ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Range Query Grouping In Spatial Networks Using Dual Distance Measurements

S.Udhayakumar M.E., (CSE) [1], D.Sureshkumar M.E., (CSE) [2]

Hindusthan College of Engineering and Technology, Coimbatore[1, 2]

ABSTRACT: Location-based services and the abundant usage of smart phones and GPS-enabled devices, the necessity of outsourcing spatial data has grown rapidly.It deals with the approximate string search in large spatial databases.Specifically, this investigate range queries augmented with a string similarity search predicate in both Euclidean space and road networks. Euclidean space is ordinary two- or three-dimensional space. These called as the spatial approximate string (SAS) query. In Euclidean space, this 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 subtree of u. This analyzes the pruning functionality of such signatures based on the set resemblance between the query string and the q-grams from the subtrees of index nodes. It also discusses how to estimate the selectivity of a SAS query in Euclidean space.Presented a novel adaptive algorithm to find balanced partitions using both the spatial and string information stored in the tree. For queries on road networks, we propose a novel exact method, RSASSOL, which significantly outperforms the baseline algorithm which is serving in basis, as for measurement, calculation or location in practice.The RSASSOL combines the q-gram-based inverted lists and the reference nodes based pruning.

I. INTRODUCTION

An increasing number of applications require the efficient execution of nearest neighbor (NN) queries constrained by the properties of the spatial objects. Due to the popularity of keyword search, particularly on the Internet, many of these applications allow the user to provide a list of keywords that the spatial objects (henceforth referred to simply as objects) should contain, in their description or other attribute.

A spatial keyword query consists of a query area and a set of keywords. The answer is a list of objects ranked according to a combination of their distance to the query area and the relevance of their text description to the query keywords.

The proposed system deals the spatial approximation string search based on the Euclidean space and road space. *k*-NEAREST NEIGHBORS ALGORITHM (*k*-NN) is a non-parametric method for classification and regression, that predicts objects' "values" or class memberships based on the *k*closest training examples in the feature space. *k*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small).

II. EXISTING SYSTEM

- The IR2-tree was proposed to perform exact keyword search with kNN queries in spatial databases.
- Many existing system studied the m-closest keywords query in Euclidean space,

- Two other relevant studies concentrates on ranking queries that combine both the spatial and text relevance to the query object.
- The LBAKtree was proposed to answer location-based approximate keyword queries.
- There was many similarity functions have been proposed to quantify the closeness between two strings.
- Several techniques have been proposed for identifying candidate strings within a small edit distance from a query string.

DRAWBACKS

- The IR2-tree cannot support spatial approximate string searches.
- IR2-Tree is that it only supports exact keyword search.
- BR_-tree failed to handle approximate string search.
- Existing string solution suffers the same scalability and performance issues.
- Query optimization problem.
- R trees suffers from high IO cost and communication overhead.
- Only considered Euclidean space or road space.

PROBLEM

The problem in the paper is different: these want to search in a collection (unordered set) of strings to find those similar to a single query string ("selection query").Selectivity estimation of range queries on road networks is a much harder problem than its counterpart in the Euclidean space.Different points may contain duplicate strings.

METHODOLOGY

- SAS queries in Euclidean space as (ESAS) queries.
- SAS queries to road networks (referred as RSAS queries).
    - o For RSAS queries, the baseline spatial solution is based on the Dijkstra's algorithm
- Edit distance concept has been introduced for Euclidean space.

III. PROPOSED SYSTEM

This work presents a novel index structure, MHR tree, for efficiently answering approximate string match queries in large spatial databases. The MHR-tree is based on the R-tree augmented with the min-wise signature and the linear hashing technique.
Introduces a new index for answering ESAS queries efficiently which embeds min-wise signatures of q-grams from subtrees into the R-tree nodes.The RSASSOL method partitions the road network, adaptively searches relevant subgraphs, and prunes candidate points using both the string matching index and the spatial reference nodes. Lastly, an adapted multipoints ALT algorithm (MPALT) is applied, together with the exact edit distances, to verify the final set of candidates.

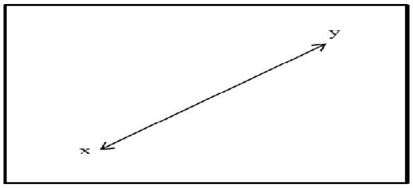
- Examining spatial approximate substring queries.
- This examines query selectivity estimation for queries in the Euclidean space.

ADVANTAGES

- Examining spatial approximate substring queries.

- Introduces a new index for answering ESAS queries efficiently which embeds min-wise signatures of q-grams from subtrees into the R-tree nodes.
- This examines query selectivity estimation for queries in the Euclidean space.

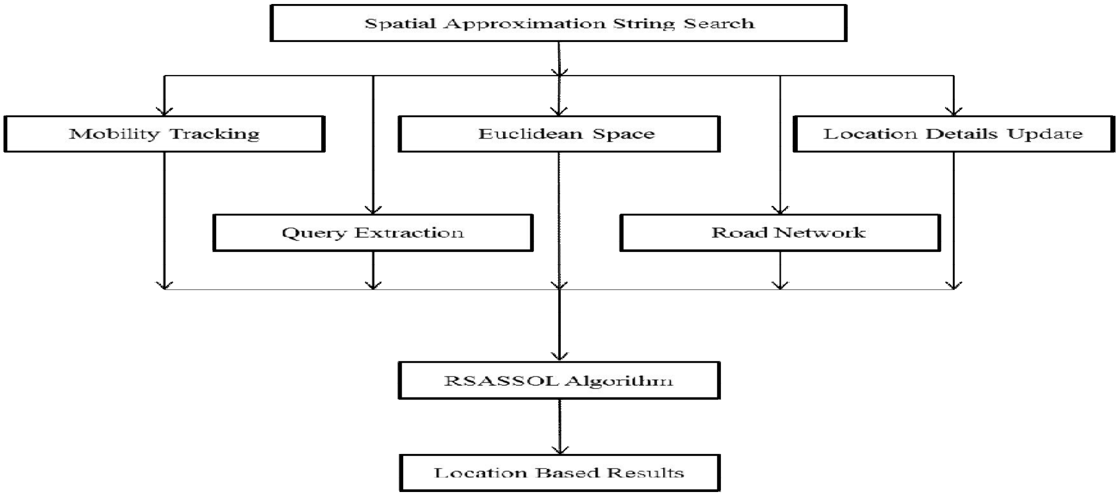
THE DEFINITION OF A EUCLIDEAN SPACE



Euclidean

Euclidean spaceis a real vector space on which is defined a fixed symmetric bilinear form whose associated quadratic form is positive definite. The vector space itself will be denoted as a rule by L, and the fixed symmetric bilinear form will be denoted by *(x, y)*. Such an expression is also called the *inner product* of the vectors *x* and *y*. Let us now reformulate the definition of a Euclidean space using this terminology.

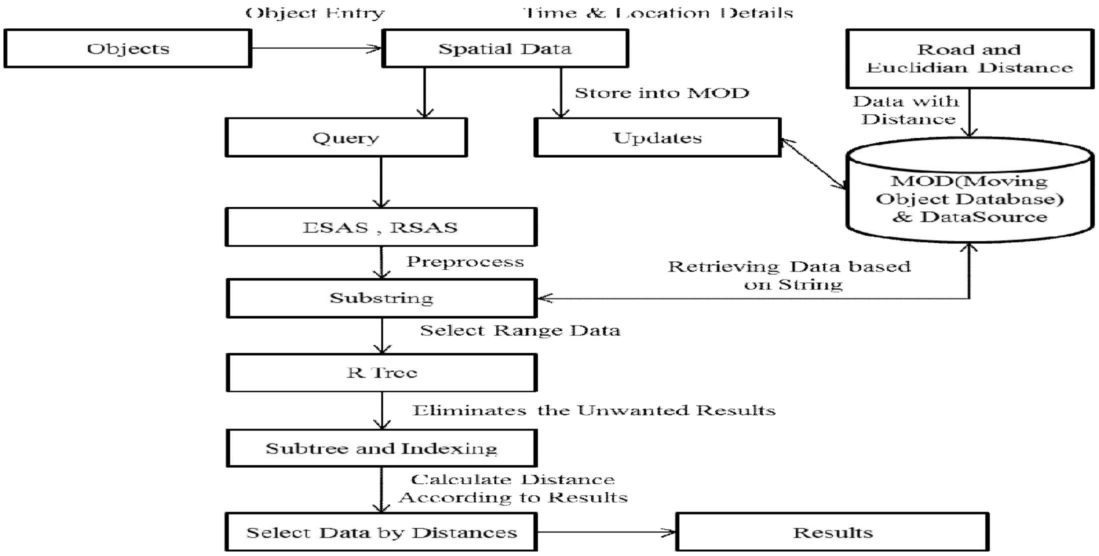
IV. ARCHITECTURE DIAGRAM



IMPLEMENTATION OF R-TREES

- R-Trees can organize any-dimensional data by representing the data by a minimum bounding box.
- Each node bounds it's children. A node can have many objects in it
- The leaves point to the actual objects (stored on disk probably)
- The height is always log n (it is height balanced)

OVERALL DIAGRAM



RSASSOL ALGORITHM

RSASSOL(network G,Q_r=(q,r),Q_s=(σ, □))

- /* Step 1: find subgraphs intersecting the query range*/

Find the set X of ids of all subgraphs intersecting(q,r);

Set A=0,Ac=0;

for each subgraph id i € X do

- /*Step 2: use the Filter Trees to retrieve points with strings that are potentially similar to the query string*/

Find all point ids in G_i whose associated strings σ' may satisfy €(σ',σ)<=□ using Filter Tree$_i$, and insert them into A_c;

- /*Step 3: prune away points by calculating the lower and upper bounds of their distances to the Query point using V_R*/

for every point p_i € A_c do

calculate $d^+(p_i , q)$ and $d^-(p_i , q)$ as discussed;

```
if $d^+(p_i , q)$<= r then
if €( $\sigma_i$ ,$\sigma$) <= r then
move $p_i$ from $A_c$ to A;
• /* Step 4: prune points using the exact edit distance between the query string and the candidate string */
else
delete $p_i$ from $A_c$ to A;
else
if $d^-$ ($p_i$ ,q)> r then
delete $p_i$ from $A_c$;
/* Step 4: same pruning as the step 4 above */
for every point $p_i$ € $A_c$ do
if €( $\sigma_i$ ,$\sigma$) > r then
delete $p_i$ from $A_c$;
• /* Step 5: check the exact distances of the remaining candidate points to the query point */
Use the MPALT algorithm to find all points p's in $A_c$ such that(p,q)<=r, push them to A;
Return A.
```

DESCRIPTION OF THE ALGORITHM:

RSAS query framework consists of five steps . Given a query, we first find all sub graphs that intersect with the query range. Next, we use the Filter Trees of these sub graphs to retrieve the points whose strings are potentially similar to the query string. In the third step, we prune away some of these candidate points by calculating the lower and upper bounds of their distances to the query point, using VR. 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. After this step, the string predicate has been fully explored. In the final step, for the remaining candidate points, we check their exact distances to the query point and return those with distances within r. We dub this algorithm RSASSOL and the rest of this section presents the details of this algorithm.

MODULES DESCRIPTION

Data Station Broadcast Launch

The Data Station Broadcast Launch module Used to start the Server. Its enable the Server Ip Address and port address to listening the client nodes. It also lists the active nodes currently in the Client side. Here the Admin only has the privileges to start the server.The server can add service details such as name, location and other contact details. Here the admin can click on the map to get the exact point of the area, which they need to add into database. Creation of spatial datasets dynamically may help the user to get the updated information from the service. The first module contains the appropriate service registration process.

Mobile Host Creation

In this module the Admin going to enter the mobile host details such as Mobile Host Id, Mobile Host Name, Password and Location .It will used to connect the clients. In later These details are used to connect the appropriate clients. The mobile host is the end user of spatial string search. The user can register their selves and can login into the system. This process helps to track the user interest and searching history from the server. The host id is an unique id, which has been created using random function. The system will automatically provides an id when registration process in progress.

Location Details Update

In this module Admin update the Category, Category name, Address and location. These detail service are fully depends on the clients. Through this details other clients are going to search the location based queries.

This module rank tuples using an aggregate score function on their attribute values. In this module we will defines the top-k spatial preference query problem and describes the index structures for the datasets.

Sending Mobile HostQueries

The sending mobile host SAS module used to send the Location based spatial query to the Server. The Clients are the privileged person to send the Query to the server. They get the Response based on the Locations .These details are resided in the spatial database of the server.

- If user enters a query first it checks with all databases.
- The query available in any of the above databases, the query and count are updated in the corresponding databases.
- If user enters the same query, It will be stored in Cache memory.
- Similarly, in order to cache and reuse the intermediate results among different queries, we utilize the materialized views in databases for Results.

Retrieving SAS and Data Filtering

In this module the client node search the nearest nodes to get the response if that node contains the particular query it will response to the corresponding client otherwise the query forwarded to the server. Then the server Filter details based on the query and those details are sending to the client.

Query Processing

This module focuses on range queries and dubs such queries as Spatial Approximate String (SAS) queries.

The RSASSOL method partitions the road network, adaptively searches relevant subgraphs, and prunes candidate points using both the string matching index and the spatial reference nodes. Lastly, an adapted multipoints ALT algorithm (MPALT) is applied, together with the exact edit distances, to verify the final set of candidates.

ALT is a well known preprocessing-based speed-up technique for Dijkstra's algorithm, which allows fast computations of shortest paths in large scale road networks. The preprocessing of the ALT algorithm has some degree of freedom, in that it must select a subset of nodes in the graph, called landmarks, that fulfill a special role

V. CONCLUSION

This paper presents a comprehensive study for spatial approximate string queries in both the euclidean space and networks. We use the edit distance as the similaritymeasurement for the string predicate and focus on the range queries as the spatial predicate. We also address the problem of query selectivity estimation for queries inthe euclidean space. Future work includes examining spatial approximate substring queries, designing methods that are more update friendly, and solving the selectivityestimation problem for RSAS queries.

REFERNECES

[1] Feifei Li, Member, IEEE, Bin Yao, Mingwang Tang, and Marios Hadjieleftheriou, "Spatial Approximate String Search", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 6, JUNE 2013.

[2] S. Acharya, V. Poosala, and S. Ramaswamy, "Selectivity Estimation in Spatial Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 13-24, 1999.

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

[4] A. Arasu, S. Chaudhuri, K. Ganjam, and R. Kaushik, "Incorporating String Transformations in Record Matching," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 1231-1234, 2008.

[5] A. Arasu, V. Ganti, and R. Kaushik, "Efficient Exact Set-Similarity Joins," Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB), pp. 918- 929, 2006.