数据挖掘 homework2 实验报告

2013011326 周建宇 计 32

一、 实验环境

- 操作系统：Windows 10 64 位
- 内存：8G
- CPU 主频：2GHz
- 编程语言&平台：Matlab

二、 算法原理

《Clustering by fast search and find of density peaks》论文中的算法原理很清楚，也比较简单，具体过为对于一组输入点，我们首先计算出每个点的 density。Density 的计算论文中给出了两种过程。一种是离散形式，一种是连续形式。离散形式即算出与某一点的距离不大于给定阈值 dc 的所有点的数量。连续形式由下式给出：

$$P_i = \sum_{j=1}^{n} e^{-\frac{dist(i,j)^2}{dc^2}}$$

其中 Pi 代表第 i 个点的密度，dist(i.j)代表第 i 个点和第 j 个点的距离，dc 为我们所设定的距离阈值。

这样我们就能在 O（n^2）的时间内计算出每个点的密度。之后对于每个点，我们计算出其δ，δi 的含义为比第 i 个点的密度大的所有点中与第 i 个点距离最小的点的距离。

接下来我们要找到聚类中心，这个过程有人为因素，我们需要找到δ和 p 都比较

大的点作为聚类中心（cluster center）。能够很直观地感觉这样的聚类中心是合理的。因为一个聚类中心所形成的聚类其性质应该具有类内部的点之间的距离足够小，而与其他类的点间的距离足够大。寻找聚类中心的过程需要 O（n）的时间。

然后对于每个点，我们找到他应该属于哪个类。显然，一个点属于第 i 个聚类当且仅当比该点密度更高且与该点距离最小的点也属于第 i 个聚类。这样我们就大致给每个点分出了其应属于的类别。然而实际数据中其实有大量的噪声或随机点存在，因此我们需要识别出噪声点，噪声点不属于任何类别。论文中识别噪声的方法不是直接定义一个噪声函数，而是首先找出每个聚类的聚类边缘。一个聚类的聚类边缘实质是一个点集，其中的点具有一些特点：每个点都存在至少一个与其距离不大于 dc 且不属于该聚类的点。找到边缘集合后，我们找到集合中密度最大的点的密度作为 pb。如果一个点的密度小与其初始聚类的 pb，则该点被判定为噪声点。

基于以上几步，我们就能在 O（n^2）的时间内实现聚类，而且该算法的鲁棒性较强，很容易扩展到高维数据，只要给除了两个点的相似度或者距离定义即可完成聚类。

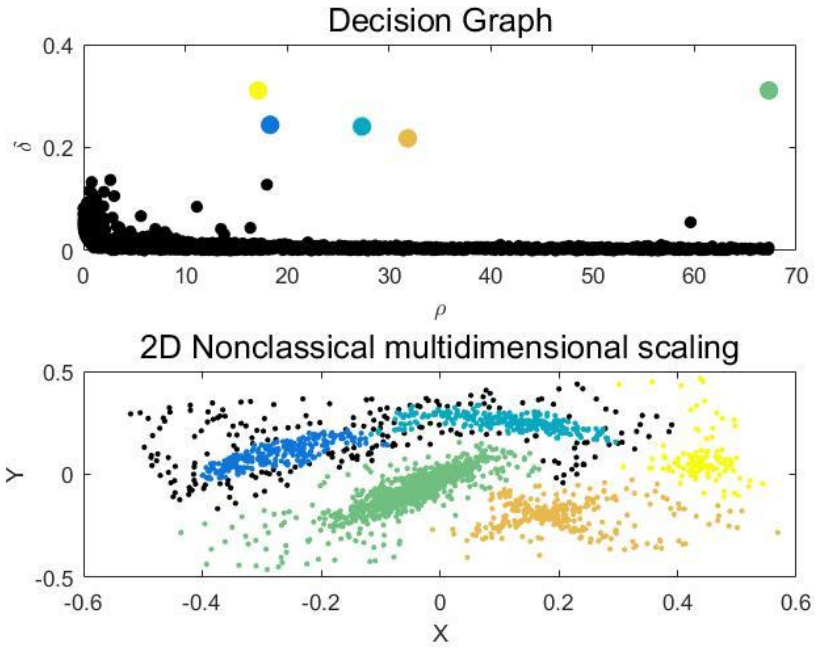
三、 实验结果与分析

我们利用 matlab 编程实现了 fast cluster 算法，并将其运用到 S1 数据集上进行评测。评测有两点需要注意：一是 dc 的选择，二是聚类中心的选择。dc 的选择既不能太小又不能过大，而且依赖于具体数据集的特征。在此我们首先将数据点距离排序，选出位于 percent%的距离值作为 dc。

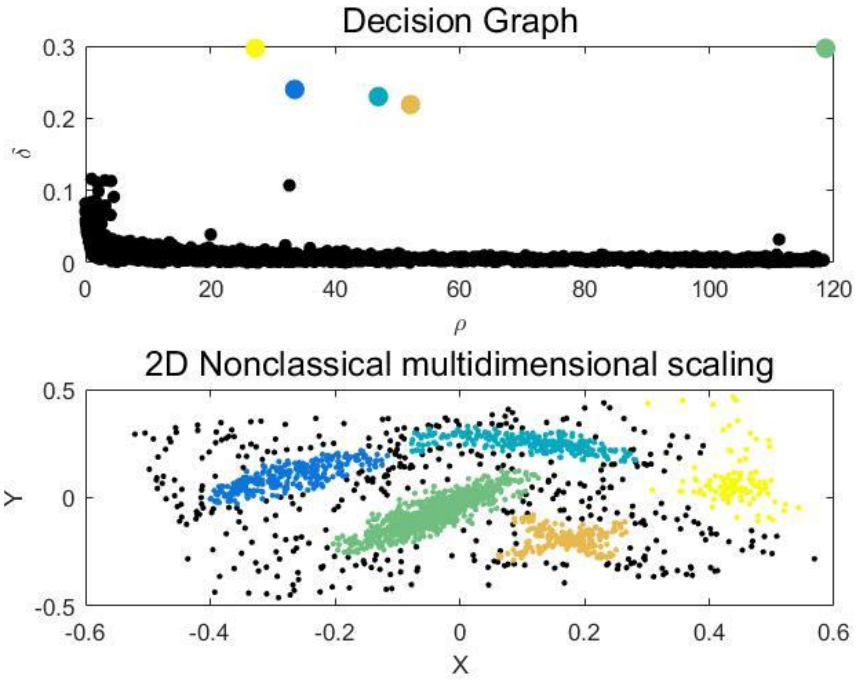
对于聚类中心的选择，论文中的方法是设置了 y=δ*p 的阈值。我们可以选出前 k 大的的 y 作为聚类中心点。

为了选出较适合的 percent 和 k 我们进行了几组数据的实验

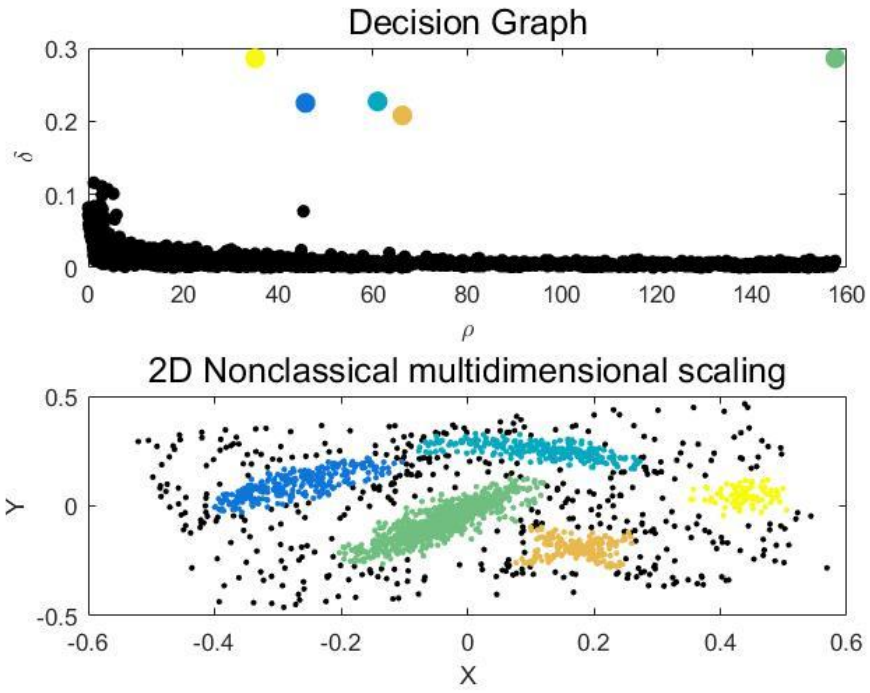
结果如下：



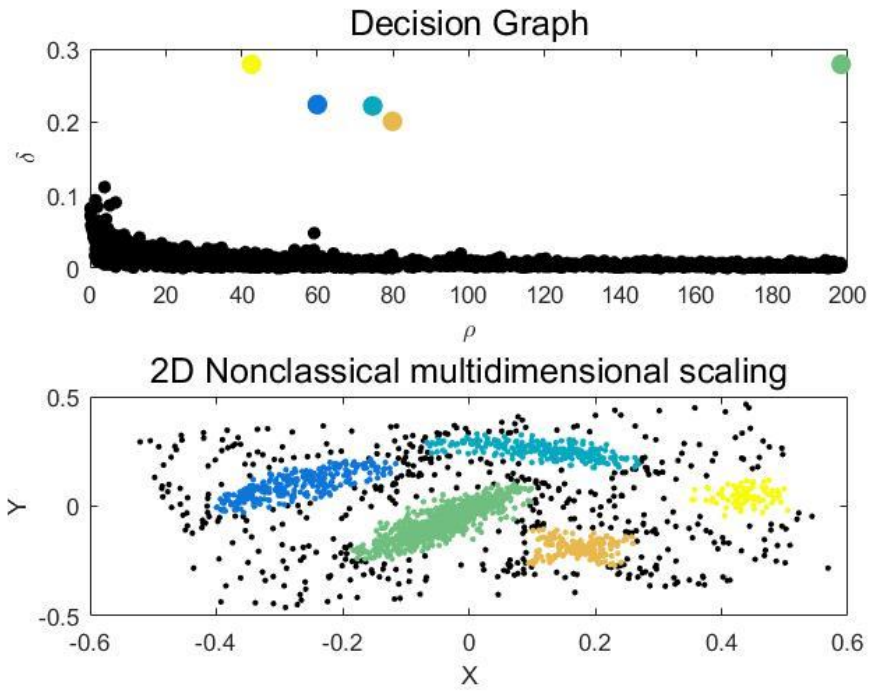
K=5，percent=0.01



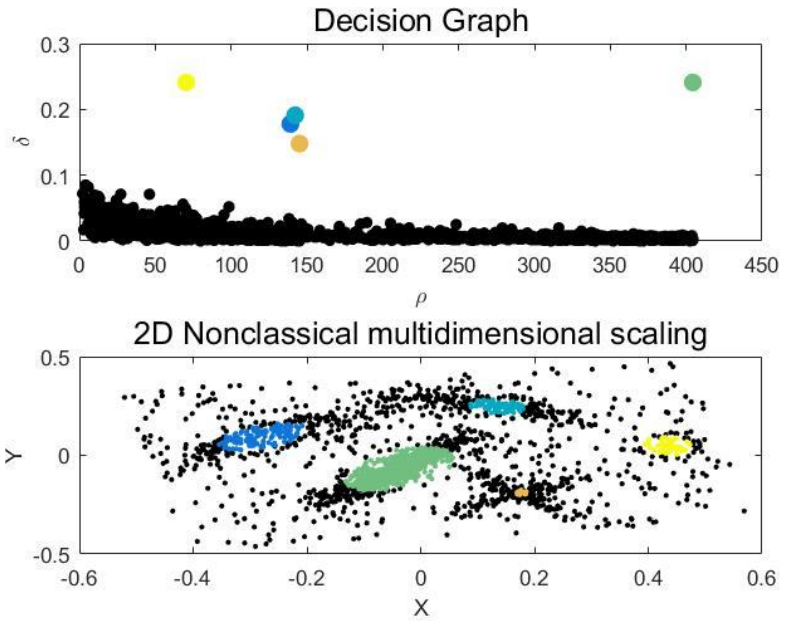
K=5,percent = 0.02



K=5, percent = 0.03



K=5,percent = 0.04



K=5, percent=0.1

我们能够看到，在 percent=0.01 时，聚类边界开始变得模糊，甚至在左下角区域区域有许多明显可以归为噪声点的点也被划归为聚类点。这一点可由高斯函数的表达式看出：$P_i = \sum_{j=1}^{n} e^{-\frac{dist(i,j)^2}{dc^2}}$，随着 percent 较小时，dc 也很小，而 dc 恰恰代表高斯分布的标准差，标准差很小则每个点的 P 也趋向于同一，自然也会有更多的噪声点被划归为聚类点。而随着 percent 的增大，分类能力会逐渐增强。但也并非越大越好，我们看到，percent 为 0.03 的聚类效果是很好的，但当到达 0.1 时，则对聚类的要求过分苛刻，大部分点都被列为了噪声。

而对于 k 的选择，我们看到其实δ对选择聚类中心的作用是更大的。这也是因为哪怕一个聚类只有很少的几个点，但只要它与外界足够远，则也可以被划归为聚类中心。

因此我们可以设定一个δ的阈值（可以为δ的 percent 分位数）。对于满足δi>=δ的值我们再进一步判断其 yi 是否满足要求。假设我们有 k1 个点的δi 满足要求，而其中 k2 个点的 pi 小于所有 p 的 k1/n 分位数，则一般来讲该点应该被剔除掉，则 k=k1-k2;

按照我们的方法设置δ和 p 的阈值同样的到 k=5.但不能保证其他数据集该参数仍有效。

因此更直观和有效的做法通常是将高维数据变换到二维平面，再由人工确定 k 值。

四、 总结

聚类方法多种多样，这篇文章中的聚类算法直观上很容易理解，合理性也有保证，效率也很不错，可扩展性也很强，唯一的难点在于 dc 的选取上，而选取 dc 的关键在于选取 percent，S1 数据中 percent 最佳值应该在 0.03 左右，对于不同的数据集，需要不断尝试 percent 取值已达到较好的分类效果。