Work Progress

kNN Search with Parallel Incremental Query Answering

Jaouhara Chanchaf

Wednesday Oct. 19th, 2022

1. Summary

Done:

AI 2 (prev. week)	Kashif: Only store a recall matrix at the coordinator thread and update NN recall incrementally.
AI 1	Read about LSH forest. and find the relationship between m (number of sampled hash functions) and delta (the error probability).
AI 2	PUFFINN: How is the $(1-\delta)*k$ the expected recall for $k-\delta$NN search.
AI 4	Kashif: finish multi-thread Parallel IQA implementation run experiments on khawarizmi / HPC.

In progress:

AI 3	PUFFINN: explain the stopping condition.

Gaps:

- Measure query time in Kashif Parallel IQA (in single thread and multi-thread code).
- Read "Data Series Progressive Similarity Search with Probabilistic Quality Guarantees"

2. Recall guarantees

Recall guarantee in PUFFINN (M. Aumüller et al. LIPIcs, 2019):

- **What:** How to perform parameter-less, probabilistic kNN search?
- **Why:** Exact search is a hard problem.
  Probabilistic approaches suffer from low recall.
  Must find the optimal parameter values to achieve good performance.
- **How:** An LSH-based Index that only requires recall guarantee and index memory size as parameters to perform kNN search.

2. Recall guarantees

Recall guarantee in PUFFINN (M. Aumüller et al. LIPIcs, 2019)

"(...) we measure the individual recall of each query, i.e., the fraction of points reported by the implementation that are among the true kNN(...)"

$$\text{Expected Recall} = (1 - \delta) * k \text{ (for one query vector)}$$

- The expected value (mean) for a random variable X is:

$$E(X) = \sum x * P(x)$$

- For $k = 1 (1NN)$ the expected recall is:

$$r_{1NN} = \sum x * P(x), \quad x \in \{0, 1\}$$
$$r_{1NN} = 1 * (1 - \delta) + 0 * \delta$$
$$r_{1NN} = (1 - \delta)$$

- For $k (kNN)$ the expected recall is:

$$r_{kNN} = r_1 + r_2 + ... + r_k$$
$$r_{kNN} = (1 - \delta) * k$$

3. Recall in Kashif

Recall in Kashif (for one query vector, $|Q| = 1$):

$$\text{Recall} = TP$$
$$\text{Recall (in \%)} = TP/k$$

- For $k = 1 (1NN)$ recall is equal to:

$$r_{1NN} = \frac{TP}{TP + FN}, \quad TP \in \{0, 1\}, \quad FN = 1 - TP$$
$$r_{1NN} = \frac{TP}{TP + (1 - TP)}$$
$$r_{1NN} = TP$$

- For $k (kNN)$ the recall is:

$$r_{kNN} = \frac{TP}{TP + FN} \quad TP \in [0, k], \quad FN = k - TP$$
$$r_{kNN} = \frac{TP}{TP + (k - TP)}$$
$$r_{kNN} = \frac{TP}{k}$$

3. Recall in Kashif

Recall in Kashif (for $|Q|$ query vectors)

$$\text{Recall} = TP'$$
$$\text{Recall (in \%)} = \frac{TP'}{|Q| * k}$$

- For $k(kNN)$ and $|Q| = 1$ the recall is:

$$r_{kNN,\ |Q|=1} = \frac{TP}{k}$$

- For $k(kNN)$ and $|Q| \geq 1$ the recall is:

$$r_{kNN,\ |Q|\geq 1} = \frac{TP_1 + ... + TP_{|Q|}}{TP_1 + FN_1 + ... + TP_{|Q|} + FN_{|Q|}}$$

$$TP_i \in [0, k], \quad FN_i = k - TP_i, \quad i \in \{1, ..., |Q|\}$$

$$r_{kNN,\ |Q|\geq 1} = \frac{TP'}{|Q| * k} \qquad TP' = TP_1 + ... + TP_{|Q|}$$

4. Discussion

- The expected recall depends on δ and δ is used in building the LSH Forest. PUFFINN predicts the stage (trie and signature length) at which the expected recall would be achieved.
- Kashif index building does not depend on k nor on TP.