####....###
```

1 (实际为6)

```
.....##...  
.....##...  
....##...  
....##...  
....##...  
....##...  
....##...  
....##...
```

```
.....#
.....##
.....#
.....##...###...
.....#...#...#
.....#...#...##
.....#.....#
.....#.....#
.....#.....#
.....#.....##...
.....#.....##...
.....#.....##...
.....#####...
```

.....

.....

.....

.....

可以发现有一部分图片本身难以确认，但的确有一部分图片是特征较为显著但却分类错的。