International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 9, November- 2012

Cluster Distance Bound for High Dimensional Indexing

Shivaprasad B[1], Dr. R.V.Krishnaiah[2]

**(Department of CSE, DRK College of Engineering and Technology, India)*

*** (Department of CSE, DRK College of Engineering and Technology, India)*

IJERT

Abstract

In data mining domain, high-dimensional and correlated data sets are used frequently. Working with high-dimensional data for data mining operations like clustering has become a common task in data mining. In this paper, we consider similarity search approaches in the presence of high-dimensional data. The existing indexing approaches such as vector approximation has some drawbacks such as ignoring dependencies across dimensions. This results in sub optimality in results. However, clustering makes use of inter-dimensional correlations and can represent a dataset used. Pruning clusters that are not relevant is done by existing algorithms. However, they are based on bounding rectangles, bounding hyper spheres and they lack in efficiency in nearest neighbor search. We propose a new algorithm for separating hyperlange boundaries of Voronoi clusters. This is known as cluster-adaptive distance bound algorithm which is complements cluster based index. It performs spatial filtering well besides reducing the storage overhead. Our method can be used with Mahalanobis and Euclidean similarity measures. We developed a prototype application for demonstrating the efficiency of the proposed method. The results revealed that the proposed method is effective and can be used in the real world data mining applications.

Index Terms *– Data mining, indexing, similarity measures, multimedia database*

1. Introduction

With the invent of new technologies in semiconductors and signal processing tools and with new digital devices information processing and data mining has gained popularity. There has been increase in the use of specialized devices apart from digital cameras, video players, music and personal media devices. In addition to this storage media has become cheaper and affordable making expensive data mining applications feasible. With this new application pertaining to multimedia, GIS (Geographical Information Systems), CAD (Computer Aided Design) and CAM (Computer Aided Manufacturing), time series analysis, medical imaging applications are able to process large volumes of data. The databases that store the data of these applications can range from 100 GB to several 100 TBs. Supporting such huge data and performing mining operations on such data is an essential task of today and in future.

In case of high dimensional data, spatial queries have been researched that include nearest neighbor queries. The nearest neighbor search with metric like Euclidean Distance Metric is not practical for high dimensional data due to the problem of "curse of dimensionality" and other metrics are known to be over pessimistic [1]. In case of R-tree also according to [2], search performance is intrinsic dimensionality of the dataset. It is not the address space dimensionality or embedding dimensionality in other words. In previous researches the assumptions include data uniform distribution of data having attributes independent. These data sets actually resulted in "curse of dimensionality" as the distance between them results in same value as described in [3]. Using techniques like furthest neighbors and nearest neighbors it is impossible to index such data. The assumptions used in the prior research can't help in real world data sets where data is not as assumed. For this reason data sets in the real world are indexable with Euclidean distances. Content based image retrieval is using such metric to retrieve data towards weighted Euclidean (Mahalanobis) as explored in [4]. In this paper the focus is on real data sets and performance is compared against the state-of-the-art indexing with real world datasets.

2. Related Work

To process queries on multidimensional data many indexing structures came into existence. These indexing structures help in processing search queries faster. Recursive partitioning is used in case of low-dimensional data using R-trees as described in [5]. Other indexing structures that are effective include SS-Tree [6], SR Tree [7] or combination of them. In case of Euclidean distance the methods described previously are specialized with it. According to [8] the very effective metrics spaces with distance functions are M-trees. In low-dimensional spaces, multidimensional indexes work well. Moreover they are able to outperform the sequential scan. However, the important observation is that the performance of them is degraded when the features dimensions are increased after cross some threshold pertaining to dimensions. It becomes inferior in case of sequential scan. As per the research made in [9] when the dimensionality is more than 10, these methods are showing less performance than sequential scan. This kind of performance is attributed to "curse of dimensionality" proposed by Bellman [10] which talks about the exponential growth of dimensionality in the space.

IJERT

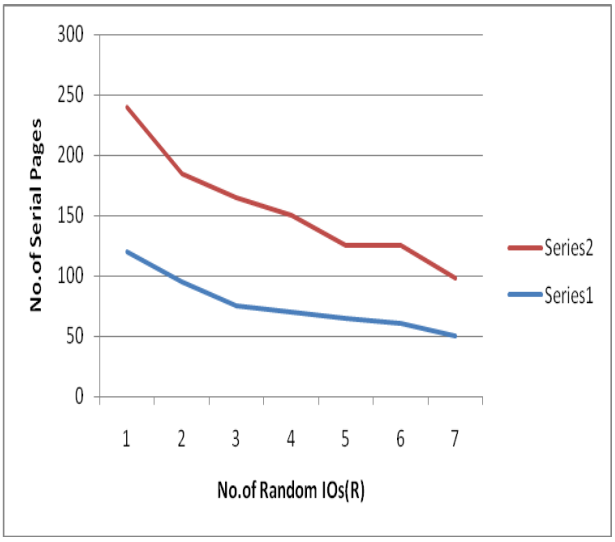

Fig. 1 - Comparing Performance Graphs of (hypothetical) Index A and Index B

In order to overcome the curse of dimensionality a new approach by name vector approximation file [9] is used and it became popular technique. It works by breaking space into some cells which are hyper rectangular in nature that can obtain quantized approximation of the data that is present in the cells. A separate approximation file is used in hard disk for storing encoded bit string in case of non empty cell locations. When nearest neighbor search is performed the VA-file is scanned sequentially and lower and upper bounds, each cell distance from the distance vector is estimated. In order to prune irrelevant cells these bounds are used. Finally the candidate vectors from hard disk are read and the nearest neighbors are found out. In case of VA-file what exactly being performed is known as scalar quantization and the terms "Vector Approximation" looks not appropriate name. After VA-file, many such techniques came into existence to overcome the problem known as curse of dimensionality. According to [11], in case of VA-File, the data set is rotated into an array of dimensions which are uncorrelated that contain many approximate bits which are given by high dimensions with different variance. According to the data distribution, the approximation cells are adaptively spaced. The recently proposed approximations [12] and the methods like LDR [13], their aim is to outperform sequential scan. There are some hybrid methods like IQ-Tree and A-Tree they are combination of approximations and tree based indexes.

Finally, distance functions and feature vectors, according to some argument, are often represent approximations of user's perception of similarity. Therefore, the results of an exact similarity search are really approximate in terms of perceptually with refinement necessary with additional rounds of query. Considerable savings in query processing can be done by performing approximate search with little less accuracy. Search strategies such as PAC-NN [14], MMDR [15], and VA-LOW [16], [17] and hashing mechanism known as locality sensitive hashing [18]. More information of approximate similarity search [19] has to be contacted. The approximate indexing has some limitations that are optimal tradeoffs between the search time and quality of search. This limitation is well explored in [20].

3. Cluster Distance Bounding

This section describes the procedure for estimating distances to clusters. Distance function to develop an effective cluster distance bound is computed as

$$d : \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty).$$

When converting to Mahalanobis distane measure the distance function is computed as

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \Sigma^{-1} (\mathbf{x}_1 - \mathbf{x}_2)},$$

IJERT

4. Clustering and index structure

As part of index construction, the first step is the creation of Voronoi/Nearest Neighbor clusters. After doing this, there are several techniques for clustering given datasets based on the fast K-means algorithm [21] and GLA (Generalized Lloyd Algorithm) [22] and BIRCH [23] that need a single scan for completing the search. The result of any of these algorithms can be used as starting point. A pivot is selected from each of K clusters detected by a generic clustering algorithm where pivot means K pivot points in all. Afterwards, the entire dataset is scanned and each data mapping is done between the data elements and the nearest pivot. In order to form Voronoi clusters, data mapping to the same pivot are grouped together. It is shown in algorithm1.

VORONOL-CLUSTERS

1: //Generic clustering algorithm returns

//K cluster centroids

$\{C_m\}^k=1 \leftarrow$ GenericCluster(X,K)

Set l=0, X_1=Φ, X_2= Φ,…….X_k= Φ;

While l< | X | do

L=l+1

//Find the centroid nearest to data element xl

K=arg min $_m$d(x1.C_m)

//Move xl to the corresponding Voronoi partition

X_k= X_k∪{Xl}

End while

Return{X^m}=1, {C_m}^{k}=1

Algorithm 1 [24]

Slight rearrangement of clusters takes place to retain preciseness. However, centroid is a good choice as pivot. With only a single scan thus, in case of quick Voronoi clustering it can be achieved using a generic clustering algorithm. An indexing scheme needs one scan at least. This means that our algorithm takes less time for creation of index.

KNN-SEARCH(q)

//Initialize

set FLAG=0, count = 0;N =0; kNN = Φ

//Evaluate query-cluster distance bounds

dLB[]←HyperplaneBound(q)

//Sort the query-cluster distance bounds in ascending

//order

while FLAG==0 do

count =count + 1

//Find the kNNs upto current cluster

{Nc; kNN←FindkNNsIn(q.$X_{0(count)}$, kNN)

//Update number of elements scanned

N =N + Nc

//Find the kNN radius

dkNN =Farthest(q. kNN)

if count < K then

if N >k then

set FLAG=1 //kNNs found, search ends

end if

end if

else

set FLAG=1 //all clusters scanned, search ends

end if

end while

return kNN

ALGORITHM 2 – KNN-SEARCH(Q) [24]

FindkNNsIn(q;A; I)

set Nc = 0, F = Open(A), kNN = I

while !(EOF(F)) do

// Load the next cluster page

C = LoadNextPageðFÞ

//Merge kNN list with current page

X_{cand} =C∪ Knn

//Find the kNNs within the candidate list

kNN[]← Find $_{kNN}$(q,X_{cand})

//Update number of elements scanned

IJERT

Nc =Nc + | C |

end while

return N_c, kNN

Algorithm 3 [24]

5. Evaluation of Results

The experiments are made using a prototype application which demonstrates the efficiency of the proposed scheme. The results of the application are analyzed and the same are presented in the form of a series of graphs.

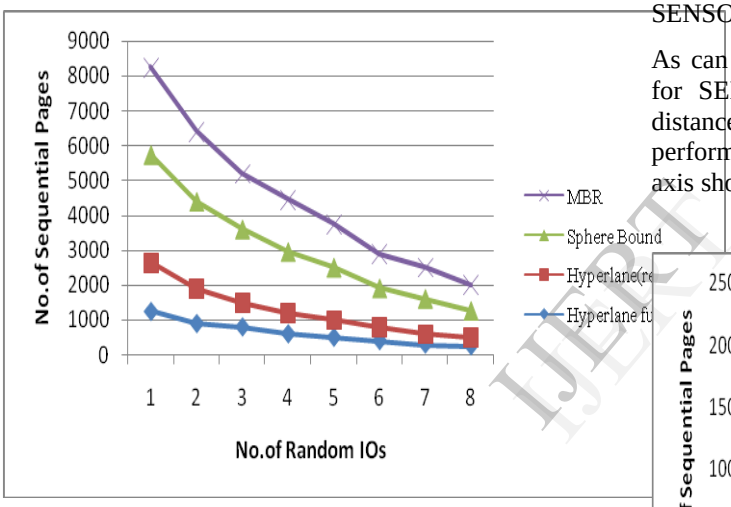

Fig. 2 - IO Performance of Distance Bounds—BIO-RETINA.

As can be seen in fig. 2, the results are presented for Bio-Retina showing the IO performance of distance bounds. The horizontal axis shows IO performance of distance bounds while the vertical axis shows the number of sequential pages.

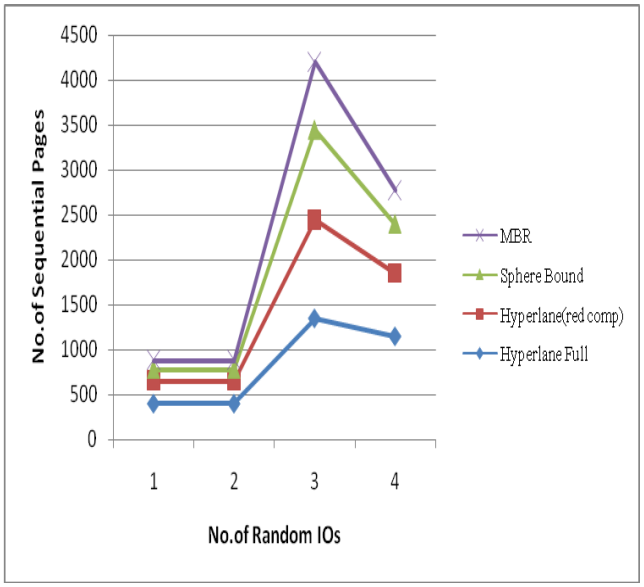

Fig. 3 - IO Performance of Distance Bounds—SENSORS.

As can be seen in fig. 2, the results are presented for SENSORS showing the IO performance of distance bounds. The horizontal axis shows IO performance of distance bounds while the vertical axis shows the number of sequential pages.

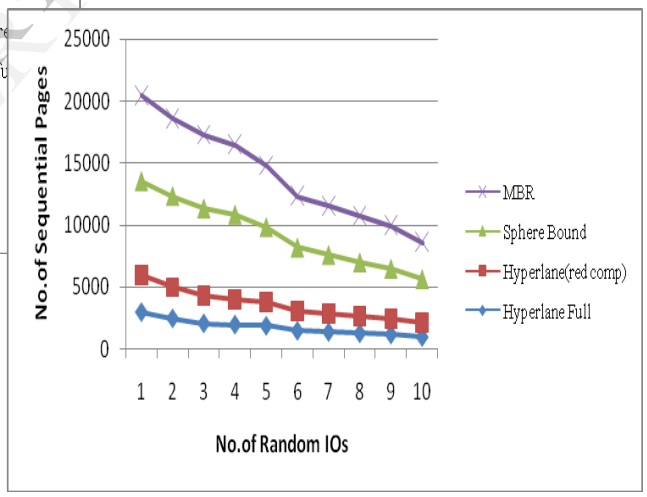

Fig. 3 - IO Performance of Distance Bounds—AERIAL.

As can be seen in fig. 3, the results are presented for AERIAL showing the IO performance of distance bounds. The horizontal axis shows IO performance of distance bounds while the vertical axis shows the number of sequential pages.

IJERT

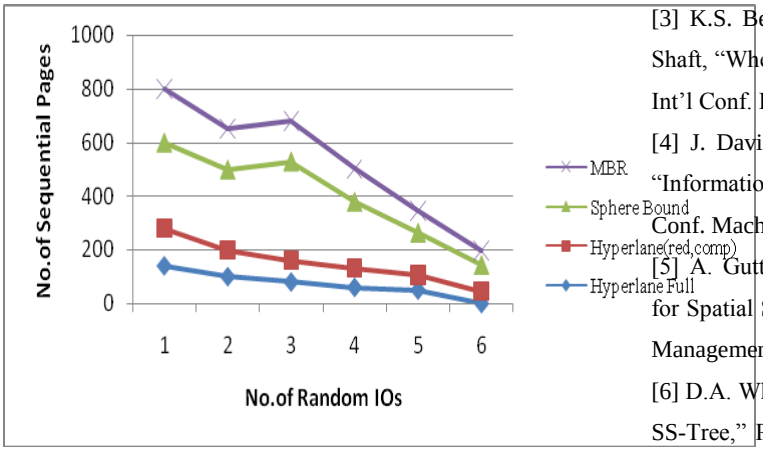

Fig. 4 - IO Performance of Distance Bounds—HISTOGRAM.

As can be seen in fig. 4, the results are presented for HISTOGRAM showing the IO performance of distance bounds. The horizontal axis shows IO performance of distance bounds while the vertical axis shows the number of sequential pages.

6. Conclusion

Non uniform distributions and significant correlations are exhibited by multidimensional datasets. Indexing such content with VA-File can't provide optimal performance. To overcome this problem, we proposed in this paper a new indexing method that is based on vector quantization in which dataset is divided into a set of voronoi clusters. Then cluster – distance bounds are developed based on the separating hyperplane boundaries. The new search index which is supported by these bounds can be used with Mahalanobis and Euclidean distance metrics. The proposed method also reduced IO cost significantly besides managing the operations with less memory. Its computational cost is low and can scale well with dimensions and size of dataset. When compared with MBR and MBS bounds, the proposed method is better. The experimental results revealed that our method is effective and can be used in real world data mining applications.

7. References

[1] C.C. Aggarwal, A. Hinneburg, and D.A. Keim, "On the Surprising Behavior of Distance Metrics in High Dimensional Spaces," Proc. Int'l Conf. Database Theory (ICDT), pp. 420-434, 2001.

[2] B.U. Pagel, F. Korn, and C. Faloutsos, "Deflating the Dimensionality Curse Using Multiple Fractal Dimensions," Proc. Int'l Conf. Data Eng. (ICDE), pp. 589-598, 2000.

[3] K.S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "Nearest Neighbor" Meaningful?," Proc. Int'l Conf. Database Theory (ICDT), pp. 217-235, 1999.

[4] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon, "Information- Theoretic Metric Learning," Proc. Int'l Conf. Machine Learning (ICML), pp. 209-216, 2007.

[5] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 47-57, 1984.

[6] D.A. White and R. Jain, "Similarity Indexing with the SS-Tree," Proc. Int'l Conf. Data Eng. (ICDE), pp. 516-523, 1996.

[7] N. Katayama and S. Satoh, "The SR-Tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 369-380, May 1997.

[8] P. Ciaccia, M. Patella, and P. Zezula, "M-Tree: An Efficient Access Method for Similarity Search in Metric Spaces," Proc. Int'l Conf. Very Large Databases (VLDB), pp. 426-435, 1997.

[9] R. Weber, H. Schek, and S. Blott, "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High- Dimensional Spaces," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 194-205, Aug. 1998.

[10] R. Bellman, Adaptive Control Processes: A Guided Tour. Princeton Univ. Press, 1961.

[11] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A.E. Abbadi, "Vector Approximation Based Indexing for Non-Uniform High Dimensional Data Sets," Proc. Int'l Conf. Information and Knowledge Management (CIKM), pp. 202-209, 2000.

[12] K. Vu, K. Hua, H. Cheng, and S. Lang, "A Non-Linear Dimensionality-Reduction Technique for Fast Similarity Search in Large Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 527-538, 2006.

[13] K. Chakrabarti and S. Mehrotra, "Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces," Proc. Int'l Conf. Very Large Databases (VLDB), pp. 89-100, Sept. 2000.

[14] P. Ciaccia and M. Patella, "PAC Nearest Neighbor Queries: Approximate and Controlled Search in High-Dimensional and Metric Spaces," Proc. Int'l Conf. Data Eng. (ICDE), pp. 244-255, 2000.

IJERT

[15] H. Jin, B.C. Ooi, H.T. Shen, C. Yu, and A. Zhou, "An Adaptive and Efficient Dimensionality Reduction Algorithm for High-Dimensional Indexing," Proc. Int'l Conf. Data Eng. (ICDE), pp. 87-98, Mar. 2003.

[16] R. Weber and K. Bo¨hm, "Trading Quality for Time with Nearest Neighbor Search," Proc. Seventh Int'l Conf. Extending Database Technology (EDBT): Advances in Database Technology, pp. 21-35, 2000.

[17] E. Tuncel, H. Ferhatosmanoglu, and K. Rose, "VQ-Index: An Index Structure for Similarity Searching in Multimedia Databases," Proc. ACM Int'l Conf. Multimedia, pp. 543-552, 2002.

[18] A. Gionis, P. Indyk, and R. Motwani, "Similarity Search in High Dimensions via Hashing," Proc. Int'l Conf. Very Large Databases (VLDB), pp. 518-529, Sept. 1999.

[19] P. Ciaccia and M. Patella, "Approximate Similarity Queries: A Survey," Technical Report CSITE-08-01, May 2001.

[20] E. Tuncel, P. Koulgi, and K. Rose, "Rate-Distortion Approach to Databases: Storage and Content-Based Retrieval," IEEE Trans. Information Theory, vol. 50, no. 6, pp. 953-967, June 2004.

[21] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis. John Wiley & Sons, 1996.

[22] A. Gersho and R.M. Gray, Vector Quantization and Signal Compression. Kluwer Academic Publishers, 1992.

[23] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 103-114, 1996.

[24] Sharadh Ramaswamy, Student Member, IEEE, and Kenneth Rose, Fellow, IEEE, "Adaptive Cluster Distance Bounding for High-Dimensional Indexing", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 6, JUNE 2011.

8. Authors

Shivaprasad B is a student of DRK College of Engineering and Technology, Ranga Reddy, Andhra Pradesh, India. He has received B.Tech degree in Computer Science and Engineering and M.Tech Degree in Computer Science and Engineering. Her main research interest includes Data Mining and Image Processing.

Dr.R.V.Krishnaiah is working as Principal at DRK INSTITUTE OF SCINCE & TECHNOLOGY, Hyderabad, AP, INDIA. He has received M.Tech Degree EIE and CSE. His main research interest includes Data Mining, Software Engineering.

IJERT