IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 1, January 2019

Survey on Various Width Clustering as per Density of Data for Efficient K Nearest Neighbor Search

Rajkumar D. Gulpatil[1], Prof. Dr. S. K. Shirgave[2]

PG Student, Department of Computer Science and Engineering, D.K.T.E.Society's Textile And Engineering Institute, Ichalkaranji, India[1]

Professor, Department of Information Technology, D.K.T.E. Society's Textile and Engineering Institute, Ichalkaranji, India[2]

Abstract: The data size is growing day by day as it has large use in industrial applications. Due to various data sizes and type, it creates interrupt to find the exact results. The K Nearest neighbor search technique is widely used to find a similar type of data, but it will result in high computational time as the data size increases. In this research, the various widths clustering is introduced to efficiently find the K Nearest Neighbor (K-NN) for a query object from a given data set. This reduces clustering time in addition to balancing the number of producing clusters and their respective sizes.

Keywords: Clustering, K-nearest neighbor, Various widths clustering, high dimensional.

I. Introduction

K-Nearest Neighbor algorithm (KNN) is a method for classifying objects based on the closest training examples in the feature space. k-NN returns the most similar objects from the query object q. It is used in many applications such as pattern recognition [1], outlier detection [2], intrusion detection [3] and classification [4]. There are two techniques to find k-NN that are approximate [5] and exact [6] [7], the exact k-NN is more accurate and expensive as compared to approximate k-NN.

In order to compute exact k-NN, an approach called Exhaustive k-NN which first scans the whole data set and then gives the result, but this result in High computational cost. To overcome this drawback, there are two types of indexing techniques are used 1) Tree-based Indexes; 2) Flat Indexes, which construct an index so that only part of the data set is accessed.

The Tree-based indexing makes use of the binary partitioning technique to construct a tree. A leaf node contains the actual values while the index node provides ordered access to the nodes. Tree-based indexing is not a good solution to find the k-NN because it has its own drawbacks. It uses the binary partition technique to build the tree structure and this separation may not be the appropriate way for the object distributions, especially when the data distribution is unknown and dimensionality is high. Most high dimensional databases are dissimilar and distributed sparsely. Internal nodes have High overlap with each other [8]. Triangle inequality fails to prune the intermediate nodes and ends by visiting the whole data set.

Flat-indexes partition the data set to form clusters using different clustering techniques. It uses the Fixed-Width clustering (FWC) [9][10] and k-means [11], where feature space is directly partitioned into a number of clusters. Center and radii are considered for upper bound and lower bound distance between the query and the objects. The quality of the clusters is determined by the radius and the number of objects in each cluster. Compact and well-separated clusters increase the efficiency of the k-NN.

The existing approaches are not compatible to produce high-quality clusters like k means assigns a sparse object to the closest cluster because of that the overlapping of clusters is high. In FWC, width is the bottleneck. In this, a survey based on the clustering method named Various Width Clustering is presented.

II. Literature survey

J. Prerau et al. [10] proposed unsupervised anomaly detection technique within the context of a network intrusion detection system. Formation of clusters is used as a tool to reduce the time required to find the K Nearest neighbors. Clustering is used as a means of breaking down the search space into smaller subsets to remove the necessity of checking every data point. This paper proposed a clustering technique using fixed width, but with some variations such

DOI 10.17148/IJARCCE.2019.8111

that each element is placed only in one cluster, therefore the overlapping of clusters is minimized. However, this approach doesn't work properly for when data distribution in a cluster is not even.

K means K Nearest Neighbor (kMkNN) algorithm is presented by Xueyi Wang [11] to obtain fast K Nearest neighbors from query object q. It uses k-means and triangle inequality to find the nearest objects. It first creates k clusters of n objects and uses the triangle inequality to prune the unnecessary clusters and only keep the required clusters and search the objects in only remaining clusters. Its performance is good for high dimensional data as compared to binary partition trees because it uses flat bases indexes were as a binary partition technique result in an unbalanced and deep tree structure which requires unnecessary computation through internal nodes. This approach works properly even when the data distribution in a cluster is not equal. However, it assigns sparse objects to the closest cluster and the clusters may have a large radius which increases the search time.

Marius Muja and David G. Lowe [5] proposes a randomized k-d forest and priority search k-means tree. K-d search algorithm is very effective in low dimensional data, but its performance decreases in high dimensional data. This approach works on randomized k-d forest using parallel search technique and k means tree. It searches nearest neighbors across all the dimensions; however, this approach gives only approximate search.

On the basis of FWC Abdul Mohsen Almalawi et al. [12] presented Various Width Clustering technique to efficiently obtain the K Nearest neighbors. First, consider the whole dataset as a single cluster and compare its size with the user-defined threshold, if size is greater than the user-defined threshold then calculate width by using width learning formula. Take this width as an input to fixed width clustering and form fixed width clusters of that width w. Again find the largest cluster and repeat this process until the size of the largest cluster is less than or equal to the user-defined threshold. Merging is used to minimize unnecessary calculations of distance. If one cluster is totally contained in another then, it’s better to merge this cluster to reduce calculations. In this way, Various Width Clusters (VWC) are formed. To find the k NN in VWC is an easy task as compared to fixed width clustering. In Fixed width clustering single cluster may contain half or more data of the dataset, therefore, to find the KNN in such a cluster is a time-consuming task. As the querying cost minimizes but the problem is that the preprocessing cost increases. So this issue can be overcome by using multithreading in preprocessing and the pruning of the cluster can be done using tree-like indexing to minimize querying cost also. The key point in recent approach is the user-defined threshold. To minimize overlapping of clusters, find the optimal radius of a cluster for a different distribution of data.

III.METHODOLODY

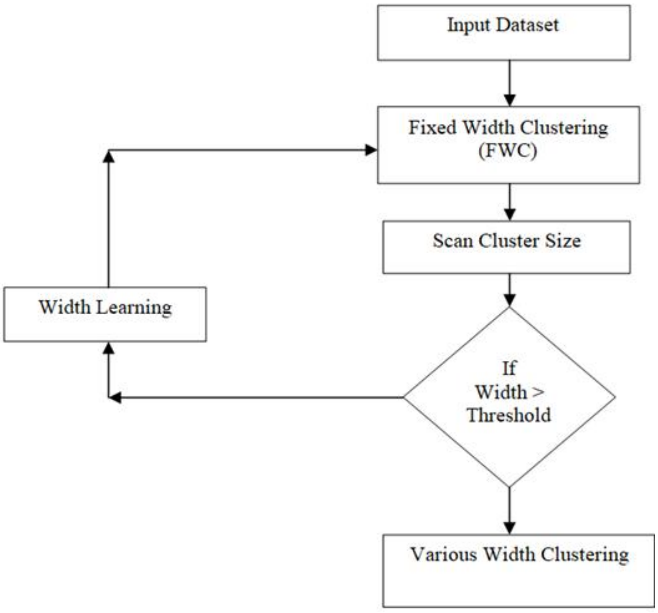

Fig. 1 Flow chart of Various Width Clustering

As shown in the flow chart using Fixed Width Clustering (FWC) the input dataset is simply partitioned into the clusters of fixed width w. Initially, the first object of the data set is considered as the centroid of one cluster, then the second object from input dataset is obtained, if its distance from the centroid is less than width w then this object is assigned to the first cluster otherwise another cluster of that object is created and treated as a centroid. This process is repeated until all the datasets are assigned in clusters. In this way suppose if there is a large dataset and the closest object from the query is to be determined, then instead of searching the whole data set, only the closest object in the neighboring cluster is searched. The problem arises when the density of data in one cluster is high then that cluster is repartitioned, by using various width as per density. To overcome the drawback of fixed width clustering Various Width Clustering is used based upon the density of the cluster.

In this algorithm, there are three parts.
1. Finding Width of the cluster
2. Partitioning of the cluster
3. Merging of the cluster.
In various width clustering finding the width is the sole of the algorithm. In this, randomly select a few objects from the dataset, compute the radius of its k-nearest neighbors and the average is used for global width. Let D be the dataset, H= {H1,H2,........,Hr} are randomly selected objects where r<<|D| then the following equation is used to compute the width.

$$w = \frac{1}{r}\sum_{i=1}^{r} clsWidth(NN_k(Hi), Hi) \qquad (1)$$

Some cluster size exceeds the user-defined threshold ß, in that case, need to partition the cluster such that whose width suits the density of the cluster and width is less than ß. In this case, the dataset and threshold ß are the inputs. First, consider the whole dataset as a single cluster and its centroid, a radius is zero. Find the largest cluster and check the size of that cluster, if its size is greater than ß then using the above equation find the width. If width is zero, then there is no need to partition because zero width means that the dataset is similar and no need to partition the dataset. Otherwise, using fixed width clustering, partition the dataset into the number of clusters. Again find the largest cluster among that and repeat the process until all the cluster size is less than threshold ß. If the produced cluster is only one, then reduce its size gradually and use it again. In this way, Various Width Clusters are formed, but the problem arises when some clusters are totally contained in another cluster. In that case, finding the k-NN from the query object need to calculate the distance of all the clusters from the query object. Merging such cluster increases the performance of the search k-NN by decreasing the number of clusters. Many of the objects are located in cluster C1 because it is a closer cluster, but they are also in cluster C2. In this case, when C1 is partitioned, it forms new clusters, which are completely located in another cluster i.e.C2, therefore, it is necessary to merge these clusters with C2. The merging process checks two conditions that are if the distance between the centroid of merging clusters plus width of the cluster, which is going to merge is less than the width of the cluster in which going to merge. Another issue is that one cluster may be contained by many clusters so merge the cluster, which is closer to the merging cluster.

$$\begin{cases} D(Ci, Cj) + wi \leq wjCi \subset Cj, & i \neq j, \\ \qquad OtherwiseC \notin Cj \end{cases} \qquad (2)$$

$$Cj = \min_{j=1} D(Ci, Cj). \qquad (3)$$

In this way the Various Width Clusters are formed which has a different size for each an every cluster as per density of data. The quality of clusters is determined by the radius and the number of objects in each cluster. Compact and well separated clusters increases the efficiency of the *k*-NN. As the quality of the clusters is improved in Various Width Clustering, minimum search time required to find out k-NN as compare to FWC

IV. CONCLUSION

This work presented how to minimize the overlapping of cluster by producing compact and well separated various width clusters, which increases the efficiency of the K Nearest neighbor search. The proposed approach produces the ideal amount of clusters as well as it has a density as per data size.

ACKNOWLEDGMENT

It gives me an immense pleasure to express my sincere and heartiest gratitude towards our guide **Prof. Dr. S. K. Shirgave**, Department of Information Technology for his guidance, encouragement and moral support during the course of our work. He has proven to be an excellent mentor and teacher. This work is also the outcome of the

blessings, guidance and support of our parents, family members and friends. Lastly our thanks to all who have contributed indirectly in completion of this work.

REFERENCES

[1]. G. Shakhnarovich, T. Darrell, and P. Indyk, "Nearest-neighbor methods in learning and vision," IEEE Trans. Neural Netw.,vol. 19, no. 2, p. 377, Feb. 2008.
[2]. F. Angiulli, S. Basta, and C. Pizzuti, "Distance-based detection and prediction of outliers," IEEE Trans. Knowl. Data Eng., vol. 18, no. 2, pp. 145–160, Feb. 2006.
[3]. P. Cunningham and S. J. Delany, "k-nearest neighbor classifiers," Multiple Classifier Systems, pp. 1–17, 2007.
[4]. S. Magnussen, R. E. McRoberts, and E. O. Tomppo, "Model-based mean square error estimators for k-nearest neighbor predictions and applications using remotely sensed data
[5]. M. Muja and D. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," IEEE Trans. Pattern Anal. Mach. Intell.,vol. 36, no. 11, pp. 2227–2240, Nov. 1, 2014.
[6]. S. A. Nene and S. K. Nayar, "A simple algorithm for nearest neighbor search in high dimensions," IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 9, pp. 989–1003, Sep. 1997.
[7]. J. H. Friedman, F. Baskett, and L. J. Shustek, "An algorithm for finding nearest neighbors," IEEE Trans. Comput., vol. C-24, no. 10,pp. 1000–1006, Oct. 1975.
[8]. C. T. Jr., A. Traina, B. Seeger, and C. Faloutsos, Slim-Trees: High Performance Metric Trees Minimizing Overlap Between Nodes. New York, NY, USA: Springer, 2000.
[9]. E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in Applications of Data Mining in Computer Security. New York, NY, USA:Springer, 2002, pp. 77–101.
[10]. M. J. Prerau and E. Eskin, "Unsupervised anomaly detection using an optimized k-nearest Neighbor's algorithm," Undergraduate thesis, Columbia Univ., New York, NY, USA, Dec. 2000.
[11]. W. Xueyi, "A fast exact k-nearest neighbor's algorithm for high dimensional search using K-means clustering and triangle inequality," in Proc. Int. Joint Conf. Neural Netw., 2011, pp. 1299.
[12]. A. Almalawi, A. Fahad and Z. Tari "An efficient k-nearest neighbors approach based on various-width clustering" in 2016

BIOGRAPHIES

Rajkumar Dadaso Gulpatil, has completed B.E. Computer Science and Engineering from Mumbai University. He is currently pursuing M.E. in Computer Science and Engineering at D. K. T. E.'s Textile and Engineering Institute, Ichalkaranji, India. His areas of interest include Data Mining and Security.

Prof.(Dr) Suresh K. Shirgave, working as a Professor in Computer Sc. & Engg. He has completed M.E. and PhD and having 25 years teaching experience in teaching. He has presented and published more than 20 research papers in International Conferences and Journals. His areas of research includes Data Mining, Web Mining, Security and Recommender Systems.

DOI 10.17148/IJARCCE.2019.8111