Multi-object Similarity Query Evaluation

Michal Batko

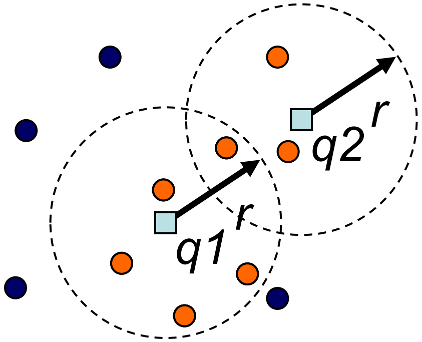
FACULTAS ARTIS INFORMATICÆ • UNIVERSITAS MASARYKIANA •

Motivation

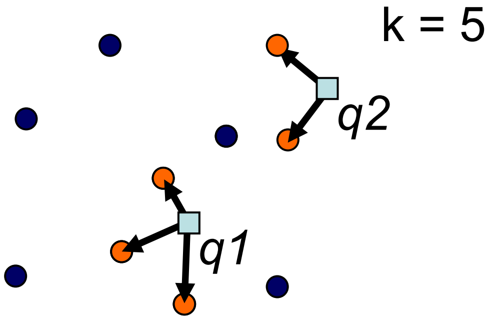
- Applications where we have
  - multiple samples of the search object
  - best match to any of the given objects
- Examples
  - Multiple camera shots
    - Tourist takes multiple pictures of a building
  - Face recognition
  - 3D scanned input
  - …

Multi-object Query Definition

- Regular similarity query-by-example
  - one query object + parameters
- Multi-object query Q({q_1, q_2, …, q_n}, *params*)
  - several query objects + parameters
  - goal is to find the results that are best with respect to **any** of the query objects



Range query – R({q1,q2}, r)



kNN query – kNN({q1,q2}, k)

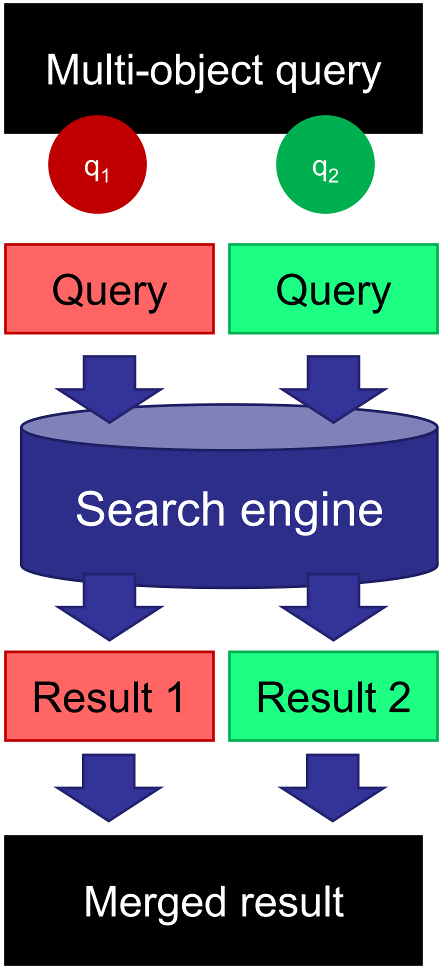
State of the Art

- Relational databases
  - Optimizations for searching multiple values in indexes
- Text search
  - Special case: multiple query documents
    - Combined vector of keywords (cannot just boost the frequencies)
- Multi-modal search
  - Specialization of the technique:
    - Only one domain, *minimum* aggregation function
- Vector space approach
  - Compute a new single-query object
- Tree structure evaluation extension
  - Multiple paths of a tree are traversed

Base-line Approach
(Multi-modal search)

- Evaluate a separate query for each query object
- Merge the results

- Pros
  - Use engine without modification
  - Can evaluate any query
- Cons
  - Same part of the index is likely visited multiple times
  - Parallelization drawback


Multi-object query

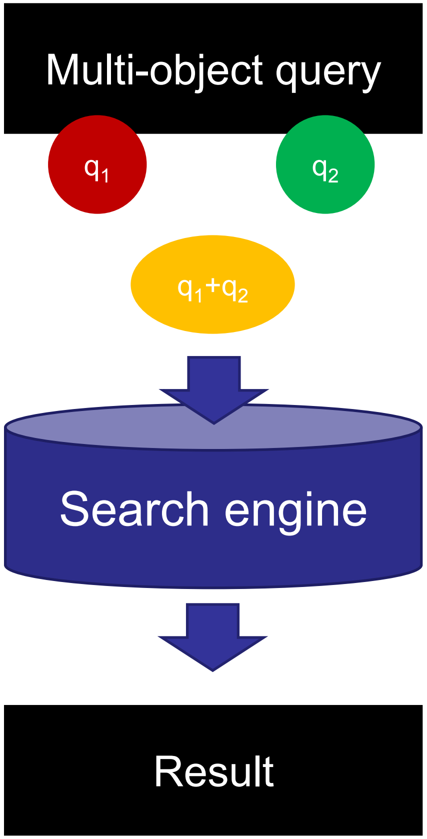
Search engine

Merged result

Vector space approach

- Compute a single query
  - Can be centroid of the queries or some specialized algorithm that utilizes other knowledge about the space
- Execute as normal single query

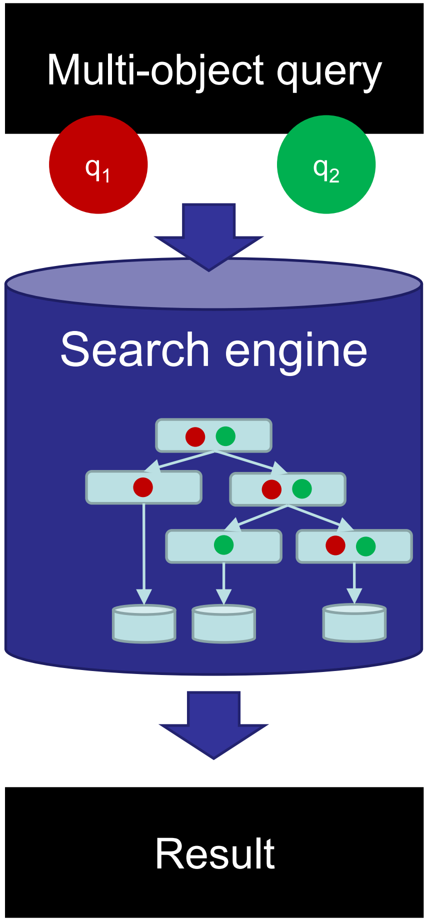
- Pros
  - Use engine without modification
- Cons
  - Need to specify the query merging algorithm
  - Approximation



Tree-based approach

- Modify tree traversal
  - In each branch, traverse if any of the query objects matches

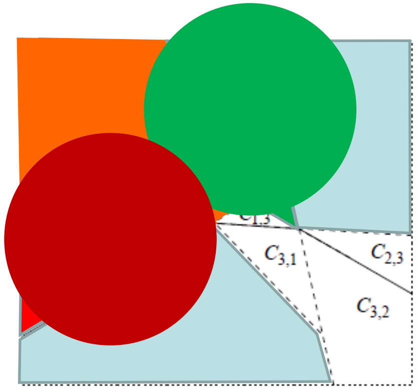
- Pros
  - Efficient, precise
  - Data are accessed once
- Cons
  - Specialized structure is needed
  - Usable for “prunable” queries
    - such as range query
    - modification for kNN exists
      - traverse to most likely leaf for each query
      - use the radius to backtrack



Hash-based approach

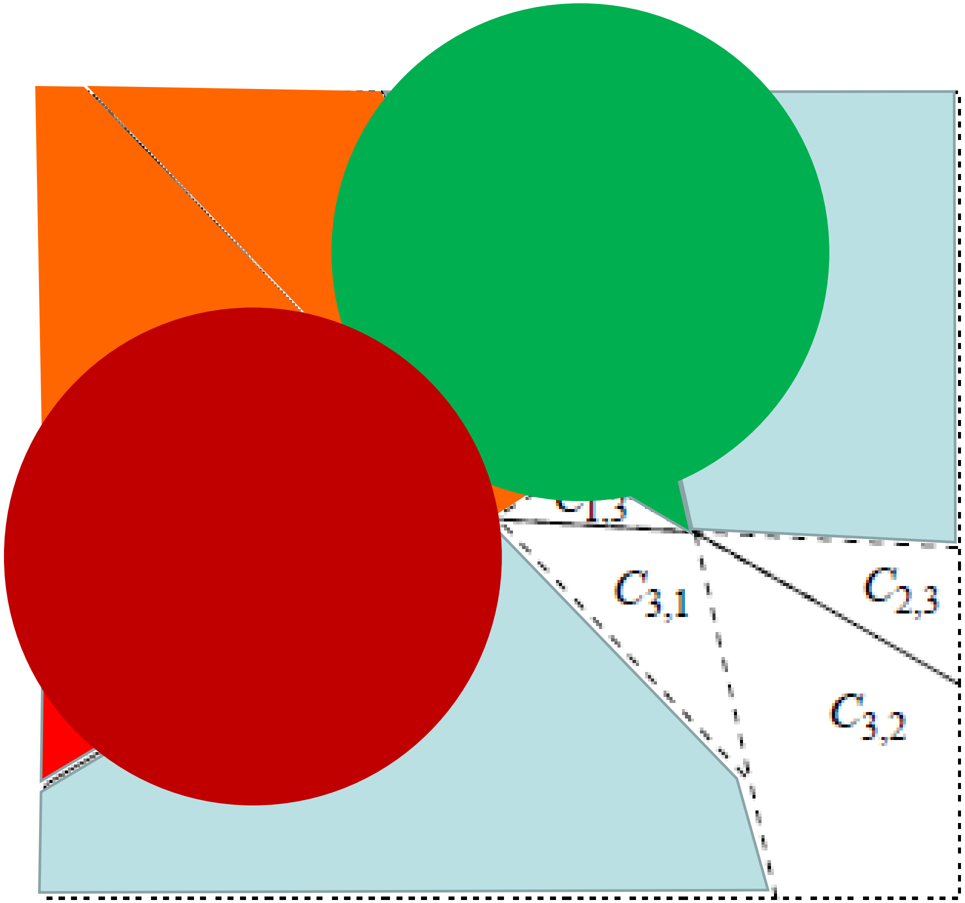
(M-Index)

- Inspired by the tree multi-object search
  - Compute the hash function for each query object
    - If a tree is used to compute the hash, use tree m-o-q
  - Expand the accessed buckets to the neighbors
    - Using the heuristic for computing the adjacent cell probability
  - Sort the buckets according to the minimum distance to the set of query objects
    - Could be incorporated into the heuristic?



Hash-based approach

(M-Index)



Conclusion

- Multi-object query
  - Potentially useful in various application
  - Not much studied in context of similarity search
- Algorithm for M-Index can be designed
  - Heuristic for finding the best-match buckets needs to be properly specified and evaluated
- Explore the properties of Voronoi space
  - Can the promising cells be identified directly?