Work Progress

kNN Search with Parallel Incremental Query Answering

Jaouhara Chanchaf

Tuesday Nov. 28th, 2022

1. Summary

Done:

AI 3	Reproduce Kashif PQA plot using query columns of the same size.
AI x	Prepare internship presentation.

In progress: Fullbright Application.

Not started:

AI 1	Kashif PQA: Fix bug in increment barrier.
AI 2	LSH Ensemble: get familiar with the code and run it on WDC.
AI 4	PEXEO: run experiments using the same settings in the paper.
AI 5	Kashif: Stop when NN distance changes. Measure recall based of results from LSH and PEXESO.
AI 6	Pick a query vector and manually label accurate NN. Measure recall and visualize the correlation between the NN accuracy and distance to the query vector.

2. Performance

Experiment settings:

Dataset: 100k tables, 494k columns, 5M vectors.
Query: 10 Queries of size 100.
k: 1,000

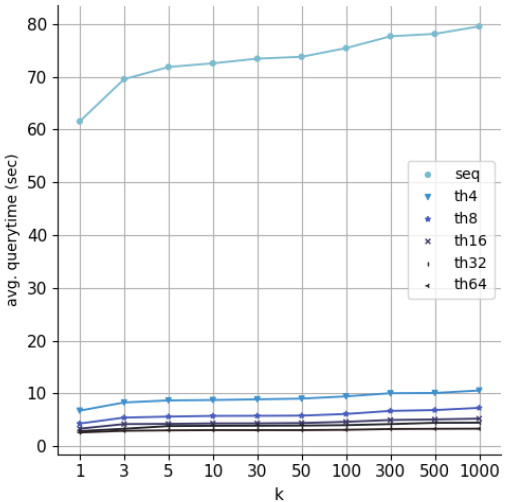

Figure: Kashif Recall

2. Performance

Experiment settings:

Dataset: 100k tables, 494k columns, 5M vectors.
Query: 10 Queries of size 100.
k: 1,000

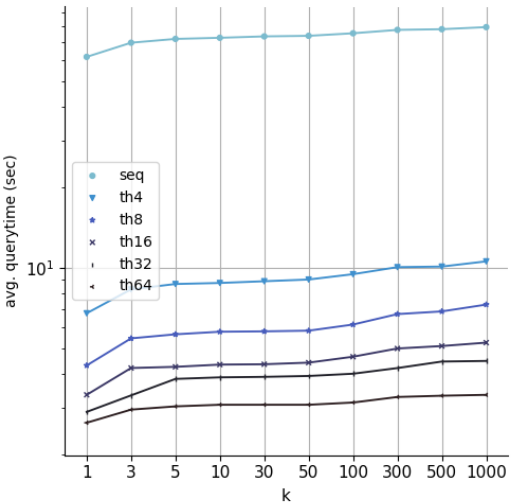

Figure: Kashif Recall

3. Discussion

- Large k values yield better recall.
- The optimal k value is data dependent.
- Solution: set $k = datasetsize$ and perform Incremental NN Search.

(!) Important:

- Suppose $\#\mathbf{workers} = 32$, $\#\mathbf{queryvectors} = 1k$, $\mathbf{k} = 10M$.
- Kashif will only perform incremental NN search for the first 32 vectors, the rest 968 vectors will not participate in improving the recall.
- The NN search for next 32 vectors will only start once all the first 32 query vectors were answered!
- But we assume that the query time is bad when k is large!
- Should workers alternate between query vectors? i.e. start answering another query vector before completing NN search for the current query vector. To ensure that we get identical (or close) NNs for all query vectors early.

4. Deadlines

- Research Proposal **December 1st**.
- Fullbright Application **December 15th**.