Work Progress

kNN Search with Parallel Incremental Query Answering

Jaouhara Chanchaf

Tuesday Nov. 8th, 2022

1. Summary

Done:

AI 1	Check the requirements for the Fulbright Joint-Supervision Program.

In progress:

AI 2	Kashif Parallel Incremental Query Answering: Measure the impact of threads on the query time.
AI 3	Read Progressive Similarity Search paper.

Done:

AI 2	Measure recall based on column ID.

Issues:

- Incremental kNN search barrier not working properly when $|Q| > \#threads$.
- Cannot create index for larger datasets (5 Million tables).

2. Performance Comparison

Experiment over 1M tables, 4.9M columns, 50.3M vectors (9 GB).

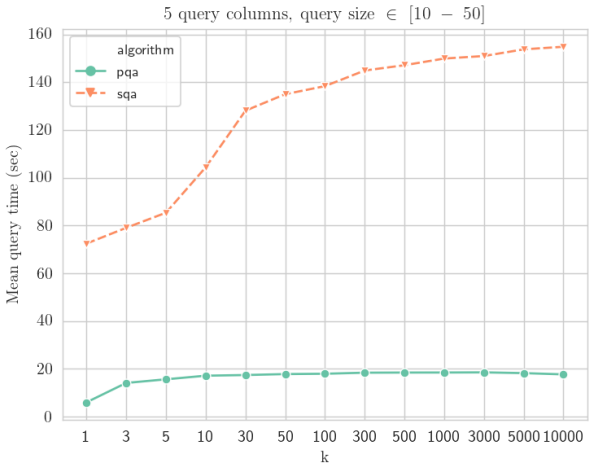

Figure: Kashif performance: Parallel (Incremental) Query Answering (pqa) Vs Sequetial Query Answering (sqa)

(!) Waiting for brute force results to measure recall and precision.

3. Discussion

1. Parallel Incremental Query Answering when $|Q| > \#thread$:
   - Submit the query column to a thread pool with a job queue.
   - Each query vector in the query column is stored as a job in the job queue.
   - A worker thread must acquire the lock to pop a job (i.e. query vector) from the job queue.
   - If the worker thread finished the current job and the job queue is empty the worker thread goes to idle stat.
   - The algorithm stops when the queue is empty and all the worker threads are in idle state.

3. Discussion

2. Approximating the pdf using the kNN distance distribution of witnesses requires defining k in the training phase (to estimate the 1nn, 2nn, ..., knn distance distributions).