ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

EFFICIENT QUERY PROCESSING BASED ON QUALITY PREFERENCES IN CLOUD DATA STORAGE

J. Lakshmi Prasanna*
* M. Tech Student, Department of Computer Science and Engineering's. R. K. R Engineering College, Bhimavaram, Andhra Pradesh, India- 534 204.

ABSTRACT
A Spatial positioning inquiry, positions items basing on the characteristics of elements in their spatial neighborhood. For instance, utilizing a land office database of flats and lands for sale, a client may need to retrieve the items basing on their spatial ranks, characterized by gathering the characteristics of different elements (e.g., shopping malls, schools, hospitals, theaters, super market, and so forth.) inside of their spatial neighborhood. For searching the spatial data i.e., some m multidimensional spaces the regular indexing systems are not suitable. To address this issue, here a novel tree structure introduced. And performs some tests on the spatial data collected from different sources and prove that this structure performs well for the multidimensional data.

KEYWORDS: Query processing, spatial databases, top-k spatial preference query.

INTRODUCTION
Geographical data, which was separated by their spatial characteristics, is maintained in our spatial database, to retrieve best spatial information for the user. The quality preferences are may be the nature of the elements that are considered. Recent days Cloud Computing has become more popular by the services it offers. Cloud Storage is one of those services, in which they maintain, manage and backup the data remotely and made available to users through a network. In this paper, a fascinating kind of inclination questions are considered, which select the best spatial area as for the nature of offices in its spatial neighborhood [2]. Given a set M of intriguing items (e.g., competitor areas), a top-k spatial inclination question recover the k objects in M with the most noteworthy scores. The score of an item is characterized by the nature of elements (e.g., offices or administrations) in its spatial neighborhood. As a persuading sample, consider a land office that holds a database with accessible pads for lease. Here "element" alludes to a class of articles in a spatial guide, for example, particular offices or administrations.

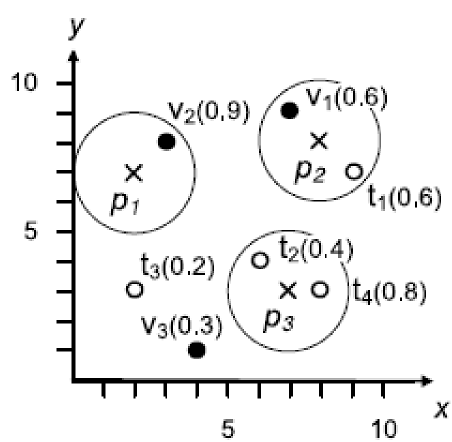

Figure 1: Spatial area containing data and feature objects.

There are large no. of web data frameworks for positioning the interested area based on the questions, with promotion of geo-labeling of data. On the other hand, a large portion of the current frameworks are restricted to plain spatial questions that arrival the items present in a given locale of the space [3]. For instance, figure 1 outlines the areas of an item information set M (Apartments) in white, and two element information sets: the set F1 (Schools) in dim, and the set F2 (Food Courts) in dark. Highlight focuses are named by quality values that can be acquired from rating suppliers. For the simplicity of dialog, the qualities are standardized to values in [0; 1]. The score (p) of an apartment p is characterized regarding: (i) the most extreme quality for every element in the area district of p, and (ii) the collection of those qualities. The visitor determines a spatial limitation (in the figure delineated as a reach around every inn) to confine the separation of the qualified element objects for every inn. Along these lines, if the real estate officer needs to rank those flats in view of the score of schools, the main 1 inn is p3 0.8) whose score 0.8 is controlled by t4. Then again, if he/she needs to rank the flats in light of food courts, the main 1 inn is p1(0.9) dictated by v2.

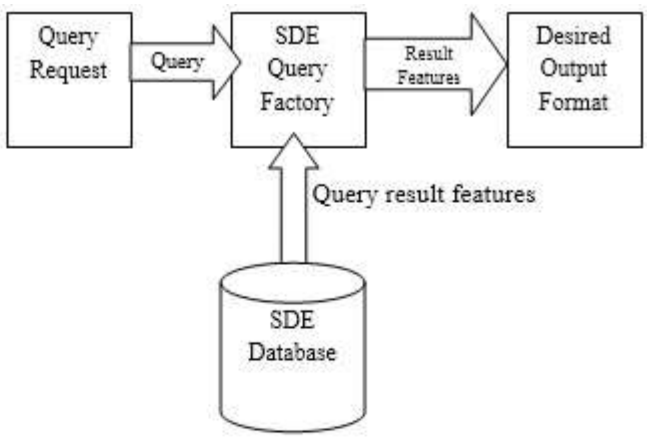

Figure 2: Interoatability of the query processing in spatial query processing.

In this paper, a novel methodology for handling spatial inclination inquiries effectively is introduced [2]. The primary process of our proposed as shown in figure 2 perform efficient data extraction in relevant data procedures in semantic data residence with relevant data in query processing Hence, matches of information and highlight items should be inspected to focus the score of an article. Our methodology depends on mapping of sets of information and highlight items to a separation score space, which permits us to recognize the negligible subset of sets that is adequate to answer all spatial inclination inquiries [6]. Gaining by the appearance of this subset of sets, a proficient calculation is done to enhance inquiry avoiding time consuming procedure .So as the execution, looks at the spatial neighborhood of information items and question execution.

RELATED WORK

A few methodologies have been proposed for positioning spatial information objects. The opposite Nearest Neighbor (KNN) inquiry was initially proposed by Korn and Muthu krishnan. At that point, Xia et al. examined the issue of recovering the top-k most compelling spatial articles in its nearest neighborhood, where the score of each spatial information object p is characterized as the scores' entirety of all component protests that have p as their closest neighbor [1]. Yang et al. considered the issue of discovering an ideal area.

The primary is that the ideal area can be any point in the information space and not so much an object of the dataset, while the score is figured in a comparative manner to. The previously stated methodologies characterize the score of a spatial information object p taking into account the scores of highlight protests that have p as their closest neighbor and are constrained to a solitary list of capabilities. In an unexpected way, [2] Joao B. Rocha Junior initially thought to be registering the score of an information object p taking into account highlight objects in its spatial neighborhood from various capabilities. To this end, three distinctive spatial scores were characterized: range, closest neighbor, and impact score; and diverse calculations were produced to register top-k spatial inclination questions for these scores.

To settle this sort of inquiries, a large portion of the specialists proposed a few strategies. In [5] 2004, F. Ilyas et.al. presented another rank-join calculation that made utilization of the individual requests of its inputs to create join results requested on a client indicated scoring capacity. They had tentatively assessed their proposed rank join administrators and investigate its execution. In 2006,[8] Rank-mindful question enhancement structure by Ihab F. Ilyas et.al. completely coordinated the rank-join administrators into social question motors and demonstrated the execution of

 ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

the proposed system. In 2007 [4] Yiu et.al. proposed the Branch and bound (BB) and Feature join calculation (FJ) that rank articles in light of the characteristics of components. They demonstrated that their proposed work is superior to anything basic and Group examining calculations with genuine and manufactured information. The top-k questions produce requested result by utilizing some ascertained score. For the most part, clients are keen on top-k join result. For this, the top-k questions oblige joins to create top-k result. The social processors ought not to transform the positioning inquiries with join proficiently.

PROPOSED WORK

A Spatial inclination question, positions the spatial items in light of nature of its neighbor offices. For instance a vacationer may recover a sorted rundown of hotels taking into account the spatial objects around that (e.g. Cafes, clinic, market, and so on.). Accept that p is our purpose of interest (e.g. a Hotel) and have a m kind of facilities (e.g. cafe implies m=1 and park implies m=2) [9]. At that point expect that n m f is n-th office from sort m (e.g. Cafe A). To begin with a rundown of contents for p are recovered, indicating how the systems picks the essential applicants. As should be obvious, Nearest Neighbor, from every sort m recovers n-th component of that (n m f) which has the base separation with p. Reach score recovers a rundown of things which have at any rate distance(d) of pre-characterized R with P. Impact score recovers every one of the things for further calculation.

Later, ranks of objects are characterized by considering the average or maximum or aggregation of the scores, w is equivalent to the weight or nature of item(e.g. lodging with 5 star can have weight of 5 and inn with one star can have weight of 1) and i is a file of recovered hopefuls. α is impact capacity which is equivalent to 1 for Nearest Neighbor and At that point the consequence of Top-K spatial inclination question is a sorted rundown of Sp for all purpose of integration.

Algorithm 1: For organizing spatial data efficiently in a tree structure.
Inputs: (i) A node *N* that has had its contents modified, (ii) The resultant split node *N*', if not NULL, that accompanies *N*.
Outputs: (i) *N* as above, (ii) *N*' as above.

- If *N* is the root Then Return {(i) *N*, (ii) *N*'}
- Let *PN* be the parent node of *N*.
- Let *EN* = (*I_N*, *child-pointer_N*) in *PN*, where *child-pointer_N* points to *N*.
- Adjust *I_N* so that it tightly encloses all entry regions in *N*.
- **If** *N*' is Not NULL **Then**
    - **If** number of entries in *PN* < *M*-1 **Then**
        - Create a new Entry *EN*' = (*I_N'*, *child-pointer_N'*)
        - Install *EN*' in *PN*
        - Return **Adjust Tree** (*PN*, NULL)
    - **Else**
        - Set {*PN*, *PN*'} = **Split Node** (*PN*, *EN*')
        - Return **Adjust Tree** (*PN*, *PN*')
    - **End If**
    - **Else**
    - Return **Adjust Tree** (*PN*, NULL)
    - Collect attributes list from leaf nodes in construction.
    - If compare each attribute with specified value c
    - If c!= null then
    - Process Spatial rank based on attributes values.
    - If srank>0 then filter spatial rank based on user selection.
    - Allocate PN to to tree leaf node
    - **End If**

Let F_c be a feature data set, in which each feature object s to F_c is associated with a quality w(s) and a spatial point.

Simple Probing Algorithm
Here this technique is used to efficiently organize the tree structure. In this when a new entry e is obtained, if the compared node is a non leaf node it is redirected to its child nodes. If it is a leaf node then upper bound is calculated.

 ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

Algorithm 2: Simple Probing Algorithm

Algorithm Sp (node N)

1. for each entry e ϵ N do
2. if N is non-leaf then
3. read the child node N' pointed by e;
4. Sp(N');
5. else
6. for c:=1 to m do
7. If T_+ (e)> γ then -> Upper bound score
8. Compute T_c (e) using tree F_c; Update T_+(e);
9. if T(e) > γ then
10. Update Wk (and γ) by e;

By using this simple probing (SP) algorithm, the data is effectively organized and the results are efficiently retrieved by computing the scores of the object points [4]. Initially, the algorithm is invoked at the root node of the object tree (i.e., N [M: root]). The procedure is recursively applied (at Line 4) on tree nodes until a leaf node is accessed.

Upper Bound Score Computation

It remains to clarify how the (upper bound) scores T of non-leaf entries (within the same node N) can be computed concurrently. Our goal is to compute these upper bound scores such that the bonds are computed with low I/O cost, and the bonds are reasonably tight, in order to facilitate effective pruning. To achieve this, only level-1 entries (i.e., lowest level non-leaf entries) are utilized in Fc for deriving upper bound scores because: 1) there are much fewer level-1 entries than leaf entries (i.e., points), and 2) high-level entries in Fc cannot provide tight bounds.

The process of the above algorithm will perform efficient and effective processing operations in organizing the spatial data with consistent data processing [4].

The overall procedure of maintaining the spatial data and performing searching is stated in below steps.

Input: Request for the desired location.

Output: Retrieves the required item in the given neighborhood.

1. Collect the Spatial data from different regions along with the quality preferences in the neighborhood.
2. Compute the spatial rank by performing some functions (like Sum, Average or Maximum) for the scores of those quality preferences.
3. Now, organize the data in tree structure by using simple probing algorithm.
4. Finally searching is performed and results are obtained by applying spatial filters that is taking into consideration of spatial ranks.

EXPERIMENTAL RESUTLS

To start with, a Real estate management system is built and trails are performed on spatial data with highlight objects and scores. Ranks are computed basing on the scores of the elements in the neighborhood of particular object. Here comparison is performed in between the spatial data that is efficiently organized in a tree structure and the spatial data that is randomly saved in a database. The time taken while using a tree structure is comparatively less, than the time taken by query search.

key words	query search	Tree based
1	0.024	0.018
2	0.022	0.014
3	0.018	0.012
4	0.021	0.015
5	0.015	0.013

Table 1: Evaluation procedure for data results.

As a matter of course, the non-spatial score of the element articles is a consistently produced worth inside of the extent [0, 1].

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

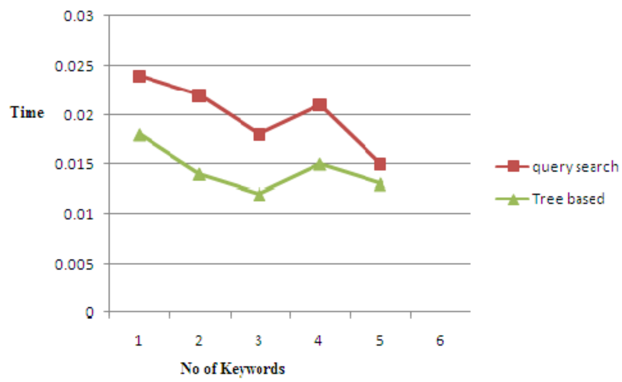

Figure 3: Time efficiency for data extraction from overall data sets.

Query Processing Performance with Respect Time

In Figure 3, the default setup is used and studies the number of I/os and the response time for all datasets, while varying k. Figure 3 shows comparison results in relevant data extraction from overall data base with respect time.

Experiments on Real Datasets

The lead experiments are done uding genuine articles furthermore; include information sets so as to exhibit the utilization of top-k spatial inclination questions. The genuine spatial information sets are taken from a real estate site Makkan.com, Areas in these information sets compare to (longitude, scope).The information sets are cleaned by disposing of records without longitude and scope. Each remaining area is standardized to a point in the 2D space [0; 10000]2. One information set is utilized as the item information set and the other two are utilized as highlight information sets. The article information set M contains 11399 outdoors areas. The component information set F1 contains 30921 inn records, each with a room value (quality) and an area. The element dataset F2 has 3848 records of Wal-Mart stores, each with fuel accessibility (quality) and an area. The area of every quality characteristic (e.g., room value, gas accessibility) is standardized to the unit interim [0; 1]. Naturally, an outdoors area is considered as great on the off chance that it is near a Wal-Mart store with high fuel accessibility (i.e., helpful supply) and an inn with high room cost (which in a roundabout way mirrors the nature of close-by open air environment). Instinctively, an outdoors area is considered as great in the event that it is near a Wal-Mart store with high gas accessibility (i.e., advantageous supply) and a lodging with high room cost (which in a roundabout way mirrors the nature of adjacent open air environment).

CONCLUSION

In this paper, a novel tree structure is implemented to obtain top-k spatial inclination inquiries, which gives best spatial results in light of characteristics of elements in their spatial neighborhood. The neighborhood of an element is defined by the nearest area for that element. By using this technique the spatial data is more efficiently searched when compared to normal query based searches.

REFERENCES

[1] C. B ohm, B. C. Ooi, C. Plant, and Y. Yan. Efficiently processing continuous k-nn queries on data streams. In Proc. of Int. Conf. on Data Engineering (ICDE), pages 156-165, 2007.
[2] “Efficient Processing of Top-k Spatial Preference Queries”, Joao B. RochaJunior, Akrivi Vlachou, Christos Doulkeridis, and Kjetil Norvag, Department of Computer and Information Science Norwegian University of Science and Technology (NTNU) Trondheim, Norway.
[3] S. Chaudhuri, N. N. Dalvi, and R. Kaushik. Robust cardinality and cost estimation for skyline operator. In *Proc. of* In Proc. of Int. Conf. on Data Engineering (ICDE), pages 64, 2006.
[4] **“**Ranking Spatial Data by Quality Preferences**”, by** Man Lung Yiu, Hua Lu, Nikos Mamoulis, and Michail Vaitis, proceedings in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.23, NO. 3, March 2011.
[5] M. L. Yiu, N. Mamoulis, and P. Karras, “Common Influence Join: A Natural Join Operation for Spatial Pointsets,” in ICDE, 2008.
[6] Y.-Y. Chen, T. Suel, and A. Markowetz, Efficient Query Processing in Geographic Web Search Engines,” in SIGMOD, 2006.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

[7] V. S. Sengar, T. Joshi, J. Joy, S. Prakash, and K. Toyama, “Robust Location Search from Text Queries,” in ACM GIS, 2007.
[8] S. Berchtold, C. Boehm, D. Keim, and H. Kriegel, “A Cost Model for Nearest Neighbor Search in High-Dimensional Data Space,” in PODS, 1997.
[9] E. Dellis, B. Seeger, and A. Vlachou, “Nearest Neighbor Search on Vertically Partitioned High-Dimensional Data,” in DaWaK, 2005, pp. 243–253.
[10] I. F. Ilyas, W. G. Aref, and A. Elmagarmid, “Supporting Top-k Join Queries in Relational Databases,” in VLDB, 2003.
[11] N. Mamoulis, M. L. Yiu, K. H. Cheng, and D. W. Cheung, “Efficient Top-k Aggregation of Ranked Inputs,” ACM TODS, vol. 32, no. 3, p. 19, 2007.
[12] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, “Efficient OLAP Operations in Spatial Data Warehouses,” in SSTD, 2001.
[13] S. Hong, B. Moon, and S. Lee, “Efficient Execution of Range Top-k Queries in Aggregate R-Trees,” IEICE Transactions, vol. 88-D, no. 11, pp. 2544–2554, 2005.