方骏-2020年2月23日-读书报告

1. 自己提出的问题的理解：
2. 提出的问题1：文中所说的k-mode究竟是属于哪一类的模型，属性有什么特征？

讨论后的理解：这里适合k-means做区别的。k-means求的是平均值，也就是说数据点的属性是可以求平均值，可以进行计算的，例如坐标，钱等。而k-mode则是对于那种分类属性的，不可以当作数值去计算，比如杯子的颜色，有红色，绿色，黄色等，这些都是分类，不能被当作数值去求平均值，标准差等，但是可以求众数等，这个也是作为k-mode的分类标准。

1. 提出的问题2：书上第129页：

https://qqadapt.qpic.cn/txdocpic/0/c195b4cace8ac61e662f46aaf74622f7/0?w=692&h=55

https://qqadapt.qpic.cn/txdocpic/0/d3249b44832d8f09d86e86aa56da765b/0?w=269&h=36

这里究竟是什么特殊情况。

讨论后的理解：这里的意思只是单纯的表示像是规则这样的矩形分割并不能满足将所有的聚类都可以完整的分割出来。如果有那种不规则形状的，比如圆形，三角形等，就不可能通过矩形分割出完整的，因此用规则表示聚类还是有缺点的，有的需要用很多的规则表示一个聚类。

1. 别人提出的问题的理解：
2. 问题3：To be safe, we may want to monitor these possible outliers over a few iterations and then decide whether to remove them. It is possible that a very small cluster of data points may be outliers. Usually, a threshold value is used to make the decision.其中，threshold value指的是什么？能具体讲一下怎么监视与这个方法的整体吗？

自己的理解：经过我们的激烈讨论，我个人的理解就是这个应该是用一个阈值来监督看是否有outliers，如果有outliers，那么可以根据这个outlier距离所有centroid的距离之和来判断这个outlier是不是真的满足异常点的情况，必须是远，而且数量很少。所以考虑距离和或者是平均距离来区别，这时候就需要定制合理的阈值来区分。如果outliers很多，那么这几个点形成的centroid肯定距离这些点很近，所以和不会特别大，所以定制合理的阈值才是根本，需要多次迭代判断阈值。

1. 问题4： 使用k-均值算法的时候，为什么全局最小值对于大规模数据集来说在计算上是不可行的？

自己的理解：每进行一次k-means算法，会得到一个局部极小值，说是局部极小值的原因，是因为每次算法初始的centroid选取是随机的，每次都可能会不一样，这样就没法确定自己根据这些centroid得到的解是不是最小值，这些值只能是极小值。当数据量大的时候，我们没办法遍历所有的初始centroid，因此不能保证得到的聚类就是全局最小值，因为还有一些极小值没有考虑到。

1. 问题5： 在4.3.1节，the centroid representation alone works well if the clusters are of the hyper-spherical shape. If clusters are elongated or are of other shapes, centroids may not be suitable.是因为可以计算出the radius，standard deviation of the cluster吗？

自己的理解：这里因为是一些规则图形，比如超球体，超正方体之类的，形状规则，那么centroid选择它的中心就很容易且有代表性。如果是不规则图形，选择其centroid就是无意义的，不具有代表性。

1. 读书计划

1、本周完成的内容章节：4.1-4.3，5.1前4页

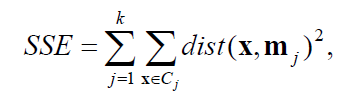
2、下周计划：第五章剩余部分和涉及的3.7章节

四、读书摘要及理解

1、读书摘要及理解

本次阅读读的内容是无监督学习。无监督学习和有监督学习的根本区别就是在于数据是没有标记的，也即没有类别。而无监督学习的目标就是把这些数据根据他们之间的数据关系进行分类，得到数据的类别。无监督学习虽然有多种，但是最具代表性的还是聚类方法。聚类方法中最简单、最有效率的就是k-means算法。

一个聚类里的数据都是性质或是属性相近的数据点，k-means算法中的k指的是划分出来k个类，means就是指一个类中的平均值作为参考或是centroid。过程就是先随机选择k个centroid，其余每一个点都要和每一个centroid计算距离，距离采用欧氏距离，然后选择距离中最短的对应的centroid代表的类加入进去，依次进行，分好类后，计算每一个类的均值，作为新的centroid，进行下一边循环迭代。知道达到我们的标准。标准有三：数据点的类别不再变化或很少变化；centroid不再变化；SSE最小。这里SSE为：



改进的disk k-means算法就是简单的找中间变量记录每一类的数据和以及对应的数据点的个数，这样在计算means的时候就不需要重新扫一遍数据了，效率更高，占的空间更小。

k-means的优缺点也比较明显，缺点是：算法只使用于数据集中的均值有明确定义的数据；用户需要提前指定k的个数，虽然可以不提前指定，但是提前指定可以减少迭代次数，更加趋向于自己想要的分割结果；算法对于outliers非常敏感，如果存在，会对类的分割产生很大的影响，因为这个outlier本不该出现在任何类里；初始种子的选取很敏感，不同的种子带来不同的分割结果；不适用于超椭圆体之类的，因为聚类中心可能会过于靠近，导致分类没有意义。

outlier的解决方法可以设定阈值，详细说明见上述问题。还可以进行采样，避开outlier，或是在迭代中观察异常点并将其剔除。

可以代表聚类的方法有如下：用每一个类的centroid来代替表示聚类，这是最常用的一种方法；用规则rules来表示聚类，虽然当聚类形状不规则时需要用多个rules表示一个聚类；对于k-mode型的分类算法，可以用一个类中的众数来表示。