**lab6实验报告**

**-------闫世杰 2020200982**

**利用priority\_queue优化KNN search**

**添加数据结构**

**struct Bin{**

**KDT \*T; //记录候选节点,即待检查平面**

**long long distance;//记录该节点的split\_line距离Target的直线距离,将其作为优先 队列的排序标准,使得距离最小的节点(及最优节点)位于最前**

**}**

**在自顶向下递归寻找Target所在的最小cell过程中,在每次进入一子树的同时,判断当前最小距范围是否和该平面有交点,如果有交集,将另一棵子树加入待考察的priority\_queue中,作为候选,当遍历到叶子节点时,从优先队列中选择最优的节点进行回溯,如果最优节点的平面与当前最小距范围无交点,说明已经达到了最优,直接清空优先队列返回即可**

**不断更改E\_Max 的值,对于测试所给出的数据,发现E\_Max=5时,BBF的准确率已经有95%,当E\_Max=20时,已经完全正确**

**由于测试数据维度较小,且数据规模不大,整个搜索过程只需不到1s,且由于BBF需要维护一个优先队列.因此在数据测试上,发现BBF必普通的KNN还要慢一点**

**时间测试表(E\_MAX=20):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 点数|搜索次数 | 10000|10000 | 20000|20000 | 50000|50000 | 100000|100000 |
| BBF\_KNN | 0.056999s | 0.087552s | 0.235076s | 0.488669S |
| NOR\_KNN | 0.033032s | 0.049679s | 0.094528s | 0.18016s |