International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 4 938 – 941

Geographical Search with Approximate String in Spatial Databases

S.Anandhi[1] , B .Anantharaj[2], R.Hariraman[3]

Student[1], HOD[2], Faculty[3] Department of CSE, Thiruvalluvar college of Engineering and Technology
Anna University, India
anandhi_soundari@yahoo.com[1], ananthu_arun72@yahoo.com[2], hariraman.vs@gmail.com[3]

***Abstract*--**This work deals with the approximate string search in large spatial databases. Specially, I investigate range queries augmented with a string similarity search predicate in both Euclidean space and road networks. I dub this query the Spatial Approximate String (SAS) query. In Euclidean space, it propose an approximate solution, the MHR-tree, which embeds min-wise signatures into an R-tree. The min-wise signature for an index node u keeps a concise representation of the union of q-grams from strings under the sub-tree of u. It analyzes the pruning functionality of such signatures based on the set resemblance between the query string and the q-grams from the sub-trees of index nodes. We analyze the pruning functionality of such signatures based on set resemblance between the query string and the q-grams from the sub-trees of index nodes. MHR-tree supports a wide range of query predicates efficiently, including range and nearest neighbor queries. We also discuss how to estimate range query selectivity accurately. We present a novel adaptive algorithm for finding balanced partitions using both the spatial and string information stored in the tree. Extensive experiments on large real data sets demonstrate the efficiency and effectiveness of our approach.

Index Terms– *Spatial Databases, Approximate String Search, Range query, Road Networks.*

*****

1. INTRODUCTION

With the explosion of Web 2.0 services, more and more user generated data have been shared on the Web. They exist in the form of user reviews on shopping or opinion sites, in posts of blogs or
customer feedback.It is the process of analyzing data from different perspectives and summarizing it into useful information. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified.

1.1 AIM OF KNOWLEDGE DISCOVERY

Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years, due to the wide availability of huge amounts of data and its significant need for turning such data into useful information and knowledge is needed.

The information and knowledge thus gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration.

1.2 SPATIAL DATABASES

A spatial database is a collection of spatial data types, operators, indices, processing strategies, etc. and can work with many post-relational DBMS as well as programming languages like Java, Visual Basic etc.

In order to visualize and analyze spatial data using spatial analysis functions such as Search Thematic search, search by region, classification, Location analysis Buffer, corridor, overlay, Terrain analysis Slope/aspect, catchment, drainage network, Flow analysis Connectivity, shortest path.

Distribution Change detection, proximity, nearest neighbor, Spatial analysis/Statistics Pattern, centrality, autocorrelation, indices of similarity, topology: hole description, Measurements Distance, perimeter, shape, adjacency, direction. An efficient algorithm to answer spatial queries uses two Common Strategy - filter and refine methods.

Approximating spatial data types such as Minimum orthogonal bounding rectangle (MOBR or MBR), approximates line string, polygons.MBRs are used by spatial indexes, e.g. R-tree, Algorithms for spatial operations MBRs are simple.

The building blocks used for query processing are Point Query-Return one spatial object out of a table, Range Query- Returns several objects within a spatial region from a table, Spatial Join-Return pairs from 2 tables satisfying a spatial predicate ,Nearest Neighbor- Return one spatial object from a collection.

The Query makes an execution plan with intermediate stopovers and makes an query tree, which is later transformed into logical tree transforms, selection strategy is used here. Once the execution plan is executed, Query answer is returned.

This paper is ordered as follows. Section 2 deals about the implementation, Section 3 focuses the ways of Algorithm used. Section 4 discusses about the related works and discussion. Section 5 discusses about the future enhancements and Section 6 reviews the conclusion of this paper.

2. IMPLEMENTATION

Based on the previous research we are taking, checking address in the road network in map as our target. Need for implementing the project is that it is very helpful for Exact Result from Non Exact keywords. To increase the selectivity estimation providing query optimization and data analysis, novel method of SAS is introduced.

Here in this I implemented modules of registration of user and admin, uploading of details about the place by admin and the user sends an key for searching the address, edit distance pruning is done with the given query and answer for the search is provided. In this registration module, Users and admin are having authentication and security to access the detail which is presented in the ontology system.

In the key module, the key of common Index can be made from the Index word given by the Data owner and File. The secure index and a search scheme to enable fast similarity search in the context of data. In such a context, it is very critical not to sacrifice the confidentiality of the sensitive data while providing functionality. We provided a rigorous security definition and proved the security of the proposed scheme under the provided definition to ensure the confidentiality.

In the Edit Distance Pruning, Computing edit distance exactly is a costly operation. Several techniques have been proposed for identifying candidate strings within a small edit distance from a query string fast. All of them are based on q-grams and a q-gram counting argument. For a string s, its q-grams are produced by sliding a window of length q over the characters of s. To deal with the special case at the beginning and the end of s, that have fewer than q characters, one may introduce special characters, such as "#" and "$", which are not in S. This helps conceptually extend s by prefixing it with q – 1 occurrences of "#" and suffixing it with q – 1 occurrences of "$". Hence, each q-gram for the string s has exactly q characters.

In the Search module, we provide a specific application of the proposed similarity searchable encryption scheme to clarify its mechanism. Server performs search on the index for each component and sends back the corresponding encrypted bit vectors it makes by the respective like commend.

In the min-wise signature module, Implementation of min-wise independent permutations requires generating random permutations of a universe and Broder et al. showed that there is no efficient implementation of a family of hash functions that guaran-tees equal likelihood for any element to be chosen as the minimum element of a permutation. Thus, prior art often uses linear hash functions based on Rabin fingerprints to simulate the behavior of the min-wise independent permutations since they are easy to generate and work well in practice

Finally, we illustrated the performance of the proposed scheme with empirical analysis on a real data.

3. ALGORITHMS

3.1 Rsassol Algorithm

RSASSOL algorithm is used in RSAS. It occurs in five steps. The first step is that for a given query, it finds all the subgraphs that intersect with the query range. The second step uses the filter-trees of these subgraphs to retrieve the points whose strings are potentially similar to the query string. In the third step, it prune away some of these candidate points by calculating the lower and upper bounds of their distance point. The fourth step is to further prune away some candidate points using the exact edit distance between the query string and strings of remaining candidates. The string predicate has been fully explored in this step. The final step, for the remaining candidate points, it checks their exact distances to the query point and return those with distances with in the query range.

3.2 Mpalt Algorithm

The MPALT algorithm is defined as the Multipoint Abbreviated List Table. This algorithm computes multiple shortest paths, within the query range, simultaneously at once between a single source point and multiple destination points. The distances computed and stored in storage model between a node to all reference nodes, which allows us to compute lower and upper distance bounds for any given node and any destination.

The basic idea is that it starts the expansion of the network from source with the two nodes from the edge containing source node and always expand the network from an explored node that has the shortest possible distance to any one of the destinations. The algorithm terminates when the priority queue becomes empty.

This algorithm minimizes the access to the network by avoiding the nodes that will not be on any shortest path distance between source and destination. It avoids repeatedly access to the explored part of the network when calculating multiple shortest paths to multiple destinations.

4. Range Query and Edit distance

A range query is a common database operation that retrieves all records where some value is between an upper and lower boundary. The Data structures for range query are Range tree.Range query consists of preprocessing some input data into a data structure to efficiently answer any number of queries on any subset of the input. a range query $q_f(A,i,j)$on an array $A=[a_1,a_2,..,a_n]$of *n* elements of some set S, denoted $A[1,n]$, takes two indices $1 \le i \le j \le n$, a function $f$defined over arrays of elements of $S$and outputs $f(A[i,j]) = f(a_i,\ldots,a_j)$. This should be done space and time efficient.

consider for instance $f = sum$and $A[1,n]$ array of numbers, the range query $sum(A,i,j)$computes $sum(A[i,j]) = (a_i + \ldots + a_j)$, for any $1 \le i \le j \le n$. These queries may be answered in constant time and using $O(n)$extra space by calculating the sums of the first $i$elements of $A$and storing them into an

 ISSN: 2321-8169
938 – 941

auxiliar array B, such that $B[i]$ contains the sum of the first i elements of A for every $0 \leq i \leq n$. Therefore any query might be answered by doing $sum(A[i,j]) = B[j] - B[i-1]$.

Edit distance is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other. Edit distances find applications in natural language processing, where automatic spelling correction can determine candidate corrections for a misspelled word by selecting words from a dictionary that have a low distance to the word in question.

4. RESULTS

We implemented the R-tree solution(Fig 4.1), the string index solution and the MHR-tree, using the widely adopted spatial index library . We do not report any results for the string index since, first, it requires linear space with respect to the

Fig 4.1

Number of data q-grams and hence space-wise it is not competitive, and, its query performance was not up to par across the board.

The adaptive R-tree algorithm(Fig 4.2) for the selectivity estimator seamlessly works for both the R-tree and the MHR-tree. The default page size is 4KB and the fill factor of all indexes is

0.7. All experiments were(Fig 4.3) executed on a Linux machine with an Intel Xeon CPU at 2GHz and 2GB of memory.

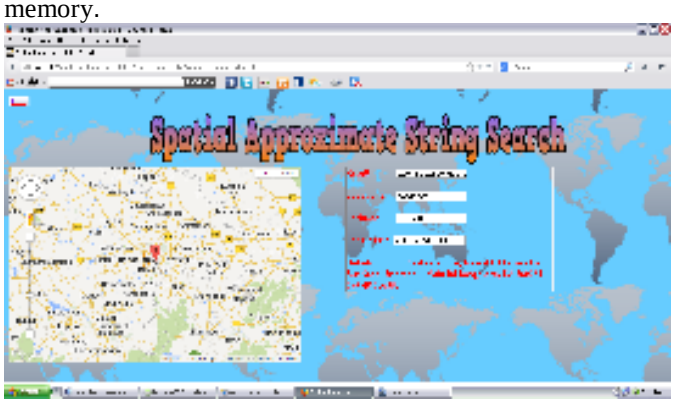

Figure 4.2

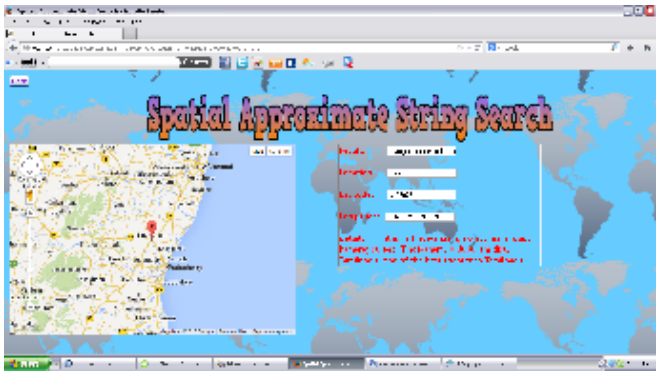

Figure 4.3

5. RELATED WORKS

Plenty of researches are done in searching data and their location with better selectivity estimation. Regarding similar research are presented in this paper.

Alsubaiee.S [1] proposed a method to provide results to the query made with their location. Many websites provide answers for spatial data queries, inconsistencies and errors may occur in either query made by the user or in the database where data stored. Search which provides results for this queries is important, but mostly it considers only spatial keywords not their location specified. He built an LBAK- tree (Location Based Approximate Keyword) which has an tree based spatial index, gram based spatial index to answer for the misspelled query made by the user and finds an optimal solution.

Felipe.I.D [7] proposes an tree structure for answering spatial database queries named IR- tree (Information Retrieval). It uses the combination of R-tree and signature files techniques, for indexing. In this it finds the nearest neighbor to the query point, and retrieves an object. An comparison is made between the object's textual description and the query's keyword, when the comparison fails the selected object is discarded and searches for next object, the process continues until the textual comparison fails at the end of the tree index, then the tree has to be trans versed and again object has to be compared in transverse order.

Hadjieleftheriou.M and Li.C, [8] uses different approximate search in finding text data which is widely used. In order to find an query string from a collection of strings, that is similar to an given query approximate string search method is used. Data cleaning, Query Relaxation, Interactive search and Spell checking are the techniques which are used mainly. Data cleaning involves in finding similar string among collections or to find an pair of strings across multiple collections. Query relaxation technique uses query predicate similarity to answer an query made. Spell checking uses an dictionary to find answers for the given query. Interactive Search is provided with suggestion box for the mistyped word.

Jin.L [9] finds the solution by developing an technique called SEPIA, selectivity estimation for string predicates in order to get better result for the fuzzy query

search. The group of strings are made in to clusters, for that cluster an histogram is made, for multiple histograms an global histogram is made. In order to find the string similarity function, edit distance is used. This supports accurate and efficient selectivity estimation in string predicates.

Various researches have been studied using the informative messages and presented as tree structure for indexing. The motivation of our study is that approximate string search in spatial databases should provide an accurate and efficient results or answers.

6. FUTURE ENHANCEMENTS

Many studies are undertaken to search for data they needed with correct spelling in the queries. Most are aimed to find the data without complete knowledge about the query they submit. Our study is to make the query in fuzzy condition. It is possible that many data's can be found without considering the spelling and its degree of uncertainty of error.

Expanding the system to detect various kinds of data's using fuzzy string search. The system includes the assumption to get a data misspelling or error. We can find the data related with misspelling query instead of the message "No Result Found".

A search query is important for searching data .we can search data to find data and its location needed which are newer to the user. Advance algorithms can be useful for our future work.

7. CONCLUSION

As described in the paper approximate string search in spatial databases is examined. The similarity measurement is found for string predicate using edit distance and for spatial predicate it is done with range queries. Based on this we find an data. As an application the search is used when users have an fuzzy search queries for data.

It is hope that it provides some future integration in searching for substrings with approximate makes user updating with selectivity estimation.

REFERENCES

[1] Alsubaiee.S., Behm.A. And Li. C, (2010) "Supporting Location-Based Approximate-Keyword Queries," Proc. SIGSPATIAL 18th Int'l Conf. Advances in Geographic Information Systems (GIS), pp. 61-70.

[2] Beckmann, Krieger H.P., Schneider, and Seeger (2000)," The R_- tree: an efficient and robust access method for points and rectangles." In SIGMOD, pages 322–331.

[3] Cao.X, Cong.G, and Jensen.C.S. (2010) "Retrieving top-k prestige-based relevant spatial web objects" Proc. VLDB Endow., 3:373–384.

[4] Chakrabarti.K, Chaudhuri.S, Ganti.V, and Xin.D (2008)" An efficient filter for approximate membership checking". In SIGMOD, pages 805–818.

[5] Chaudhuri.S, Ganjam.K, Ganti.V, and Motwani.R (2003)" Robust and efficient fuzzy match for online data cleaning". In SIGMOD, pages 313– 324.

[6] Chaudhuri.S, Ganti.V, and Gravano.L (2004)" Selectivity estimation for string predicates: Overcoming the underestimation problem." In ICDE, pages 227–238

[7] Felipe.I.D., Hristidis.V, and Rishe.N, (2011) "Keyword Search on Spatial Databases," Proc. IEEE Int'l Conf. Data Eng. (ICDE), pp. 656-665.

[8] Hadjieleftheriou.M and Li.C, (2009)"Efficient Approximate Search on String Collections," Proc. VLDB Endowment, vol. 2, no. 2, pp. 1660-1661.

[9] Jin.L and Li.C, (2011) "Selectivity Estimation for Fuzzy String Predicates in Large Data Sets," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 397-408.

[10] Lee.H, Ng.R.T, and Shim.K, (2012) "Approximate Substring Selectivity Estimation," Proc. 12th Int'l Conf. Extending Database Technology: Advances in Database Technology (EDBT), pp. 827-838.

S.Anandhi received her B.Tech degree in Information Technology (IT) in 2011 from Anna University,Trichy and post graduate degree in Computer Science and Engineering from Anna University Chennai, India. Her areas of interest are Mobile Computing, Software Engineering and Computer Networks. she has presented many papers in national conferences in various fields. As part of this paper, she is working on finding approximate string search in spatial databases—When users have an fuzzy search condition for searching in spatial databases.She also describes to increase the selectivity estimation of spatial databases.