LSM-Tree Phase3 报告

2025 年 4 月 27 日

1 背景介绍

本阶段承接上一阶段。上一阶段中的 search_knn 函数支持精确匹配下的语义检索功能，效率较低。在此基础上，本阶段实现了 search_knn_hnsw 函数，采用 HNSW 算法，用更快的效率寻找与查询字符串较接近的 k 个字符串。

search_knn_hnsw 函数与四个关键参数相关，如表 1。

参数名	作用
M	被插入节点与图中其他节点建立的连接数
M_max	每个节点与图中其他节点建立的最大连接数
efConstruction	候选节点集合数量
m_L	最大层数

表 1: search_knn_hnsw 函数的关键参数

本实验探究了 search_knn_hnsw 函数与 search_knn 函数的正确率、性能差异，并探究了 search_knn_hnsw 函数中，参数对正确率和性能的影响。

2 测试

本实验的测试分为两个部分：正确率测试和性能测试。

在正确率测试中，实验改变相关参数，探究参数对正确率的影响；此外，与 search_knn 函数的正确率比较。

在性能测试中，改变相应参数探究参数对性能的影响，并测试 search_knn_hnsw 函数与 search_knn 函数相比带来的性能提升。

2.1 实验设置

实验运行在 Linux 环境下。

实验中，以 data/trimmed_text.txt 的每一行作为一个句子输入，以 data/test_text.txt 的每一行作为查询语句，比较 search_knn_hnsw 函数与 search_knn 函数的正确率、性能差异。此外，改变表 1中参数，探究 search_knn_hnsw 函数的正确率和性能变化。

计时使用 std::chrono 库。

2.2 预期结果

search_knn_hnsw 函数与 search_knn 函数相比，应该正确率下降，性能提升。

M 增大时，插入节点可以与更多节点建立连接，此时在不考虑其它参数相互作用的情况下，应该正确率提高，性能下降；M_max 增大时，应该正确率提高，性能下降；efConstruction 增大时，可以从更多的候选节点中选取邻居，应该正确率提高，性能下降；m_L 增大时，应该性能提升。

除此之外，参数之间涉及相互作用。如果 M 的值和 M_max 的值过于相近，每次插入都更容易导致之前的边被删除，可能导致边聚集在最后插入的点附近，从而降低正确率。

2.3 实验结果与分析

调整 search_knn_hnsw 函数的参数，使其正确率达到可接受范围后，与 search_knn 函数相比的性能及正确率如表 2，具体测试结果如图 1。

函数 / 操作	search_knn	search_knn_hnsw
正确率	84.4%	75.0%
插入用时（360 次，ms）	22	118075
删除用时（360 次，ms）	95612510	73590
总用时（ms）	95612532	191665

表 2: knn 与 hnsw_knn 正确率及性能比较

可以看到，以正确率下降 10% 左右为代价，新增函数 search_knn_hnsw 的用时只用原函数的 0.2% 左右。考虑到原函数的插入性能较高，所以在数据量极大但所需查询次数较少或需要精确匹配的情况下，原函数更合适；在其它情况下，search_knn_hnsw 可以带来非常明显的性能提升。

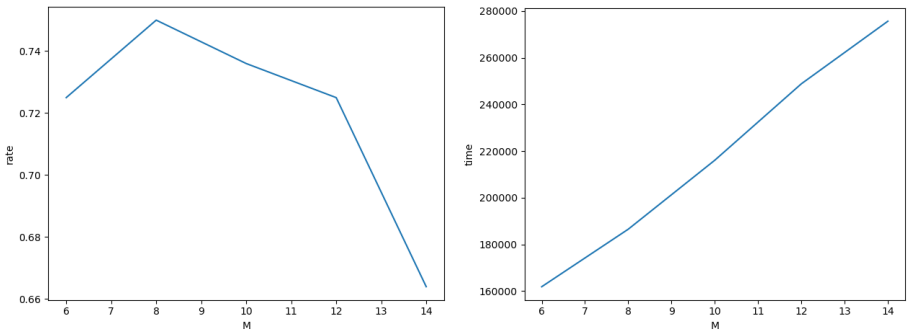
改变参数时，对应的正确率与用时变化分别如图 2、图 3、图 4、图 5。

```
 knn: 304/360, rate: 0.844
hnsw: 270/360, rate: 0.750

knn:
insert time: 22ms
 query time: 95612510ms

hnsw:
insert time: 118075ms
 query time: 73590ms
```

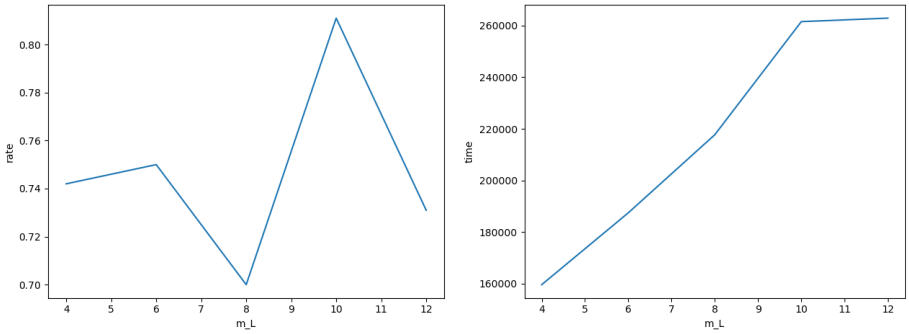
图 1: 测试结果



(a) M 与正确率关系

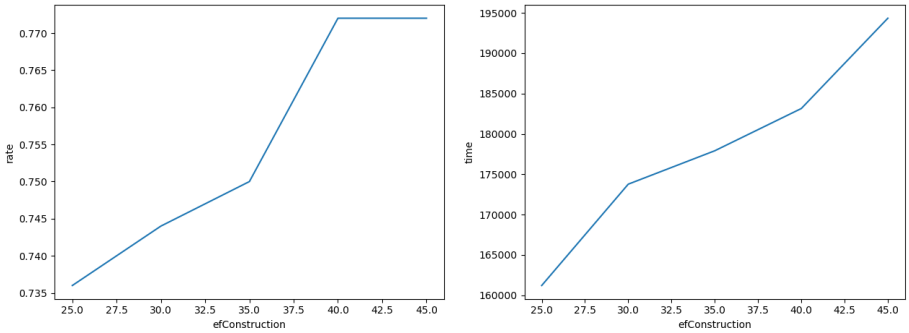
(b) M 与性能关系

图 2: 参数 M 与正确率、性能的关系



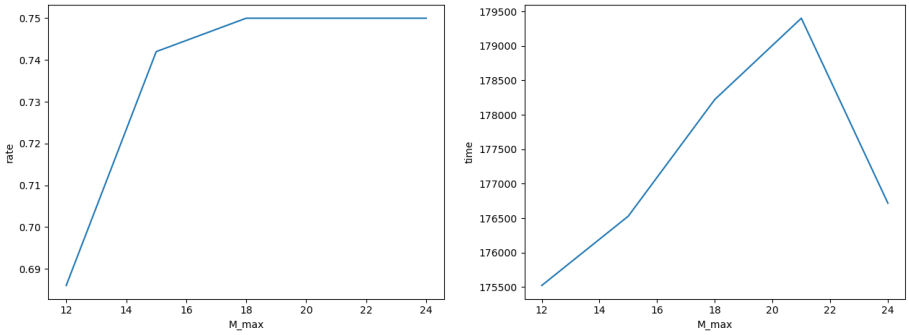
(a) m_L 与正确率关系　　(b) m_L 与性能关系

图 3: 参数 m_L 与正确率、性能的关系



(a) efConstruction 与正确率关系　　(b) efConstructio 与性能关系

图 4: 参数 efConstruction 与正确率、性能的关系



(a) M_max 与正确率关系　　(b) M_max 与性能关系

图 5: 参数 M_max 与正确率、性能的关系

基本上，在各个参数增大时，性能都会随之下降。efConstruction 和 M_max 增大时正确率会上升，但在上升到一定程度时，不会继续带来正确率的提升，可能是因为数据量较小，参数提高的作用无法凸显。M 和 M_max 接近会导致正确率下降，可能是因为每个点连接的边过多，导致靠前插入点的较多边被删除，靠后插入点连接的边较多，数据表现出一定的聚集性。图中无法看出 m_L 对正确率的显著影响，理论上层数增大会使图结构更复杂、增大正确率，数据表现可能与测试用例的数据分布有关。

3 结论

与精确匹配相比，新增的 search_knn_hnsw 以 10% 以内的正确率下降为代价，带来了极大的性能提升。

此外，关键参数对 HNSW 算法的正确率及性能均有显著影响。参数增大时，基本上性能都会下降；efConstruction 和 M_max 在一定范围内增大时会带来正确性提升；M 和 M_max 接近会导致正确率下降；m_L 对正确率的影响在实验中表现的不明显。

HNSW 算法的性能还受到随机种子以及数据集本身性质的影响。

4 致谢

感谢知乎、维基百科等博客、网站提供的参考；感谢 deepseek、kimi 等大模型提供的思路与帮助。

感谢提供支持的朋友们。

5 其他和建议

感觉大模型太受各种参数以及乱七八糟的因素影响了……不同随机数种子可以带来 10% 的正确率差异

希望以后可以有更详细的问题文档，描述一下不同平台上不同参数可能的正确率范围

以及感觉文档有些地方不是很清楚（我最开始读的时候，受文档中图的影响，以为高层的连线要和底层重合），建议以后可以添加参考链接