|| Volume 2 ||Issue 8 ||FEB 2017||ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH
AND ENGINEERING TRENDS

Reduction the Dimensional Dependence Using Rank-Based Similarity Search

Miss. Manisha R. Ghunawat[1], Prof. R. A. Auti[2]

P.G. Student, Computer Science & Engineering, Everest Educational Society's Group of Institutions, Maharashtra, India[1].
HOD Computer Science & Engineering, Everest Educational Society's Group of Institutions, Maharashtra, India[2].

***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. In K-NN performance of classification is depend on K and it can be determined by the choice of K as well as the distance metric of query. The performance of K-NN classification is largely affected by selection of K which is having a suitable neighborhood size. It is a key issue for classification. This paper proposed a data structure which is for K-NN search, called as RANK COVER TREE to increase the computational cost of K-NN Search. In RCT pruning test involves the comparison of objects similar values relevant to query. In Rank Cover Tree each object can assign a specific order and according to that order object can selected which can be relevant to the respective object query. It can control the overall query execution cost .It provides result for Non-metric pruning methods for similarity search and when high dimensional data is processed it provides the same result. 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**

Keywords: K-Nearest neighbor search, intrinsic dimensionality, rank-based search, RCT.

I 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.

Various application of data mining is depend on that. Similarity search can accesses an unacceptable –huge part of the data object elements, unless the other data can be distributed having special properties of data elements. Various Data mining application which uses common neighborhood knowledge of data which is useful and having great meaning. High data dimensional tends to make this common information which very costly to gain. In Similarity search indices selection and identification of objects which is relevant to query objects depend on similarity values of information. This can measure the performance of similarity search. In distance-based similarity search make use of numerical constraints of similar values of data objects for building pruning and selection of data objects such types include the triangle inequality and additive distance bounds. 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.

II LITERATURE SERVEY

For clustering, various effective and common methods require the finding of neighborhood sets of data objects which is depend on mostly at a required proportion of data set objects[1][2]. Various examples consists such as hierarchical (agglomerative) methods like ROCK [3] and CURE [4]; another

Sr No	Technique	Key Idea	Advantages	Disadvantages	Target Data
1.	k Nearest Neighbor (kNN)	Uses nearest neighbor rule	1. training is very fast 2. Simple and easy to learn 3. Robust to noisy training data 4.Effective if training data is large	1. Biased by value of k 2.Computation Complexity 3.Memory limitation 4. Being a supervised learning lazy algorithm i.e. runs slowly 5.Easily fooled by irrelevant attributes	large data samples
2.	Weighted k nearest neighbor (WkNN)	Assign weights to neighbors as per distance calculated	1. Overcomes limitations of kNN of assigning equal weight to k neighbors implicitly. 2. Use all training samples not just k. 3. Makes the algorithm global one	1. Computation complexity increases in calculating weights 2. Algorithm runs slow	Large sample data
3.	Condensed nearest neighbor (CNN)	Eliminate data sets which show similarity and do not add extra information	1. Reduce size of training data 2. Improve query time and memory requirements 3.Reduce the recognition rate	1. CNN is order dependent; it is unlikely to pick up points on boundary. 2. Computation Complexity	Data set where memory requirement is main concern
4.	Reduced Nearest Neigh (RNN)	Remove patterns which do not affect the training data set results	1. Reduce size of training data and eliminate templates 2. Improve query time and memory requirements 3.Reduce the recognition rate	1.Computational Complexity 2.Cost is high 3.Time Consuming	Large data set
5.	Ball Tree k nearest neighbor (KNS1)	Uses ball tree structure to improve kNN speed	1.Tune well to structure of represented data 2.Deal well with high dimensional entities 3.Easy to implement	1.Costly insertion algorithms 2.As distance increases KNS1 degrades	Geometric earning tasks like robotic, vision, speech,graphics
6.	k-d tree nearest neighbor (kdNN)	divide the training data exactly into half plane	1.Produce perfectly balanced tree 2.Fast and simple	1.More computation 2.Require intensive search 3.Blindly slice points into half which may miss data structure	organization of multi dimensional points
7.	Locality Sensitive Hashing	Locality sensitive hashing is based on the idea of random projections	1. Hash tables to improve search speed. 2. Locality-sensitive hashing offers sub-linear time search by hashing highly similar examples together.	1.LSH requires a vector representation	Near-duplicate detection

Method density-based example as DBSCAN [5], OPTICS [6], and SNN [7] and also non-agglomerative shared-neighbor clustering [8].

A recommender systems and anomaly detection technique used content based filtering approach [9], k-NN method also used in normal condition build, by making direct use of method k-NN cluster analysis. A another very popular local density-based measure that is method of local Outlier factor (LOF), totally rely on data set of k-NN whose computation to obtain the denseness of used data which is present in the test point of that section [10].

III. RANK COVER TREE

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.

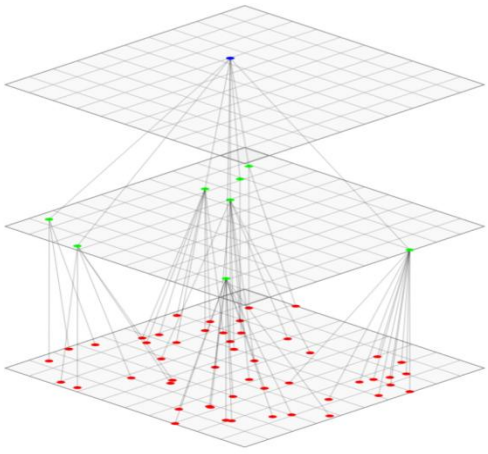

Figure 1: RCT Construction

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:

1. 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.
2. To increase the computational cost of K-NN Search.
3. Using RCT user can minimize the average amount of time required for execution .to obtain a great query accuracy.
4. It provides tighter control on overall execution costs. Provides best result for similarity search

Ii. Necessity:

1. 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.
2. It improves computational cost of similarity search

IV 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.

i. 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 be

held above. It can be provided to different alternatives which is consist of min value of ball object of a size set.

ii. 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 .

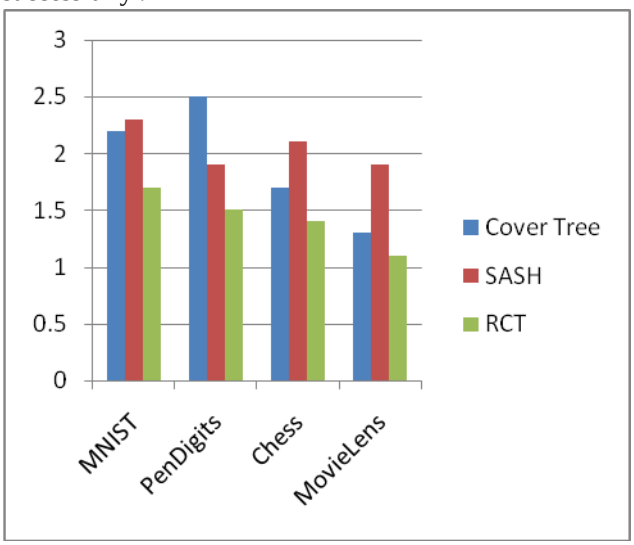

Chart 1: Average Execution time versus No. of records

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.

Results:

The RCT of fixed-height variants can provide experimental results which give great speed-ups as compared to sequential search. It can be more than 10 times of that. Whenever we compared the performance of other methods in competition of RCT and SASH for lower dimension data sets provides best results. FLANN method is used for high dimensional data sets which is competitor of RCT and SASH. But FLANN can reduce its performance when Euclidean distance is used. To overcome the drawback of FLANN rank based similarity search method means RCT work very well.RCT can find the precision and recall error of high dimensional data sets

The execution query times of various datasets are consistent for RCT and SASH as compared to different approximation algorithms method. Space required to store database prevent a full exposition of the finding the variation in query processing performance. We can compare the performance of RCT and KD tree on database MNIST data set by providing the variation in query processing time.

Table 1: Construction Times required (in ms) for the Various Methods Tested over a various data sets

Data Set Names	Size in	Dim.	BD Tree	KD-Tree	Cover Tree	SASH	RCT
MNIST	69,000	784 Features	110.29 ms	33.18ms	135.28ms	39.69 ms	219.48ms
Pen digits	11,992	16 Features	0.05ms	0.02ms	0.04ms	0.41 ms	0.57ms
Chess	31	056 Features	5ms	-	0.05 ms	0.88ms	2.34ms
Cover Type set	581	012 Features	53ms	53.35ms	4.57ms	82.04 ms	884.09ms
Movie Lens Data	199,229	568 Features	33ms	21.5ms	2.35ms	0.69 ms	698.82ms

Table 2: KD Tree and RCT performed on MNIST and finding the construction time (in Seconds) needed for both methods having different size and dimension

Size	Dimension	KD Tree	RCT
92,827	512 (426) Features	22.53 S	22.14S
150,257	1024 (832) Features	110.69 S	60.64 S
242,320	2,048 (1657) Features	528.93 S	156.04 S
321,547	4096 (3285) Features	2481.48 S	229.87 S

V RESEARCH WORK

In RCT completely avoid numerical calculations that contain numerical constraints value can overcome the problem present in methods like triangle inequality and distance ranges in object data count actually proceed or examined having very high variation of objects, because of it the overall or complete time required for execution cannot be easily determined or predicted. Using RCT we can easily predict execution time. To increase the scalability and efficiency of data mining applications that fully rely on similarity search values. Finding the best methods for efficiently speed up the computational power of nearest neighborhood information at the great expense of accuracy

VI CONCLUSION

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 RCT construction is independent on the representational high dimension of the data. But it can be probabilistically analyzed in the form of approach a measure values of intrinsic dimensionality. The RCT can be build by using two main methods –that means the cover Tree and SASH structures techniques.

REFERENCES

[1] J. Han, and M. Kamber, Data Mining: Concepts and Techniques, San Francisco, CA, USA: Morgan Kaufmann, 2006. [1] J. Han, and M. Kamber, Data Mining: Concepts and Techniques, San Francisco, CA, USA: Morgan Kaufmann, 2006.

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

[3] S. Guha, R. Rastogi, and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," Inf. Syst., vol. 25, no. 5,pp. 345–366, 2000.

[4] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," Inf. Syst., vol. 26, no. 1, pp. 35–58,2001.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowl. Discov. Data Mining, 1996,pp. 226–231.

[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] L. Ert€oz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in Proc. 3rd SIAM Int. Conf. Data Mining, 2003, p. 1.

[8] M. E. Houle, "The relevant set correlation model for data clustering," Statist. Anal. Data Mining, vol. 1, no. 3, pp. 157–176,2008.

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

[10] 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.

[11] T. de Vries, S. Chawla, and M. E. Houle, "Finding local anomalies in very high dimensional space," in Proc. IEEE Int. Conf. Data Mining,2010, pp. 128–137.

[12] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in Proc. 25th Int. Conf. Very Large Data Bases, 1999, pp. 518–529.

[13] P. Indyk, and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in Proc. 30th ACM Symp. Theory Comput., 1998, pp. 604–613.

[14] M. E. Houle and J. Sakuma, "Fast approximate similarity search in extremely high-dimensional data sets," in Proc. 21st Intern. Conf. Data Eng., 2005, pp. 619–630.

[15] M. E. Houle and M. Nett, "Rank cover trees for nearest neighbor search," in Proc. Int. Conf. Similarity Search Appl., 2013,pp. 16–29.