Solving k-Closest Pairs in High-Dimensional Data using Locality-Sensitive Hashing

Martin Aumüller and Matteo Ceccarello

IT University of Copenhagen, University of Padova

maau@itu.dk

Problem Formulation

Input: Let (X, d) be a metric space. Let $S \subseteq X$ be a set of n points, and let $k \geq 1$ be an integer.

Task: Find k closest, distinct pairs $(r, s) \in S^2, r \neq s$.

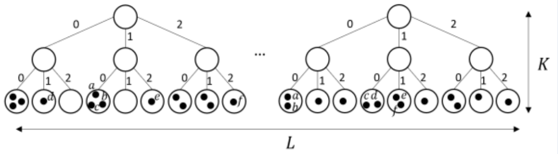
Naïve Approach: Compute all pairwise distances.

Running time: $O(n^2)$

Goal: *Subquadratic* running time with *probabilistic* guarantees.

Technique

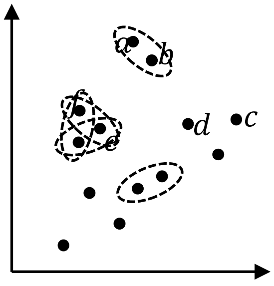
Preprocessing: Build L LSH tries each of depth K, initialize empty PQ to keep track of k points pairs and their distance.



Traversal strategy:

1. Collect all *colliding pairs* in *all leaves of all tries*, keep track of closest points.
2. Check termination: If current k-closest pair is at distance D, did we check enough repetitions to ensure result quality? If yes, return pairs.
3. Otherwise: Traverse trie one level up.

Example



13 points in $\mathbb{R}^2$. 5-closest pairs are marked in ellipses.

Results

Theory

1. Adaptivity: Knowing all pairwise distances, there exists a best trie level to query. If OPT is the expected cost on that level, our algorithm carries out work O(OPT).
2. Expected subquadratic running time: $O\left(n^{2\rho} k^{1-\rho} \log \frac{n}{\delta}\right), \rho \leq 1$ depends on *contrast* in distance distribution.

Practice

dataset	n	dimensions	RC @ 100	RC @ 10 000
DeepImage	10 000 000	96	7 615.56	2 343.25
Glove	1 193 514	200	38.04	5.15
DBLP	2 773 660	4 405 478	22.52	7.83
Orkut	2 732 271	8 730 857	20.97	2.99

Table 1: Datasets used in the experimental evaluation. The last two columns report the relative contrast at 100 pairs and 10 000 pairs [17].

Table 2: Running times. Missing values are for runs that timed out after 8 hours. The last column reports the time for the index construction (not applicable to XiaoEtAl), which is also included in the total time reported in the other columns

		Total time (s) for different k					indexing (s)
dataset	algorithm	1	10	100	1 000	10 000	
Glove	faiss-HNSW	68.1	132.8	551.7	-	-	63.8
	LSBTree	18.2	136.7	2028.4	2127.4	959.3	3.1
	PUFFINN	5.0	5.0	5.0	5.1	6.3	4.7
DeepImage	faiss-HNSW	299.7	533.8	2632.9	-	-	255.4
	LSBTree	112.0	93.4	114.6	176.2	368.6	13.6
	PUFFINN	37.2	37.5	37.1	37.4	37.4	18.9
DBLP	XiaoEtAl	9.3	14.0	9.8	12.1	58.3	0.0
	PUFFINN	4.9	4.9	4.9	4.9	5.0	4.2
Orkut	XiaoEtAl	118.0	122.0	142.3	1170.3	-	0.0
	PUFFINN	24.7	24.8	24.7	24.5	73.3	23.9

DIREC Digital Research Centre Denmark

IT UNIVERSITY OF COPENHAGEN