谱聚类

【概述】

谱聚类是一种基于图论的聚类方法，它将样本看作无向图中的结点，根据样本之间的相似度构建边，然后根据图内点相似度将图分为多个子图，使子图内部的点相似度最高，子图之间点的相似度最低。

【算法原理】

谱聚类利用最小割原理将图分割成多个子图，使得不同子图之间的连接权值最小。谱聚类有两种不同的最小化准则：

$$\mathrm{RCut}(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K) = \sum_{k=1}^{K} \frac{\mathrm{cut}(\mathcal{A}_k, \bar{\mathcal{A}}_k)}{|\mathcal{A}_k|}$$

$$\mathrm{NCut}(\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K) = \sum_{k=1}^{K} \frac{\mathrm{cut}(\mathcal{A}_k, \bar{\mathcal{A}}_k)}{\mathrm{vol}(\mathcal{A}_k)}$$

其中，第一个准则为Rcut准则， 第二个准则为NCut准则。经过推导，两种准则的目标函数如下：

1. RCut

$$\min_{F \in R^{N \times K}} \mathrm{Tr}(F^T L F), \quad \text{s.t. } F^T F = I,$$

   其中L为非标准化拉普拉斯矩阵

2. NCut

$$\min_{F \in R^{N \times K}} \mathrm{Tr}(F^T L_{sym} F), \quad \text{s.t. } F^T F = I,$$

   L_{sym}称作为标准化拉普拉斯矩阵。

上述准则的最小化问题可以通过求解拉普拉斯矩阵L的前K个特征向量所构成的矩阵。最后采用k-means方法对该矩阵的行进行行聚类，每一行对应一个样本。这种通过解除F后再使用K-means进行聚类，相当于先做了一个特征提取。

【算法流程】

【代码实现】

1. 首先构建样本间的邻接矩阵

```
def get_dis_matrix(data):
    """
    获得邻接矩阵
    :param data: 样本集合
    :return: 邻接矩阵
    """
    nPoint = len(data)
    dis_matrix = np.zeros((nPoint, nPoint))
    for i in range(nPoint):
        for j in range(i + 1, nPoint):
            dis_matrix[i][j] = dis_matrix[j][i] = m.sqrt(np.power(data[i] - data[j],
2).sum())
    return dis_matrix
```

2. 通过邻接矩阵，利用KNN方法获得相似矩阵

```
def getW(data, k):
    """
    利用KNN获得相似矩阵
    :param data: 样本集合
    :param k: KNN参数
    :return:
    """
    dis_matrix = get_dis_matrix(data)
    W = np.zeros((len(data), len(data)))
    for idx, each in enumerate(dis_matrix):
        index_array = np.argsort(each)
        W[idx][index_array[1:k+1]] = 1
    tmp_W = np.transpose(W)
    W = (tmp_W+W)/2
    return W
```

3. 通过相似度矩阵获得度矩阵

```
def getD(W):
    """
    获得度矩阵
    :param W:  相似度矩阵
    :return:   度矩阵
    """
    D = np.diag(sum(W))
    return D
```

4. 通过度矩阵和相似度矩阵获得拉普拉斯矩阵

```
def getL(D, W):
    """
    获得拉普拉斯举着
    :param W: 相似度矩阵
    :param D: 度矩阵
    :return: 拉普拉斯矩阵
    """
    return D - W
```

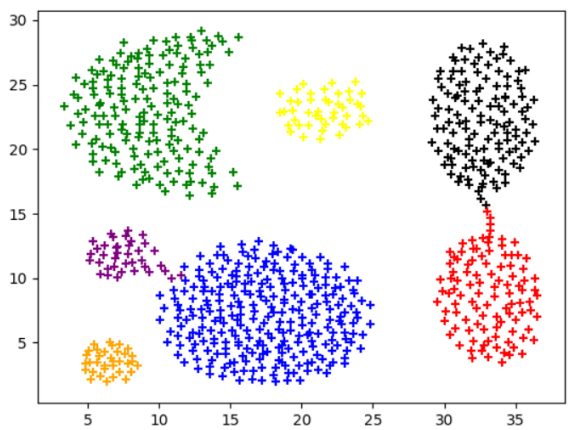
5. 求解拉普拉斯矩阵的特征向量

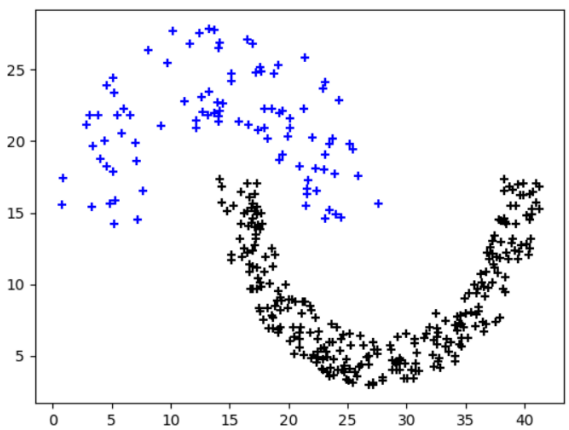
```
def getEigen(L):
    """
    从拉普拉斯矩阵获得特征矩阵
    :param L: 拉普拉斯矩阵
    :return:
    """
    eigval, eigvec = np.linalg.eig(L)
    ix = np.argsort(eigval)[0:cluster_num]
    return eigvec[:, ix]
```

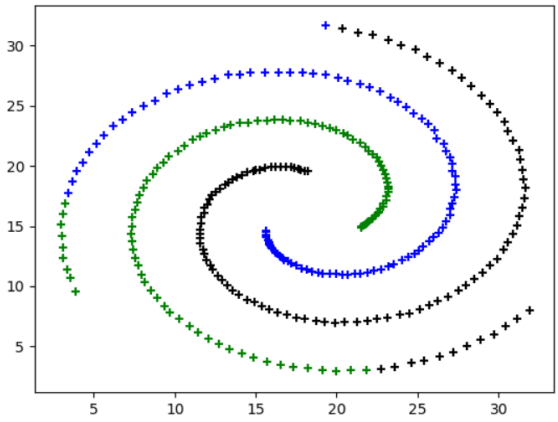
6. 根据特征向量对特征向量构成的矩阵进行行聚类，实现对数据集的k聚类。

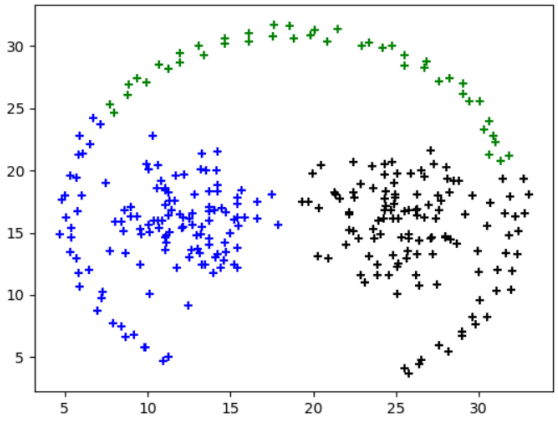
```
if __name__ == '__main__':
    cluster_num = 2
    KNN_k = 5
    data = load_data()
    data = np.asarray(data)
    W = getW(data, KNN_k)
    D = getD(W)
    L = getL(D, W)
    eigvec = getEigen(L)
    clf = KMeans(n_clusters=cluster_num)
```

【实验结果】






30
25
20
15
10
5
5
10
15
20
25
30


30
25
20
15
10
5
5
10
15
20
25
30