机器学习与数据挖掘作业一

KNN算法

数据科学与计算机学院　17大数据与人工智能

17341015　陈鸿峥

问题 1. How to measure the distance for categorical attributes with more than two values? (Please consider from both sequential and non-sequential aspects.)

解答. 对于有序[1]的离散(categorical)属性，可以直接将每个类别定为一个数字。举个例子，假设有一年龄属性分为三个类别，那么可以按下表进行数字化处理，因为年龄各类别间存在大小关系。

年龄	<18岁	$18\sim30$岁	30岁以上
数值编码	0	1	2

对于无序的离散属性，则可以采用独热码(one-hot encoding)，以确保两两类别之间距离相同，因为类别之间并没有大小差异，而只是指代的东西不同。比如，对于下列三种颜色，可以进行如下编码。

颜色	红	黄	蓝
数值编码	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$

当categorical属性转换成numerical属性后（离散转连续），那么就可以正常按照欧式距离、曼哈顿距离等进行度量。

问题 2. How to quickly retrieve K nearest neighbors of a given query (sample)?

解答. 假设询问的点是$\mathbf{x}_n$，特征的维度为$d(n >> d)$。简单的算法是$\mathbf{x}_n$与$\mathbf{x}_i(i = 1,\dots,n-1)$分别算出距离后，对距离数组升序排序，取出最小的$k$个点，即为$k$个邻居。算距离复杂度为$O(n)$，排序复杂度为$O(n\log n)$，顺序选邻居复杂度为$O(1)$，总的复杂度为$O(n\log n)$。

可以看到时间开销都花在排序上面，而事实上我们并不关心整体数组的序，而只关心最短距离的那k个点，故在$n >> k$的情况下，直接遍历k次数组每次记录最小距离，时间复杂度开销仅为$O(kn)$，关于n的线性复杂度。

通过维护一个大小为k优先队列/最大堆，我们可以做得更快。每次插入元素时进行比较堆顶元素，如果当前距离小于堆顶，则将当前值放入堆中，这样子建堆的复杂度为$O(n\log k)$，k邻居即为该堆中的k个结点。

[1]问题中的sequential和non-sequential理解成有序(ordered)和无序(unordered)

最后，目前比较常用的方法是KD树，用超平面对高维空间的点进行二划分（以均值为界），构建一棵平衡二叉树，建树复杂度为$O(dn\log n)$且可以重用，查询复杂度为$O(\log n)$。因此KD树增大了预处理时间，但是却能够让KNN的查询效率大大提升。