International Journal of Multidisciplinary and Current Research

Review Article

ISSN: 2321-3124
Available at: http://ijmcr.com

Review of Approximate String Search in Spatial Dataset

Pratiksha Nikam[a] and Prof. Srinu Dharavath[b]

[a]Lecturer,GSMIT, Pune, [b]Professor, GSMCOE,Pune

Accepted 20 April 2014, Available online 25 April 2014, **Vol.2 (March/April 2014 issue)**

Abstract

Many research articles and methods presented over the problem of approximate string search, however most this methods suffered from the accuracy and speed. Our work deals with the approximate string search in large spatial databases. Especially, this paper investigates range queries augmented with a string similarity search predicate in both euclidean space and road networks. This query is called spatial approximate string (SAS) query . This paper presents a study for spatial approximate string queries in both the euclidean space and road networks. For ESAS query Spatial solution is done using MHR-tree , which embeds min-wise signatures into an R-tree in Euclidean space. For RSAS query, spatial solution is done using RSASSOL algorithm..

Keywords: *Approximate string search etc.*

Introduction

Text data is ubiquitous. Managing such string data in databases and information systems has taken on particular importance recently.

Approximate String Search :- Suppose given a collection of strings, we have to efficiently find those string in the collection that are similar to a query string. Such type of query is called an "approximate string search." This kind of problem is of great interests to a variety of application s like,

- *Spell Checker :-*, a spell checker needs to find words similar to the words by searching in its dictionary those are possibly mistyped words. Thus, for each word that is not in the dictionary, we need to find potentially matched candidates to recommend.
- *Data Cleaning:-* Information from different data sources often has various inconsistencies. There is slightly different formats for the same real-world entity could be represented in. There could also be errors in the original data introduced in the data collection process. So, data cleaning needs to find from a collection of entities those similar to a given entity.

We have studied many research articles and methods presented over the problem of approximate string search, however most this methods suffered from the accuracy and speed. There are two fundamental problems in research on approximate string search:

(1) How to build a model that can archive both high accuracy and efficiency, and

(2) How to develop a data structure and algorithm that can facilitate efficient retrieval of the top *k* candidates.

Spatial Approximate String Queries :- In this particular research we are focusing to work over range queries and dub such queries as Spatial Approximate String (SAS) queries.

ESAS Queries:- There are many solutions presented to answer SAS queries efficiently and fast. We denote SAS queries in Euclidean space as (ESAS) queries. A key issue in SAS queries is to define the similarity between two strings. The edit distance metric is often adopted for the same.

In the euclidean space, we can instantiate the spatial solution using R-trees. While being simple, this R-tree solution could suffer from unnecessary node visits (higher IO cost) and string similarity comparisons (higher CPU cost).In R-tree Pruning power from string match predicate has been completely ignored.

RSAS Queries:-. We denote SAS queries on Road network as (RSAS) queries. On Road network, spatial solution is done using dijkastra's algorithm, but performance degrades quickly when data on network increases. There is no method for selectivity estimator of RSAS queries.

Literature Survey

IR^2-Tree :- The IR^2-Tree (Information Retrieval R Tree)[2] is an R Tree where a signature is added to each node v of the IR^2 Tree to denote the textual content of all spatial objects in the sub tree rooted at v. Felipe et al. used IR^2 – Tree to spatial databases [1], where keyword search becomes a fundamental building block for an increasing number of practical, real-world applications. A major limitation of the IR^2-Tree is that it only supports exact keyword search with kNN queries in spatial databases. The IR2-tree cannot support spatial approximate string searches, neither their selectivity estimation was addressed there in.

MCK:- MCK**,** a novel spatial keyword query called the m-closest keywords (mCK) query. Suppose database of spatial objects is given, each tuple is associated with some descriptive information represented in the form of keywords. The main aim of mCK query is to find the spatially closest tuples which match m user-specified keywords. Given a set of keywords from a document, mCK query can be very useful for comparing the keywords of geotagging document to other geotagged documents in a database. To answer mCK queries efficiently, we introduce a new index called the bR*-tree, which is an extension of the R*-tree. Based on bR*-tree, we exploit a priori-based search strategies to effectively reduce the search space. Authors in [3] study the m-closest keywords query in Euclidean space, where the proposed bR tree cannot handle the approximate string search.

Two other relevant studies appear in [4], [5] where ranking queries that combine both the spatial and text relevance to the query object was investigated.

Approximate string search alone has been extensively studied in the literature These works generally assume a similarity function to quantify the closeness between two strings. There are a variety of these functions such as edit distance and Jaccord. Many approaches leverage the concept of q-grams. Our main pruning lemma is based upon a direct extension of q-gram-based pruning for edit distance that has been used extensively in the field [6], [7], [8], [9]. Improvements to the q-grams-based pruning has also been proposed, such as v-grams [10], where instead of having a fixed length for all grams variable length grams were introduced, or the two-level q-gram inverted index [11].

MHR Tree

Rather than using R-tree here we are using MHR-Tree for ESAS queries rather. Suppose the disk block size is B. The R-tree and its variants (e.g., R_-tree) share a similar principle. They first group B points that are in spatial proximity into a minimum bounding rectangle (MBR), which is stored in a leaf node. When all points in P are assigned into MBRs, the resulting leaf node MBRs are then further grouped together recursively till there is only one MBR left. Each node u in the R-tree is associated with the MBR enclosing all the points stored in its sub tree, denoted MHR Tree. Each internal node also stores the MBRs of all its children.

MHR-Tree Construction

For a leaf level node u, let the set of points contained in u be up. We compute its q-grams gp for every point p in up, and the corresponding min-wise signature s(gp). Note that in order to compute the min-wise signature s(gp) for the node u, we do not need to compute gu (which can be costly, space wise). The QUERY-MHR algorithms for the MHR-tree generally follow the same principles as the corresponding algorithms for the spatial query component. We would like to incorporate the pruning method based on q-grams without the explicit knowledge of gu for a given Rtree node u. We need to achieve this with the help of s(gp).

Building Inverted List

There are many selectivity estimators for approximate string matching, none though in combination with spatial predicates. Various techniques have been proposed specifically for edit distance [13], [14], [15]. A state-of-the-art technique based on q-grams and min-wise signatures is VSol [15]. It builds inverted lists with q-grams as keys and string ids as values; one list per distinct q-gram in input strings. Each list is summarized using the min-wise signature of the string ids in the list. VSol selectivity estimator for a data set P id the collection on min-wise signatures and their corresponding q-grams (one signature per distinct q-gram).

Algorithm for RSAS Queries

For RSAS queries rather than using Dijkstra's algorithm here we are using RSASSOL algorithm. We partition a road network G={V,E} into m edge disjoint sub graphs G1,G2, . . .,Gm. where m is a user parameter, and build one string index (FilterTree) for strings in each sub graph. We also select a small subset VR of nodes from V as reference nodes: they are used to prune candidate points/nodes whose distances to the query point q are out of the query range r. Conceptually, our RSAS query framework consists of five steps:-

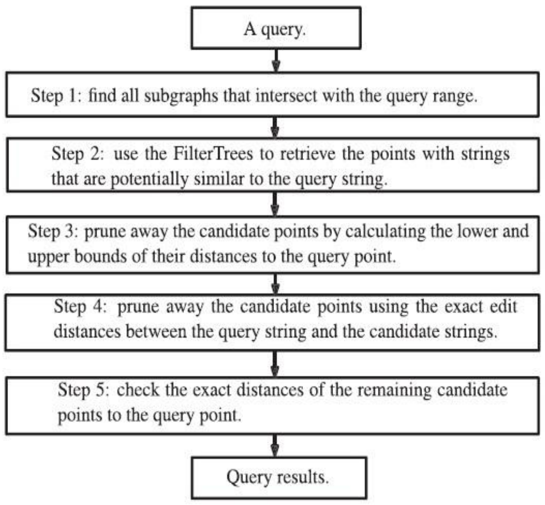
A query.
Step 1: find all subgraphs that intersect with the query range.
Step 2: use the FilterTrees to retrieve the points with strings that are potentially similar to the query string.
Step 3: prune away the candidate points by calculating the lower and upper bounds of their distances to the query point.
Step 4: prune away the candidate points using the exact edit distances between the query string and the candidate strings.
Step 5: check the exact distances of the remaining candidate points to the query point.
Query results.

Fig Overview of RSASSOL algorithm

Selectivity estimation for RSAS queries:- Selectivity estimation of range queries on road networks is a much harder problem than its counterpart in the Euclidean space. Several methods were proposed in [12]. However, they are only able to estimate the number of nodes and edges in the range. None can be efficiently adapted to estimate the number of points in the range.

Conclusion

This paper gives the review of Approximate String Search in Spatial Dataset. . Approximate string matching can be generally define as searching for substrings of a text that are within a predefined edit distance threshold from a given pattern. We presented here previous method for string search as well as Spatial Approximate string queries in both euclidean space and road network.

Future work

There is not any efficient method to estimate the number of points in the spatial range on road network. So our future work includes solving the selectivity estimation for RSAS queries on road network.

References

[1] I. D. Felipe, V. Hristidis, and N. Rishe, "Keyword searc h on spatial databases," in ICDE , 2008.
[2] Christos Faloutsos, Stavros Christodoulakis: Signature Files: An Access Method for Documents and Its Analytical Performance Evaluation. In ACM Trans. Inf.Syst.2(4):267 288(1984).
[3] D. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, and M. Kitsuregawa, "Keyword Search in Spatial Databases: Towards Searching By Document," Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE), pp. 688-699, 2009.
[4] X. Cao, G. Cong, and C.S. Jensen, "Retrieving Top-k Prestige Based Relevant Spatial Web Objects," Proc. VLDB Endowment, vol. 3, pp. 373-384, 2010.
[5] G. Cong, C.S. Jensen, and D. Wu, "Efficient Retrieval of the Top-K Most Relevant Spatial Web Objects," Proc. VLDB Endowment, vol. 2, no. 1, pp. 337-348, 2009.
[6] E. Sutinen and J. Tarhio, "On Using Q-Gram Locations in Approximate String Matching," Proc. Third Ann. European Symp.Algorithms (ESA), pp. 327-340, 1995.
[7] E. Tiakas, A.N. Papadopoulos, A. Nanopoulos, and Y. Manolopoulos, "Node and Edge Selectivity Estimation for Range Queries in Spatial Networks," Information Systems, vol. 34, pp. 328-352, 2009.
[8] E. Ukkonen, "Approximate String-matching with Q-Grams and Maximal Matches," Theoretical Computer Science, vol. 92, no. 1, pp. 191-211, 1992.
[9] X. Yang, B. Wang, and C. Li, "Cost-Based Variable-Length-Gram Selection for String Collections to Support Approximate Queries Efficiently," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 353-364, 2008.
[10] X. Yang, B. Wang, and C. Li, "Cost-Based Variable-Length-Gram Selection for String Collections to Support Approximate Queries Efficiently," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 353-364, 2008.
[11] M.-S. Kim, K.-Y.Whang, J.-G.Lee, and M.-J. Lee, "N-Gram/2l: A Space and Time Efficient Two-Level N-Gram Inverted Index Structure," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 325-336, 2005.
[12] E. Tiakas, A.N. Papadopoulos, A. Nanopoulos, and Y. Manolopoulos, "Node and Edge Selectivity Estimation for Range Queries in Spatial Networks," Information Systems, vol. 34, pp. 328-352, 2009.
[13] L. Jin, C. Li, and R. Vernica, "Sepia: Estimating Selectivities of Approximate String Predicates in Large Databases," The VLDB J., vol. 17, no. 5, pp. 1213-1229, 2008.
[14] H. Lee, R.T. Ng, and K. Shim, "Extending Q-Grams to Estimate Selectivity of String Matching With Low Edit Distance," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 195-206, 2007.
[15] A. Mazeika, M.H. Bo¨hlen, N. Koudas, and D. Srivastava, "Estimating the Selectivity of Approximate String Queries," ACM Trans. Database Systems, vol. 32, no. 2, pp. 12-52, 2007.