Chapter 21
Algorithmic Approach for Spatial Entity and Mining

Priya Govindarajan
SASTRA University (Deemed), India

Balakrishnan R.
https://orcid.org/0000-0002-4752-6400
SASTRA University (Deemed), India

Rajesh Kumar N.
https://orcid.org/0000-0001-5394-218X
SASTRA University (Deemed), India

ABSTRACT

Mining has gained its momentum in almost every arena of research. The mining can be either spatial or non-spatial based on the search query. For classifying or for grouping the spatial data, algorithms with extended perspectives are projected in this chapter. Besides framing algorithms, one can also provide mass points based on the required attributes as well as indexing techniques. The extended algorithms can also be manipulated for efficient and robust solution with respect to different parameters.

MOTIVATION

Many databases contain both spatial and non-spatial information, for which algorithms are being devised with prolonged comparative study and also keeping in perspective of different parameters for analyzing spatial information (Zhang et.al., 2022, Medad et.al., 2020). The wide spread knowledge (Germanaite et.al., 2021, Govindarajan & K.S., 2014) indicates there is no such system which drastically reduces the fault cost and time during the search.

We have ventured an extended technique, which efficiently reduces the privilege for accessing the featured dataset (Cheng & junli.,2013) and the searched object. The indexed objects higher score values

DOI: 10.4018/978-1-7998-8892-5.ch021

are pruned while accessing the spatial dataset. The extended versions of algorithms are being proposed for efficient outcome.

For accessing and retrieving data in different applications, objects are being ranked. (Isaj et.al., 2019) Until the identification of summed up value, the no. of times an object are being accessed should be minimized or brought down. The algorithms are being devised, keeping an eye on the above constraints.

ALGORITHMIC SCHEMA

Trouble-Free Probing (TP)

Trouble-free Probing (TP), computes the mass of every object (*Algorithm 1*). It uses two globalized term (assuming) X_k for manipulating the top-k results and Y calculates the value of top-k. Algorithm invokes at root node of the tree. The algorithm works recursively on the nodes of the tree, until it encounters (Sivakami & G.M.K, 2011) a terminating node. Once the non-terminating node is reached the mass M is calculated through the execution of a range of search on the tree T. The point b is ignored if its upper bound mass $M_+(b)$ can't be greater than best-k mass Y. The term X_k and Y are overwritten when the mass M(b) is greater than Y.

Algorithm TP (node N) (Algorithm 1)

For all entries in B (belongs to) N do
If N is terminating node Then
Take the subsidiary node N'-B
SP (N')
Else
A = 1 to m do
If $M_+(b) > Y$ Then
Calculate M using tree T; Update $M_+(b)$
If M>Y Then
Overwrite X_k (and Y) by B

Extended Branch and Bound (EBB)

While computing, Trouble-free algorithm seems to be inefficient for huge data sets. Hence a version of Extended Branch and Bound (EBB) is proposed (*Algorithm 2*) for accessing huge data sets. Let N be the root node of T, If N comes under the terminating category f, and then mass M(f) is manipulated. With the aid of mass for the component $M_c(f)$ one can drive the $M_+(f)$, an upper bound of M(f). If $M_+(f)<=Y$, (Thakoor N. et.al., 2008) then one cannot fetch best result with the sub-tree of f than those in X_k and removed from Z(subset of T).To fetch high mass, M(f) is sorted in descending order then the nodes of the subsidiary are called recursively by the entries in Z. X_k of top-k results (You wan, et.al., 2008) are updated by manipulating the mass of all N simultaneously. Whenever EBB is called recursively the globalised variables X_k and Y are updated then and there.

Related Content