1. 密度峰值聚类(DPC / density peaks clustering)算法

历史：2014年6月在《Science》上被首次提出

效果：自动选出任意形状数据的聚类中心

聚类中心：① 局部密度, 高；②与其他高密度点的距离, 大。

选取原则：密度高、距离大。

缺陷：

(1) 局部密度的计算引入了截断距离，截断距离的参数设置和选取由手工完成，影响聚类质量

(2) 对多维度数据进行计算时，维度的增加使得距离计算过程中误差对结果的影响增大

(3) 聚类中心点的自动选取依赖于人工设置的重要参数。

一、截断距离设置：

将距离按递增方式进行排序并编号，找出所有距离个数的1%~2%所对应的序号，将dc设置为该序号对应的距离值。



二、密度设置：

截断密度：



高斯密度：



三、距离设置：



算法：

(1) 生成欧式距离矩阵

(2) 计算各点的**局部密度**和**与高密度点间距离**

(3) 生成决策图，直观选择出合理的聚类中心和异常点，直接清除异常点

(4) 分配剩余点

2. 范、杨等人的工作：

针对问题：聚类中心密度高、距离大的选取原则使得密度峰聚类算法不适用于密集数据(缺乏对密集数据的定义，多大算密集)

改进动机：对各样本点的局部密度，进行平均操作可以缓解这一问题(缺乏理论证明)

改进点：算法(2) 计算各点的**局部密度**和**与高密度点间距离**

详细改进内容：

(1) 基于高斯核函数得到各个样本点的局部密度(理解为截断密度就是截断范围内的样本点个数)

(2) 对于各个样本点，选取m个(人工设定m = 1、2、…)欧式距离最近的样本点，对其密度求平均作为该样本点的新密度

3. 亟待补充的内容(重点完成1、2、3，5；4、6、7可作为待选补充项)

1. 密集数据的定义，多大算密集数据
2. 原始密度峰聚类算法在稀疏数据集和密集数据集上的表现、以及对比分析
3. 平均局部密度的理论证明，为什么平均局部密度缓解了密集数据问题？在多大程度上缓解了密集数据问题？
4. 方向的说明，为什么选取正确率作为评价指标？正确率为什么可以替代指导密集数据的复杂程度缓解？该算法的核心意义是什么？(基于表述，我认为该算法偏向于正确率提升而不是缓解密集数据问题)
5. 基于的实验都是小样本公开数据集，对数据的密集程度如何看待？在密集程度的定义的基础上在自己加设几个人工数据集，使得密集程度递增，看看效果。(说明该方法是确实解决密集程度的，而不是只在本文提到的几个数据集上有提升效果，是一个可以泛化的方法而不是一个特定类型的特定方法)
6. m取值的不同对效果的影响，为什么会有这样的效果。
7. 首先我们知道，如果上述方法确实成立，那么说明原始的局部密度计算存在缺陷，那么为什么要在有缺陷的方法上去掩盖缺陷，该方法确实掩盖了部分缺陷，但实验结果又说明该方法在某些数据集上反而不如原始DPC算法，说明缺陷并未被解决，或者说，该方法只是在进行人工微调，那么还需要证伪：是人工微调在起作用，而不是m在起作用。

4. 缺点(假设该方法确实成立，原始三大缺陷仍未解决)

(1) 局部密度的计算引入了截断距离，截断距离的参数设置和选取由手工完成，影响聚类质量

(2) 对多维度数据进行计算时，维度的增加使得距离计算过程中误差对结果的影响增大

(3) 聚类中心点的自动选取依赖于人工设置的重要参数。

(4) 增加了又一个人工参数，使得模型稳定性更差

(5) 方法缺少理论支撑，以结果为导向

5. 改进方向(假设该方法确实成立, 可进行的改进)

(1) 对多维度数据的各维度进行重要性分析，加权计算欧式距离

(2) 在加权欧式距离步骤后引入高斯核函数进行高维距离求解

(3) 局部平均局部密度(而不是全局)

6. 对于原老师说的中心域问题

原老师，上述方法已经优点偏向中心域范围了，上述方法的中心点的局部密度通过一个平均操作，其实就是在将各个节点进行密度的传播，使得临近的节点更加区域化， 特征化为区域特征。但上述方法一个较失败的点是，该区域化过程是面向全局的，而不是面向局部的，这导致了全局区域都被进行了一次smooth化操作，缺失了样本的独立特征，使得上述方法在稀疏数据上出现一些偏差。如果上述方法设置一个局部化操作，比如说中心概率较大的几个点或者几十个点，进行一步局部的平均局部密度操作，可能能达到老师说的效果。

7. PyTorch 1.3发布：能在移动端部署，支持Colab云TPU，阿里云上也能用

<https://mp.weixin.qq.com/s/BpEyfLjOfrao7CDrqcyqjA>

8. .gitignore

利用路径来控制哪些文件夹不被push

9. MarginNote3

bilibili 【学习神器Marginnote3】简介及学习实操流程