实验2- K-means聚类实验报告

陈意扬 计96 2019011341

实验目的

1）实现k-means算法

2）在真实数据集上评估k-means聚类性能

3）分析实验结果

实验原理

K-Means算法步骤：

1.确定k个类的类中心向量

2.对于集合中的每个对象，计算出距离最近的类中心，并将对象归于该类

3.重复上述步骤

对于“确定k个类的类中心向量”，第二轮及以后的轮次的相应策略是计算类中所有向量的均值作为新的类中心向量，初始的类中心向量一般是随机设置。基于一些策略来确定初始的类中心向量可以减少收敛所需要的迭代次数，这也是接下来会讨论的话题。

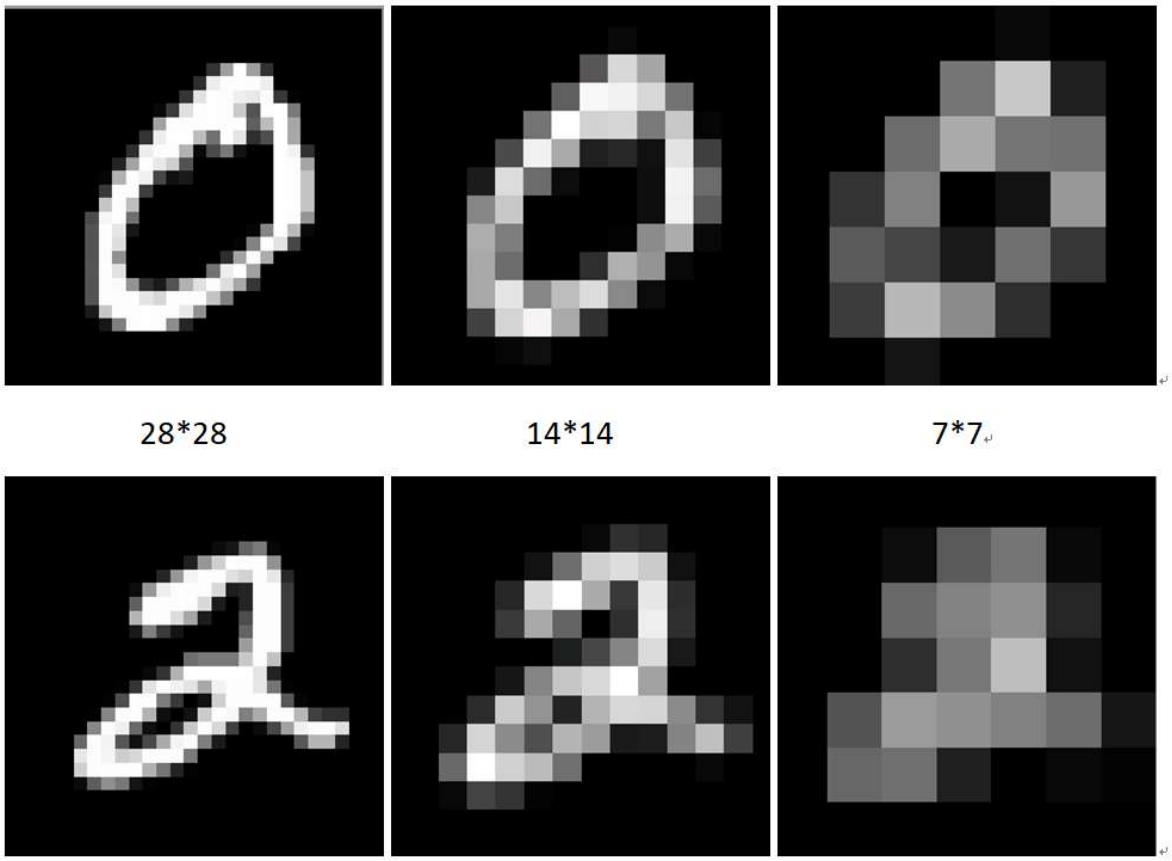
实验步骤

1.切分数据集并提取特征（parse_email.py）

实验数据集来自经典的MNIST dataset。training set中总共60000个手写数字以28x28的image object形式存储。

1.向量化图片

一种想当然的做法是将28x28像素展平成784维的向量。对比28x28的图片和resize成14x14的图片，我发现抽样的所有图片在resize成14x14后都能保留原来的所有信息，7x7则丢失大部分信息。于是我第一步将图片resize成14x14来进行降维。



但是我考虑到一个问题，在实际图片中，像素之间的差别有大小之分。例如（1,1）和（1,2）像素，它们在图片中的差别并不是很大，而（1,1）和（14,14）的差别就会比较明显。展成向量后，（1,1）和（1,2）之间的关系与（1,1）和（14,14）之间的关系在计算距离时就变得等价了，这是不合理的。所以我的策略是在保留已有信息的同时加入近一步模糊化的信息（模糊化可以使靠得很近的像素合并）。经过简单的调参我发现6倍的7x7像素图片+1倍的14x14像素图片合并起来的向量可以得到最好的`acc`（加上6倍的7x7像素信息之前是55%左右，加上后达到58%左右），近一步的模糊化已经几乎丢掉所有信息（如4x4），所以没有必要。

2.选择初始类中心

在选择类中心之前首先要确定K-Means中的k值。首先原数据集是0~9的手写数字，所以k=10是必要的选择，且可以作为一个临界值。小于10的k值我选择了5，大于10的k值我选择了一个20。过于小的k值聚类上意义不大，过于大的k值会导致收敛极慢。

根据聚类的个数有相应的初始类中心的选择策略。对于10聚类，我的策略是在原数据集10种label中每种随机抽取一张图片向量化作为初始类中心，这是因为原数据集本身就有ground truth的分类，从ground truth的分类中抽取作为类中心从理论上可以获得更快的收敛速度。对于5聚类，我同样随机抽取5张图片作为初始类中心，同时保证5张图片的label不同。对于20聚类，将数据集切为两半，用10聚类的选取方法在两半中各取一次合在一起。

3.计算向量之间的距离

我在10聚类中分别尝试了欧氏距离和曼哈顿距离并在最终的模型中选择了欧氏距离来进一步尝试5聚类和20聚类，调用`torch`库的`dist`方法可以加快速度。

10聚类模型中欧氏距离和曼哈顿距离的表现（每种重复5次取平均）：

	欧氏距离	**曼哈顿距离**
acc	57.37%	47.79%
收敛轮数	88	110

可见欧氏距离相比较曼哈顿距离收敛更快且准确率更高。

4.终止迭代的时机

我设置迭代轮数上限为200轮。每轮迭代过后我会计算所有类中心相比前一次的移动距离，当移动距离全部为0后即表示完全收敛，此时我会终止迭代。若200轮后未收敛也会终止迭代。5聚类和10聚类大多在100轮内收敛，20聚类大多会在150轮内收敛。

模型的评价

1.准确率（`acc`）

首先需要确定聚类的label。我选取该类中占比最大的label（不一定超过半数，即不是majority voting）作为该类的label。

以下列举5means、10means和20means各三次独立实验的最终收敛情况。

实验编号	[label]：[单类的`acc`]				总acc
1	0：91%	1：41%	3：39%	6：44%	7：31%	43.37%
2	0：92%	1：43%	3：36%	6：45%	7：30%	43.38%
3	0：92%	1：43%	3：36%	6：45%	7：30%	43.38%

实验编号	[label]：[单类的`acc`]									总acc
1	0：80%	0：88%	1：48%	1：33%	2：89%	3：47%	4：39%	6：83%	7：93%	9：37%	57.12%
2	0：90%	1：44%	1：43%	2：88%	3：45%	4：41%	6：76%	7：88%	8：31%	9：42%	57.32%
3	0：91%	1：44%	1：43%	2：79%	3：43%	4：43%	6：85%	7：91%	8：31%	9：41%	57.45%

实验编号	[label]:[单类的acc]									总acc
1	0：94%	0:96%	1:75%	1:79%	2:95%	2:93%	3:89%	3:52%	3:43%	4:55%	
	4：24%	5:66%	6:90%	6:87%	7:87%	7:95%	8:57%	8:75%	9:51%	9:47%	69.45%
2	0：93%	0:91%	0:97%	1:73%	1:81%	2:93%	2:95%	3:49%	3:47%	4:28%	
	4：35%	5:53%	5:61%	6:91%	6:91%	7:67%	7:96%	8:57%	8:75%	9:50%	69.93%
3	0：94%	0:96%	1:75%	1:79%	2:95%	2:93%	3:89%	3:52%	3:43%	4:55%	
	4：24%	5:60%	5:70%	6:91%	6:91%	7:67%	7:96%	8:57%	8:75%	9:48%	69.17%

整体的准确率

	5-means	10-means	20-means
acc	43.38%	57.30%	69.52%
收敛所需轮数	76	88	157

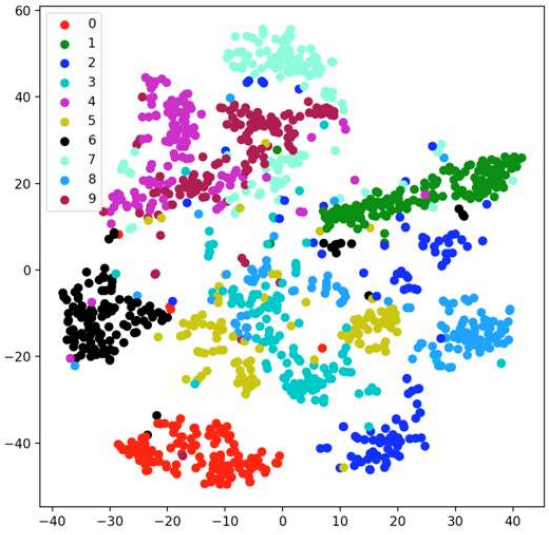
分析实验结果可以得到以下信息：

1.准确率随着k值升高而提升，且最终收敛后相同k值的准确率波动极小、几乎确定——不同初始类中心的收敛结果几乎相同；

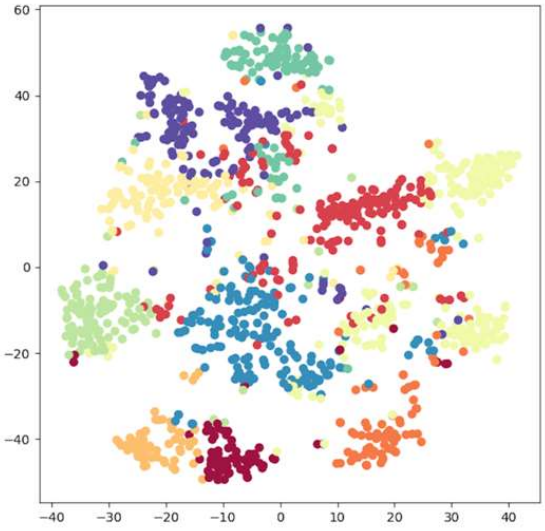
2.最终收敛后相同k值类的label分布比较稳定，5-means完全固定（0、1、3、6、7），10-means出现5消失的情况，20-means10个label都有，大概率每种label出现2次，但会在0、3、5、9上波动。值得注意的是有一些label在3种聚类中都有较好的效果，如0、2、6、7，而有一些label普遍较差，如3、4、8、9，还有一些label在不同的聚类上有差异性的表现，如1。结合实际推测3、4、8、9在手写体中的确形似（3和8，4和9）。在接下来的可视化部分会进一步讨论。

2.可视化（tsne.py）

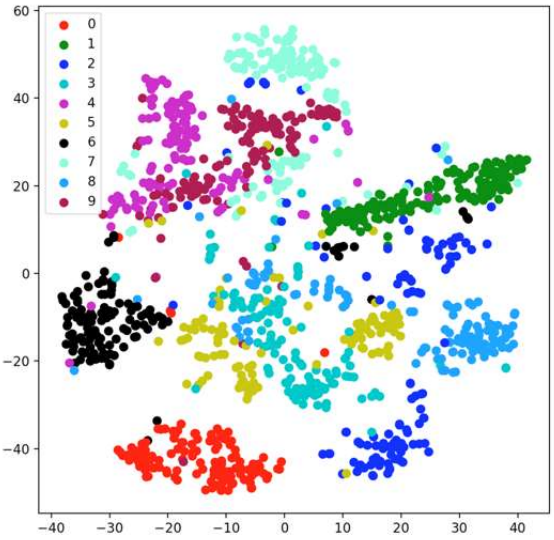
在数据集中我按各个label的比例（50取1）从60000个样本中sample出1200张图片。利用`t-SNE`算法将14x14=196维向量降成2维。利用label区别颜色，调用`matplotlib`库绘制图片作为ground truth（`./tsne`目录下）。在训练过程中，我每5轮迭代进行一次可视化，可视化选择之前sample的点，根据训练时的聚类给它们分配颜色。实验结果如下（收敛时的结果）：



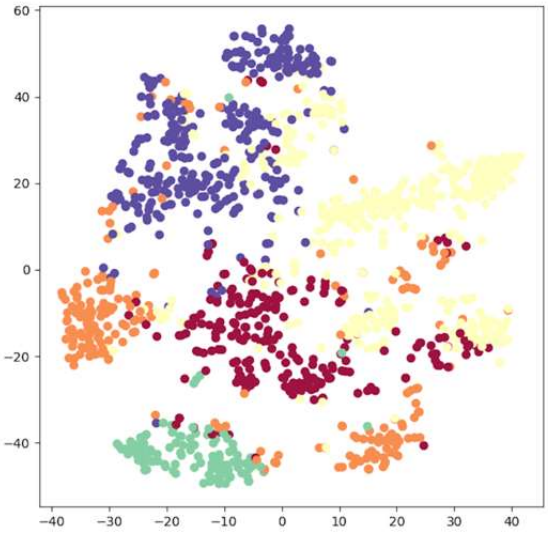
ground truth



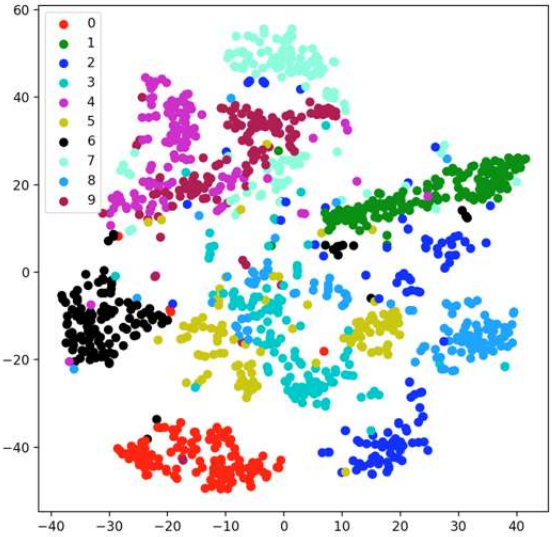
10-means



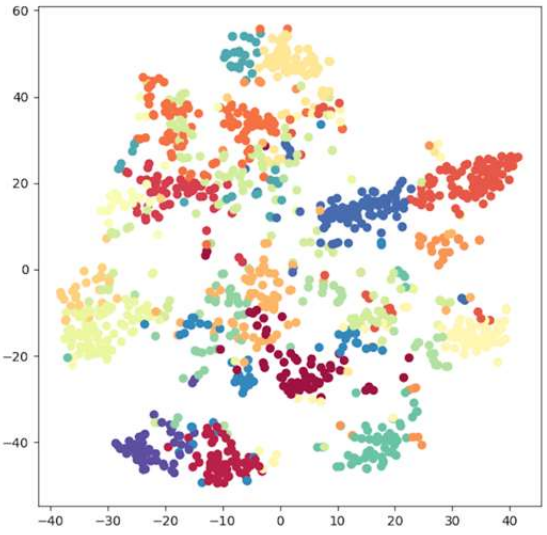
ground truth



5-means

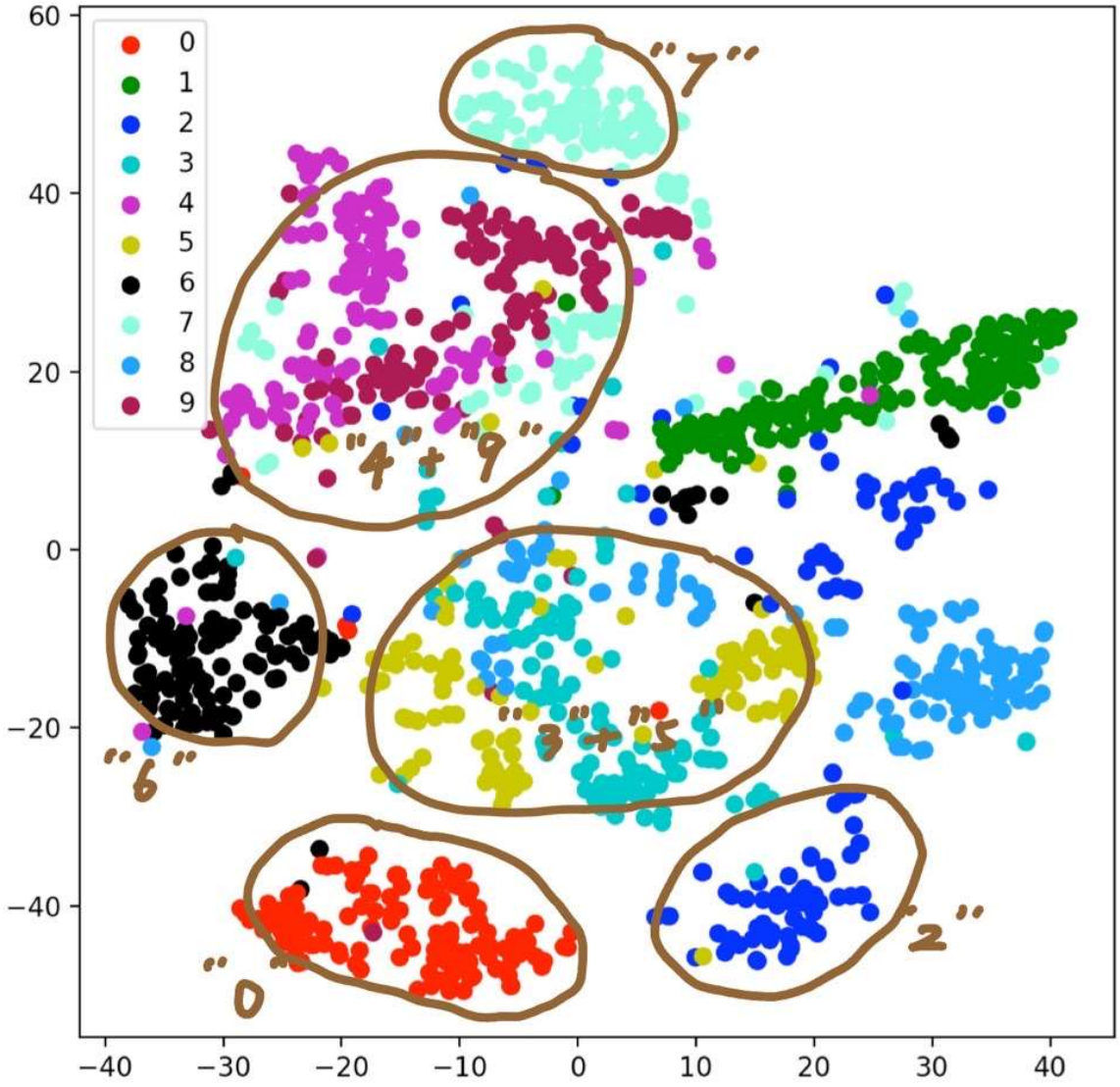


ground truth



20-means

通过观察ground truth可知0、2、6、7这4个label有明显独立的、离其他点较远的cluster，在各种聚类上都能把这些cluster很好的分出来，解释了`acc`上普遍较高的实验结果；而3和5混杂在一起，4和9混杂在一起，在5-means和10-means上几乎被聚为一类，也解释了这几类`acc`上普遍较低的实验结果。



实验结论

在MNIST dataset上10聚类的准确率并不高，20聚类准确率稍有提高但收敛所需轮数会骤增，k值更大的聚类收敛所需时间进一步增加且由于k值和ground truth的label数差距太大而失去实际意义。