1.聚类部分不具有随机性，实际上采用的是图论中“K步可达”的原理，不断优化边与边之间的权重使得联系紧密的节点间的权重越来越大，

而不紧密的节点间的权重越来越小。这种紧密程度也不是单纯指两个节点的直接相似度，而是两个节点所构成的向量（邻居）见得相似度，

可以理解成两个朋友圈的相似度。体现在伪代码的16行。

2.时间复杂度，我需要认真分析下。

3.效果的话，目前只跑了很少量的数据集。



**时间复杂度分析1：**

1. 寻找邻居，每个对象m个属性，每个对象需要需要与其他n-1个对象计算相似度，此步的复杂度为mn(n-1),记为O(mn2).计算相似度后，比较并找到K个邻居，此步的复杂度为kn,记为O(n).
2. 每轮迭代时.总的时间复杂度为O(n+k2n+kn2);则进行T轮的时间复杂度为O(n+k2n+kn2)
3. 计算初始权重矩阵，n个对象，每个对象有k个邻居，需要计算k次，此步的复杂度为kn,记为O(n).
4. 然后计算至少P步可达的概率, 此步的复杂度为O((p-1)n\*k\*k)
5. 根据步骤2，更新邻居间的权重。n个对象，K个邻居，每个邻居进行n次乘法运算，记为O(kn2)

时间都咋都为O(n+k2n+kn2)+ O(n2+n)

**时间复杂度分析2：**

1. 寻找邻居，每个对象m个属性，每个对象需要需要与其他n-1个对象计算相似度，此步的复杂度为mn(n-1),记为O(mn2).计算相似度后，比较并找到K个邻居，此步的复杂度为kn,记为O(n).

2. 每轮迭代时.总的时间复杂度为O(nkp(k+p));则进行T轮的时间复杂度为O(Tnkp(k+p))

1. 计算初始权重矩阵，n个对象，每个对象有k个邻居，需要计算k次，此步的复杂度为kn,记为O(n).
2. 然后计算至少P步可达的概率。一步可达的复杂度为nk;2步可达的复杂度为nk2;三步可达的复杂度为nk3，p步可达的概率为nkp。至少P步可达的概率为O(nkp)p.
3. 更新原始图的权重，计算两两向量间的距离。每个对象有k个邻居，向量长度最大为(k+k2+kp=k(1-kp)/(1-k)),复杂度记为O(n\*k\* k(1-kp)/(1-k))=O(nkp+1)

总的时间复杂度为O(nkp(k+p))