孤立森林算法的增量学习

感谢 https://github.com/shubhomoydas/ad_examples

孤立森林算法本身以及他的无监督学习特性，导致其不能够向随机森林那样对每棵树进行更新。

随机森林通过统计每个节点的信息，计算信息增益，将节点继续分裂，这种操作对森林中的每棵树进行处理。孤立森林算法本身令难以通过对每棵树本身进行处理、分裂、达到增量学习的目的，其更多采用了森林层面的处理方法。

孤立森林的在线学习主要有两种方案:

1) 将森林中时间最久的 20%（超参数）的树删除，用新的数据集训练得到的树进行替代。

2) 利用 KL 散度将树进行替换

利用 KL 散度将树进行替换：

假定所有样本都是独立同分布的。这样每批样本通过孤立森林打分得到的分数的分布应该是类似的，利用 KL 散度计算不同分布间的差异。将差异大的树进行替换。也就是说，先检测是否出现了数据集的偏移，如果发生了偏移，那就将影响大的树重新训练。

步骤：

1. 将原始数据集随机分成两个部分 A 和 B
2. 对孤立森林中的每棵树计算 A 和 B 上的预测分数，将分数放到 bins 中
3. 每棵树都根据 A 和 B 得到了 bins，相当于预测的得分的分布，计算 A、B 得分分布间的 KL 散度
4. 重复 10 次步骤 1-3， 并将结果求平均值
5. 这样得到了 T 个 KL 散度的平均值
6. 给定参数 α，默认 0.025，得到（1-α）分位数的 KL 值， 称作 KL-q
7. 这次每棵树对整个原始数据集进行打分，得到 bins 的分布，将这 T 个分布作为 baseline，称作 P
8. 接收一个新的数据集，用每棵树对新数据集进行打分，得到了 T 个分布成为 Q
9. 对每棵树，计算 P、Q 中对应分布的 KL 散度，如果有 2*α*T 棵树的 KL 散度超过了 KL-q，那么执行 10-12
10. 用新数据集生成新的树，替换掉之前 KL 散度大于 KL-q 的树
11. 用新的模型和新的数据集重新计算 KL-q 和 P（步骤 6,7）
12. 使用一些可用的有标签的样本进行模型权重调整。这个步骤对模型发生改变以后的效 果提升有一些帮助

关于 KL 散度，这有一篇生动形象的解释。

https://www.jianshu.com/p/43318a3dc715?from=timeline&isappinstalled=0