International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 03 | Mar-2018 www.irjet.net p-ISSN: 2395-0072

Priority Based Search by Reducing Techniques

Palak D. Pande[1], Ankita N. Rawale[2], Ankita S. Sachdev[3], Kajal V. Sawadh[4]

[1,2,3,4] *Student, Department of CSE, Des'scoet, Dhamangaon Rly, Maharashtra, India*

Abstract - *The K-NN is a technique in which objects are classified depends on nearest training examples which is present in the feature query space. The K-NN is the simplest classification method in data mining. In K-NN objects are classified when there is no information about the distribution of the data objects is known. Objects are selected according to their ranks with respect to the query object, allowing much tighter control on the overall execution costs. A formal theoretical analysis shows that with very high probability, the RCT returns a correct query result in time that depends very competitively on a measure of the intrinsic dimensionality of the data set. It returns corrects query execution result in required time that relies on a intrinsic dimensionality of objects of the data set. RCT can exceed the performance of methods involving metric pruning and many selection tests involving distance values having numerical constraints on it.*

Key Words: **K-Nearest neighbor search, intrinsic dimensionality, rank-based search, RCT.**

1. INTRODUCTION

In Data mining there is a various tools of data analysis which can find patterns of objects and relationships among the data. These tools make use valid prediction of object data. There are various fundamental operations such as cluster analysis, classification, regression, anomaly detection and similarity search. In all of which the most widely used method is of similarity search. Similarity search having built in principal of k-Nearest Neighbor (K-NN) classification. k-NN is founder of it. When number of data object classes is too large then similarity search produces low error rate as compare to other methods of analysis. Error rate of Nearest Neighbor classification shows when training set size increased is „asymptotically optimal „.In similarity search feature vectors of data objects attributes are modelled for which similarity measure is defined.

The performance of similarity search indices depends crucially on the way in which they use similarity information for the identification and selection of objects relevant to the query. Virtually all existing indices make use of numerical constraints for pruning and selection. Such constraints include the triangle inequality (a linear constraint on three distance values), other bounding surfaces defined in terms of distance (such as hypercubes or hyperspheres) range queries involving approximation factors as in Locality-Sensitive Hashing (LSH) [19], [30], or absolute quantities as additive distance terms [6]. One serious drawback of such operations based on numerical constraints such as the triangle inequality or distance ranges is that the number of objects actually examined can be highly variable, so much so that the overall execution time cannot be easily predicted. In an attempt to improve the scalability of applications that depend upon similarity search, researchers and practitioners have investigated practical methods for speeding up the computation of neighborhood information at the expense of accuracy. For data mining applications, the approaches considered have included feature sampling for local outlier detection, data sampling for clustering, and approximate similarity search for k-NN classification (as well as in its own right). Examples of fast approximate similarity search indices include the BD-Tree, a widely-recognized benchmark for approximate k-NN search; it makes use of splitting rules and early termination to improve upon the performance of the basic KD-Tree. One of the most popular methods for indexing, Locality-Sensitive Hashing can also achieve good practical search performance for range queries by managing parameters that influence a tradeoff between accuracy and time. The spatial approximation sample hierarchy (SASH) similarity search index has had practical success in accelerating the performance of a shared-neighbor clustering algorithm, for a variety of data types.

The use numerical constraints shows large variations in the numbers of objects that can be examined in the execution of a query, It is difficult to manage the execution costs. To overcome the problem of large variation in objects analysis in execution. We build a new data structure, the Rank Cover Tree (RCT), used for k-NN. This can totally exclude the use of elements of data objects having numerical constraints. All selection operation of RCT can be performed using a specific assigned ranks of each objects according to the query, having strict control of execution of data query. By using a rank of objects it gives rank-based search analysis provides best probability of analysis, the RCT gives a correct result of query in required time that fully depends on data set intrinsic dimensionality. The RCT is similarity search method use the ordinal pruning method and provides correct analysis of performance of the query result.

2. RELATED WORK

This paper consists of two most important and recently-developed approaches that are quite dissimilar from each other which is consider to proposed RCT data structure. The SASH heuristic is used for approximate searching of similarity, and second approach that is the cover Tree used for exact searching of similarity. RCT can used method of combinatorial search similarity approach. the SASH also used an combinatorial similarity search approach, whereas In the cover tree numerical constraints are used for selection

and pruning of data objects. Description of SASH and Cover Tree as given below.

A. Cover tree:

In Cover Tree the intrinsic dimensionality performance can be analyzed by a common search method for determining nearest neighborhood data queries example. In this approach, a randomized structure can found like to be skip list which can be used to recognized pre-determined samples of data elements which is surrounding points object of interest. In CT sample data elements can be shift by applying the same procedure which is nearest to the relevant object query and finding new samples set which can be in surrounding point of nearest interest. The sample elements S having minimum value of expansion rate δ as it needs required condition which is to beheld above. It can be provided to different alternatives which is consist of min value of ball object of a size set.

B. Spatial Approximation Sample Hierarchy (SASH)

The huge amount of data sets objects that used a data structures providing the better performance for an amount of N data items within given database. The R-Tree play an efficient role for efficiency of DBSCAN. To handle very massive data sets, use SASH technique. The SASH method can build minimal number of assumptions about associative objects queries metric. SASH does not regulate a partition of the query search space, as the instance of R-Trees can done. For similarity search of approximation of k-NN (k-ANN) queries present on the huge data sets, the similarity search SASH can systematically provide a huge part of k-NNs truth of queries at specific speeds of randomly of two different orders of relative size which is faster than regular sequential search method. For clustering method and navigation of very huge , very large dimensional text, image sets of data on which The SASH can perform successfully .

C. Performance of Rank Cover Tree

We can compare execution time required for processing of high dimensional data of fixed-height variant of Rank Cover Tree against various algorithms such as Cover Tree, SASH. RCT can provide measure accuracy as compared to CT and SASH. It also considers the E2LSH implementation of Locality Sensitivity Hashing method. It is most common method which is also used to Speed up execution cost of KNN. Following graph shows how RCT work for No. of Records against CT and SASH. There are different algorithm was developed to increase the efficiency of KNN like KD-Tree and BD Tree. the first library, ANN provides implementations which is approximate search of KNN in the KD-Tree and BD-Tree (box decomposition tree). We measured the accuracy of the methods in terms of both distance error and recall, the latter perhaps being a more appropriate measure for k-NN query performance. The recall is defined as the proportion of true nearest neighbors returned by an index structure.

D. Rank Function

Tree-based strategies for proximity search typically use a distance metric in two different ways: as a numerical (linear) constraint on the distances among three data objects (or the query object and two data objects), as exemplified by the triangle inequality, or as an numerical (absolute) constraint on the distance of candidates from a reference point. The proposed Rank Cover Tree differs from most other search structures in that it makes use of the distance metric solely for ordinal pruning, thereby avoiding many of the difficulties associated with traditional approaches in high-dimensional settings, such as the loss of effectiveness of the triangle inequality for pruning search paths [12]. Let U be some domain containing the point set S and the set of all possible queries. We assume the existence of an oracle which, given a query point and two objects in S, determines the object most similar to the query. Note that this ordering is assumed to be consistent with some underlying total order of the data objects with respect to the query. Based on such an oracle we provide an abstract formulation of ranks.

E. Rank Cover Tree

The proposed Rank Cover Tree blends some of the design features of the SASH similarity search structure and the Cover Tree. Like the SASH (and unlike the Cover Tree), we shall see that its use of ordinal pruning allows for tight control on the execution costs associated with approximate search queries. By restricting the number of neighboring nodes to be visited at each level of the structure, the user can reduce the average execution time at the expense of query accuracy.

3. PROPOSED SYSTEM

We proposed a new data structure which is a probabilistic used for similarity search index; the rank-based search means Rank Cover Tree (RCT), in which no involvement of numerical constraints for selection and pruning of data element objects. All internal operation such as selections of objects are made by consider to specified ranks of that objects element according to that query , having strict control on query execution costs. A rank-based probabilistic method having huge probability, the RCT perform a correct result of query execution in specific time that relies on a high portion of the intrinsic dimensionality of that data set. *Construction:* 1. Consider each item **x** To X, provides **x** into levels 0,. . . x. Height of tree is h, x can follows technique of a geometric distribution with q = jXj-1=h. 2. A partial RCT can be build by connecting each items in that level to an artificial root of tree on the highest level. 3. In partial RCT by using approximate nearest neighbors method which is found in the partial RCT can connect the next level of tree. 4. A RCT can be well-build with very high probability.

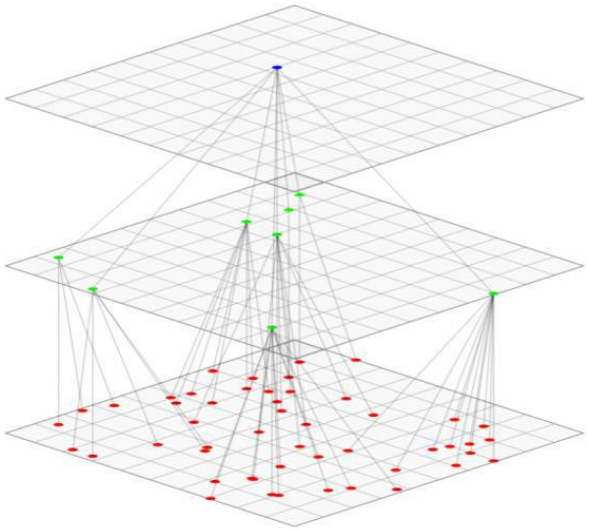

Figure 1: RCT Construction

To implement Rank Cover Tree it consists of design features of similarity search SASH and also design feature of Cover Tree. SASH can be used for approximate searching and cover tree for exact search of objects. both of these make use of a ordinal strategy for pruning of objects and it allows for strict control on query execution cost which is obtained with method of queries of approximate search . At each and every level of the tree structure visited the number of neighboring nodes can be restricted, the user also reduces average required execution time of that query at the each level of that query accuracy. The proximity search of Tree-based strategies make use of distance metric method in two ways in which numerical constraint of objects among three data objects on its the distances as it is examined by the method of triangle inequality, or distance of data candidates from its a reference point of numerical (absolute) value constraint present on it.

i. Objective:

- The RCT can increase the performance of methods that involves metric pruning strategy or other type of selection tests having numerical constraints on distance values.
- To increase the computational cost of K-NN Search.
- Using RCT user can minimize the average amount of time required for execution .to obtain a great query accuracy.
- It provides tighter control on overall execution costs. Provides best result for similarity search *Ii.*

ii. Necessity:

- In RCT Rank thresholds method specifically calculate the number of data objects which is to be selected for pruning it avoid and reduce a major of variation of data elements objects in the overall execution time of query.
- It improves computational cost of similarity search.

4. CONCLUSIONS

In this paper, the Rank Cover Tree is a new search data structure for KNN which completely avoid numerical calculation and increase the efficiency of algorithm. It is a rank based similarity search. In which ordinal pruning approach is used that involves direct distance values of data objects comparisons. The Rank Cover Tree, whose ordinal pruning strategy makes use only of direct comparisons between distance values. The RCT construction and query execution costs do not explicitly depend on the representational dimension of the data, but can be analyzed probabilistically in terms of a measure of intrinsic dimensionality, the expansion rate.

5. REFERENCES

[1] Abraham, D. Malkhi, and O. Dobzinski, "LAND: Stretch (1 + epsilon) locality-aware networks for DHTs," in Proc. 15th Annu. ACM-SIAM Symp. Discrete Algorithm, 2004, pp. 550–559.

[2] A. Andoni and P. Indyk. (2005). E2LSH 0.1: User Manual.[Online].Available:www.mit.edu/andoni/LSH/, 2005.

[3] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in Proc. ACM SIGMOD Int. Conf. Manag. Data, 1999, pp. 49–60.

[4] A. Asuncion and D. J. Newman. (2007). UCI machine learning repository. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[5] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in Proc. 7th Int. Conf. Database Theory, 1999, pp. 217–235.

[6] A. Beygelzimer, S. Kakade, and J. Langford, "Cover trees for nearest neighbor," in Proc. 23rd Int. Conf. Mach. Learn., 2006, pp. 97– 104.

[7] T. Bozkaya and M. Ozsoyoglu, "Indexing large metric spaces for similarity search queries," ACM Trans. Database Syst., vol. 24, no. 3, pp. 361–404, 1999.

[8] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," SIGMOD Rec., vol. 29, no. 2, pp. 93–104, 2000.

[9] S. Brin, "Near neighbor search in large metric spaces," in Proc. 21th Int. Conf. Very Large Data Bases, 1995, pp. 574–584.

[10] H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou, "On hierarchical routing in doubling metrics," in Proc. 15th Annu. ACMSIAM Symp. Discrete Algorithm, 2005, pp. 762–771.

[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, 2009.

[12] E. Ch_avez, G. Navarro, R. Baeza-Yates, and J. L. Marroqu_ın, "Searching in metric spaces," ACM Comput. Surv., vol. 33, no. 3, pp. 273–321, 2001.

[13] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in Proc. 23rd Int. Conf. Very Large Data Bases, 1997, pp. 426–435.

[14] T. Cover, and P. Hart, "Nearest neighbor pattern classification, IEEE Trans. Inf. Theory, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.