ISSN No. 2395-0900
International Journal of Engineering Studies and Technical Approach
IJESTA

Survey of Efficient and Fast Nearest Neighbor Search For Spatial Query on Multidimensional Data

Ms. Pranati Waghodekar* & Prof. (Ms) Kavita Bhosle**

***Research Scholar, GSM's Maharashtra Institute of Technology, Aurangabad,**

****Professor & Head, Department of Computer Science &Technology, GSM's Maharashtra Institute of Technology, Aurangabad.**

ABSTRACT

Spatial data mining is a special kind of data mining. Patterns, clusters, classifications, etc., can be derived from the big data available. Especially, nearest neighbor search approach with respect to a query point plays a key role in arriving at the final decision making. Like Computer Integrated Manufacturing, Facility Layout, Cellular Manufacturing, nearest neighbor search has been found several applications in searching the nearest hospitals, restaurants, jogging parks, wedding halls, cinema theaters, schools, etc. This paper presents a brief literature review of efficient and fast nearest neighbor search. The older approach is banked upon IR^2 –Tree that usually follows two strategies: R Tree and Signature files. But during the last couple of years, several research papers have been published for fast and efficient nearest neighbor search (FNN) optimizing space; accuracy for handling geometric properties and documents, etc, SI-Index is one of the latest techniques that deal efficiently with multidimensional large scale problems in real time. This paper, therefore, will find interesting to all concerned.

Keywords: Nearest Neighbor, IR^2 -Tree, Spatial Geometric Dimensions, Signature file, Text, SI-Index.

INTRODUCTION

Data mining is concerned with the process of analyzing data available or generated from various sources and it's bunching into some useful information to be used to attain some predetermined goals like increase revenue, cuts costs or some other useful end purpose [1]. Data mining in a way is the process of finding correlations or patterns among dozens of fields in large relational databases. Data are any facts, numbers, or text usually subjected to a computer processing. In today's information era, organizations do generate big data in growing amounts in different formats and databases like operational or transactional data, nonoperational data, and macro economic data or Meta data - data about the data itself-, etc. For instance, big data is available in Computer Integrated Manufacturing, Facility Layout Planning, social media, health care, tourism, sales, etc., that can be utilized for arriving at a better/optimum decision making. The patterns, associations, or relationships among all this data can provide information that can be converted into knowledge about historical patterns and future trends, helping organization for optimum decision making on real-time basis.

Unprecedented advances in data capture, processing power, data transmission, and storage capabilities are enabling organizations in integrating their various databases into data warehouses. Data warehousing, therefore, is defined as a process of centralized data

www.ijesta.com

management and retrieval. For instance, Wal-Mart is pioneering massive data mining to transform its supplier relationships capturing point-of-sale transactions from over 2,900 stores in 6 countries and continuously transmits this data to its massive 7.5 terabyte data warehouse. While large-scale information technology has been evolving separate transaction and analytic systems, data mining provides the link between the two.

Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Generally, any of four types of relationships are sought: Classes, Clusters, Associations and Sequential patterns. Data mining consists of five major elements:

- Extract, transform, and load transaction data onto the data warehouse system.
- Store and manage the data in a multidimensional database system.
- Provide data access to business analysts and information technology professionals.
- Analyze the data by application software.
- Present the data in a useful format, such as a graph or table.

Data mining applications today are available on all size systems for mainframe, client/server, and PC platforms. Enterprise-wide applications generally range in size from 10 gigabytes to over 11 terabytes. NCR has the capacity to deliver applications exceeding 100 terabytes. This calls for two critical technological drivers:

- **Size of the database**: the more data being processed and maintained, the more powerful the system required.
- **Query complexity**: the more complex the queries and the greater the number of queries being processed, the more powerful the system required.

Relational database storage and management technology is adequate for many data mining applications less than 50 gigabytes. However, this infrastructure needs to be significantly enhanced to support larger applications. Some vendors have added extensive indexing capabilities to improve query performance. Others use new hardware architectures such as Massively Parallel Processors (MPP) to achieve order-of-magnitude improvements in query time. For example, MPP systems from NCR link hundreds of high-speed Pentium processors to achieve performance levels exceeding those of the largest supercomputers.

For large data, different levels of analysis like Artificial neural networks, Genetic algorithm, Decision trees, Rule induction, and Data visualization are available. Another significant tool for data analysis is the Nearest Neighbor (NN) method: a technique that classifies each record in a data set based on a combination of the classes of the *k* record(s) most similar to it in a historical data set. Sometimes it is termed the *k*-nearest neighbor technique and found to be very handy tool in such situations as traveling salesman problem. This involves:

- : The extraction of useful if-then rules from data based on statistical significance.
- The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

The following Section presents a literature survey in the area of fast nearest neighbor search method.

www.ijesta.com

ISSN No. 2395-0900
IJESTA International Journal of Engineering Studies and Technical Approach

THE LITERATURE REVIEW

In situations like Facility Layout Planning, locating nearby hospitals or restaurants, traveling salesman problem, etc., one searches for the nearest neighbor for minimizing both travel cost and time. For achieving this, researchers, based on such criteria as similarity, closeness of relationship, distance, etc, have introduced a technique called "Nearest Neighbor (NN)". This Section is an attempt to present in brief a literature survey of NN in respect of such matters as evolution, growth, algorithms, etc. Emergence of computer in mid-1970 has opened up new avenues further that are found to be quite capable of handling big data in real time.

Anjum and Saktel [2] have presented a survey of such various techniques as collective spatial keyword query, the combined notion of keyword search with reverse nearest neighbor query, hybrid indexing structure bR*-tree, efficient method to answer top-k spatial keyword query, computing the relevance between the documents of an object and a query, spatial inverted index, etc., for nearest neighbor search for spatial database, The authors have proposed, to overcome the drawbacks of previous methods, like, expensive space consumption, unable to give real time answer, etc., a new method based on variant of inverted index and R-tree and algorithm of minimum bounding method to reduce the search space. The approach accommodates a query with both spatial data and associated text. For this type of query, a variant of inverted index is used that is effective for multidimensional points and comes with an R-tree built on every inverted list, and uses the algorithm of minimum bounding method that can answer the nearest neighbor queries with keywords in real time.

Yufei Tao and Cheng Sheng [3] have proposed a variant of inverted index that optimizes multidimensional points called the spatial inverted index (SI-index). IR^2 -tree inherits a drawback of signature files: file hits, e.g., a signature file by virtue of its conservative nature may still direct to search to some objects, though they do not posses all the keywords. Thus, only in full text description, it becomes possible to verify if a query is satisfied or not. False hit problem may exist in other methods too for approximate set membership tests with compact storage. The SI-index approach successfully incorporates point's coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. This is achieved in two competing ways for query processing, namely, sequentially merge multiple lists very much like merging traditional inverted lists by ids or alternatively, leverage the R-tree to browse the points of all relevant lists in descending order of their distance to the query point. It is claimed that SI-index significantly outperforms the IR^2 -tree in query efficiency usually by a factor of magnitude.

C Usha Rani and N Munisankar [4] have developed a new access method called Spatial Inverted Index that extends the conventional inverted index meeting with multidimensional data and comes with keywords in real time. The authors have proposed the following SI-index algorithm:

Input: Query, Cache Queries

Output: Result set generated for query

Procedure

If Query available in cache, Result related to query is:

Forward to Tree Process (Query)

www.ijesta.com

Else,

Result related to query = Gecoding Tree Process (Query)

Geocoding Tree Process (Query)
Parameters:

Qi: Input spatial query

Qj (j=1,2,....n): A set of queries containing same location

Distj (j=1,2,...n): Array for set of distances

Procedure:

(Xi,Yi): Geocoding of Qi

(Xj,Yi): Geocoding of all queries with respect to location

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 features and distance

Return list

Forward to Tree Process ()

1. Build an empty list
2. Make a root node
3. If Qi in cache and status ==false

    For j= 0 to

    Compare features(Qi,Qj) status==true

    For Each child in tree

    If (status==true)

    Geocodebyfeature (Qi);

    Geocodebyfeature (Qj);

    End

    Else

    Empty list ()

    End For Each

4. Add nodes to list
5. **Return list**

www.ijesta.com

The authors have claimed through experimentation that the proposed algorithm yields the optimal results compared to traditional approaches.

Chandrashekhar [5] has presented some features of data mining like fast analysis, complex data set, etc., and also has briefly presented data mining applications, operations, techniques and algorithms. The author has presented three issues, namely, queries focus on object's geometric properties, modern applications call for geometric coordinates and their associated texts and that major approaches being straightforward fail to provide real time answers on difficult inputs. The author has proposed the spatial inverted index that extends the standard inverted index to address multidimensional information and claims that the algorithm answers the nearest neighbor query with key words in real time. The system design proposed by the author consists of four modules, namely, Location Manager, HTTP Communicator, Spatial Inverted Index and Spatial Data Display.

Sophiya and Sounderrajan [6] present that spatial data mining is a special kind of data mining. On reviewing the existing system having two strategies, i.e., R trees and signature files, the authors have proposed SI-index and enhanced search. This approach improves both space and query efficiency.

Rajkumar R, et al, [7] have proposed a system using the Euclidean distance for large scale computer vision problems. The data points are embedded nonlinearly onto a low-dimensional space by simple computations proving that the distance between two points in the embedded space is bounded by the distance in the original space. The proposed algorithm is claimed to be well suited for high-dimensional and large-scale problems because a lot of candidates are rejected based on distances in the low-dimensional embedded space. The algorithm is further improved by partitioning input vectors recursively.

Sonal and Bongale [8] have presented in their literature survey that spatial index is used for creating indices, the inverted index data structure is a central component of a typical search engine indexing algorithm and nearest neighbor search identifies as closeness search, parallel search optimizing closest points in metric space. The authors also have presented a comparative study of the Depth first search and inverted index shown in Table 1.

Table 1: Comparison of Depth first search and inverted index [8].

Parameters/Methods	Depth first search	Inverted index
Space	More	Less
Time complexity	O(n+m)	O(n)
Working	Slow	Efficient

Nitin and Vandana [9] have presented a survey of Nearest Neighbor Techniques that is presented in Table 2.

www.ijesta.com

ISSN No. 2395-0900
IJESTA International Journal of Engineering Studies and Technical Approach

Table 2: Comparison of Nearest Neighbor Techniques. [9].

No.	Technique	Key Idea	Advantages	Disadvantages	Target Data
1	k Nearest Neighbor (kNN)	Uses nearest neighbor rule	Training very fast, simple &easy to learn, robust to noisy training data and effective if mining data large.	Biased for value k, computational complexity, memory limitation, runs slowly, easily fooled by irrelevant attributes.	Large data sample.
2	Weighted k Nearest Neighbor (kNN)	Assign weights to neighbors as per distance calculated.	Overcomes limitations of kNN of assigning equal weight to k neighbors implicitly, use all training samples not just k, algorithm global one.	Computation complexity increases in calculating weights, algorithm runs slow.	Large sample data
3	Condensed Nearest Neighbor (CNN)	Eliminate data sets that show similarity and do not add any extra information.	Reduce size of training data, improves query time& memory requirements, reduce the recognition rate.	CNN order dependent, unlikely to pick up points on boundary and computational complexity.	Data set where memory requirement is main criteria.
4	Reduced Nearest Neighbor (RNN)	Remove patterns which do not affect the training data set results.	Reduce size of training data and eliminate templates, improve query time and memory requirements, reduce the recognition rate.	Computational complexity, high cost, time consuming.	Large data set.
5	Model based k Nearest neighbor (MkNN)	Model constructed from data & classifies new data, using model.	More classification accuracy, value of k selected automatically, high efficiency as reduces number of data points.	Do not consider marginal data outside the region.	Dynamic web mining for large repository.
6	Rank Nearest Neighbor (RkNN)	Assign ranks to training data for each category.	Perform better when there are too many variations between features, robust as based on rank, less computational complexity as compared to kNN.	Multivariate kRNN depends on distribution of data.	Class distribution of Gaussian nature.

www.ijesta.com

ISSN No. 2395-0900
IJESTA International Journal of Engineering Studies and Technical Approach

7	Modified k nearest neighbor (MkNN)	Use weights & validity of data point to class-ify nearest neighbor.	Partially overcome low accuracy of WkNN, stable and robust.	Computational complexity.	Methods facing outlets.
8	Pseudo/Generalized Nearest Neighbor (GNN)	Utilizes information of (n-1) neighbors also instead of that of the nearest neighbor only.	Uses (n-1) classes that consider the whole training data set.	Does not hold good for small data, computational complexity.	Large data set.
9	Clustered k Nearest Neighbor	Clusters formed to select nearest neighbor.	Overcome defects of uneven distribution of training samples, robust in nature.	Selection of threshold parameter difficult before running algorithm, biased by value of k for clustering.	Text classify-cation.
10	Ball Tree k Nearest Neighbor (kNSI)	Uses ball tree structure to improve kNN speed.	Tune well to structure of represented data, deal well with high dimensional entities, easy to implement.	Costly insertion algorithm, as distance increases KNSI degrades.	Geometric Learning task like robotics, vision, speech, graphics.
11	k-d tree Nearest Neighbor (kdNN)	Divide the training data exactly into half plane.	Produce perfectly balanced tree, fast and simple.	More computation, require intensive search, blindly slice points into half that may miss data structure.	Organizati on of multi dimension al points.
12	Nearest Feature Line Neighbor (NFL)	Take advantage of multiple templates per class.	Improve classification accuracy, highly effective for small size; utilize information ignored in nearest neighbor, i.e., templates per class.	Fail when prototype in NFL far away from query point, computational complexity, to describe features points by straight line hard task.	Face recognitio n problems.

www.ijesta.com

ISSN No. 2395-0900
International Journal of Engineering Studies and Technical Approach
IJESTA

13	Local Nearest Neighbor	Focus on nearest neighbor prototype of query.	Cover limitations of NFL.	Number of computation.	Face recognition.
14	Tunable Nearest Neighbor (TNN)	A tunable metric used.	Effective for small data sets.	Large number of computation.	Discrimination problems.
15	Center based Nearest Neighbor (CNN)	A center line calculated.	Highly efficient for small data sets.	Large number of computation	Pattern recognition.
16	Principal Axis Tree Nearest Neighbor (PAT)	Uses PAT.	Good performance, fast search.	Computation time.	Pattern recognition
17	Orthogonal Search Tree Nearest Neighbor	Uses orthogonal trees.	Less computation time, effective for large data sets.	Query time more.	Pattern recognition

Yufei Tao's contribution to the area of the Nearest Neighbor Search over the past 15 years has been commendable. References at Sr. No. 10 to 18 are representative of Yufei's work and included in the Reference Section as a part of bibliography.

CONCLUSIONS

Spatial data mining is a special type of data mining. Geometric properties along with text play a key role in meeting a query for searching the nearest neighbor, like, the nearest vegetarian restaurant with vegetarian food with Paratha, rice, dal and Curd. In this paper, a literature survey is presented of the recent research work for fast, efficient, accurate, large-scale problems with multi-dimensional points with respect to the Geo-coordinates of a query point. The paper brings out the special features of the latest SI-Index approach with documents or signature files. An exhaustive list of Nearest Neighbor Search as reported is provided. The paper, it is believed, will of interest to all concerned.

www.ijesta.com

REFERENCES

i. http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm

ii. Anjum Zareen and Saktel Priti, 2014, Survey on Nearest Neighbor Search for Spatial Database, International Journal of computer Science and Information Technologies, Vol. 5, No. 6, pp. 7101-7103.

iii. Yufei Tao and Cheng Sheng, 2014 , Fast Nearest Neighbor Search with Keywords, IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 4, April,0000-0000/008800.00 (c) 20XX IEEE, pp.878-888.

iv. C. Usha Rani and N. Munisankar, 2014, Spatial Index Keyword Search in Multidimensional Database, International Journal of Computer Science and Information Technologies, Vol. 5, No. 5, 6468-6471, ISSN: 0975-9646.

v. Chandrashekhar, 2014, Fast Searching With Keywords Using Data Mining, International Journal of Computer Science and Information Technology Research, April-June, Vol. 2, o. 2, pp.82-89, ISSN 2348-120X.

vi. Sophiya K and Sounderrajan T, 2014, Implements the Spatial Inverted Index to Perform Quick Search, International Journal of Innovating Research in Computer and Communication Engineering, Vol. 2, No. 1, March, pp. 2528-2531.

vii. Rajkumar R, Manimekalai P, Mohanpriya M and Vimalarani C, 2014, Efficient Nearest and Score Based Ranking for Keyword Search, International Journal of Advanced Research in Computer Engineering & Technology, Vol. 3, No. 3, March, pp. 1023-1027.

viii. Sonal K Kasare and Anup Bongale, 2013, Efficiently Searching Nearest Neighbor in Documents using Keywords, International Journal of Research in Engineering and Technology, Vol. 2, No. 12, December, pp. 559-561.

ix. Nitin Bhatia and Vandana, 2010, Survey of Nearest Neighbor Techniques, International Journal of Compute Science and Information Security, Vol. 8, No.2, pp. 302-305.

x. Yufei Tao, Dimitris Papadias, and Qiongmao Shen. 2002, Continuous Nearest Neighbor Search. Proceedings of the 28th Very Large Data Bases conference (VLDB), Hong Kong, China, pp. 287-298.

xi. Jun Zhang, Manli Zhu, Dimitris Papadias, Yufei Tao, and Dik Lun Lee. 2003, Location- based Spatial Queries. Proceedings of ACM Conference on Management of Data (SIGMOD), pp. 443-454.

xii. Yufei Tao, Dimitris Papadias, and Xiang Lian. 2004, Reverse kNN Search in Arbitrary Dimensionality. Proceedings of the 31st Very Large Data Bases conference (VLDB), pp. 744-755

xiii. Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui, 2005, Aggregate Nearest Neighbor Queries in Spatial Databases, ACM Transactions on Databases Systems (TODS), 30(2), pp. 529-576.

www.ijesta.com

xiv. Yufei Tao, Man Lung Yiu, and Nikos Mamoulis. 2006, Reverse Nearest Neighbor Search in Metric Spaces, IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 18, No. 9, pp. 1239-1252.

xv. Yufei Tao, Ke Yi, Cheng Sheng, and Panos Kalnis. 2009, Quality and Efficiency in High Dimensional Nearest Neighbor Search. Proceedings of ACM Conference on Management of Data (SIGMOD), pp. 563-576

xvi. Yufei Tao, Ke Yi, Cheng Sheng, and Panos Kalnis. 2010, Efficient and Accurate Nearest Neighbor and Closest Pair Search in High Dimensional Space. ACM Transactions on Databases Systems (TODS), Vol. 35, No. 3.

xvii. Sze Man Yuen, Yufei Tao, Xiaokui Xiao, Jian Pei, and Donghui Zhang. 2010, Superseding Nearest Neighbor Search on Uncertain Spatial Databases. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 22, No. 7, pp. 1041-1055

xviii. Yufei Tao, Stavros Papadopoulos, Cheng Sheng, and Kostas Stefanidis. 2011, Nearest Keyword Search in XML Documents., Proceedings of ACM Conference on Management of Data (SIGMOD), pp. 589-600

www.ijesta.com