Module 5: Disk-based Index Structures

Lecture 27: M-trees

Prev topic | Next topic | Prev page | Next page

The Lecture Contains:

- M-tree
  - Structure
  - Range queries
  - kNN queries
  - Algorithms
- Non-metric distances

M-tree

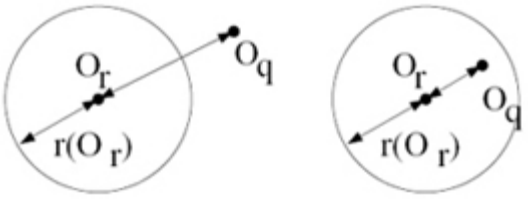
- Feature vectors are not available, or
- L_2 or L_1 is not the distance measure
- Only, distances among objects are available

M-tree

- Feature vectors are not available, or
- L_2 or L_1 is not the distance measure
- Only, distances among objects are available
- Distance function is a metric
- Balanced
- Dynamic
- Disk page-based design
- Distance computation may be expensive

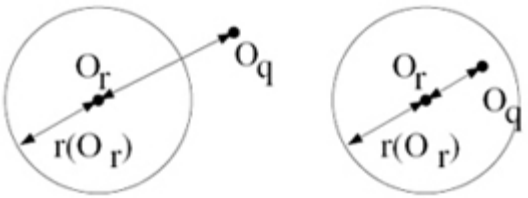
Structure

- Each internal node entry contains
  - Routing object O_r
  - Covering tree $T(O_r)$: subtree of O_r
  - Covering radius $r(O_r)$: all objects in T is within distance $r(O_r)$ from O_r
  - Distance of O_r to its parent O_p: $d(O_r, O_p)$
- Leaf nodes
  - Covering radius $r(O_r) = 0$
  - $T(O_r)$ point to actual data object



Structure

- Each internal node entry contains
  - Routing object O_r
  - Covering tree $T(O_r)$: subtree of O_r
  - Covering radius $r(O_r)$: all objects in T is within distance $r(O_r)$ from O_r
  - Distance of O_r to its parent O_p: $d(O_r, O_p)$
- Leaf nodes
  - Covering radius $r(O_r) = 0$
  - $T(O_r)$ point to actual data object
- Minimum distance from object O_q to any object in $T(O_r)$ is $\max\{0, d(O_r, O_q) - r(O_r)\}$
- Maximum distance from object O_q to any object in $T(O_r)$ is $\{d(O_r, O_q) + r(O_r)\}$



Range queries

- Query Q and range $r(Q)$
- Prune a subtree $T(O_r)$ if
  - $|d(O_p, Q) - d(O_r, O_p)| > r(Q) + r(O_r)$
  - $d(O_r, Q) > r(Q) + r(O_r)$
- Otherwise, recurse

Range queries

- Query Q and range $r(Q)$
- Prune a subtree $T(O_r)$ if
  - $|d(O_p, Q) - d(O_r, O_p)| > r(Q) + r(O_r)$
  - $d(O_r, Q) > r(Q) + r(O_r)$
- Otherwise, recurse
- First condition implies second condition
  - $|d(O_p, Q) - d(O_r, O_p)|$ is a lower bound of $d(O_r, Q)$
- Minimum distance of any object O_c in $T(O_r)$ to Q is $d(O_r, Q) - d(O_c, O_r) \geq d(O_r, Q) - r(O_r)$
- Second condition is tighter but requires additional calculation of $d(O_r, Q)$

kNN queries

- Query Q and a positive integer k
- For $T(O_r)$, compute minimum distance to any object in it
  - $d_{min}(T(O_r)) = \max\{0, d(O_r, Q) - r(O_r)\}$
- Maintain a heap with distance to k^{th} object as d_k
- Prune $T(O_r)$ if
  - $d_{min}(T(O_r)) > d_k$
- Initially, $d_k = \infty$
- Traverse children in sorted order of d_{min}

kNN queries

- Query Q and a positive integer k
- For $T(O_r)$, compute minimum distance to any object in it
  - $d_{min}(T(O_r)) = \max\{0, d(O_r, Q) - r(O_r)\}$
- Maintain a heap with distance to k^{th} object as d_k
- Prune $T(O_r)$ if
  - $d_{min}(T(O_r)) > d_k$
- Initially, $d_k = \infty$
- Traverse children in sorted order of d_{min}
- Also compute maximum distance to any object in $T(O_r)$
  - $d_{max}(T(O_r)) = \{d(O_r, Q) + r(O_r)\}$
- Prune $T(O_r)$ if
  - $d_{min}(T(O_r)) > d_{max}(T(O_s))$ and $|T(O_s)| \geq k$
  - At least k objects in $T(O_s)$ have lower distances

Algorithms

- Insertion: choose subtree with least radius enlargement
- Splitting: metric to minimize
  - Sum of radii
  - Maximum radius
- Splitting: choosing routing objects
  - Old routing object and farthest
  - Sampling
  - Random
- Splitting: distributing objects
  - To nearest routing object
  - Routing object grabs nearest neighbors
- Deletion: mark as deleted

Non-metric distances

Non-metric distances

- Sequential scan

Non-metric distances

- Sequential scan
- Approximations

Non-metric distances

- Sequential scan
- Approximations
- Embedding into vector spaces

Non-metric distances

- Sequential scan
- Approximations
- Embedding into vector spaces
- Make the distance function metric

Non-metric distances

- Sequential scan
- Approximations
- Embedding into vector spaces
- Make the distance function metric
  - Constant shift embedding
    - Assumes only triangular inequality is violated
    - Identify triplets x, y, z where $d(x, y) + d(y, z) < d(z, x)$
    - Add a constant c to each distance such that $(d(x, y) + c) + (d(y, z) + c)$ becomes $\geq (d(z, x) + c)$
    - Pruning power suffers for large c