Kmeans 实验报告

191250111 裴为东

算法原理：

1. 首先在样本中随机选取 k 个点作为簇的中心点
2. 然后计算每个样本点与各个簇中心点的距离并将他们划入距离最近的中心点所在的簇
3. 在每个划分好的簇中重新计算中心点
4. 重复 2，3 直到中心点位置不在变化或者达到了设定的迭代次数

由此 Kmeans 算法完成了对无标签数据集的划分

核心代码：

1.生成随机数据集

```
#随机生成在区间[0.0，100.0]的100个点
def load_data():
    data = np.random.rand(100, 2)*100
    return data
```

2.计算两点的欧式距离

```
#计算两点之间的欧氏距离
def cal_distance(p1,p2):
    return np.sqrt(np.sum(np.square(p1-p2)))
```

3.第一次根据随机选取的中心点对数据集进行划分，划分的方法是计算某个点与每个中心点的距离，选择距离最小的中心点所在的类簇作为该点所在的类簇

```python
#划分类簇
def divide(data,centroids):
    cluster = dict()
    k = len(centroids)
    for p in data:
        cluster_index = -1
        min_dis = float("inf")
        for index in range(k):
            c = centroids[index]
            dis = cal_distance(p,c)
            if dis < min_dis:
                min_dis = dis
                cluster_index = index
        if cluster_index not in cluster.keys():
            cluster[cluster_index] = []
        cluster[cluster_index].append(p)
    return cluster
```

4.每次划分完成之后，再在每个划分好的簇内重新计算中心点

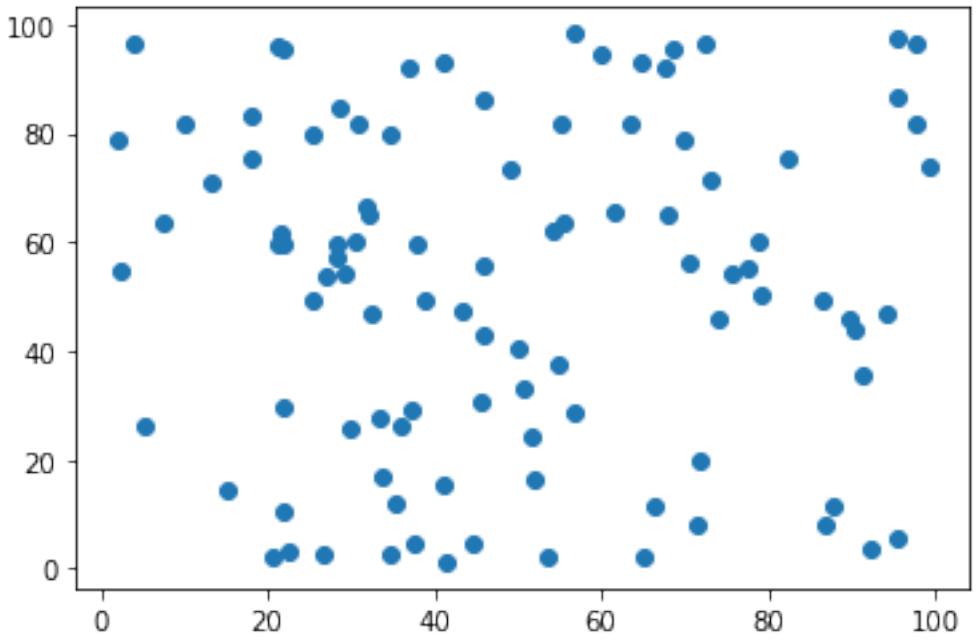
```python
#重新计算中心点
def update_centroids(cluster):
    new = []
    for i in cluster.keys():
        c = np.mean(cluster[i],axis=0)
        new.append(c)
    return new
```

5．主循环，结束条件为中心点变化误差小于 0.00001 或者迭代次数达到 10000 次，每迭代 10 次输出一次点的分布图和中心点坐标

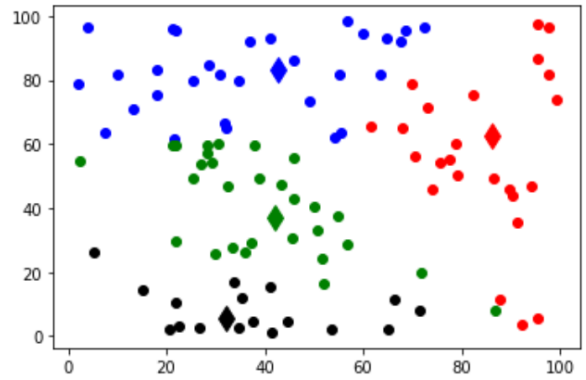
```
while (abs(new - old) >= 0.00001) and times < 10000:
    centroids = update_centroids(cluster)
    cluster = divide(data,centroids)
    old = new
    new = cal_variance(centroids,cluster)
    if times % 10 == 0:
        show_cluster(centroids,cluster)
        print(centroids)
        print("\n")
    times += 1
print("final result:\n")
show_cluster(centroids,cluster)
print(centroids)
```

实验结果:

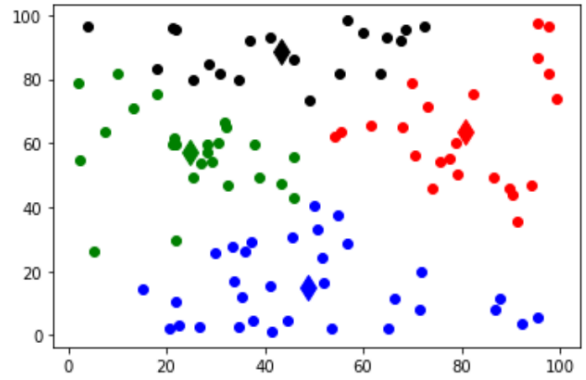
初始点分布图



迭代过程中的分布图



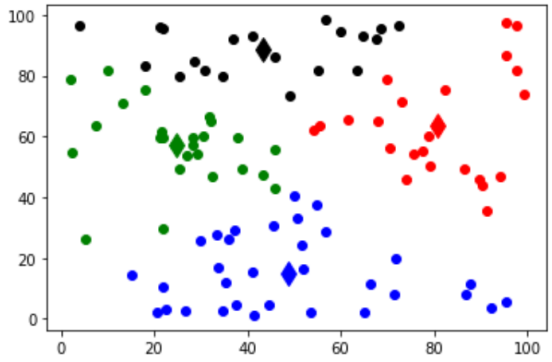
[array([86.09280262, 62.5469437]), array([42.67901773, 83.16900185]), array([42.189604
31, 36.88031593]), array([32.16958138, 5.44303946])]



[array([80.82157501, 63.5134281]), array([48.90189754, 15.07340046]), array([24.917959
43, 57.14470996]), array([43.32424025, 88.74086668])]

最后结果分布图

final result:



[array([80.82157501, 63.5134281]), array([48.90189754, 15.07340046]), array([24.917959
43, 57.14470996]), array([43.32424025, 88.74086668])]