关于论文《Weighted Numerical and Categorical Attribute Clustering in Data Streams》的学习报告

论文提出的算法主要针对具有数值型属性与标称型属性混合的数据流的聚类问题。算法的具体流程如下：

1. 种子生成算法：首先选择数据集中的任意一个对象加入到种子集中，每次从数据集取出一个对象加入到种子集中，取对象的准则是基于数据集中的某个对象与种子集中所有对象的距离和，作为加入到种子集的可能性。以上步骤迭代直到种子集的大小达到预设的数值。
2. 权重k-meas++算法：对于每一个包含s个对象的数据块，首先基于种子生成算法生成k'个聚类，然后把数据集中的每个数据对象加入到最近的聚类中并对每个聚类的中心进行更新。同时，这一步算法还会完成权重的更新，权重的更新是把以前数据块的权重以及当前数据块得到的新权重进行加权求和，并且以前的权重占的权值更大，这是因为以前存在的总的数据块包含的信息更多。
3. 流聚类算法：对于第一个数据块，我们先生成k个宏聚类，生成的方法为把第一个数据块采用权重k-meas++算法生成k'个微聚类，再把这些微聚类根据距离合并直到生成k个宏聚类。此后对于每一个新的数据块，采用权重k-meas++算法生成k'个微聚类，然后把每个微聚类加入到最近的超聚类中或生成一个新的超聚类进而完成当前数据块的聚类，然后更新宏聚类。最后根据阈值把距离相近的两个宏聚类合并，直到生成最终的聚类结果。

细节讨论：

1. 计算数值型属性之间的距离：

首先定义属性的非均匀性：对于某一属性，该属性的非均性定义为该属性下的原始序列与基于原始序列构建的等差数列的差值平方和。很容易想象得出，假如原始序列的分布特性类似于等差序列，那么计算出来的非均匀性将会接近0，对于这种分布特性的序列，是无法把序列进行正确的聚类划分的，因为找不到明显的划分界线。而用

具有非均匀性较大的属性来进行聚类划分的话，得到的效果就会很明显。（论文的作者们通过一个简单的实验对这一特性进行了验证，也很形象）。

两个对象之间的数值属性的距离的计算公式如下：

$$D_1(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{r} \omega_i (x_i - y_i)^2$$

根据公式，每个属性对距离的贡献并不是相等的，而是根据该属性的非均匀性来进行确定，前面提到应该用更具有非均匀性的属性来进行聚类划分，因此，非均匀性大的属性应该有更大的权重，权重的值为该属性的非均匀性除以所有属性的非均匀性。

2. 计算标称型属性之间的距离：

两个对象在同一标称型属性下的距离定义为：

$$D_{A_i}(a_{iu}, a_{iv}) = \sum_{z_j \in \mathcal{Z}} \left| \frac{count(z_j, S_{iu})}{|S_{iu}|} - \frac{count(z_j, S_{iv})}{|S_{iv}|} \right|$$

其中S_{iu}定义为标称型属性A_i的值为a_{iu}的对象的集合，z_j为聚类的标签，也就是对于历史数据块进行聚类的结果。因此标称型属性的距离是通过当前数据块的值以及该值对应的历史数据块中的结果来进行相似度计算的。最终两个对象的标称型属性的距离为每个属性的聚类之和。（我觉得这里可以基于香农熵来赋予每个标称型属性一定权重再进行计算，可能会达到更好的效果）。

3. 计算微聚类之间的距离：

微聚类的定义：微聚类可以通过三个参数进行表征，分别是微聚类的中心c、微聚类的半径r以及微聚类的数据点的个数p。c向量的计算公式是c的数值属性部分是通过每个属性的平均值来计算的，标称属性部分是通过出现最多次的值来进行表征的。r的计算公式为聚类的每个数据点与聚类中心c的平均距离。

两个微聚类之间的距离为聚类中心的距离除以两个距离的半径之和

4. 计算宏聚类ma与微聚类mi之间的距离：

宏聚类的定义：宏聚类就是一系列微聚类的集合。

一个宏聚类与一个微聚类之间的距离的计算公式为宏聚类ma中的每一个微聚类与微聚类mi之间的距离中最小的值。