BIRCH 与层次聚类算法的比较

一、研究目的

BIRCH 算法（Balanced Iterative Reducing and Clustering using Hierarchies）一次扫描能够产生一个基本聚类，多次扫描能够改善聚类结果。它是一个增量的聚类方法，对于数据的聚类决策是基于已经处理过的数据点，而不是全部样本空间，因此能够提高计算速度，所以它天生就是为处理大规模的数据集和数据流聚类而设计的。本文使用普通的层次聚类法与 BIRCH 算法作比较，观察两者在静态数据流和动态数据流聚类上的表现，凸显 BIRCH 算法在运行速度、在有标签和无标签两种情况上更优的聚类效果。

二、算法及评价标准简介

2.1 层次聚类法

层次聚类法是传统的统计聚类分析方法之一。先计算样本之间的距离。每次将距离最近的点合并到同一个类。然后，再计算类与类之间的距离，将距离最近的类合并为一个大类。不停的合并，直到合成了一个类。其中类与类的距离的计算方法有：最短距离法，最长距离法，中间距离法，类平均法等。比如最短距离法，将类与类的距离定义为类与类之间样本的最短距离。

由于 BIRCH 算法是层次聚类方法的一种，两者的聚类思想有诸多相似之处，如两种方法均无需事先指定聚类个数，聚类个数可根据设定的阈值选取，因此选传统的层次聚类法与 BIRCH 算法作比较，查看两者对数据流的聚类效果。

2.2 BIRCH 算法

BIRCH（Balanced Iterative Reducing and Clustering using Hierarchies，利用层次结构的平衡迭代归约和聚类）是由 T. Zhang 等人[1]于 1996 年为处理超大规模聚类设计的一种层次聚类方法。Birch 算法是一种非常有效的、传统的层次聚类算法，该算法能够用一遍扫描有效地进行聚类，并能够有效地处理离群点。Birch 算法是基于距离的层次聚类，综合了层次凝聚和迭代的重定位方法，首先用自底向上的层次算法，然后用迭代的重定位来改进结果。层次凝聚是采用自底向上策略，首先将每个对象作为一个原子簇，然后合并这些原子簇形成更大

的簇，减少簇的数目，直到所有的对象都在一个簇中，或某个终结条件被满足。

2.3 轮廓系数

轮廓系数（Silhouette Coefficient），是聚类效果好坏的一种评价标准，适用于实际类别信息未知的情况。最早由 Peter J. Rousseeuw 在 1986 提出。它结合内聚度和分离度两种因素。可以用来在相同原始数据的基础上用来评价不同算法、或者算法不同运行方式对聚类结果所产生的影响。其基本原理为：对于单个样本，设 a 是与它同类别中其他样本的平均距离，b 是与它距离最近不同类别中样本的平均距离，其轮廓系数为：

$$sw_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$$

对于一个样本集合，它的轮廓系数是所有样本轮廓系数的平均值。

$$\overline{sw} = \frac{1}{n}\sum_{i=1}^{n} sw_i$$

轮廓系数的取值范围是[-1,1]，同类别样本距离越相近不同类别样本距离越远，分数越高。如果观测值的 sw_i 值接近 1，则这个数据点比邻近点更靠近自己的类；如果观测值的 sw_i 值接近-1，则这个数据点没有被很好的聚类；如果观测值的 sw_i 值接近 0，则这个数据点可以归于当前的类或离这个数据点最近的一个类。Kaufman 和 Rousseeuw 提出 $\overline{sw}$>0.5，则可以进行合理的聚类，$\overline{sw}$<0.2 时，则表示数据集不存在很好的聚类结构。

2.4 纯度

纯度（purity）是已知样本真实标签时极为简单的一种聚类评价方法，只需计算正确聚类的文档数占总文档数的比例。其中 $X = \{x_1, x_2, ..., x_k\}$ 是聚类的集合 x_i 表示第 i 个聚类的集合。$Y = \{y_1, y_2, ..., y_r\}$ 是真实标签集合，y_j 表示第 j 个真实标签，N 表示文档总数。

$$purity = \frac{1}{N}\sum_{k} max|x_i \bigcap y_j|$$

三、BIRCH 算法在鸢尾花数据集的聚类分析

对鸢尾花数据集的 Petal.Length 和 Petal.Width 两个变量使用 BIRCH 算法聚类，当 BIRCH 算法的阈值为 0.65 时能聚成 3 类。图 1 中颜色为鸢尾花真实的分类，形状为 BIRCH 算法聚类标签，可以看出位于左下方 setosa 类型的鸢尾花的点因为离其他类型点较远，因此很好地被聚成了一类，图中有几个蓝色的三角形的点，说明有少数 virginica 类别的鸢尾花因与 versicolor 类型的鸢尾花距离较近而被聚到此类。该聚类的轮廓系数为 0.6613，从数据结构层面来说聚类效果较好，纯度为 0.9467，仅 8 个点被分错。

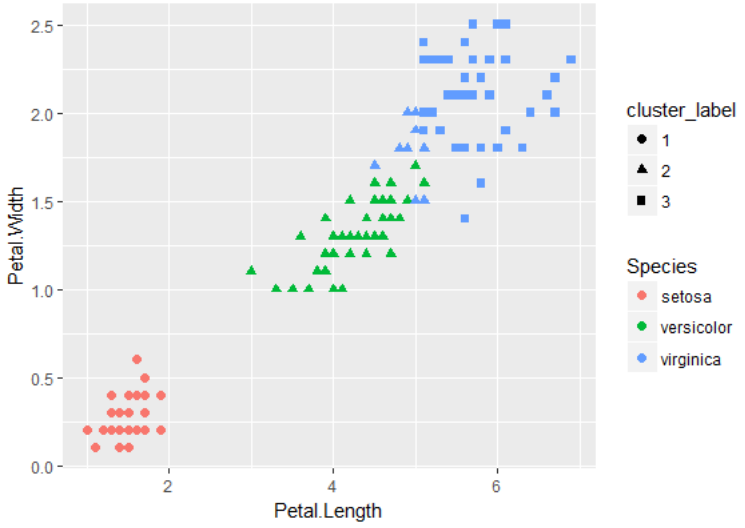

图 1 鸢尾花数据集聚类效果图

四、BIRCH 算法与层次聚类法的比较

4.1 基于静态数据流

使用 R 语言 stream 包中的 DSD_Gaussians 函数随机生成 500 条服从高斯分布的数据流样本，维数为 2，中心点个数为 3。如图 1 所示，当 BIRCH 算法的阈值为 0.18，层次聚类法的阈值为 0.65 时，两种方法均把样本聚成 3 类，且效果相似。但 BIRCH 算法聚类结果的轮廓系数为 0.5778，层次聚类结果的轮廓系数为 0.5613，说明在未知样本标签时，仅从数据结构上衡量聚类效果上还是 BIRCH 算法稍胜一筹。BIRCH 算法聚类结果的纯度为 0.9858，层次聚类结果的纯度为 0.9748，说明在已知样本标签时，BIRCH 算法仍比层次聚类法表现要稍好一些。

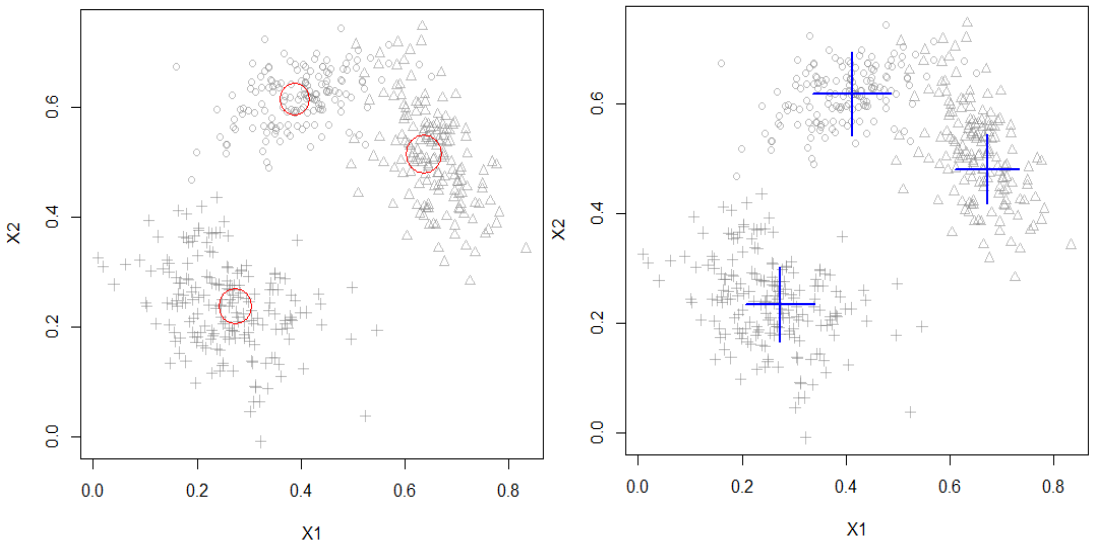

图 2 BIRCH 算法（左）与层次聚类法（右） 的聚类效果图

4.2 基于动态数据流

使用 R 语言 stream 包中的 DSD_Gaussians 函数随机生成 25000 条服从高斯分布的数据流样本，维数为 2，中心点个数为 3。每次取 500 条用 BIRCH 和层次聚类法分别进行聚类，记录轮廓系数和纯度。在 50 次聚类中，使用默认参数（BIRCH 算法的阈值为 0.1，层次聚类法的阈值为 0.2）的两种算法聚类个数均为 3 类。

从图 2 图 3 可以看出，BIRCH 算法的轮廓系数、纯度均在层次聚类法之上，表 2 显示，BIRCH 算法的轮廓系数均值为 0.3595，方差为 0.0001，波动性小，并且有轻微上升的态势，层次聚类法的轮廓系数均值为 0.2341，方差为 0.0004，波动性相对较大。BIRCH 算法的纯度均值为 0.9816，方差为 3.72e-05，十分稳定，层次聚类法的纯度均值为 0.9614，方差为 0.0002，波动性相对较大。说明无论是从数据结构角度，还是已知样本标签角度，BIRCH 算法都比层次聚类法聚类效果更好更稳定。

从运行时间上看，BIRCH 算法比层次聚类少约 2 秒，由于数据量少显得两者运行时间相差不是很大，但在处理大规模数据聚类时，BIRCH 算法节省的时间很可观。

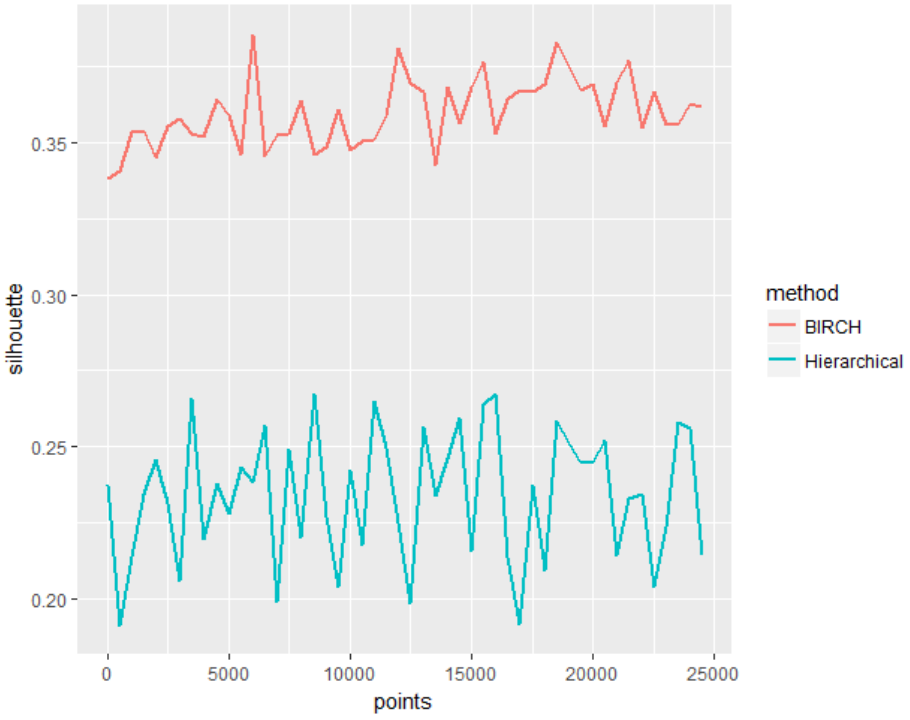

图 3 两种聚类算法在动态数据流聚类时的轮廓系数对比

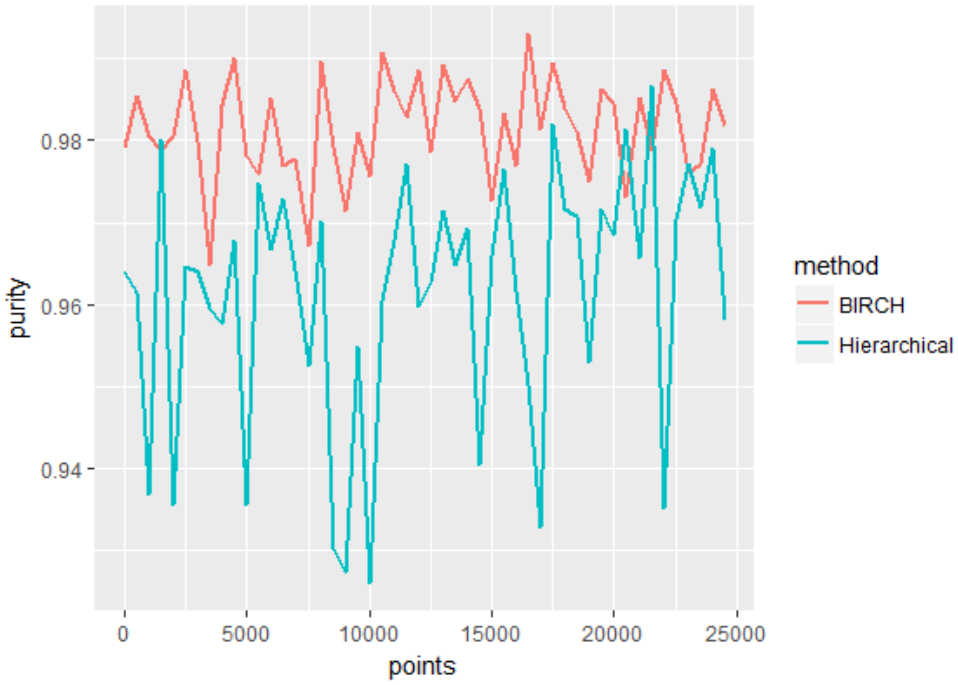

图 4 两种聚类算法在动态数据流聚类时的纯度对比

表 1 两种聚类算法在动态数据流聚类时轮廓系数与纯度的对比

方法	Birch	Hierarchical
轮廓系数均值	0.3595	0.2341
轮廓系数方差	0.0001	0.0004
纯度均值	0.9816	0.9614
纯度方差	3.72e-05	0.0002
运行时间	4.61s	6.83s

五、结论

相比于传统的层次聚类法，BIRCH 算法运行速度快，只需一遍扫描数据就可以有效地进行聚类，且聚类效果更好。但在调参方面，BIRCH 要比层次聚类法复杂得多，因为它需要对 CF Tree 的几个关键的参数进行调参，这几个参数对 CF Tree 的最终形式影响很大。