本科生实验报告

（2017-2018 学年秋季学期）

一、 实验题目

聚类

二、 算法解析

1. K-means

K-Means 算法是一种线性聚类算法，主要是以不断地取离聚类中心点最近均值为目标的算法。K-means 算法能够将数据集分成 K 个不相交的类，需要指定类的个数 K,初始化 K 个中心点。

算法流程如下：

（1）随机从对象集中选择 K 个对象作为初始聚类中心；

（2）对于每一个样本点，分别计算其到各个聚类中心的距离，将样本点划分到距离其最近的类。

（3）根据划分的类，对每个类重新计算聚类中心。

（4）重复步骤 2-3，直至聚类中心不再发生改变，最终得到 K 各类。

优缺点：

优点：

（1）算法易于理解且实现方法简单易行

缺点：

（1）只能用于线性聚类

（2）需要人工选择初始聚类数目 K，类的数目确定往往非常复杂和具有不确定性，不容易确定。

（3）聚类中心的初始化是随机的，结果不稳定，初始化方式对聚类结果影响大。

（4）容易受到噪声点的干扰

2. DBSCAN

DBSCAN 是一种基于密度的空间聚类算法。该算法将具有足够密度的区域划分为簇，并在具有噪声的空间数据库中发现任意形状的簇，它将簇定义为密度相连的点的最大集合。

（1）参数

DBSCAN 需要指定两个参数：

Eps: 半径，划定了样本点的邻域范围

Minpts: 密度阈值，核心点的密度下界

（2）分类

所有样本点可以分为三类：核心点、边界点、噪声点。

核心点：在 Eps 半径范围内，密度>=Minpts 的样本点

边界点：落在一个或者多个核心点的邻域范围内的点

噪声点：除了核心点和边界点的点

（3）主要算法流程

① 初始化所有样本点的状态为未访问状态，选择合适的两个参数；

② 对于每一个未访问的点，如果是核心点，则继续访问它的邻域内的全部样本点。对于邻域内的点，如果为核心点，则继续搜索其邻域，若为边界点则停止搜索。

③ 如此搜索得到的全部样本点属于一个类，继续执行上一步，直到没有未访问的点。

3. 谱聚类

谱聚类的主要思想是把所有的数据看做空间中的点，这些点之间可以用边连接起来。距离较远的两个点之间的边权重值较低，而距离较近的两个点之间的边权重值较高，通过对所有数据点组成的图进行切图，让切图后不同的子图间边权重和尽可能的低，而子图内的边权重和尽可能的高，从而达到聚类的目的。

谱聚类中权重的高低是通过距离的远近来近似邻接矩阵的近似矩阵 W 来表示的，构建邻接矩阵 W 的方法有三类。ϵ -邻近法，K 邻近法和全连接法。在实际的应用中，使用第三种全连接法来建立邻接矩阵是最普遍的，而在全连接法中使用高斯径向核 RBF 是最普遍。高斯核函数的公式如下：

$$W_{\mathrm{ij}} = \exp(-\frac{\| x_i - x_j \|_2^2}{2\sigma^2})$$

度数矩阵 D 为 nxn 的对角矩阵，其中：

$$\mathrm{d}_i = \sum_{j=1}^{n} w_{ij}$$

拉普拉斯矩阵：L=D-W

算法流程：

（1）计算相似度矩阵 W 和度数矩阵 D

（2）计算拉普拉斯矩阵

（3）构建标准化拉普拉斯矩阵 $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$

（4）标准化后的拉普拉斯矩阵求 k 个最小的特征值对应的特征向量，合并为特征矩阵；

（5）将特征矩阵按行进行标准化得到 NxK 维的矩阵，使用 K-means 算法进行聚类

三、 代码实现

1. K-means

（1）主要变量

```
[N,dim]=size(X);%总点数
c=zeros(N,k);%第i个样本是否属于第k类
u1=zeros(k,dim);%上一个的中心点
u2=zeros(k,dim);%新计算的中心点
```

（2）随机初始化中心点

```
%随机初始化中心点u
for i=1:dim %对于每一个维度
    t_h=max(X(:,i));%该维度范围上限
    t_l=min(X(:,i));%该维度范围下限
    u1(:,i)=(t_h-t_l)*rand(k,1);
end
```

（3）对每个点 X_{ik} 划分类，更新 c_{ik}

```
c=zeros(N,k);
t_d=zeros(k,1);
for i=1:N
    %计算样本点和每个聚类中心的距离
    for j=1:k;
        t_d(j)=norm(X(i,:)-u1(j,:),2).^2;
    end
    [~,t_k]=min(t_d);
    c(i,t_k)=1;%将样本点划分到最小距离的位置
end
```

（4）更新中心点 μ_k

```
for j=1:k
    u2(j,:)=zeros(1,dim);
    %求属于第K类的点的坐标和
    for i=1:N
        u2(j,:)=u2(j,:)+X(i,:).*c(i,j);
    end
    if sum(c(:,j))~=0
        u2(j,:)=u2(j,:)./sum(c(:,j));
        %除以点数
    end
end
```

（5）获得标签

```
for i=1:N
    label(i)=find(c(i,:));
end
```

2. DBSCAN

（1）主要变量的初始化

```
[N,~]=size(X);%总点数
Dis=zeros(N,N);%点与点之间的距离矩阵
for i=1:N
    for j=1:N
        Dis(i,j)=sqrt(sum((X(i,:)-X(j,:)).^2));%计算距离矩阵
    end
end
visited=false(N,1);%是否已经访问
noise=false(N,1);%是否为噪点
NEps=zeros(N,N);%每个点的eps半径内的点
CNum=0;%簇的个数
label=zeros(N,1);%每个簇的label
for i=1:N
    tem=find(Dis(i,:)<=Eps);
    NEps(i,1:length(tem))=tem;%寻找每个点EPS半径内的点的下标
end
```

（2）核心点的邻域拓展访问过程

```
while true%对可达区域内的每一个点
    t_vis=Neighbors(j);
    if ~visited(t_vis)
        visited(t_vis)=true;
        tem=NEps(t_vis,:);tem(tem==0)=[];
        if numel(tem)>=MinPts
            Neighbors=[Neighbors tem];%拓展邻域
        end
    end
    %为样本打标签
    if label(t_vis)==0
        label(t_vis)=CNum;
    end
    j = j + 1;
    if j > numel(Neighbors)
        break;
    end
end
```

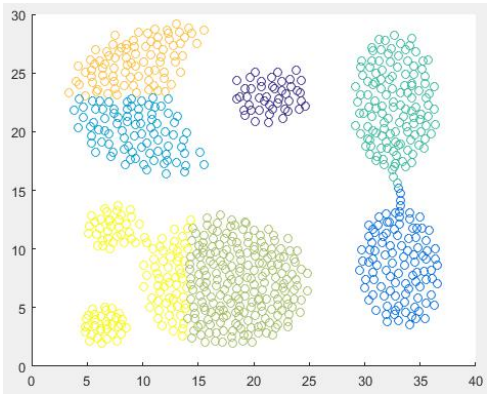
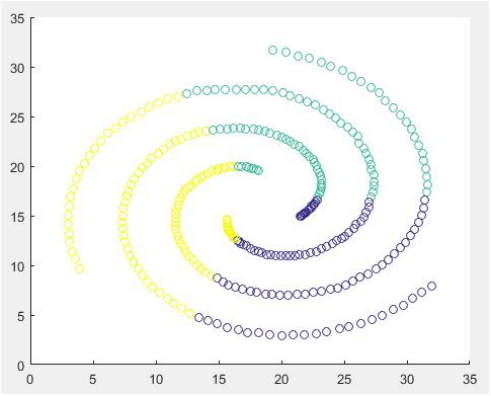
3. 谱聚类

```
%计算相似矩阵
W=zeros(N,N);
for i=1:N
    for j=1:N
        W(i,j)= exp(-norm(X(i,:)-X(j,:)))/(2*sigma^2);
    end
end
%计算对角矩阵
D=eye(N);
for i=1:N
    D(i,i) = sum(W(i,:));
end
%计算拉普拉斯矩阵
L=D-W;
L = D^(-.5)*L*D^(-.5);
% %计算特征值特征向量
[eigVectors,~] = eigs(L,k1,'SA');
% 构造归一化矩阵U从获得的特征向量
for i=1:N
    n = sqrt(sum(eigVectors(i,:).^2));
    U(i,:) = eigVectors(i,:) ./ n;
end
label=k_means(U,k2);
```

四、 效果分析

- **小数据集**

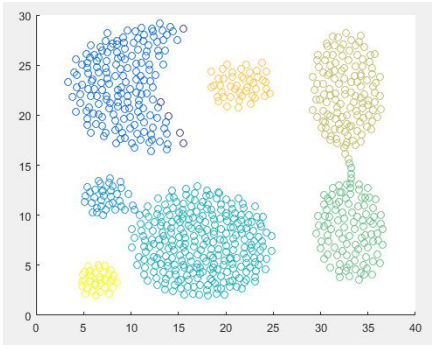
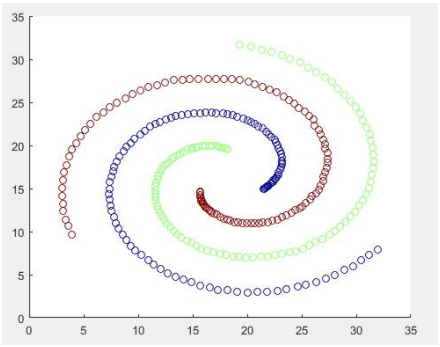
1. K-means

K-means 算法由于只能进行线性聚类，所以对于 Aggregation_cluster 数据集中通过线性划分能够划分出部分类别。但是对于 Spiral_cluster 数据集，可以很明显看到其线性划分的特性并不能够满足需求。

2. DBSCAN

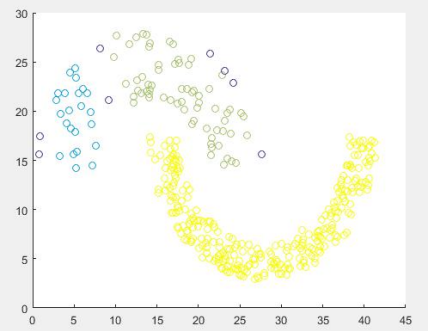
和 K-means 相比，DBSCAN 作为非线性聚类算法，通过调节参数，能够使得给定的一些小数据集实现比较完美的划分效果，如下两个数据集均能够得到比较理想的结果：

下表为对不同数据集调到效果比较好的参数：

DBSCAN					
指标	Aggregation_cluster	flame_cluster	Jain_cluster	Pathbased_cluster	Spiral_cluster
Eps	1.6	1	2.5	1.8	2
Minpts	10	6	7	4	4

不过，参数的调节是比较困难的，因为有两个参数需要调节，这位调节参数带来了一些困难，比如 *Jain_cluster* 样例调参所得最优结果如下：



可以看到，黄色部分密度比较高，能够很好的聚合，但是另一部分由于比较稀疏，被拆分成了两个类，很难调节参数得到不错的结果。

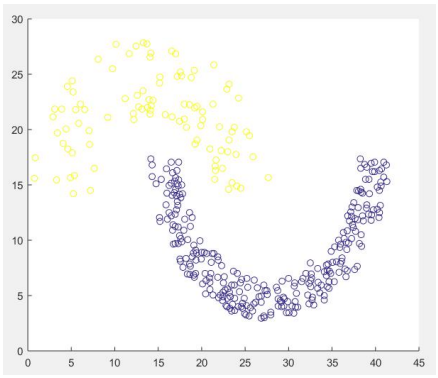
在网上博客上看到一种 k-距离选择参数的方法。k-距离是点 p(i)到所有点（除了 p(i)点）之间距离第 k 近的距离。根据得到的所有点的 k-距离集合 E，对集合 E 进行升序排序后得到 k-距离集合 E'，需要拟合一条排序后的 E'集合中 k-距离的变化曲线图，然后绘出曲线，通过观察，将急剧发生变化的位置所对应的 k-距离的值，确定为半径 Eps 的值。其中 K 为 Minpts. 实验过程中画出了 4-距离曲线，但是发现在急剧变化的位置的 Eps 半径处，效果并不好。所以在调参的过程中仍然采用了多次尝试调整的方式。

调整两个参数时，基本如下：

总的来说，虽然能够通过调参获得比较好的结果，但是 DBCSAN 在实际应用中选择合适的参数仍然是非常困难的。

3. 谱聚类

谱聚类在进行特征提取之后也进行了 K-means，但是实验效果明显好于 K-means.对于每一个数据集都能够调参获得比较好的聚类效果。并且和 DBSCAN 相比，调节的参数只有在高斯相似度计算时的一个 σ，调参更加容易。下图为在 DBSCAN 没能获得良好效果的 *Jain_cluster*，显然，它对于聚类的分隔非常好。



（$k=2,\sigma=0.9$）

- **Mfeat**

在 Mfeat 手写数据集 0-9 上，直接采用了 data 进行聚类。

对比以上三种聚类算法，在 Mfeat 数据集上进行运行所得结果如下表所示：

	ACC	NMI	PUR	参数
K-means	0.52	0.55	0.693	k=10
DBSCAN	0.49	0.5182	0.741	MinPts=10;Eps=600;
谱聚类	0.6275	0.6719	0.8205	k1=k2=10;sigma=300;

实验结果从 K-means,DBSCAN,谱聚类逐个变好，和在小数据集上的实验结果非常相似，谱聚类在参数调节方面更容易，在实验效果上达到三种算法当中最好的效果。