On Dynamicity of Metric Hull Trees

Josef Podaný

advised by doc. RNDr. Vlastislav Dohnal, Ph.D

Faculty of Informatics, Masaryk University

June 29, 2022

Outline

Motivation

- Interesting data to be stored, queried, and analyzed not often structured – images, documents, video, audio, webpages...
- Traditional data retrieval methods are lacking in this direction
- Challenge: **manage complex data efficiently** and evaluate similarity queries faster than by sequential scans
- $\Longrightarrow$ Utilize **metric spaces** to build **efficient indexing structures**

Metric Space

- Metric space $\mathcal{M} = (\mathcal{D}, d)$
  - Domain $\mathcal{D}$ consisting of multidimensional vectors
  - Distance function $d : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_0^+$ satisfying metric postulates

$a = (1.3, 12.8, \dots, 9.21)$

$b = (1.8, 16.3, \dots, 12.1)$

Figure: Image of a dog.[1]

$d(a,b) = 32.42$

Figure: Image of a wolf.[1]

[1] Images taken from https://www.pixabay.com/.

Metric Hull Representation

- A **Hull Representation**[2] of a group C is defined as $\mathcal{H}(C) = \{p_i \mid p_i \in C\}$ and any other object $o \in C$ is **covered** by hull. Each p_i corresponds to a boundary object of C referred to as *hull object*.

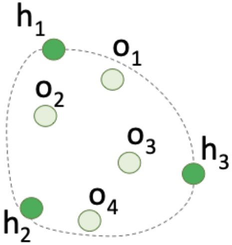
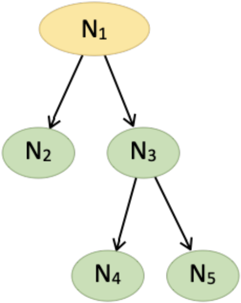


Figure: The hull representation $\mathcal{H} = \{h_1, h_2, h_3\}$ covering objects $h_1, h_2, h_3, o_1, o_2, o_3, o_4$

[2] Antol, Jánošová, and Dohnal [1]

Metric Hull Tree

- A **Metric Hull Tree**[3] is a hierarchical n-ary index structure
- Leaf nodes store vectors in buckets
- Internal nodes contain metric hulls
- Supports static bulk-loading of objects, insertion of new objects, approximate kNN search, exact-match query



[3] Jánošová, Procházka, and Dohnal [2]

Drawback: MH-Tree Not Balanced

- When bucket of node n overflows, it is split into two
- New leaf nodes for each bucket are created
- Leaves are then appended as children of node n, deepening the tree locally
- **Path length from the root to leaf linear** w.r.t. number of nodes

Growing MH-Trees at the Root

- Goal: Ensure the tree has depth of $\mathcal{O}(\log n)$, where n is the number of nodes
- Idea: **Follow insertion techniques used in B+ trees and M-trees**
  - Propagate splits upward and grow the tree at the root

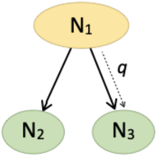


Figure: (1): $a = 2$. Store q in the bucket of N_3.

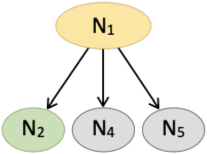


Figure: (2): $a = 2$. Split N_3 into N_4, N_5.

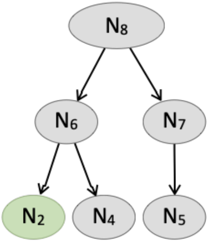


Figure: (3): $a = 2$. Repair N_1 by splitting.

Drawback: Bounding Regions of Metric Hulls Deteriorate

- During insertion, hulls are recomputed
- Recomputation may cause the hull to cease covering some objects
- **Recall of approximate kNN queries deteriorates** over time

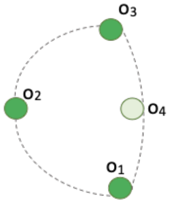


Figure: Hull $\mathcal{H} = \{o_1, o_2, o_3\}$ covering o_4

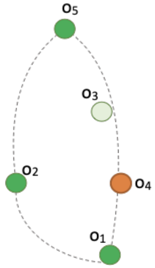


Figure: Hull $\mathcal{H} = \{o_1, o_2, o_5\}$ not covering o_4

Candidate Hull Objects

- Goal: Improve the approximate bounding regions of hulls
- Idea: Keep additional information about the bounding region

1. When routing the inserted object to the appropriate leaf, store the object alongside visited hulls, calling it a **candidate hull object**
   - Maximum number of candidates bounded by m for each internal node
   - Only the best ranking candidates are kept
2. When recursively repairing the tree, utilize such objects to build new hulls

Measured Results

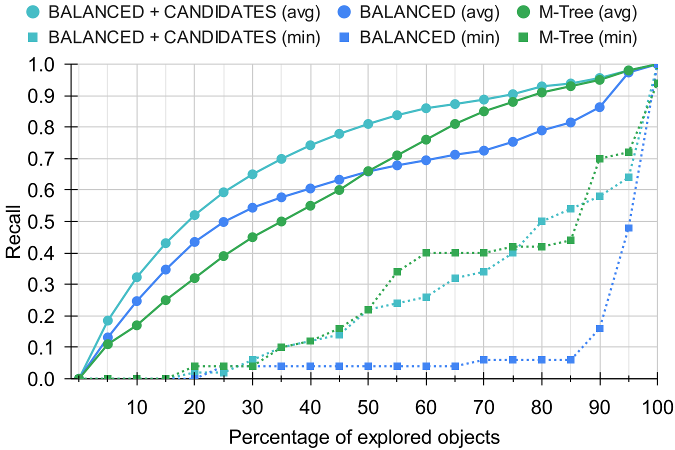


Figure: Average and minimum recall of 50NN queries in Deep 250 configuration. 5000 objects bulk-loaded, 5000 objects inserted.

Conclusion

- Introduced repairing procedure **making Metric Hull Trees self-balancing trees**
- Introduced the concept of **candidate hull objects** to **better approximate bounding regions** of hulls
- Showed that such modifications improve the average recall of 50NN queries by up to 20% when compared to M-Trees when a small number of objects is explored

Bibliography I

[1] Matej Antol, Miriama Jánošová, and Vlastislav Dohnal. “Metric hull as similarity-aware operator for representing unstructured data”. In: *Pattern Recognition Letters* 149 (2021), pp. 91–98. ISSN: 0167-8655.

[2] Miriama Jánošová, David Procházka, and Vlastislav Dohnal. “Organizing Similarity Spaces using Metric Hulls”. eng. In: *14th International Conference on Similarity Search and Applications (SISAP 2021)*. Springer, 2021, pp. 1–14.

Question 1: Time Complexity of Insertion

- Time complexity of original insertion after bulk-loading:

$$\mathcal{O}(n \cdot T_R(o) \cdot a + T_S(a) + 2T_H(a \cdot o)) = \boldsymbol{\mathcal{O}(n)}$$

- Time complexity of balanced insertion:

$$\mathcal{O}(\log n \cdot T_R(o) \cdot a + \log n \cdot (T_S(a) + 2T_H(a \cdot o))) = \boldsymbol{\mathcal{O}(\log n)}$$

- Time complexity of balanced insertion with candidates:

$$\mathcal{O}(\log n \cdot T_R(o) \cdot a \cdot (1 + m) + \log n \cdot (T_S(a) + 2T_H(a \cdot o))) = \boldsymbol{\mathcal{O}(\log n)}$$

- n: Number of objects
- a: Tree arity
- m: Candidate limit
- o: Hull objects per hull

- T_R: Time complexity of a ranking function
- T_S: Time complexity of a splitting strategy
- T_H: Time complexity of computing new hull from a set of objects

Question 1: Time Complexity of Insertion

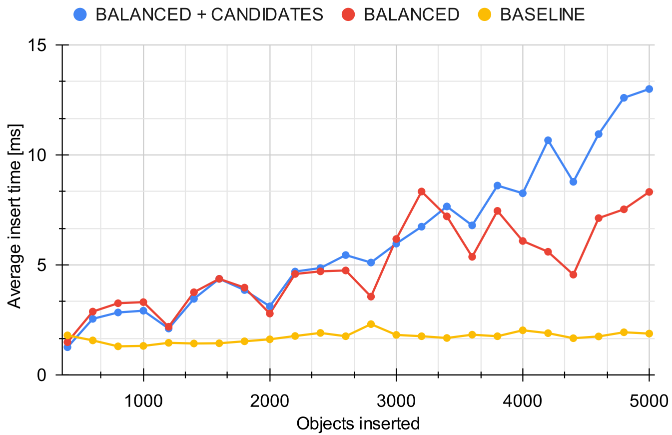


Figure: Total time to insert 5000 objects to MH-Trees in Deep 250 configuration. Hardware: 2019 MacBook Pro, 2.6 GHz 6-Core Intel Core i7, 32 GB 2667 MHz DDR4.

Question 1: Space Complexity

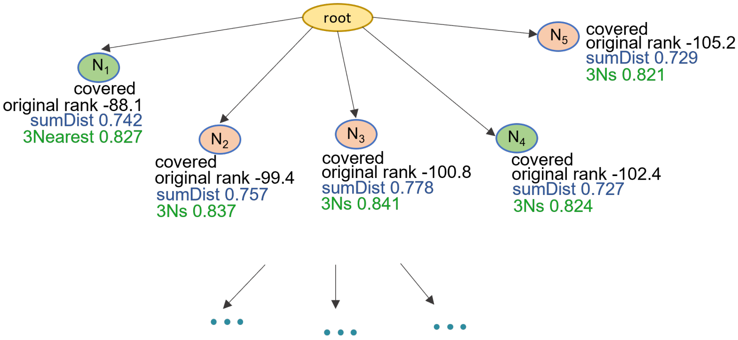
- Space complexity of original MH-Tree: $\mathcal{O}(n)$
- Space complexity of MH-Tree with candidates: $\mathcal{O}(m \cdot n) = \mathcal{O}(n)$

Question 2: M-Tree & MH-Tree Recommendations

- Exact-match queries are important $\Longrightarrow$ M-Tree
- Majority of objects not available before building the index $\Longrightarrow$ M-Tree
- Majority of objects available before building the index:
  - Recall of kNN queries is important $\Longrightarrow$ balanced MH-Tree with candidates
  - Insertion time and exact-match queries are important $\Longrightarrow$ balanced MH-Tree

Ranking Functions

- Formally, *rank* : $\mathcal{X} \times \mathcal{H} \times \mathbb{N} \rightarrow \mathbb{R}$
- Defines relevance of an object to a given hull



MUNI
FACULTY
OF INFORMATICS