**Clustering by Synchronization**

1. **论文概述**：

论文主要介绍了一种基于同步的聚类算法，同步是指相类似的物体的运动会趋于一致，而不同类别之间的物体运动有较大的差异，利用同步的特性，我们可以很容易的来解决聚类问题以及发现数据中的孤立值。

算法思想：将每一个对象看作一个相位振荡器，用一个坐标向量来表示（这里指描述花的四个指标），坐标向量在“空间”中的初始分布具有三类：聚类中心部位、聚类边缘部位以及孤立点。对于聚类边缘部分的点来说，其的邻居点有部分偏向中心而一部分没有，根据与邻居点的位置关系来确定聚类方向，改变坐标值，且越边缘，移动约远；对于聚类中心周围的点，其周围的邻居多且仅，所以各对象间的影响较小；独立的点由于没有在邻居节点检测范围内，所以保持基本不变。

算法步骤：

1. 获取并整理数据（从文件读取并去掉标签）
2. 首先，计算各对象到其余对象的距离，通过假设半径来确定各粒子对应的邻居，通过公式：



来计算对象新坐标，并通过公式：



计算cluster order参数来判断是否满足终止条件，当时可以退出

1. 若 与1差距较大，那么重复步骤（2）
2. **遇到问题：**
   1. 如何定义距离及如何计算邻居距离?

距离问题个人觉得是任何学习算法最基础也是最关键的问题，这里的数据都是数值且单位统一，大小相当，所以可以直接考虑欧式距离即可，如果数据项非数值或数据项间差距过大，还需要先进行类别型特征编码或特征归一化处理。

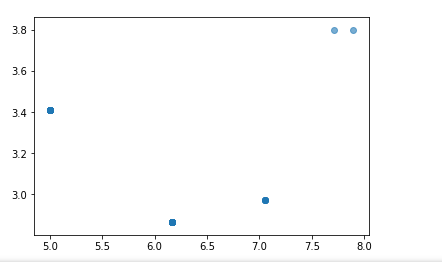
确定用欧式距离后，如何计算邻居距离，之前学习的时候知道用矩阵来处理的速度比直接用循环来处理要快得多，所以我一直思考和查找如何通过矩阵直接计算距离而不是使用循环，找到scipy库提供了用于求距离矩阵的工具，只不过得到的距离是以上三角排列展开的，还需要进行转换，转换后得到一个对角线对称的距离矩阵。

* 1. 邻居矩阵设计：

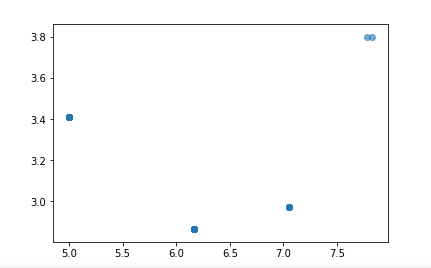
同样由于矩阵的方便，这里设计邻居矩阵为：NxNx4的矩阵，初始化为每个矩阵的邻居都是自己，然后在通过a）中得到的距离矩阵判断是否为邻居后得到的邻居的矩阵下标，更改邻居矩阵中的对应位置。这里将初始的矩阵邻居设置为自己，是因为后面在计算x(t+1)和cluster order参数时，都要用到邻居矩阵减去对应点的矩阵，如果初始化邻接矩阵为0那么这里无法统一处理，初始化为邻居矩阵则可以直接相减，这样除邻居外的所有节点都为0，对计算无影响。

1. **执行结果并分析：**

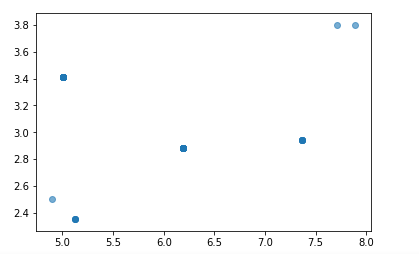
1）邻居半径=0.8，当cluster order参数与1的差值小于1e-2时结束运行，得到结果如下：



2）邻居半径=0.8，当cluster order参数与1的插值小于1e-3时结束运行得到结果如下：

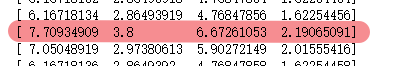


3）邻居半径=0.8，当cluster order参数与1的插值小于1e-3时结束运行得到结果如下：



分析总结：【iris数据有三个聚类】

1. 由1）和2）可知cluster order参数越接近1，越准确，这是因为cluster order参数表示的是各个点的邻居与自己越接近，那么差值之和越接近0，取e的指数越接近1，然后每个点都接近1，其总和就接近N，最后除以N所以越接近1，效果越好。
2. 有2）和3）有，邻居半径对聚类结果的影响是比较大的，如果选择的半径较小，那么聚类效果就不明显；如果半径越大，那么最后很可就就只集中在一个点，如何选择一个合理的半径是问题的关键。
3. 1）存在的其他信息，因为我们是已知聚类有三类，而1）中含右五类，通过查看数据我们可以看到有两个点的结果比较特殊：





所以我们基本数可以得出结论，就是样本中存在噪声点，即画出的两个样本。

1. **总结与反思：**

**总结**：本次复现是从收到任务的第一天开始，阅读了一下论文的摘要及部分内容，然后从第二天早上开始重读论文，此次论文确实相对来说很“简单”，但是确实然我很兴奋，iris聚类我之前做过，是使用k-means算法实现的，k-means要求我们先确定要生成的聚类个数k，然后随机选择k个点作为聚类中心来计算归类，在归类过程中检查聚类中心是否变化，对于iris分类来说，我们提前知道有多少个类，所以能得到比较好的效果，但是对于不知道聚类个数的样本来说则不太合适，需要取多次不同的k来观察，也引出了基于k-means的半监督聚类算法；除此之外，k-means的运算效率也和初始点有关，初始点的选取不同可能导致聚类效率不同；

而本次基于同步的算法则不需要选取初始值，计算出邻居后通过同步使对象间更靠近中心，不需要随机选初值，所以不存在初值对效率的影响。但是半径对该算法的影响还是存在的，不过sycn算法更为简单，也确实很厉害了。

**反思**：虽然说比较简单，不管是论文还是计算，但是我还是花了不少时间，主要是因为很久没用，一些接口忘记了导致花了更多的时间来查阅资料。还有就是代码中还存在使用循环的地方并未全部用矩阵来实现，因为想早点交差给个好印象所以暂时没有管，后序会尝试改进的。

**附上代码GitHub链接：。**