信息处理技术 作业6

From: 梁鑫宇 3160104494

题目：聚合聚类

1. 题目分析：

完成由底向上的非传统聚合聚类，允许不完全嵌套

2. 思路分析

聚类依据动态维护的相似度矩阵，使用嵌套列表用物理上的一维实现逻辑上的二维。

为减少不必要的时间空间浪费，嵌套列表时采用变长方式，保证两个类的相似度仅被存储一次。

相当于在一个行向量中存储了一系列不等长的列向量，列向量中的分量记录了此类与此前各类之间的相似度

- 示意图如下

\	C1	C2	C3	C4
C1		?	?	?
C2			?	?
C3				?
C4				

源代码

Matrix.py

```
class matrix:
    m = []  # proximity matrix
    PointsDic = {}  # cluster result
    def __init__(self,points):
        self.centerpoints = points
        for i in range(0,len(points)):
            self.PointsDic[i] = points[i]
            self.m.append([])   # append empty list to m thus initiate an empty proximity matrix
        for i in range(0,len(points)):
            for j in range(0,i):
                self.m[i].append(abs(points[i]-points[j]))

    def nextcluster(self):
        min = float('Inf')
        minx = -1
        miny = -1
        # find the min in the proximity matrix and note these two clusters
```

```
        for i in range(0,len(self.m)):
            for j in range(0,len(self.m[i])):
                if self.m[i][j]<min:
                    min = self.m[i][j]
                    minx = j
                    miny = i
        # update the cluster result
        self.PointsDic[minx] = (self.PointsDic[minx], self.PointsDic[miny])
        # update the centerpoints
        self.centerpoints[minx] = (self.centerpoints[minx]+self.centerpoints[miny])/2
        # update the proximity matrix
        del self.m[miny]
        del self.centerpoints[miny]
        for i in range(minx+1,len(self.m)):
            self.m[i][minx] = abs(self.centerpoints[minx]-self.centerpoints[i])
        for i in range(miny,len(self.m)):
            del self.m[i][miny]
            self.PointsDic[i] = self.PointsDic[i+1]
```

Main.py

```
import Matrix

str = input("Please input points, split by blank\n")
str = str.strip()   # eat the white before or after the valid input
strpoints = str.split(" ")  # divide the input and store it in a string list
points = []
for point in strpoints:
    try:
        points.append(float(point)) # transfer the string point to float
    except:
        print("Input error")

iMatrix = Matrix.matrix(points)
# call the cluster function until there is only one cluster
while(len(iMatrix.m) > 1):
    iMatrix.nextcluster()
# show the cluster result
print(iMatrix.PointsDic[0])
```

测试样例