The MM-tree

A Memory-Based Metric Tree Without Overlap Between Nodes

Ives R. V. Pola
Caetano Traina Jr.
Agma J. M. Traina
Outline

- Introduction
- Background
- Motivation
- The MM-tree
- Experiments and Results
- Conclusions
Introduction

Similarity

■ Are they similar?
  ▶ Based on what?
Introduction

Similarity measure

- We can define a similarity function
  - Compare pairs of elements
  - Based on the elements attributes.
Background

Metric Space

- A metric space is defined by $M < S, d >$

Let $s_1, s_2, s_3 \in S$, then $d: S \times S \rightarrow \mathbb{R}^+$ must hold:

- **Identity**
  - $d(s_1, s_1) = 0$

- **Symmetry**
  - $d(s_1, s_2) = d(s_2, s_1)$

- **Non negativity**
  - $0 < d(s_1, s_2) < \infty$, to $s_1 \neq s_2$

- **Triangular inequality**
  - $d(s_1, s_2) \leq d(s_1, s_3) + d(s_3, s_2)$
Background

Similarity queries

- Most common:
  - Range query
  - K-nearest neighbor query
Background

Metric Access Methods

- Index data in a Metric Space
  - Distance-based trees

- Classification
  - Disk-based trees
    - Slim-tree, M-tree, MVP-tree, OMNI-family, DF-tree, DBM-tree
  - Main memory-based trees
    - GH-tree, VP-tree, GNAT, MM-tree

- Pruning of subtrees
  - Exploring the triangular inequality property.
Background

Memory-based vs Disk-based Metric Trees

- Advantages of Memory-based trees
  - The partition of space is flexible
    - Not fixed number of elements per node
  - Do not perform disk I/O
    - Fast to build
    - Fast to answer queries

- Disadvantages
  - They are not persistent
  - There must be enough memory for data
Motivation

MM-tree

- A height-balanced tree
  - Reduces nodes retrieval on disk-based trees
    - Less disk accesses (they are computational expensive)

- But a main-memory tree
  - Do not perform disk access
  - We can choose to build a tree not fully balanced
    - In order to form disjoint regions
The MM-tree

Two levels example
The MM-tree

Building the tree
The MM-tree

Building the tree

Insert: i, j, k
The MM-tree

Building the tree
The MM-tree

Building the tree
The MM-tree

Memory metric tree

- It holds 2 elements per node
  - Divides the space into 4 disjoint regions
  - Only 2 distances per node are calculated

\[ \text{node}\left[s_1, s_2, d(s_1, s_2), Ptr_1, Ptr_2, Ptr_3, Ptr_4\right] \]
The MM-tree

Inserting elements - choosing subtrees

- **MM-tree is Dynamic**
  - **On Insertion**
    - Which region the new element will belong to?

<table>
<thead>
<tr>
<th>$d(s_i, s_1)$</th>
<th>$\theta$</th>
<th>$r$</th>
<th>$d(s_i, s_2)$</th>
<th>$\theta$</th>
<th>$r$</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt;$</td>
<td>$&lt;$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>I</td>
</tr>
<tr>
<td>$&lt;$</td>
<td>$\geq$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>II</td>
</tr>
<tr>
<td>$\geq$</td>
<td>$&lt;$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>III</td>
</tr>
<tr>
<td>$\geq$</td>
<td>$\geq$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IV</td>
</tr>
</tbody>
</table>
The MM-tree

Balancing control on leaf nodes

Extra level not needed

Empty node

Pivots changed
The MM-tree

Range Queries

- Range Query (Sq, rq)
  - At each node detect which subtree to visit

Visiting conditions:

<table>
<thead>
<tr>
<th>Region</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region I</td>
<td>((d(s_q, s_2) &lt; r_q + r) \land (d(s_q, s_1) &lt; r_q + r))</td>
</tr>
<tr>
<td>Region II</td>
<td>((d(s_q, s_2) + r_q \geq r) \land (d(s_q, s_1) &lt; r_q + r))</td>
</tr>
<tr>
<td>Region III</td>
<td>((d(s_q, s_2) &lt; r_q + r) \land (d(s_q, s_1) + r_q \geq r))</td>
</tr>
<tr>
<td>Region IV</td>
<td>((d(s_q, s_2) + r_q \geq r) \land (d(s_q, s_1) + r_q \geq r))</td>
</tr>
</tbody>
</table>
The MM-tree

Guided k-NN Query

- The sequence of subtrees visited depends on where the query center is.

- Visit order:
  
  \[ \text{Visit order:} \quad \text{II} \rightarrow \text{I} \rightarrow (\text{III, IV}) \]
The MM-tree

Guided k-NN Query

- The sequence of subtrees visited depends on where the query center is.

- Visit order:

  II → I → (III, IV)
The MM-tree

Guided k-NN Query

- The sequence of subtrees visited depends on where the query center is.

- Visit order:

  II → I → (III, IV)
The MM-tree

Guided k-NN Query

- Generalizing, there are different sequences for each region

<table>
<thead>
<tr>
<th>region $s_q$ lies</th>
<th>condition C</th>
<th>visit order when C is true</th>
<th>visit order when C is false</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>$d_1 \leq d_2$</td>
<td>$I \rightarrow II \rightarrow (III, IV)$</td>
<td>$I \rightarrow III \rightarrow (II, IV)$</td>
</tr>
<tr>
<td>II</td>
<td>$d_2 - d \leq d - d_1$</td>
<td>$II \rightarrow I \rightarrow IV \rightarrow III$</td>
<td>$II \rightarrow IV \rightarrow IV \rightarrow II$</td>
</tr>
<tr>
<td>III</td>
<td>$d_1 - d \leq d - d_2$</td>
<td>$III \rightarrow I \rightarrow IV \rightarrow II$</td>
<td>$III \rightarrow IV \rightarrow I \rightarrow II$</td>
</tr>
<tr>
<td>IV</td>
<td>$d_1 \leq d_2$</td>
<td>$IV \rightarrow II \rightarrow I \rightarrow III$</td>
<td>$IV \rightarrow III \rightarrow I \rightarrow II$</td>
</tr>
</tbody>
</table>
Experiments

Construction Statistics

- The MM-tree was compared with
  - Slim-tree
  - VP-tree

<table>
<thead>
<tr>
<th></th>
<th>Points</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Cities</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Color Histograms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dist</td>
<td>Time (ms)</td>
<td>Dist</td>
<td>Time (ms)</td>
<td>Dist</td>
<td>Time (ms)</td>
<td>Dist</td>
<td>Time (ms)</td>
<td>Dist</td>
<td>Time (ms)</td>
<td>Dist</td>
<td>Time (ms)</td>
<td></td>
</tr>
<tr>
<td>MAM MM-tree</td>
<td>161143</td>
<td>190</td>
<td>89783</td>
<td>126</td>
<td>167705</td>
<td>737</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slim-tree</td>
<td>633374</td>
<td>297</td>
<td>451830</td>
<td>156</td>
<td>665453</td>
<td>1234</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP-tree</td>
<td>2381532</td>
<td>1625</td>
<td>1203897</td>
<td>640</td>
<td>2346300</td>
<td>6188</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiments

Construction Statistics

Number of Distances

Not acceptable!
Experiments

MM-tree structure statistics

Height of MM-tree

Two levels in average
Experiments

Color Histograms dataset

k-NN query

26%
Experiments

Color Histograms dataset

k-NN query 24%
Experiments

Color Histograms dataset

Range query

![Graph showing Avg # of distance calculations vs radius for Slim-Tree MST, VP-Tree, and MM-Tree. The graph indicates a 62% improvement in performance.]
Experiments

Color Histograms dataset

Range query

The figure shows a graph plotting the total time (ms) against the radius for different data structures: Slim-Tree MST, VP-Tree, and MM-Tree. The graph illustrates the performance of these structures under range query operations. The data indicates that the Slim-Tree MST has the lowest total time compared to VP-Tree and MM-Tree, suggesting it performs better under the specified conditions.

60%
Conclusions

The MM-tree

- Useful for emerging applications that require the DBMS to provide fast ways to build indexes on data that fit in main memory

- The MM-tree is fast to build and provide fast similarity queries, partitioning the metric space into disjoint regions.

- Compared to the Slim-tree
  - KNN = 26% less distance calculations, 24% faster
  - RQ = 62% less distance calculations, 60% faster
The MM-tree

A Memory-Based Metric Tree Without Overlap Between Nodes

Thank You.

Open to questions.