误差平方和函数集群

夏厚 PB18051031

一、 实验目的

1、就不同的初始划分，观察集群结果对初始划分的敏感性；
2、比较各群样本都很密集并且彼此明显分开的情况与各群样本数目相差很大时，误差平方和集群效果；
3、了解误差平方和准则函数的优势与缺陷；
4、使对算法进行适当改进使其可以对样本数目相差很大的情况进行集群；

二、 实验原理

1、聚类分析

分类：用已知类别的样本训练集来设计分类器（监督学习）
聚类（集群）：用事先不知样本的类别，利用样本的先验知识构造分类器（无监督学习）

聚类过程的基本步骤：
①特征选择，尽可能多地包含任务关心的信息；
②近邻测度，定量测定两特征如何“相似”或“不相似”；
③聚类准则，以蕴涵在数据集中类的类型为基础；
④聚类算法，按近邻测度和聚类准则揭示数据集的聚类结构；
⑤结果验证，常用逼近检验验证聚类结果的正确性；
⑥结果判定，由专家用其他方法判定结果的正确性。

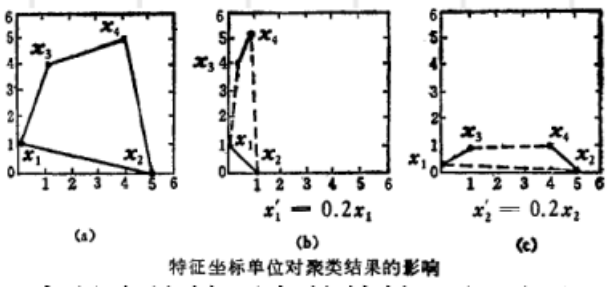
2、相似性准则（相似性度量）

用于描述各模式之间的特征的相似程度

距离测度：欧氏距离相似性度量

$$D_e(x,y)=\| x-y \|=\sqrt{\sum_{i=1}^{d}| x_i-y_i |^2}$$

模式特征坐标单位的选取会强烈影响聚类结果；
欧式距离具有旋转不变的特性，但对于一般的线性变换不是不变的，需要进行标准化；
标准化方法又有总和标准化、标准差标准化、极差标准化等；
距离测度还包括马氏距离、明氏距离等，本实验使用欧氏距离，故不再赘述；
相似性准则除距离测度外，还有相似测度、匹配测度等。



特征坐标单位对聚类结果的影响

样本的相似性度量是聚类分析的基础，针对具体问题，选择适当的相似性度量是保证聚类质量的重要问题。但有相似性度量还不够，必须要有适当的聚类准则函数。

3、聚类准则函数

一般有误差平方和准则函数、加权平均平方距离和准则函数、类间距离和准则函数散射矩阵。本实验使用误差平方和函数集群。

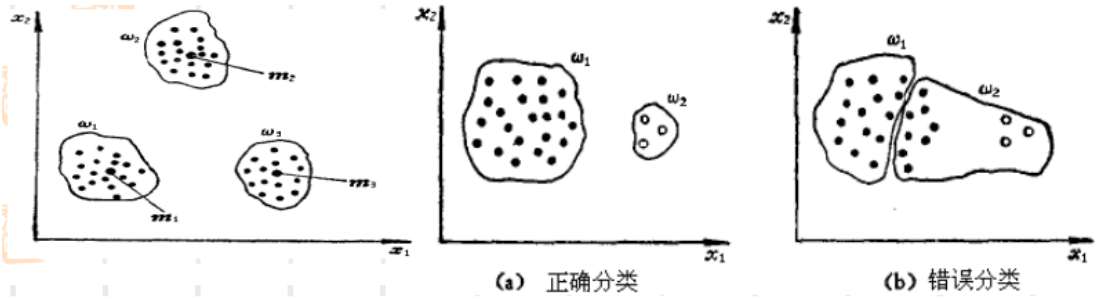
误差平方和准则是最常用的聚类准则函数:

$$J_c = \sum_{j=1}^{c}\sum_{k=1}^{n_j} \| x_k - m_j \|^2$$

式中m_j为类型w_j中的样本均值：$m_j=\frac{1}{n_j}\sum_{j=1}^{n_j} x_j$，j=1，2，……，c;

m_j是 c 个集合的中心，可以用来代表 c 个类型;

误差平方和准则函数适用于各类样本比较密集且样本数目悬殊不大的样本分布。



4、C-均值聚类算法

聚类准则函数是误差平方和准则

1）C-均值算法（一）

①、给出 n 个混合样本，令 I=1，表示迭代运算次数，选取 c 个初始聚合中心Z_j(1)，j=1，2，……，c;

②、计算每个样本与聚合中心的距离 D(x_k, Z_j(I))，k=1，2，……，n;

若 D(x_k, Z_j(I))=min{ D(x_k, Z_j(I))，k=1,2,……,n }，则$x_k \in w_j$。

③、计算 c 个新的集合中心：Z_j(I+1)=$\frac{1}{n_j}\sum_{k=1}^{n_j} x_k^{(j)}$，j=1，2，……，c。

④、判断：若Z_j(I+1)≠Z_j(I)，j=1，2，……，c，则 I=I+1，返回②，否则算法结束。

算法特点:

①、每次迭代中都要考查每个样本的分类是否正确，若不正确，就要调整，在全部样本调整完之后，再修改聚合中心，进入下一次迭代。如果在某一个迭代运算中，所有的样都被正确分类，则样本不会调整，聚合中心也不会有变化，也就是收敛了。

②、c 个初始聚合中心的选择对聚类结果有较大影响。

2）C-均值算法（二）

①、给定 n 个混合样本，令 I=1（迭代次数），选取 c 个初始中心Z_j(1)，j=1，2，……，c

②、计算每个样本与每个聚合中心的距离 D(x_k, Z_j(I))，k=1，2，……，n；j=1，2，……，c。

③、令 I=I+1=2，计算新的聚合中心。$Z_j(2)=\frac{1}{n_j}\sum_{k=1}^{n_j}x_k^{(j)}$，j=1，2，……，c；计算误差平方和$J_c$值：$J_c(2)=\sum_{j=1}^{c}\sum_{k=1}^{n_j}\|x_k^{(j)}-Z_j(2)\|^2$。

④、对每个聚合的每个样本，计算：

$$\rho_{ii}=\frac{n_i}{n_i-1}\|x_k^{(i)}-Z_i(I)\|^2,\quad i=1,2,...c$$

表示J_c减少的部分

$$\rho_{ij}=\frac{n_j}{n_j+1}\|x_k^{(i)}-Z_j(I)\|^2,\quad j=1,2,...c,\quad j\neq i$$

表示J_c增加的部分

令：$\rho_{il}=min\{\ \rho_{ij}\ \}$，若$\rho_{il}<\rho_{ii}$,则把样本$x_k^{(i)}$移到聚合中心$w_l$中，并修改聚合中心和$J_c$值。

$$Z_i(I+1)=Z_i(I)+\frac{1}{n_i-1}[Z_i(I)-x_k^{(i)}]$$

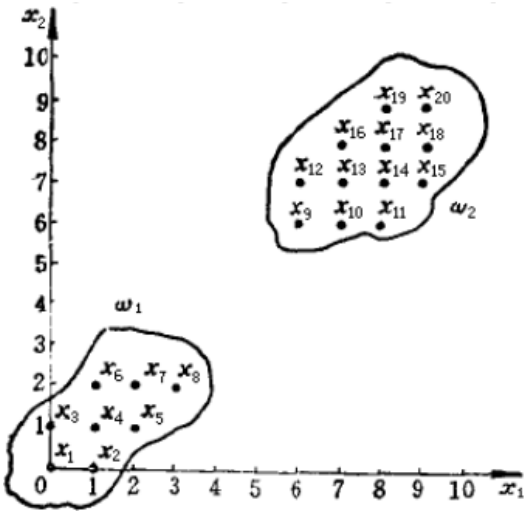
$$Z_l(I+1)=Z_j(I)-\frac{1}{n_l+1}[Z_l(I)-x_k^{(i)}]$$

$$J_c(I+1)=J_c(I)-(\rho_{ii}-\rho_{il})$$

⑤、判断：若J_c(I+1)< J_c(I)，则 I=I+1,返回④。否则，算法结束。

三、实验内容

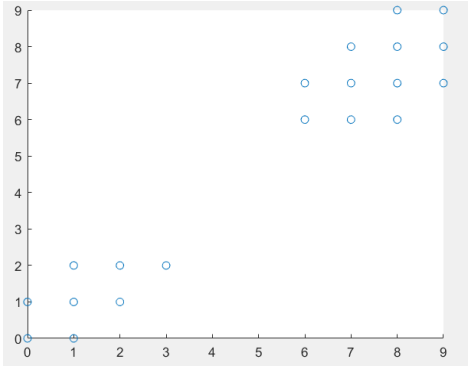
本实验使用 C-均值算法（二）实现代码
首先对如下样本进行聚类：



运行程序结果：

```
初始聚合中心=(7,8)
初始聚合中心=(9,7)
W1样本个数：15,类聚合中心=(3.93333,4)
W2样本个数：5,类聚合中心=(8.6,7.4)
误差平方和为：295.333

73.9192
37.7083
37.7083
最终聚类结果：
W1样本个数：8,类聚合中心=(1.25,1.125)
W2样本个数：12,类聚合中心=(7.66667,7.33333)
误差平方和为：37.7083

--------------------------------
```

选择两个初始聚合中心（7，8），（9，7）；
第一次聚合划分之后，聚合中心为（3.93，4），（8.6，7.4），误差平方和为 295.33；
经过三次迭代，误差平方和稳定在 37.7，成功完成聚类，聚类结果：

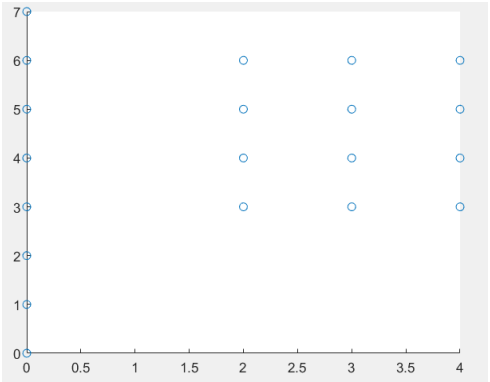


左下角 8 个点聚为一类，右上角 12 个点聚为一类。
改变初始聚合中心，只是迭代次数发生了变化，聚类结果没有改变：

```
初始聚合中心=(0,0)
初始聚合中心=(9,8)
W1样本个数：8,类聚合中心=(1.25,1.125)
W2样本个数：12,类聚合中心=(7.66667,7.33333)
误差平方和为：37.7083

37.7083
最终聚类结果：
W1样本个数：8,类聚合中心=(1.25,1.125)
W2样本个数：12,类聚合中心=(7.66667,7.33333)
误差平方和为：37.7083
```

```
初始聚合中心=(0,1)
初始聚合中心=(1,1)
W1样本个数：2,类聚合中心=(0,0.5)
W2样本个数：18,类聚合中心=(5.66667,5.33333)
误差平方和为：320.5

37.7083
37.7083
最终聚类结果：
W1样本个数：8,类聚合中心=(1.25,1.125)
W2样本个数：12,类聚合中心=(7.66667,7.33333)
误差平方和为：37.7083
```

改变样本分布，使其中一类样本分布不密集：



```
初始聚合中心=(0,2)
初始聚合中心=(0,0)
W1样本个数：19,类聚合中心=(1.89474,4.31579)
W2样本个数：1,类聚合中心=(0,0)
误差平方和为：91.8947

91.8947
最终聚类结果：
W1样本个数：19,类聚合中心=(1.89474,4.31579)
W2样本个数：1,类聚合中心=(0,0)
误差平方和为：91.8947
```

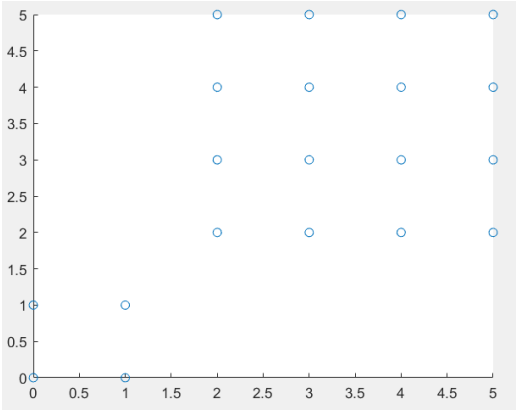
```
初始聚合中心=(0,0)
初始聚合中心=(0,7)
W1样本个数：7,类聚合中心=(1.28571,2.14286)
W2样本个数：13,类聚合中心=(2.07692,5.15385)
误差平方和为：68.9011

67.25
67.25
最终聚类结果：
W1样本个数：8,类聚合中心=(1.125,2.375)
W2样本个数：12,类聚合中心=(2.25,5.25)
误差平方和为：67.25
```

```
初始聚合中心=(4,4)
初始聚合中心=(3,4)
W1样本个数：4,类聚合中心=(4,4.5)
W2样本个数：16,类聚合中心=(1.25,4)
误差平方和为：88

72.3333
66.6
65.2323
65.0769
63.2857
62
62
最终聚类结果：
W1样本个数：15,类聚合中心=(2.4,4.8)
W2样本个数：5,类聚合中心=(0,2)
误差平方和为：62
```

```
初始聚合中心=(0,4)
初始聚合中心=(4,5)
W1样本个数：10,类聚合中心=(0.4,3.5)
W2样本个数：10,类聚合中心=(3.2,4.7)
误差平方和为：66.6

65.2323
65.2323
最终聚类结果：
W1样本个数：9,类聚合中心=(0.222222,3.44444)
W2样本个数：11,类聚合中心=(3.09091,4.63636)
误差平方和为：65.2323
```

当样本集中各类样本不是比较密集时，从程序结果可以看到初始划分对结果影响很大，这说明算法对初始划分敏感性较大。不同的初始划分得到的误差平方和极小值也有所不同，这说明，C-均值算法找到的基于误差平方和最小化的聚类结果是局部最优。由于误差平方和准则适用于各类样本比较密集的样本分布，此处 C-均值算法给出了错误的分类。

再改变样本分布使样本数目悬殊较大：




```
W2样本个数：8,类聚合中心=(1.25,2)
误差平方和为：52.5

42.9048
40.8
40.8
最终聚类结果：
W1样本个数：15,类聚合中心=(3.6,3.6)
W2样本个数：5,类聚合中心=(0.8,0.8)
误差平方和为：40.8
```

```
初始聚合中心=(4,4)
初始聚合中心=(4,5)
W1样本个数：16,类聚合中心=(2.75,2.375)
W2样本个数：4,类聚合中心=(3.5,5)
误差平方和为：75.75

62.3736
53
46.9167
43.4725
43.4725
最终聚类结果：
W1样本个数：7,类聚合中心=(1.28571,1.28571)
W2样本个数：13,类聚合中心=(3.76923,3.76923)
误差平方和为：43.4725
```

```
初始聚合中心=(0,0)
初始聚合中心=(0,1)
W1样本个数：2,类聚合中心=(0.5,0)
W2样本个数：18,类聚合中心=(3.16667,3.22222)
误差平方和为：68.1111

42
40.8
40.8
最终聚类结果：
W1样本个数：5,类聚合中心=(0.8,0.8)
W2样本个数：15,类聚合中心=(3.6,3.6)
误差平方和为：40.8
```

```
初始聚合中心=(0,0)
初始聚合中心=(5,2)
W1样本个数：5,类聚合中心=(0.8,0.8)
W2样本个数：15,类聚合中心=(3.6,3.6)
误差平方和为：40.8

40.8
最终聚类结果：
W1样本个数：5,类聚合中心=(0.8,0.8)
W2样本个数：15,类聚合中心=(3.6,3.6)
误差平方和为：40.8
```

当样本数目相差很大时，从实验结果可以看到，C-均值算法对初始划分有一定敏感性，不同的初始划分可能会造成不同的聚类。并且由于误差平方和准则的原因，算法对样本数目相差较大的情况进行了错误的聚类。

由此我们可以总结误差平方和函数集群的优势与缺陷，误差平方和函数集群，计算简单比较直观，对各类样本比较密集且样本数目悬殊不大的样本分布可以进行很好的划分。但当样本分布不够密集，比如成线性时，或者样本数目悬殊比较大时，采用误差平方和最小准则会给出错误的聚类。

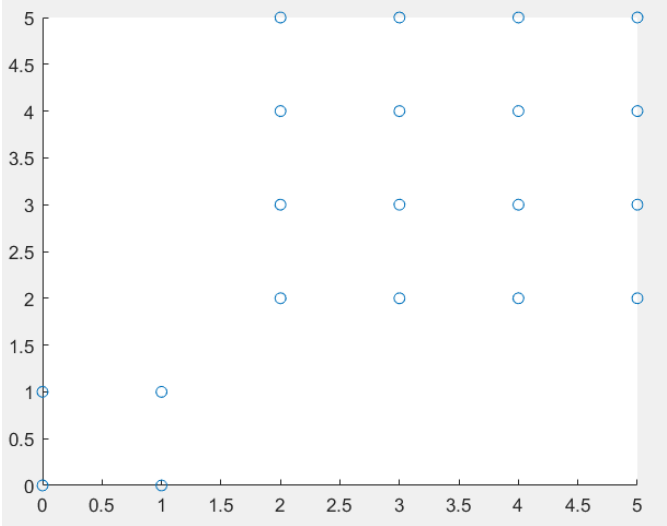
因为算法难以对样本数目相差很大情况进行正确集群的原因主要是误差平方和函数集群的特性决定的，所以要改进算法使对样本数目相差很大的情况进行集群，一方面可以改用其他函数集群，如类间距离和准则、散度矩阵等；另一方面，可以在程序中指出较小类别的样本数目；并把按照误差平方和最小集群得到的较小数目的样本，对每个样本计算：假设该样本移到到另一类中，计算新的误差平方和，在所有这些误差平方和中找到最小的那个对应的样本点，划分到另一类别中，直到该类样本数目等于程序给定的样本数目为止。

改进部分代码：

```
1.  //算法改进部分
2.      double tt[21]={0};
3.      if(W1.w>W2.w){
4.          for(int i=1;i<=20;i++){
5.              if(point[i].w==2){//假设类别 2 中的每个样本移动到类别 1 中，计算误差平
    方和
6.                  point[i].w=1;
7.                  W1.w++;W2.w--;
8.                  center2(W1,W2,point);
9.                  tt[i]=fesos(W1,W2,point);
10.                 point[i].w=2;
11.                 W2.w++;W1.w--;
12.             }
13.         }
14.     }
15.     int i=0,j=0;
16.     while(1){
17.         if(tt[i]!=0){//找到误差平方和最小，所对应的样本点
18.             tt[0]=tt[i];
19.             j=i;
20.             break;
```

```
        }
        i++;
    }
    for(int i=1;i<=20;i++){//将该样本点移动到类别 1
        if(tt[i]!=0&&tt[i]<tt[0]){
            tt[0]=tt[i];
            j=i;
        }
    }
    cout<<j<<endl;
    point[j].w=1;
    W1.w++;W2.w--;
    center2(W1,W2,point);//计算新的聚类中心和误差平方和
    esos=fesos(W1,W2,point);
```

对于前面错误聚类的样本数目相差较大的样本进行了正确聚类



改进之前的聚类结果:

```
初始聚合中心=(4,4)
初始聚合中心=(2,4)
W1样本个数：12,类聚合中心=(4,3.5)
W2样本个数：8,类聚合中心=(1.25,2)
误差平方和为：52.5

42.9048
40.8
40.8
最终聚类结果：
W1样本个数：15,类聚合中心=(3.6,3.6)
W2样本个数：5,类聚合中心=(0.8,0.8)
误差平方和为：40.8
第一类：
(0,0)
(0,1)
(1,0)
(1,1)
(2,2)
```

改进之后的聚类结果:

```
初始聚合中心=(4,4)
初始聚合中心=(2,4)
W1样本个数：12,类聚合中心=(4,3.5)
W2样本个数：8,类聚合中心=(1.25,2)
误差平方和为：52.5

42.9048
40.8
40.8
5
最终聚类结果：
W1样本个数：16,类聚合中心=(3.5,3.5)
W2样本个数：4,类聚合中心=(0.5,0.5)
误差平方和为：42
第一类：
(0,0)
(0,1)
(1,0)
(1,1)
```

改进之后对样本数目相差较大的样本，也进行了正确的聚类。

实验总结：

通过本次实验，对聚类方法有了深一步的了解，尤其是误差平方和准则和 C-均值聚类算法。虽然只是使用 c 语言实现了对一个比较简单的二类问题进行划分，但多类问题可以建立在二类问题的基础上进行分析，对准则函数以及算法的优缺点都有了一个从实践角度的认识。

样本聚类是一个很灵活的过程，不仅体现在采用什么相似性度量、什么聚类准则函数、什么聚类算法上面；还体现在若何对样本特征进行选择与归一化处理，包括量纲、角度等等。所以实际分类要选择合适的方法，或者几种方法同时分类，择优选取分类结果，也可以动态的调整。