# 模式识别实验报告

## 实验一 K-均值聚类

学院：计算机学院

姓名： 张景润

学号：1172510217

## 一、实验内容

1、 使用 python 或 Matlab 编程实现 K-均值聚类算法：要求独立完成算法编程，禁止调用已有函数库或工具箱中的函数；

2、 使用仿真数据测试算法的正确性：将下列 19 个样本聚成 2 个聚类：

$$\mathbf{x}_1=(0,0)^t,\mathbf{x}_2=(1,0)^t,\mathbf{x}_3=(0,1)^t,\mathbf{x}_4=(1,1)^t,$$
$$\mathbf{x}_5=(2,1)^t,\mathbf{x}_6=(1,2)^t,\mathbf{x}_7=(2,2)^t,\mathbf{x}_8=(3,2)^t,$$
$$\mathbf{x}_9=(6,6)^t,\mathbf{x}_{10}=(7,6)^t,\mathbf{x}_1=(8,6)^t,\mathbf{x}_{12}=(7,7)^t,$$
$$\mathbf{x}_{13}=(8,7)^t,\mathbf{x}_{14}=(9,7)^t,\mathbf{x}_{15}=(7,8)^t,\mathbf{x}_{16}=(8,8)^t,$$
$$\mathbf{x}_{17}=(9,8)^t,\mathbf{x}_{18}=(8,9)^t,\mathbf{x}_{19}=(9,9)^t$$

3、 MNIST 数据集测试：ClusterSamples 中的 10000 个 784 维特征手写数字样本聚类为 10 个类别，根据 SampleLabels 中的标签统计每个聚类中不同样本的数量。测试不同初始值对聚类结果的影响。

## 二、程序代码

（K-均值算法部分代码）

```
"""
作者：张景润
学号：1172510217
"""
import csv
import random
import numpy as np
Num = 10  # 聚类的类别数目
Width = 784  # 样本的特征维度
Data_Path = './ClusterSamples.csv'  # 样本所在的文件
Ground_Path = './SampleLabels.csv'  # 真实聚类结果文件
Label_Path = './ClusterLabels.csv'  # 样本聚类标签输出文件
def read_data():
    global Width
    with open(Data_Path, 'r') as f:
        data_lst = []
        reader = csv.reader(f)
        for lst in reader:
            data_lst.append([int(item) for item in lst])
        Width = len(data_lst[0])
        return data_lst
def init_cluster_center(data_lst, init=1):  # 初始化聚类中心
    if init == 0:  # 随机初始化聚类中心
        center_lst = random.sample(data_lst, Num)
    elif init == 1:  # 选取批次距离尽可能远的 Num 个点
        center_lst = [get_center(data_lst)]  # 第一个样本点为所有样本点的质心
        while len(center_lst) < Num:
            dis, new_center = 0, []
```

```
            for data in data_lst:
                new_dis = min([cal_dis(data, center) for center in center_lst])
                if new_dis >= dis:
                    dis = new_dis
                    new_center = data
            center_lst.append(new_center)  # 选取距离已有中心点的最近距离最大的点作为新的中心点
    else:  # 待开发
        center_lst = init_cluster_center(data_lst, 0)
    return center_lst
def cal_dis(data1, data2):  # 计算两个样本的中心
    return sum([(item1 - item2) ** 2 for item1, item2 in zip(data1, data2)])
def get_center(data_lst):  # 获取 data_lst 的中心
    center = [0] * Width
    for data in data_lst:
        for idx, item in enumerate(data):
            center[idx] += item
    center = [float(item) / len(data_lst) for item in center]
    return center
def cluster(data_lst, center_lst):
    label_lst = [-1] * len(data_lst)
    flag, iter_count = True, 0
    while flag:
        flag = False
        for idx, data in enumerate(data_lst):  # 更新每一个聚类的标签
            dis_lst = [cal_dis(data, data2) for data2 in center_lst]
            min_idx = dis_lst.index(min(dis_lst))  # 找到与当前样本距离最近的聚类中心
            if min_idx != label_lst[idx]:
                label_lst[idx] = min_idx
                flag = True
        for idx in range(Num):  # 更新聚类中心
            tmp_lst = list(map(lambda item: item[1], filter(lambda item: item[0] == idx, zip(label_lst, data_lst))))
            center_lst[idx] = get_center(tmp_lst)
        iter_count += 1
        print('当前是第%d 次迭代' % iter_count)
    return label_lst
def write_data(label_lst):  # 将聚类结果写入文件
    with open(Label_Path, 'w', encoding='utf-8') as f:
        f.write('\n'.join(map(str, label_lst)))
def evaluate():  # 查看聚类结果与标准答案比较
```

```
    with open(Ground_Path, 'r', encoding='utf-8') as f0, open(Label_Path, 'r', encoding='utf-8') as f1:
        ground_lst, predict_lst = [], []
        cluster_lst = [[0 for _ in range(Num)] for _ in range(Num)]
        for line in f0:
            line = line.strip('\n')
            if line:
                ground_lst.append(int(line))
        for line in f1:
            line = line.strip('\n')
            if line:
                predict_lst.append(int(line))
        for ground, predict in zip(ground_lst, predict_lst):
            cluster_lst[ground][predict] += 1
        for idx, lst in enumerate(cluster_lst):
            print('聚类%d: %s' % (idx, '\t'.join(map(str, lst))))
        res = np.array(cluster_lst)  # debug 模式下以表格形式查看 res
        print(res)
def main():
    data_lst = read_data()
    center_lst = init_cluster_center(data_lst)
    label_lst = cluster(data_lst, center_lst)
    write_data(label_lst)
    # evaluate()
if __name__ == '__main__':
    main()
```

## 三、实验结果

1、仿真数据实验结果：（可以列出每个聚类中包含的样本，也可以画图显示不同聚类）

聚类 1 中心点：**(1.25, 1.125)**；包含样本 **$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$**

聚类 2 中心点：**(7.818181818181818, 7.363636363636363)**；包含样本 **$x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}$**

2、MNIST 数据集实验结果：

**每个聚类中包含不同类别样本数量统计表-随机初始化中心点**

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 14 | 583 | 3 | 2 | 0 | 24 | 303 | 28 | 2 |
| 1 | 2 | 3 | 0 | 1105 | 1 | 11 | 0 | 0 | 3 | 1 |
| 2 | 13 | 27 | 6 | 117 | 20 | 722 | 29 | 22 | 29 | 9 |
| 3 | 5 | 172 | 6 | 72 | 28 | 25 | 6 | 62 | 608 | 6 |
| 4 | 0 | 1 | 1 | 29 | 421 | 3 | 19 | 8 | 0 | 477 |
| 5 | 1 | 155 | 8 | 111 | 40 | 1 | 15 | 211 | 313 | 86 |
| 6 | 0 | 6 | 11 | 71 | 14 | 15 | 742 | 112 | 7 | 1 |
| 7 | 685 | 6 | 1 | 86 | 157 | 10 | 2 | 0 | 1 | 110 |
| 8 | 1 | 574 | 4 | 117 | 28 | 9 | 19 | 28 | 161 | 38 |
| 9 | 36 | 12 | 5 | 34 | 406 | 3 | 1 | 0 | 13 | 504 |

**每个聚类中包含不同类别样本数量统计表-选取批次距离尽可能远的 K 个点**

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 2 | 452 | 2 | 423 | 28 | 24 | 2 | 2 | 23 |
| 1 | 1100 | 10 | 0 | 0 | 0 | 0 | 3 | 8 | 2 | 3 |
| 2 | 118 | 721 | 1 | 30 | 17 | 34 | 38 | 9 | 5 | 21 |
| 3 | 73 | 32 | 1 | 5 | 18 | 12 | 625 | 37 | 8 | 179 |
| 4 | 30 | 2 | 0 | 359 | 1 | 23 | 0 | 248 | 294 | 2 |
| 5 | 148 | 5 | 5 | 29 | 36 | 26 | 310 | 37 | 66 | 279 |
| 6 | 73 | 23 | 14 | 34 | 27 | 784 | 8 | 0 | 0 | 16 |
| 7 | 68 | 5 | 0 | 100 | 1 | 1 | 0 | 413 | 467 | 3 |
| 8 | 113 | 13 | 4 | 24 | 4 | 14 | 182 | 24 | 35 | 566 |
| 9 | 22 | 1 | 5 | 270 | 0 | 1 | 13 | 381 | 310 | 11 |