Distributed Top-k Query Processing on Multi-dimensional Data with Keywords

Daichi Amagata, Takahiro Hara, Shojiro Nishio
Graduate School of Information Sciences and Technology
Osaka University
Japan

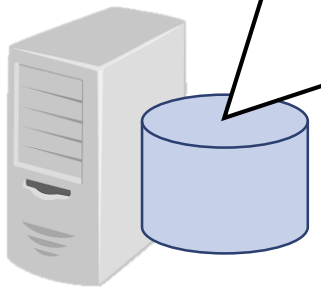
Outline

- Introduction
  - Motivation, Challenge, and Contribution
- Preliminary
  - Models, Problem definition, and Baseline algorithm
- Our algorithms
  - GeoEst, Prog-GeoEst, and GeoEst+
- Experimental results
  - Real and synthetic datasets
- Conclusion and future work

Introduction

- Preference query processing over multi-dimensional data
  - **Top-k queries** [1]
  - Skyline queries [2]
  - Top-k dominating queries [3]

Car data	Price	Fuel cost	⋯	Attribute d
o_1	0.1	0.5	⋯	-
o_2	0.2	0.3	⋯	-
⋮	⋮	⋮	⋮	⋮
o_n	0.8	0.9	⋯	-



Car data	Score
o_1	0.30
o_2	**0.25**
⋮	⋮
o_n	0.85

Top-k query form

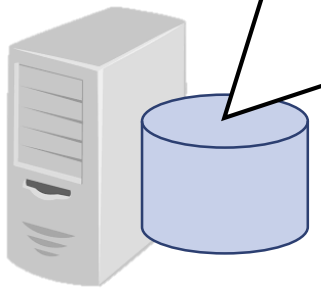
Scoring Function	$0.5 \times \text{Price} + 0.5 \times \text{Distance}$
k	1

[1] A survey of top-k query processing techniques in relational database systems, *ACM CSUR*, 2008.
[2] The skyline operator, In *ICDE*, 2001.
[3] Efficient processing of top-k dominating queries on multi-dimensional data, In *VLDB*, 2007.

Introduction (motivation)

- Real-life data: multi-dimensional data **with keywords**
  - "A set of numerical attribute values" + "A set of keywords"

Car data	Price	Fuel cost	**Keywords**
o_1	0.1	0.5	Hybrid, 4WD, Toyota
o_2	0.2	0.3	Diesel, BMW
⋮	⋮	⋮	⋮
o_n	0.8	0.9	Navigation, Mercedes



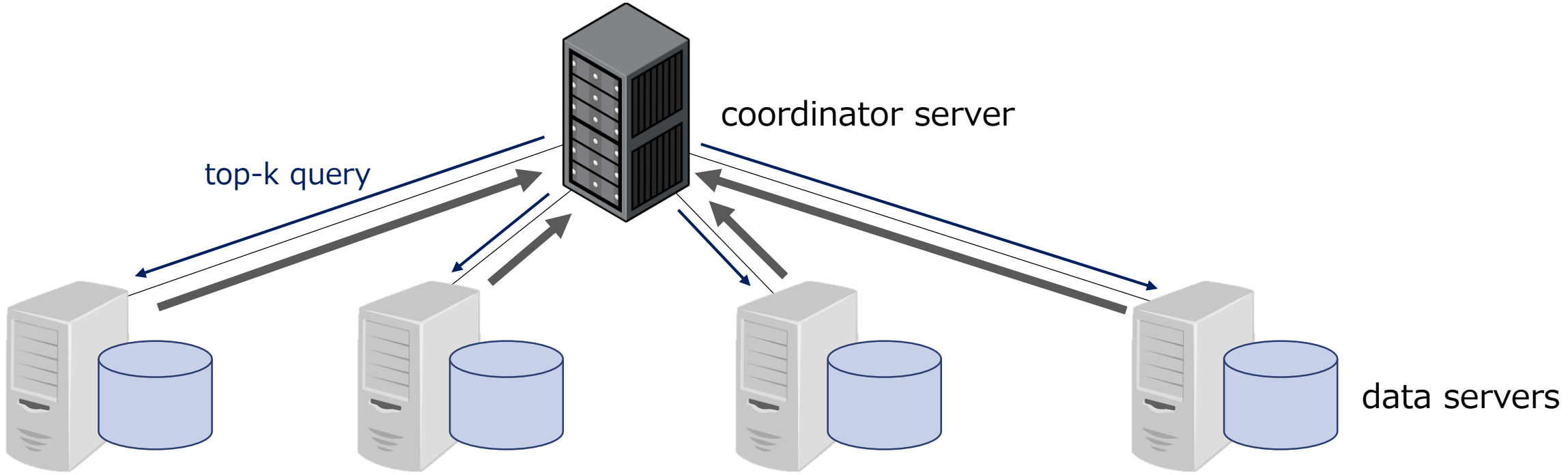
Car data	Score
o_1	**0.30**
o_2	-
⋮	⋮
o_n	-

Top-k query form

Scoring Function	$0.5 \times \text{Price} + 0.5 \times \text{Distance}$
k	1
keywords	Hybrid

Introduction (motivation)

- Distributed databases
  - A single coordinator server + m data servers
    - Ex.) cluster nodes and data centers
  - Data objects are retrieved through communication



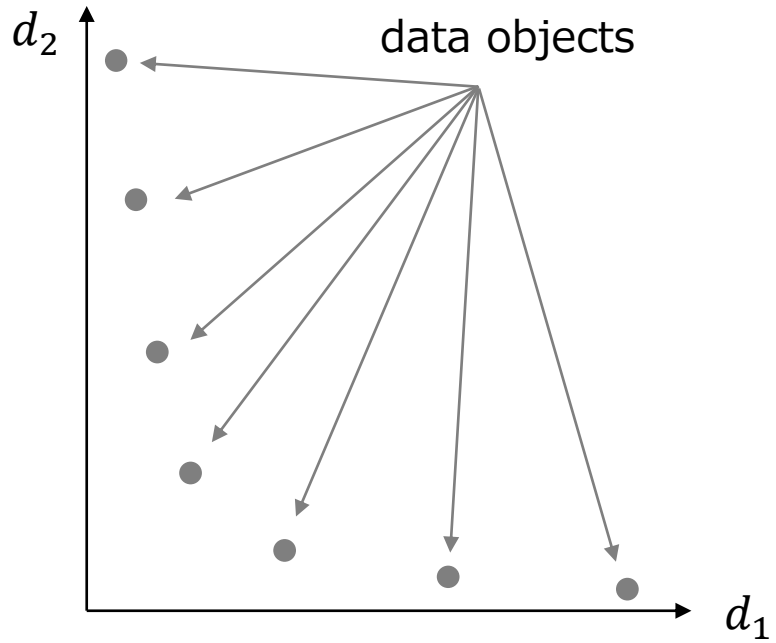
Top-k query processing over “distributed”, “multi-dimensional,” and “keyword” data

Introduction (challenge)

- Question:
  - Are any existing techniques for (distributed) top-k query processing good solution for our problem?
- Answer:
  - **NO!** They don't consider keywords.

Ex. 1) skyline on a 2-dimensional dataset
- Property: **Skyline contains the top-1 data objects for any monotone scoring functions** [4,5]

Ex. 2) - The 2-dimensional dataset holds 1,000 distinct keywords
- The combination number is $\binom{1,000}{c}$.
  - If $c = 3$, $\binom{1000}{3} = \mathbf{166,167,000}$!



[4] Efficient dual-resolution layer indexing for top-k queries, In *ICDE*, 2012.
[5] Pareto-based dominant graph: an efficient indexing structure to answer top-k queries, *IEEE TKDE*, 2011.

Contributions

- Define a Keyword Matching Top-k (KMTop-k) query
  - Retrieve k data objects holding all query keywords with the best score

- Propose novel solutions
  - **GeoEst**: Geometric-based threshold Estimation algorithm
  - **Prog-GeoEst**: Progressive GeoEst
  - **GeoEst+**: Extension of GeoEst

- Extensive experiments
  - Using both real and synthetic datasets

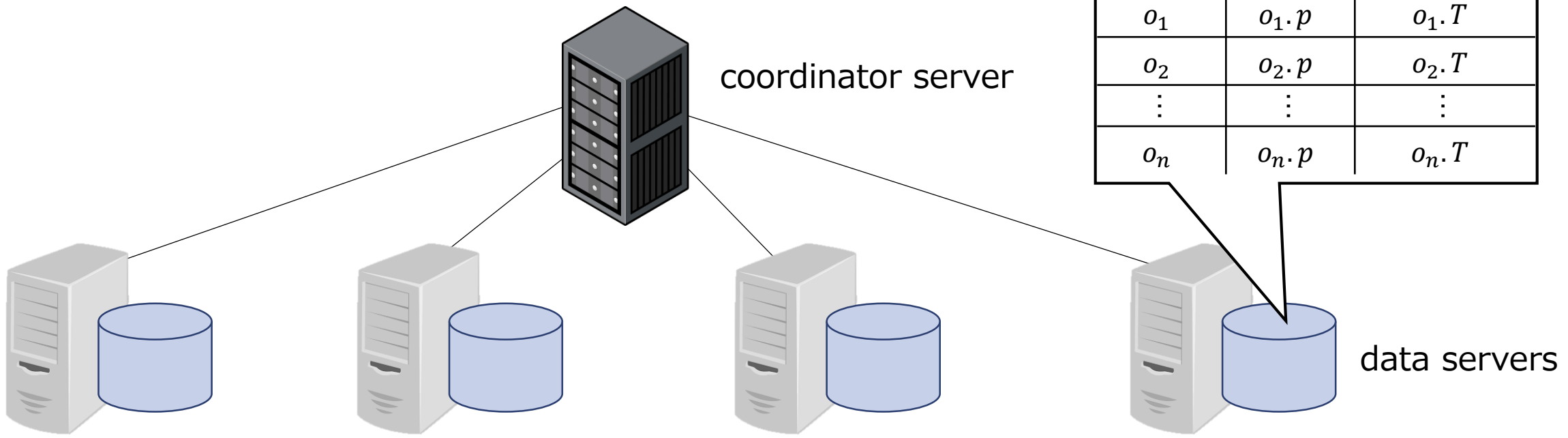
Outline

- Introduction
  - Motivation, Challenge, and Contribution
- Preliminary
  - Models, Problem definition, and Baseline algorithm
- Our algorithms
  - GeoEst, Prog-GeoEst, and GeoEst+
- Experimental results
  - Real and synthetic datasets
- Conclusion and future work

Preliminary (system and data models)

- System
  - A distributed database (shown below)
- Data object o_i
  - A pair of d-dimensional point $o_i.p$ and a set of keywords $o_i.T$
    - $o_i.p =< p[1], p[2], \ldots, p[d] >$
    - $o_i.T = \{t_x, \ldots, t_y\}$
  - Indexed by “on-memory” inverted file

Data	Point	Keywords
o_1	$o_1.p$	$o_1.T$
o_2	$o_2.p$	$o_2.T$
⋮	⋮	⋮
o_n	$o_n.p$	$o_n.T$



Preliminary (problem definition)

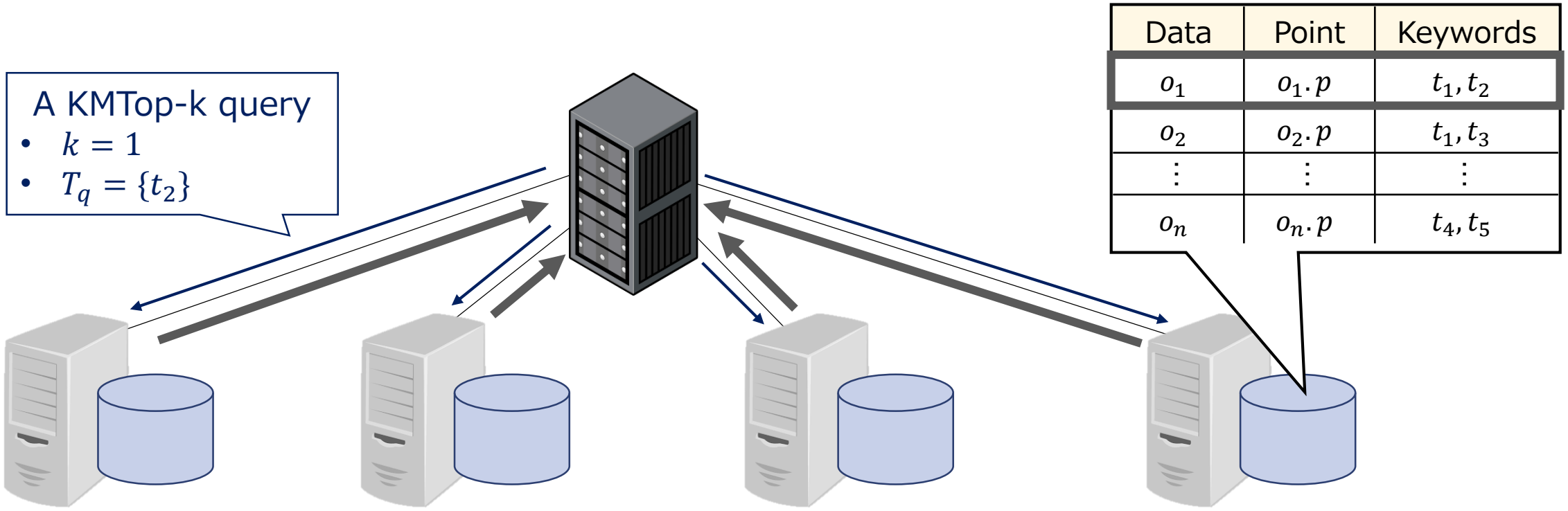
- Score of a data object o_i
  - $\boldsymbol{score(o_i) = f(w, o_i) = \sum_{j=1}^{d} w_j \times o_i.p[j]}$
    - Smaller values are preferable.
    - $w =< w_1, w_2, \ldots, w_d >$: a weighting vector ($w_j$ is the weighting factor for j-th dimension)
- KMTop-k query
  - Input
    - A set of local datasets $O = O_1 \cup O_2 \cup \cdots \cup O_m$ each of which is held by distributed server
    - k : result size
    - w: weighting vector
    - A set of query keywords T_q
  - Output (*TOPK*)
    - A set of k data objects holding **all** query keywords with the smallest score
- Objective
  - Minimizing ***communication cost*** and ***running time***

Preliminary (baseline algorithm)

- Coordinator server
  - Forward a KMTop-k query to *all* data servers
- Data servers
  - Compute the scores of data objects holding T_q
  - Forward their local *TOPK* (i.e., at most k data)

Drawback

- Redundant query forwarding
- Unnecessary data forwarding
  -> Not scalable to m



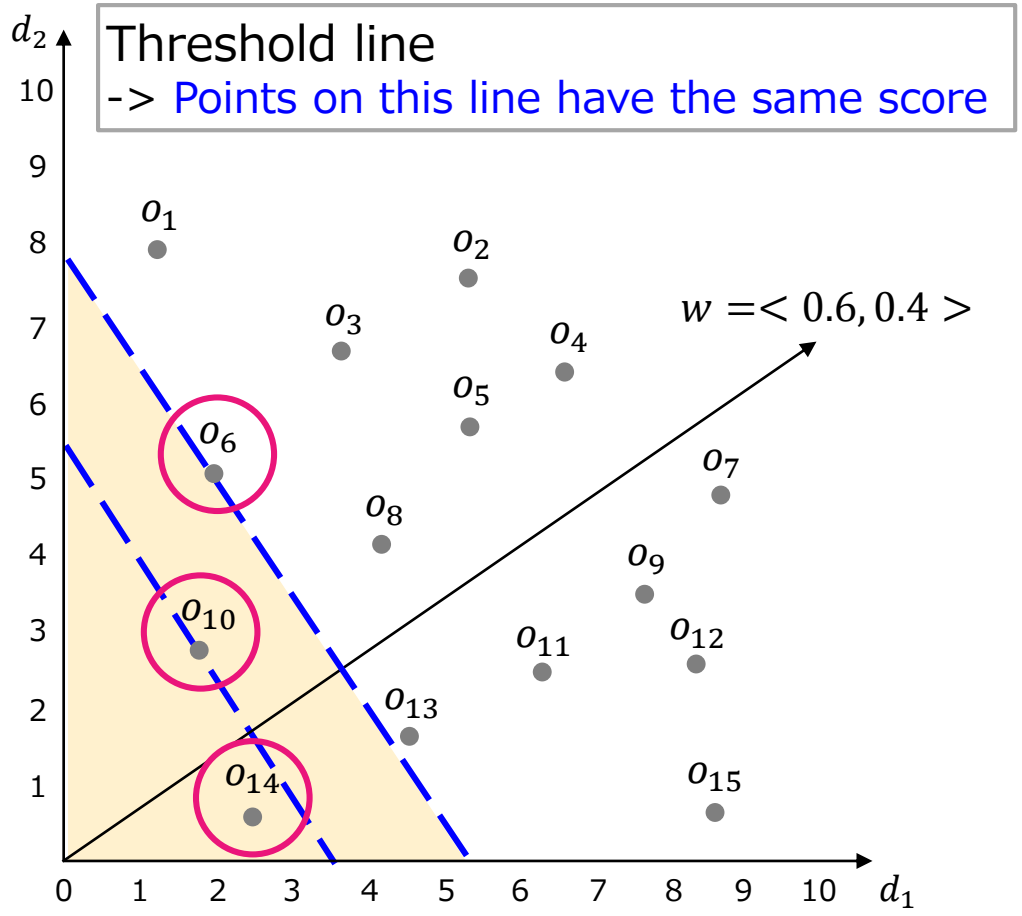
Theory

- *TOPK* exists within the half space that is perpendicular to w.
  - Ex.) a top-2 query with $w =< 0.6, 0.4 >$
  - Ex.) a KMTop-2 query ($w =< 0.6, 0.4 >$, and $T_q = \{t_1\}$)
    - $\exists t_1 \in o_6.T,\ o_{14}.T,\ \nexists t_1 \in o_{10}.T$

1. **Estimate the threshold line (plane)**

2. **Compute the threshold ρ (k-th smallest score)**

3. **Retrieve $\forall o$ where $score(o) \leq \rho$ and $T_q \subseteq o.T$**



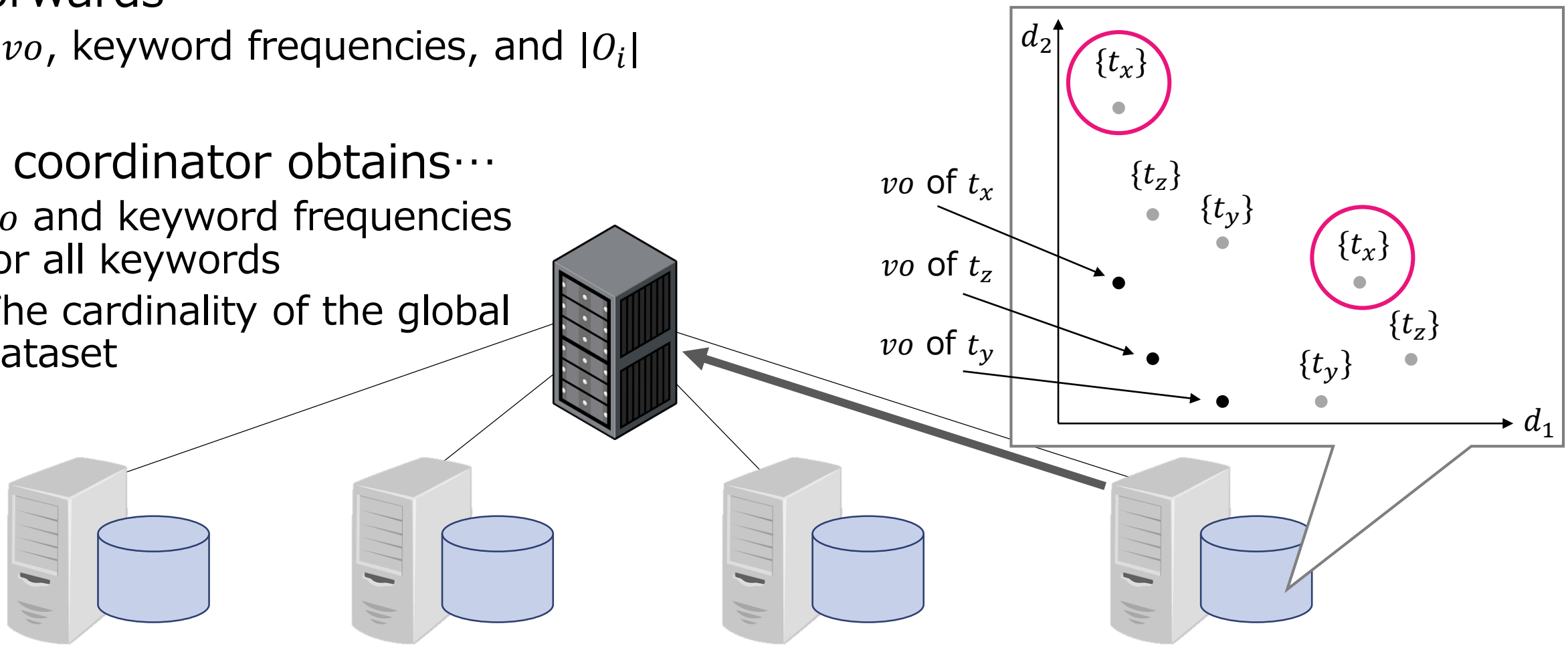
GeoEst

- Pre-processing
  - The coordinator collects information on data objects for the threshold estimation

- KMTop-k query processing
  - The coordinator estimates the threshold ρ
  - The coordinator forwards a query with ρ
  - Query receiving servers forward the corresponding data

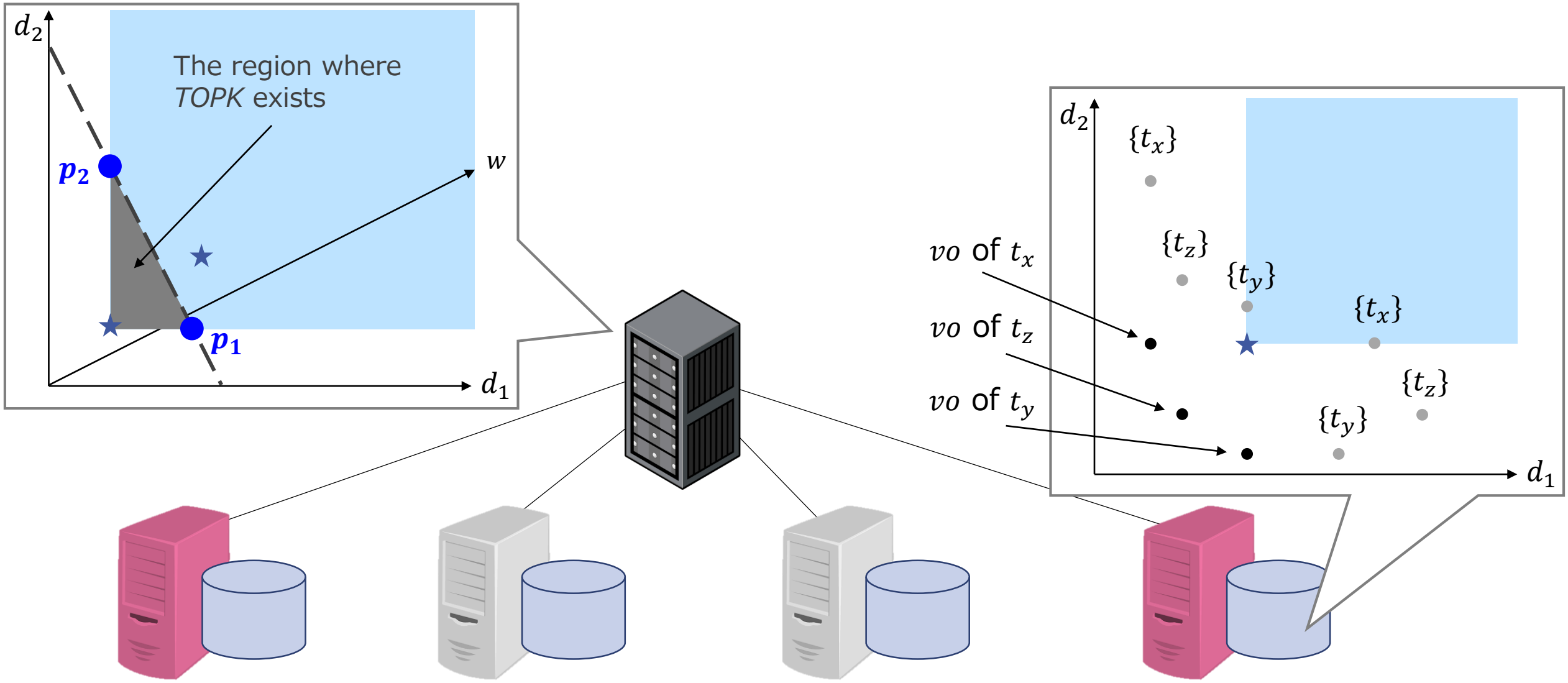
GeoEst (pre-processing)

- Each data server S_i generates a ***virtual object, vo, for each keyword***.
  - Ex.) vo for a keyword t_x: $\boldsymbol{vo.p[j] = \min(o_{i'}.p[j] \mid o_{i'} \in O_i, t_x \in o_{i'}.T)}$
- S_i forwards…
  - $\forall vo$, keyword frequencies, and $|O_i|$
- The coordinator obtains…
  - vo and keyword frequencies for all keywords
  - The cardinality of the global dataset



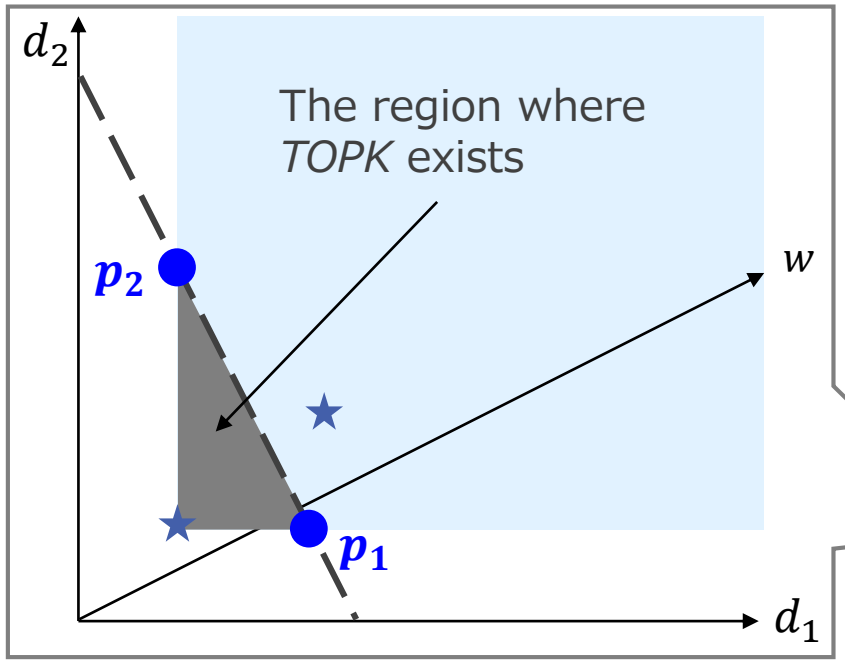
GeoEst (threshold estimation)

- The coordinator identifies the space & local database holding T_q.



GeoEst (threshold estimation & data retrieval)

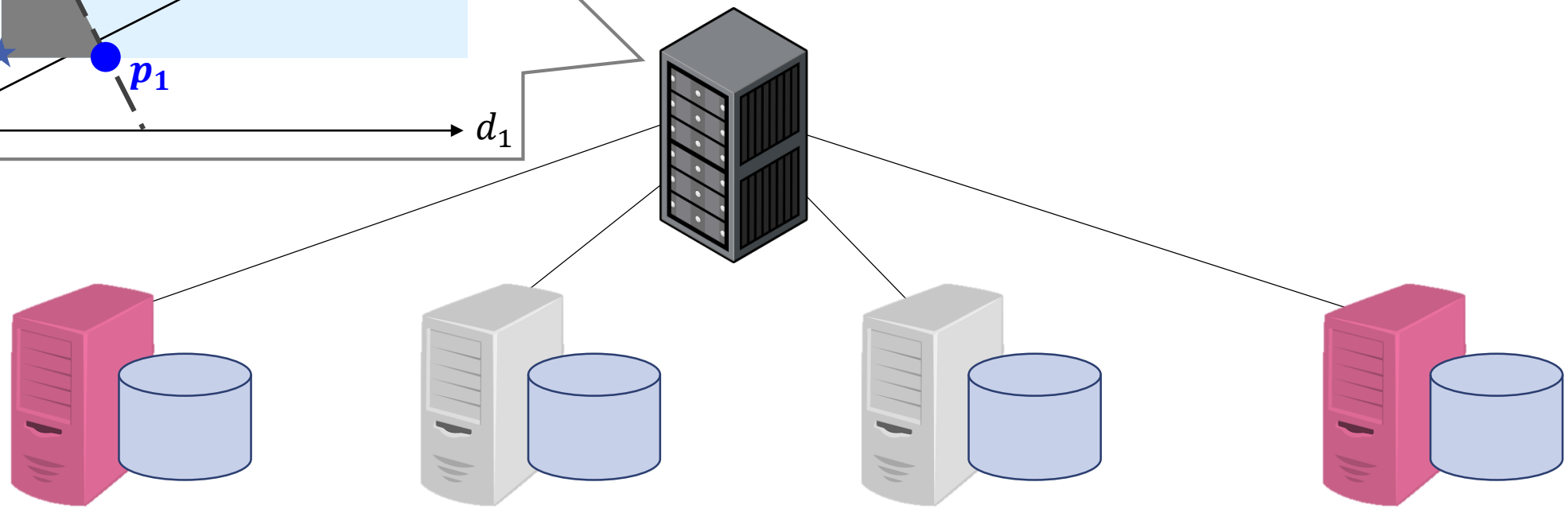
- The coordinator computes the threshold ρ.



Density-based computation

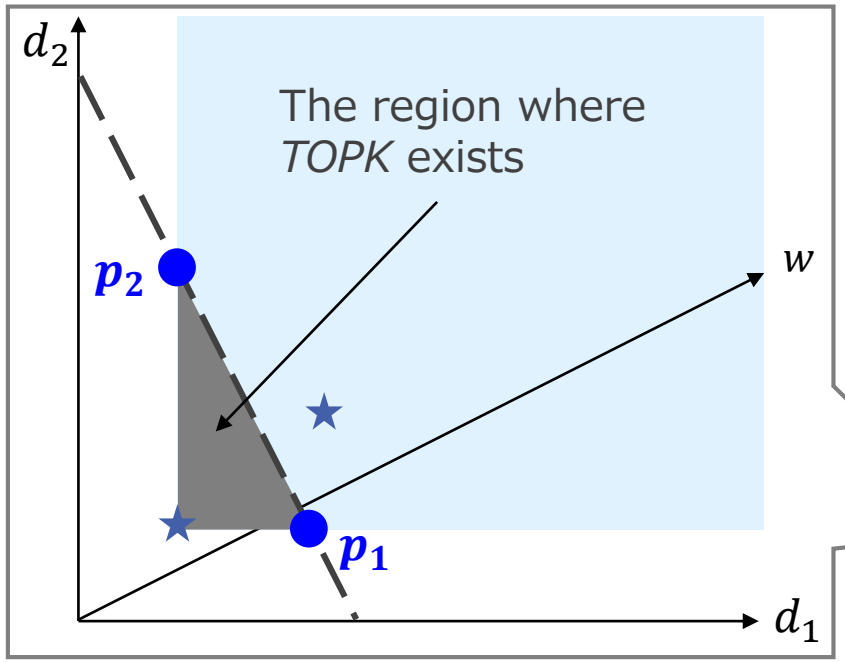
- $L^{TOPK} = k \times \frac{1}{t.freq} \times \frac{1}{|o.T|_{avg}} \times \frac{1}{\phi}$
  - $t.\text{freq} = \min(t_i.\text{freq} \mid t_i \in T_q)$
  - $\phi = \frac{\sum |o_i|}{v_{max}}$

- **The space largeness of the shaded region**
- Assumption of uniform data distribution



GeoEst (threshold estimation & data retrieval)

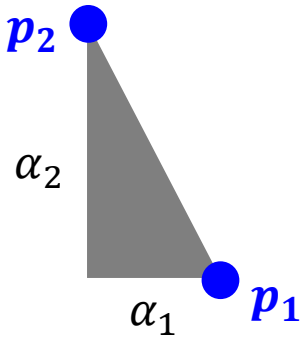
■ The coordinator computes the threshold ρ.

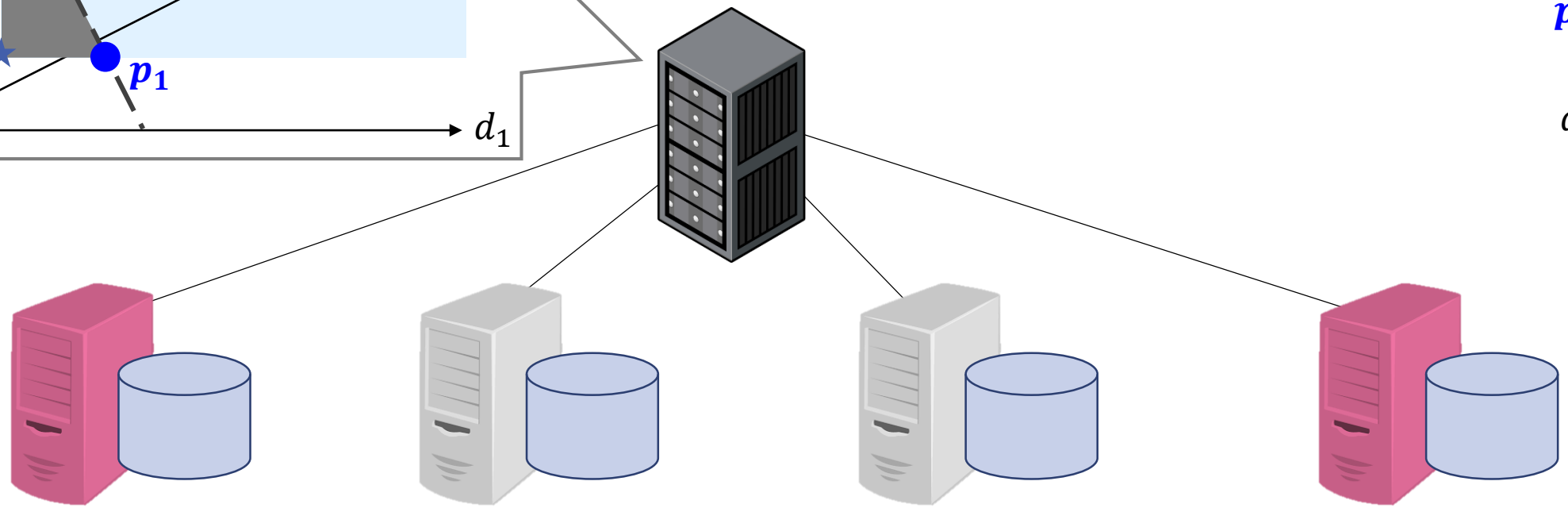


- **The space largeness of the shaded region**
- From the shape of the shaded region (triangular pyramid)

Geometry-based computation

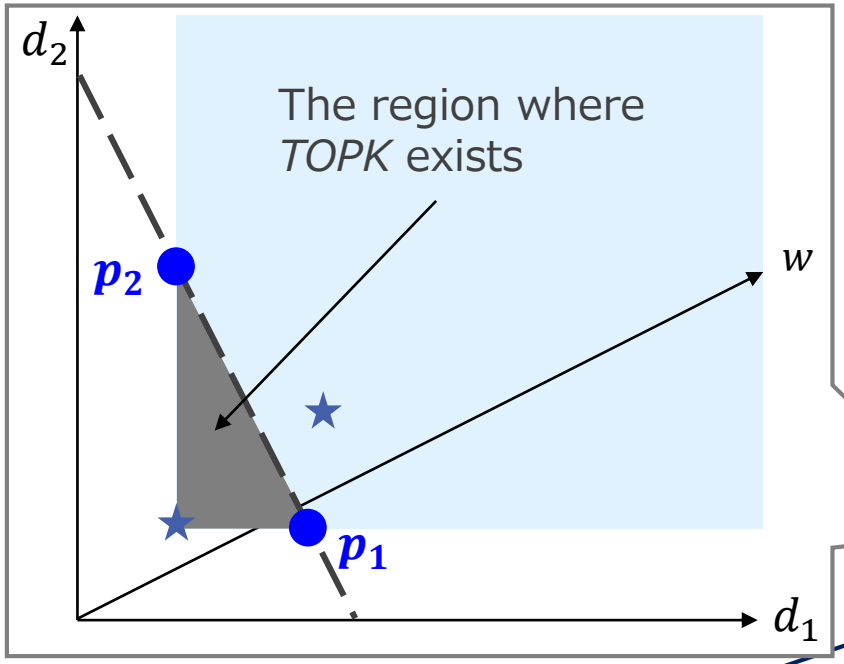
- $L^{TOPK} = \frac{\prod_d \alpha_i}{d!}$
  - $\alpha_j = \frac{w_1}{w_j}\alpha_1$
  - $p_i[j] = \begin{cases} vo.p[j] \ (i \neq j) \\ vo.p[j] + \alpha_i \ (i = j) \end{cases}$





GeoEst (threshold estimation & data retrieval)

- The coordinator computes the threshold ρ.



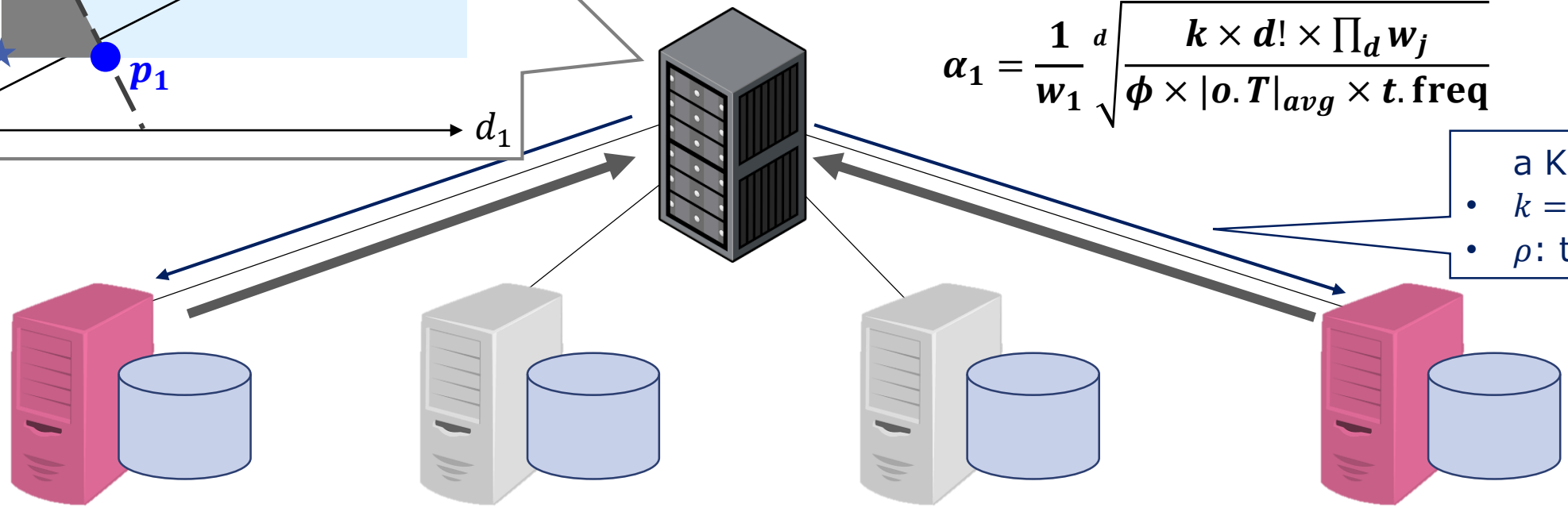
Density-based computation

- $L^{TOPK} = k \times \frac{1}{t.freq} \times \frac{1}{|o.T|_{avg}} \times \frac{1}{\phi}$
  - $t.\mathrm{freq} = \min(t_i.\mathrm{freq} \mid t_i \in T_q)$
  - $\phi = \frac{\sum |o_i|}{v_{max}}$

Geometry-based computation

- $L^{TOPK} = \frac{\prod_d \alpha_i}{d!}$
  - $\alpha_j = \frac{w_1}{w_j}\alpha_1$
  - $p_i[j] = \begin{cases} vo.p[j] \ (i \neq j) \\ vo.p[j] + \alpha_i \ (i = j) \end{cases}$

$$\alpha_1 = \frac{1}{w_1} \sqrt[d]{\frac{k \times d! \times \prod_d w_j}{\phi \times |o.T|_{avg} \times t.\mathrm{freq}}}$$

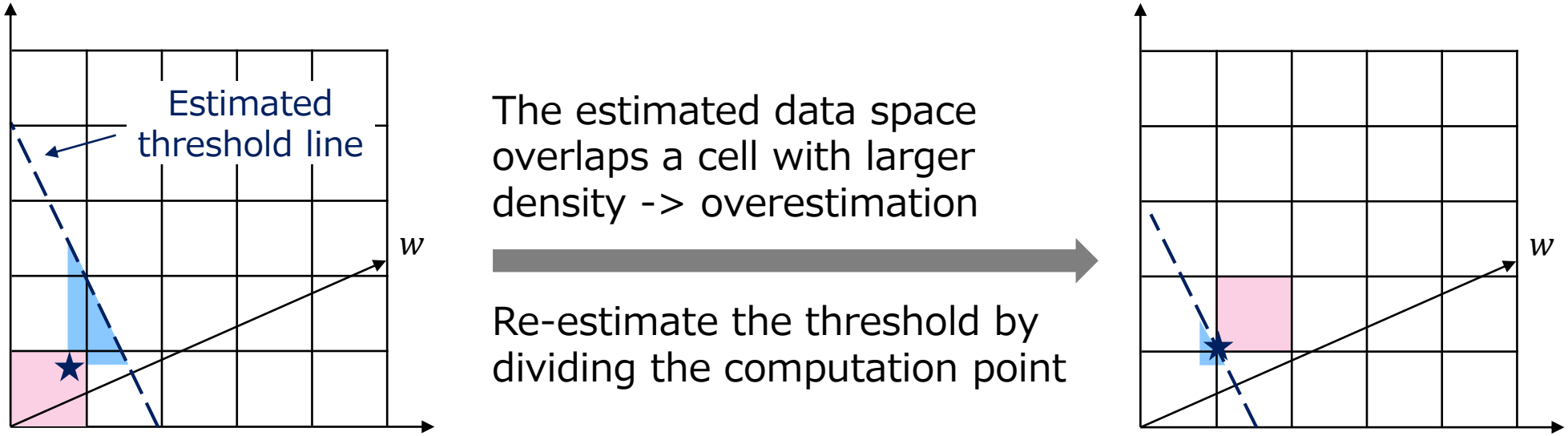


a KMTop-k query
- $k = 1$, $T_q = \{t_x, t_y, t_z\}$
- ρ: threshold

Prog-GeoEst & GeoEst+

- Drawback of GeoEst
  - Estimate the k-th smallest score **under the assumption of uniform distribution**

- Prog-GeoEst
  - Estimate the i ($\leq k$)-th smallest score progressively
    - i.e., the smallest score estimation at the first iteration
- GeoEst+
  - Collect an approximate density of each grid cell by sampling in the pre-processing



Experiment using real data

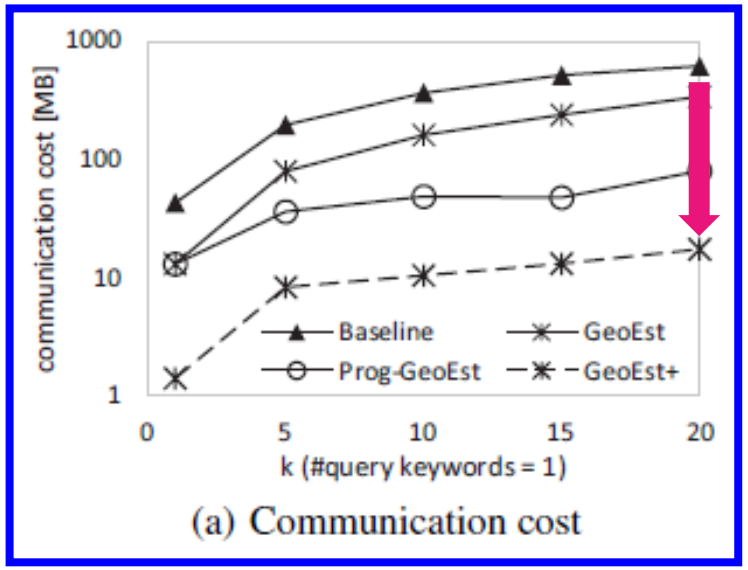
- 2,156,621 POI data in USA*
  - 2 numerical attribute values
    - distances to the nearest restaurant** and airport***
  - 5 keywords per a data object

- Parameters
  - 100 servers connected with a coordinator server
    - The dataset is evenly distributed.
    - 100[Mbps] network

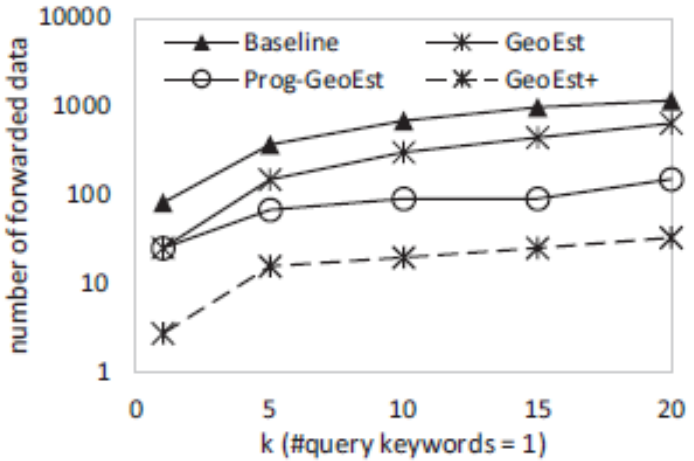
- Evaluation criteria (in KMTop-k query processing)
  - Communication cost
  - #forwarded data
  - Running time

* http://geonames.usgs.gov/
** http://www.poi-factory.com/
*** http://openflights.org/data.html/

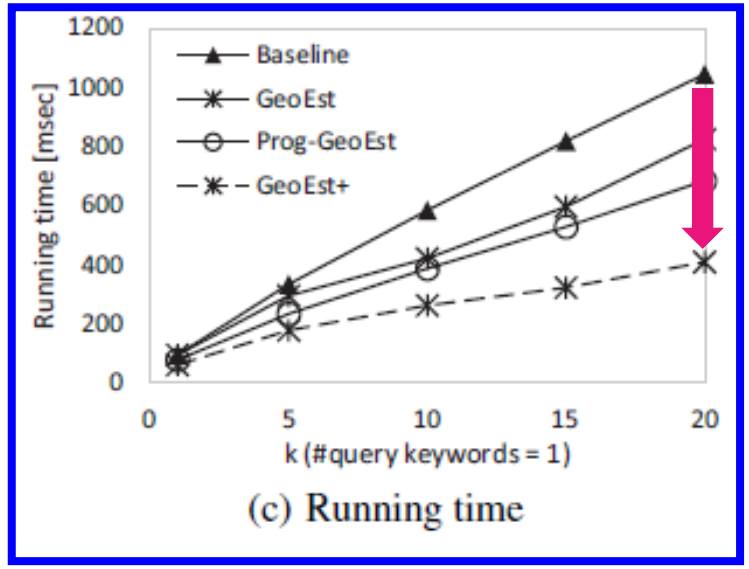
Results



(a) Communication cost

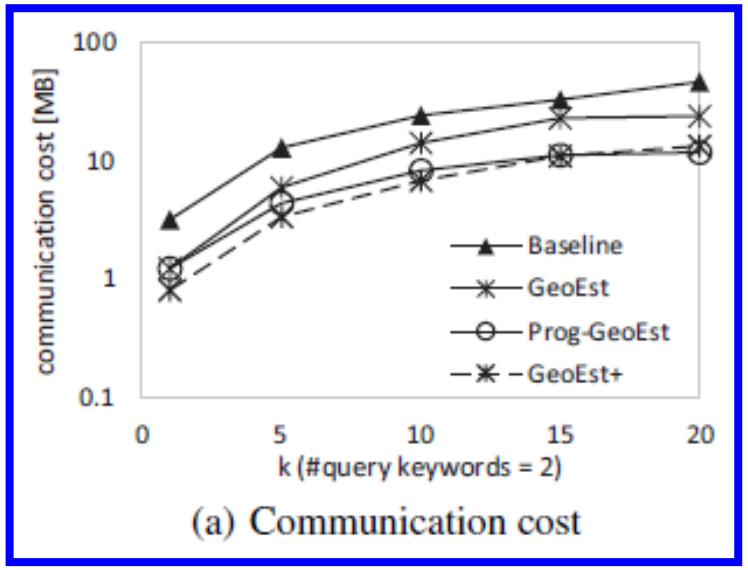


(b) #forward data

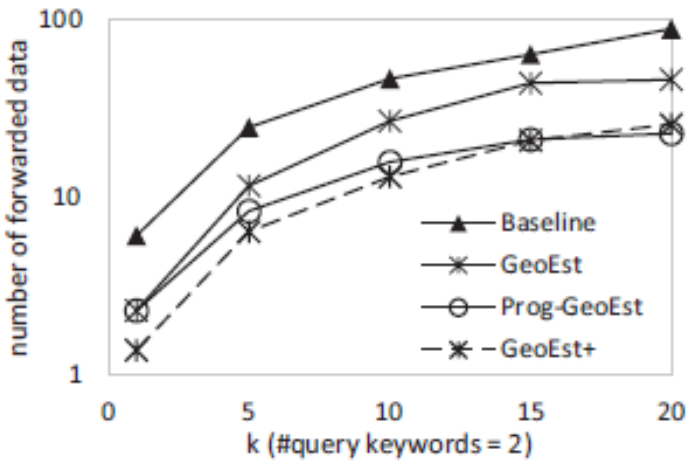


(c) Running time

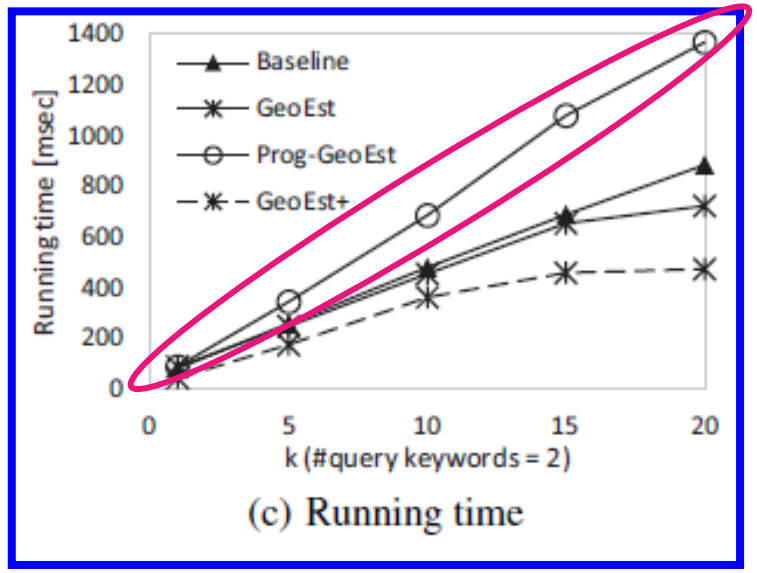
Figure 6: Impact of k ($|T_q| = 1$)



(a) Communication cost



(b) #forward data



(c) Running time

Figure 7: Impact of k ($|T_q| = 2$)

Conclusion

- KMTop-k query processing over distributed databases
  - Define KMTop-k queries
  - Propose GeoEst and its variants
  - Validation of our algorithms by experiments

- Future work
  - Develop Centralized technique
    - An efficient indexing development for I/O minimization
    - An algorithm for centralized KMTop-k query processing
    - Continuous KMTop-k query processing over data streams

THANK YOU!