An Efficient Approach for Multi-Dimensional Dataset Search Using Keywords

[1]Susmitha Dasari, [2]Loshma Gunisetti

[1]Mtech Scholar, [2]Assosiate Professor
[1]Computer Science and Engineering,
[1]Sri Vasavi Engineering College, Tadepalligudem, India

***Abstract*— Conventional spatial queries, such as range search and nearest neighbor retrieval, involve only conditions on objects' geometric properties. Today, many modern applications call for novel forms of queries that aim to find objects satisfying both a spatial predicate, and a predicate on their associated texts. Currently, the best solution to such queries is based on the IR2 -tree, which, as shown in this paper, has a few deficiencies that seriously impact its efficiency. Motivated by this, we develop a new access method called the spatial inverted index that extends the conventional inverted index to cope with multidimensional data, and comes with algorithms that can answer nearest neighbor queries with keywords in real time. As verified by experiments, the proposed techniques outperform the IR2 -tree in query response time significantly, often by a factor of orders of magnitude.**

***Index Terms*— Nearest neighbor search, spatial index, keyword search.**

I. Introduction

A spatial information could be an information that stores multidimensional objects like points, rectangles, and etc. some spatial databases permit representing easy geometric objects like lines, points and polygons. Some spatial databases handle additional advanced structures like3D objects, topological coverage's, linear networks. Based on totally different choice criteria spatial information provides quick access to 3D objects.

In spatial information real entities are sculptured in geometric manner, as an example location of hotels, hospital, restaurants are described as points on maps, whereas larger space like landscapes, lakes, parks is described as a mix of rectangles .Spatial information system will utilize in geographic information systems, during this vary search may be utilized to seek out all restaurants in a very sure space, whereas nearest neighbor retrieval will notice the building nearer to a given address. Queries about abstraction information became more and more important to recent years with the increasing quality of some services like Google Earth and Yahoo Maps, as well as different geographic applications.

Today, wide used of search engines has created it realistic to put in writing abstraction queries in an exceedingly new means. Historically, queries target objects solely geometric properties, as an example whether or not a point is in parallelogram or however two purposes are shut from every other. Some new application permits users to browse objects based on each of their geometric coordinate and their associated texts. Such sort of queries referred to as abstraction keyword question. For example, if a groundwork engine will be used to find nearest building that provides facilities like pool and internet at identical time. From this question, we have a tendency to might the initial obtain the complete building whose services contain the set of keywords, then notice the closest one from the retrieved restaurant.

The foremost disadvantage of this approach is that, on the troublesome input they are doing not offer real time answer. For example, from the question purpose the important neighbor lies quite far away, whereas all the nearer neighbors square measure missing a minimum of one of the question keywords. Abstraction keyword queries have not been wide explored. Within the past years, the cluster of people has shown interest in learning keyword searches in relational databases. Recently the eye has preoccupied to flat knowledge [5][6]. The simplest technique for nearest neighbor search with keywords is owing to Felipe et al. [5]. They mix the abstraction index R-tree [7] and signature file [8]. So that they developed a structure referred to as IR2-tree. This tree has the flexibility of each R-tree and signature files. Like R-tree it stores the abstraction proximity of object and like signature file it filters those objects that don't include all question keywords.

II. Related Work

A study was undertaken by S. Agrawal, S. Chaudhuri, and G. Das [18] Internet search engines have popularized the keyword-based search paradigm. While traditional database management systems offer powerful query languages, they do not allow keyword-based search. In this paper, we discuss DBXplorer, a system that enables keyword-based searches in relational databases. DBXplorer has been implemented using a commercial relational database and Web server and allows users to interact via a browser front-end. We outline the challenges and discuss the implementation of our system, including results of extensive experimental evaluation.

Keyword Searching and Browsing in Databases Using Banks, [2] by this with the growth of the Web, there has been a rapid increase in the number of users who need to access online databases without having a detailed knowledge of the schema or of query languages; even relatively simple query languages designed for non-experts are too complicated for them. We describe BANKS, a system which enables keyword-based search on relational databases, together with data and schema browsing. BANKS enables users to extract information in a simple manner without any knowledge of the schema or any need for writing complex queries. A

user can get information by typing a few keywords, following hyperlinks, and interacting with controls on the displayed results. BANKS models tuples as nodes in a graph, connected by links induced by foreign key and other relationships. Answers to a query are modeled as rooted trees connecting tuples that match individual keywords in the query. Answers are ranked using a notion of proximity coupled with a notion of prestige of nodes based on inlinks, similar to techniques developed for Web search. We present an efficient heuristic algorithm for finding and ranking query results.

Spatial keyword querying, the web is increasingly being used by mobile users. In addition, it is increasingly becoming possible to accurately geo-position mobile users and web content. This development gives prominence to [24] spatial web data management. Specifically, a spatial keyword query takes a user location and user-supplied keywords as arguments and returns web objects that are spatially and textually relevant to these arguments. This paper reviews recent results by the authors that aim to achieve spatial keyword querying functionality that is easy to use, relevant to users, and can be supported efficiently. The paper covers different kinds of functionality as well as the ideas underlying their definition.

G. Cong, C.S. Jensen, and D. Wu dialect [4] planned to associate approaches that computes the relevancy between the documents of associate objects and a question. This relevancy is then incorporated with the geometrician distance between an object and question to calculate associate overall similarity of object to query.

The location-aware keyword query returns ranked objects that are near a query location and that have textual descriptions that match query keywords. This query occurs inherently in many types of mobile and traditional web services and applications, e.g., Yellow Pages and Maps services. Previous work considers the potential results of such a query as being independent when ranking them. However, a relevant result object with nearby objects that are also relevant to the query is likely to be preferable over a relevant object [25] without relevant nearby objects. The paper proposes the concept of prestige-based relevance to capture both the textual relevance of an object to a query and the effects of nearby objects. Based on this, a new type of query, the Location-aware top-k Prestige-based Text retrieval (LkPT) query, is proposed that retrieves the top-k spatial web objects ranked according to both prestige-based relevance and location proximity. We propose two algorithms that compute LkPT queries. Empirical studies with real-world spatial data demonstrate that LkPT queries are more effective in retrieving web objects than a previous approach that does not consider the effects of nearby objects; and they show that the proposed algorithms are scalable and outperform a baseline approach significantly.

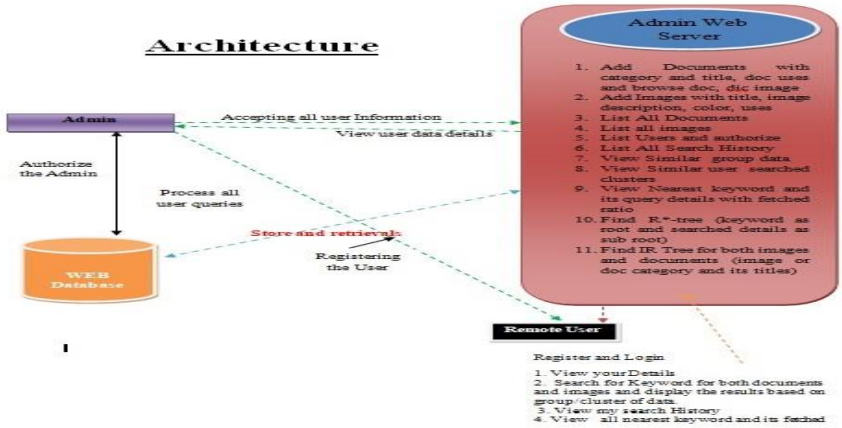
Objects (e.g., images, chemical compounds, documents, or experts in collaborative networks) are often characterized by a collection of relevant features, and are commonly represented as points in a multi-dimensional feature space. For example, images are represented using color[20] feature vectors, and usually have descriptive text information (e.g., tags or keywords) associated with them. In this paper, we consider multi-dimensional datasets where each data point has a set of keywords. The presence of keywords in feature space allows for the development of new tools to query and explore these multi-dimensional datasets.

III. Proposed Work

In this paper, a variant of inverted index that is optimized for multidimensional points, and is thus named the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. Meanwhile, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead.

As a result, it offers two competing ways for query processing. We can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also leverage the R-trees to browse the points of all relevant lists in ascending order of their distances to the query point. As demonstrated by experiments, the SI-index significantly outperforms the IR2 -tree in query efficiency, often by a factor of orders of magnitude. Distance browsing is easy with R-trees. In fact, the best-first algorithm is exactly designed to output data points in ascending order of their distances. It is straight forward to extend our compression scheme to any dimensional space.

System Architecture



Spatial Inverted Index Algorithm:

Spatial data mining is a special kind of data mining. The main difference between data mining and spatial data mining is that in spatial data mining tasks we use not only non-spatial attributes (as it is usual in data mining in non-spatial data), but also spatial attributes. Spatial data mining is the application of data mining methods to spatial data. The objective of spatial data mining is to find patterns in data with respect to geography. So far, data mining and Geographic Information Systems (GIS) have existed as two

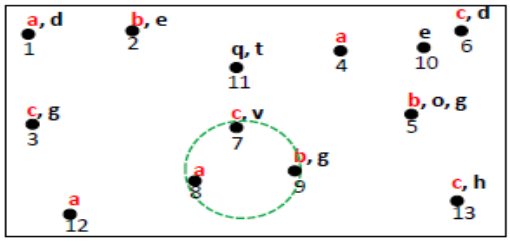
separate technologies, each with its own methods, traditions, and approaches to visualization and data analysis.
Input: Query, Cache Queries
Output: Result set generated for query
Procedure:
If Query available in cache
Result related to query: =ForwardToTreeprocess (Query)
Else
Result related to query: = GeocodingtreeProcess(Query)
Geocoding process(Query):
Parameters
Qi—Input Spatial Query
Qj (j=1…n) ---Set of Queries contains same Location
Dist[j] (j=1…..n)-----Array for set of distances
Procedure:
(xi,yi)---Geocodings of Qi
(xj,yj)--- Geocodings of all queries with respect to location
Dist[i]=Euclidean distance between the geocodes
While not leafnode
Read nodes from tree For Q.features
If Q.features[i]==Q.features[j] Add to list
End while
Sort list by feature and distance
Return list.

ForwardToTreeprocess ()

1. Build an empty list
2 .Make a root node
3. if Qi in cache and status=false
For j=0 to n
Compare features(Qi,Qj) status=true;
For Each child in tree
If(status==true)
Getnodebyfeature (Qi);
Getnodebyfeature (Qj);
End
Else
Empty list ()
End For Each
4.Add nodes to list
5.Return list

PRUNING ALGORITHM:-

Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of over fitting.



IV Conclusion and Future Enhancement

In this paper, we implemented solutions to the problem of top- k nearest keyword set search in multi-dimensional datasets. We develop a new access method called the spatial inverted index that extends the conventional inverted index with multidimensional data, and comes with algorithms that can answer nearest neighbour queries with keywords in real time. Moreover, our techniques scale well with both real and synthetic datasets. In the future, we plan to explore other scoring schemes for ranking the result sets. In one scheme, we may assign weights to the keywords of a point by using techniques like tf-idf. Then, each group

of points can be scored based on distance between points and weights of keywords. Furthermore, the criteria of a result containing all the keywords can be relaxed to generate results having only a subset of the query keywords.

REFERENCES

[1] A. Khodaei, C. Shahabi, and C. Li, "Hybrid indexing and seamless ranking of spatial and textual features of web documents," in DEXA, 2010, pp. 450–466.
[2] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in ACM SIGMOD, 1984, pp. 47–57.
[3] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in VLDB, 1999, pp. 518–529.
[4] B. Martins, M. J. Silva, and L. Andrade, "Indexing and ranking in geo-ir systems," in workshop on GIR, 2005, pp. 31–34.
[5] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, "Collective spatial keyword queries: a distance owner-driven approach," in SIGMOD, 2013.
[6] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," PVLDB, vol. 2, pp. 337–348, 2009.
[7] H. He and A. K. Singh, "Graphrank: Statistical modeling and mining of significant subgraphs in the feature space," in ICDM, 2006, pp. 885–890.
[8] I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in ICDE, 2008, pp. 656–665.
[9] J. Bourgain, "On lipschitz embedding of finite metric spaces in Hilbert space," Israel J. Math., vol. 52, pp. 46–52, 1985.
[10] J. M. Kleinberg, "Two algorithms for nearest-neighbor search in high dimensions," in STOC, 1997, pp. 599–608.
[11] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in SCG, 2004.
[12] M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Top-k spatial preference queries," in ICDE, 2007, pp. 1076–1085.
[13] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," in SIGMOD, 1990, pp. 322–331.
[14] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in VLDB, 1994, pp. 487–499.
[15] R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and perfomance study for similarity-search methods in high-dimensional spaces," in VLDB, 1998, pp. 194–205.
[16] S. Agrawal, S. Chaudhuri, and G. Das Dbxplorer: A System for Keyword-Based Search over Relational Databases".2011.
[17] T. Xia, D. Zhang, E. Kanoulas, and Y. Du, "On computing top-t most influential spatial sites," in VLDB, 2005, pp. 946–957.
[18] Vishwakarma Singh, Bo Zong, Ambuj K. Singh: Nearest Keyword Set Search in Multi- dimensional Datasets, vol 28, pp, 3-1, 2016.
[19] V. Singh, S. Venkatesha, and A. K. Singh, "Geo-clustering of images with missing geotags," in GRC, 2010, pp. 420–425.
[20] V. Singh, A. Bhattacharya, and A. K. Singh, "Querying spatial patterns," in EDBT, 2010, pp. 418–429.
[21] W. Li and C. X. Chen, "Efficient data modeling and querying system for multi-dimensional spatial data," in GIS, 2008, pp. 58:1–58:4.
[22] X. Cao, L. Chen, G. Cong, C.S. Jenson, Q.Qu, A. Skoys gaard, D. Wu, & M, L.Yiu: Spatial Keyword Querying.
[23] X. Cao, G. Cong, and C.S. Jensen: Retriving Top-k prestige-Based Relevent Spatial web objects.
[24]Y.-Y.Chen, T.Suel, and A. Markowetz: Efficient Query processing in geographic web search engines.
[25] Z. Li, H. Xu, Y. Lu, and A. Qian, "Aggregate nearest keyword search in spatial databases," in Asia-Pacific Web Conference, 2010.