统计分析第四章作业

16337183 孟衍璋

实验目的

理解K-均值聚类方法。

实验要求

手写体字符识别MNIST数据集是计算机视觉最为经典的一个数据库。里面包含0~9的10个手写字符。每张图由28×28的灰度图片组成。本题为了简化问题，只考虑训练集60000张图（见附件的数据train-images.idx3-ubyte），每个字符6000张图，共10个字符。请用kmeans进行对手写体字符进行聚类，其中固定k=10，即聚成10簇。要求：

1） kmeans的代码必须自己写，不能利用相应的kmeans库函数。

2） 聚类完，请分析聚类的精度。

3）提示：每张图看出28×28=784维度的高维空间的数据点，60000张图就相当于60000个数据点。

实验步骤

读取数据

读取数据分为两个函数，第一个是读取图片，第二个是读取标签：

```
def images_loading(file):
    data = file.read()
    head = struct.unpack_from('>IIII', data, 0)
    offset = struct.calcsize('>IIII')
    number = head[1]
    width = head[2]
    height = head[3]

    bits = number * width * height
    bitsString = '>' + str(bits) + 'B'
    images = struct.unpack_from(bitsString, data, offset)
    images = np.reshape(images, [number, width *height])
    return number,images
```

```
def labels_loading(file):
    data = file.read()
    offset = 0
    header = '>II'
    head = struct.unpack_from(header, data, offset)
    offset = struct.calcsize('>II')
    labels = struct.unpack_from('>60000B', data, offset)
    labels = np.reshape(labels, [60000])
    return labels
```

设置初始类

读取数据之后，就可以开始设置初始类，先使用随机数选取一个起点，再选取这个起点往后的10张图片，每张图片分别划分到10个初始类中。

```
k = 10  # 初始类的个数
start_from = random.randint(0,59989)
cluster = images[start_from:(start_from+k),:]  # 随机使用连续的k张图初始化k个类
```

迭代

然后便进入迭代，每轮迭代需要完成以下步骤：

计算距离

计算总体中每个样品到每个类均值的距离，与哪个类之间的距离最小，便加入那个类，此信息存储在*cluster_belong_to*中：

```
# 计算每张图片应该归于哪个类
for i in range(60000):
    distance = []
    for j in range(k):
        distance.append(european_distance(images[i,:], cluster[j,:]))
    cluster_belong_to[i] = np.argmin(distance)
```

计算新的类均值

```
for i in range(60000):
    n = cluster_belong_to[i]
    new_cluster[n,:] = new_cluster[n,:]+ images[i,:]
```

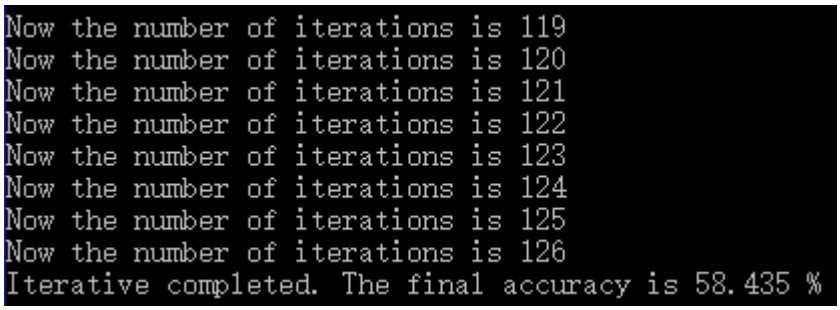
计算每个类对应的标签

```
G = np.zeros([10,10],dtype = np.int)  # 计算每个类中标签最多的是哪个
for i in range(60000):
    G[cluster_belong_to[i],labels[i]] = G[cluster_belong_to[i],labels[i]] + 1
labels_of_cluster = np.argmax(G, axis = 1)
```

判断继续迭代与否

根据各个类的均值是否有修改，判断是否继续迭代，如果均值没有修改，就跳出循环停止迭代，输出结果。

实验结果



通过126轮迭代，最后的正确率为58.435%