INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

DISTRIBUTED NEAREST NEIGHBOR MATCHING ALGORITHM FOR COMPUTER VISION

MISS. P. Y. MAHURKAR, DR. R. D. RAUT, DR. V. M. THAKARE

SGBAU, Amravati, India.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract: Highly efficient query processing on high-dimensional data is important and still a challenge nowadays as the curse of dimensionality makes efficient solution very difficult. On the other hand, there have been suggestions that it is better if one can return a solution quickly, that is close enough, to be sufficient. *K* Nearest Neighbors (*k*-NN) search is a widely used category of algorithms with applications in domains such as computer vision, machine learning and data analysis. This paper presents various nearest neighbor search techniques such as Randomized k-d forest, priority search k-means tree, R-forest and Subspace Clustering for Filtering (SCF), K-flat, degree-reduced k-nearest neighbor (k-DR) graph. To increase speed and efficiency in finding nearest neighbor for high dimensional data, this paper propose a distributed nearest neighbor matching algorithm for computer vision. The proposed method can give better performance in result.

Keywords: Randomized k-d forest, priority search k-means tree, R-forest SCF, K-flat, distributed nearest neighbor matching DKNN

Corresponding Author: MISS. P. Y. MAHURKAR

Access Online On:

www.ijpret.com

How to Cite This Article:

P. Y. Mahurkar, IJPRET, 2016; Volume 4(9): 1441-1448

PAPER-QR CODE

INTRODUCTION

Similarity search is an application that demands an efficient result. Through finding similar items within a known database, existing knowledge can be used for predicting unknown information. The most computationally expensive part of many computer vision algorithms consists of searching for the most similar matches to high-dimensional data, also referred to as nearest neighbor matching.

When working with high dimensional features, as with most of those encountered in computer vision applications, there is often no known nearest-neighbor search algorithm that is exact and has acceptable performance. To obtain a speed improvement, many practical applications are forced acceptable performance. To obtain a speed improvement, many practical applications are forced to settle for an approximate search, in which not all the neighbors returned are exact, meaning some are approximate but typically still close to the exact neighbors. Nearest-neighbor search is a fundamental part of many computer vision algorithms and of significant importance in many other fields, so it has been widely studied.

The nearest neighbor search problem can be defined as follows: given a set of points P = {p1, . . ., pn} in a vector space X, these points must be preprocessed in such a way that given a new query point q to X, finding the points in P that are nearest to q is performed efficiently. This paper discusses four methods i.e, priority search k-means tree, R-forest, SCF, K-flat and degree-reduced k-nearest neighbor (k-DR) graph. To improve the speed and performance of the algorithm, a distributed nature is adopted. The new method gives improvements to the existing ones. There is an interest in finding not only first neighbor but also several neighbors.

BACKGROUND

The randomized k-d forest algorithm is an approximate nearest neighbor search algorithm that builds multiple randomized k-d trees which are searched in parallel. When high precision is required, the priority search k-means tree is more effective for finding approximate nearest neighbors. Choosing k is important for obtaining a good search performance. This increases the performance due to less memeroty overhead [1]. In R-Forest method, the space is divided into multiple disjoint sub-regions and build an R-tree for data in each region. This method is useful for approximate high dimensional nearest neighbor search. The quality of result actually improves with larger value of k. Each R-tree will store a sub-set of points in a nonoverlapping space, which is maintained throughout the life of the forest. Median point used for ordering and searching a pruning parameter, as well as restricted access [2]. The SCF is used for K-NN search on multicore

platforms accurately. Instead of finding the likely candidates, this data filtering strategy excludes those unlikely features based on distance estimation. The SCF-based distance estimation depends on two data structures: the SCF index and a matrix of partial distances for the query feature [3]. The goal of k-flat method is to process the data so that it can give efficiently approximately nearest neighbor queries. Bur this is applicable only for the low dimensional data. When the dimension d is small, there are efficient solution. As d increases, these algorithms quickly become inefficient [4]. A fast approximate similarity search method based on a neighborhood-graph index. It works in two steps. First is for constructing a graph as a search index and the second for searching on the graph. The graph-construction builds a special neighborhood graph, which is called a degree-reduced k-nearest neighbor (k-DR) graph, from a given data set with a dissimilarity [5].

This paper introduced the efficient methods for nearest neighbor matching i.e. section I Introduction. Section II discuss Background. Section III discuss Previous work done. Section IV. discuss Existing methodology. Section V Analysis And Discussion. Section VI Proposed methodology and outcomes possible result finally section VII Conclude paper.

PREVIOUS WORK DONE

Marius Muja et al [1] has worked on randomized k-d forest, priority search k-means tree for high dimensional data. In randomized k-d forest search, a single priority queue is maintained across all the randomized trees. The priority queue is ordered by increasing distance to the decision boundary of each branch in the queue, so the search is to explore first the closest leaves from all the trees. The priority search k-means tree tries to better exploit the natural structure existing in the data, by clustering the data points using the full distance across all dimensions, in contrast to the Randomized k-d tree algorithm which only partitions the data based on one dimension at a time. Michael Nolen et al [2] has worked on R-Forest method for approximate high dimensional nearest neighbor queries. This method comprised of a set of disjoint R-trees built over the domain of the search space. In this method, d dimensions are selected to partition the space. But it is not suitable for higher dimensional data as well as insertion and deletion of points affects the overall R-Forest. Xiaoxin Tang et al [3] has worked on data filtering method named SCF for scalable high dimensional k-NN search. This method can effectively reduce the computation and memory footprints. SCF divides the space into subspaces and then create an index. Clustering is an important task as it aims at organizing data into homogeneous groups or clusters. Wolfgang Mulzer et al [4] has worked on a K-flat algorithm based on the two data structure the projection structure and the clusters. The projection structure works by projecting the point set to a space

of constant dimension and by answering the nearest neighbor query in that space. Also partition the point set into a sequence of clusters. A cluster consists of m points and a k-flat K such that all points in the cluster are close to K, where m is a parameter to be optimized. Kazuo Aoyama et al [5] has worked on a degree-reduced k-nearest neighbor (k-DR) graph constructed from the object set. In this method, there is a construction of graph as a search index and then searching is performed. The k-DR graph along its edges is constructed using a greedy search (GS) algorithm starting from multiple initial vertices with parallel processing. In the graph construction stage, the structural parameter k of the k-DR graph is determined. It works only for small and simple searching.

The proposed method is focused on a nearest neighbor matching in a distributed manner. It is a novel and robust method to mitigate problem. This interactive methodology can find nearest neighbor match very fast. It allows a user to achieve high performance for the desired data. When dealing with a large datasets, this method works by distributing the data into clusters. In future, the distributed nearest neighbor matching technique can be applied to the additional larger real data sets and higher dimensional data to enhance their robustness and quality. Experiments show that proposed system can handle a large variety of input data. For future studies, researchers are trying to minimize the time required for matching nearest neighbor.

EXISTING METHODOGY

In randomized k-d forest, a single priority queue is maintained across all the randomized trees. The priority queue is ordered by increasing distance to the decision boundary of each branch in the queue, so the search will explore first the closest leaves from all the trees. Once a data point has been examined as compared to the query point inside a tree, it is marked in order to not be reexamined in another tree. The degree of approximation is determined by the maximum number of leaves to be visited across all trees, returning the best nearest neighbor candidates. In priority search k-means tree, the tree is searched by initially traversing the tree from the root to the closest leaf, following at each inner node the branch with the closest cluster centre to the query point. The priority queue is sorted in increasing distance from the query point to the boundary of the branch being added to the queue [1]. In R-Forest, initially pick D dimensions (out of d) to partition the space. For each dimension we split the space into B non-overlapping regions. It cannot allow to have the number of trees to grow exponentially with the number of dimensions [2]. SCF divides the space into S subspaces, each of which has D/ S dimensions. The remainder of D/S can either be treated as an additional subspace, or these dimensions can be distributed to the other subspaces. Usually when S and C are increased, the estimation accuracy improves [3].

In k-flat method, for given a query flat f, the query can be answered directly in clusters whose radius is small compared to d(f,p). Clusters are classified as small and large depending upon their radius. The points in the large clusters are spread out and can be handled through projection [4]. In k-DR graph, there is a controlled k value as a search index, and explored the k-DR graph along its edges by using a greedy search method starting from multiple initial vertices with parallel processing [5].

DATASET:

In randomized kd-tree experiment, 100 SIFT features dataset is used. It is showed that the randomized kd-trees have a significantly better performance for true matches, when query features are likely to be significantly closer than other neighbors. High the FR, the more computation and memory accesses it reduces, which leads to better performance. On AMD64 multicore platform, after applying SCF to RKD algorithm, FR is high for some dataset SIFT and Digits by reducing both computation and memory accesses. For degree reduced graph, MNIST database of handwritten digits is used. It contains object set X, query se Q and quasi query set Q' which are disjoint from each other. Q' is used for graph construction and Q used to evaluate the search performance.

V. ANALYSIS AND DISCUSSION

When analyzing the complexity of the priority search kmeans tree, consider the tree construction time, search time and the memory requirements for storing the tree. For each internal node, the algorithm needs to find the branch closest to the query point, so it needs to compute the distances to all the cluster centers of the child nodes. The cost is measured as a combination of the search time, tree build time, and tree memory overhead. Depending on the application, each of these three factors can have a different importance [1]. The R-Forest locates significantly more exact nearest Neighbor. First, the quality of the results actually improves with larger value of K – i.e. the more aggressive the pruning, the more likely the R-Forest returns the better solution. Insertion and deletion are not possible in R-Forest [2]. The SCF method can effectively reduce computation and memory footprint. A metric are used to evaluate the performance and precision of SCF that is filtering rate which represents the percentage of features that can be filtered by SCF [3]. The k-flat method is applied efficiently only when dimension is small. There is a time complexity. It does not worked for very large dataset. Large radius cluster can not find the nearest neighbor [4]. k-DR graph performed a similarity search at an extremely low cost but with a high search success probability. But it is not used on a very large scale because of large time consumption [5].

Nearest neighbor search techniques	Advantages	Disadvantages
Randomized k-d tree	1) In this method, the trees are searched in parallel. 2) randomized k-d tree performed better for low dimensional data.	1)The main drawback of randomized k-d tree method is that increasing the number of random trees leads to decreasing the performance. 2) It casues memory overhead due to increasing multiple random trees.
Priority search k-means tree	1) When high precision is required, this method performs better solution. 2) In this method, clustering of data points is done across all the dimensions.	1) It requires large memory and more time to perform.
R-Forest	1)Data distribution is possible without any additional parameters. 2) It is effective, produces superior result and there is not duplication of data.	1) When k is small, this method fails.
Degree reduced k nearest neighbor	1) This method is used for small problems which want quick answer. 2) It is easy to implement.	1) Large and complex cases do not give suitable answer. 2) It requires longer time.
K-flat NN search	1)For small data, it is easy to search nearest neighbor.	1)Inefficient for large data. 2)Space complexity and time complexity.

Table 1: Comparison between different nearest neighbor searching techniques

PROPOSED METHODOLOGY: Distributed nearest neighbor matching

The proposed method DKNN allows the distributed processing of K-NN queries. For a given query q, DKNN algorithm use the clustering concept to directly locate the k neighbors of q. In this manner, the k nearest neighbors are matched from the datasets. DKNN avoids having an uncertain number of expensive iterations and is thus more efficient and more predictable.

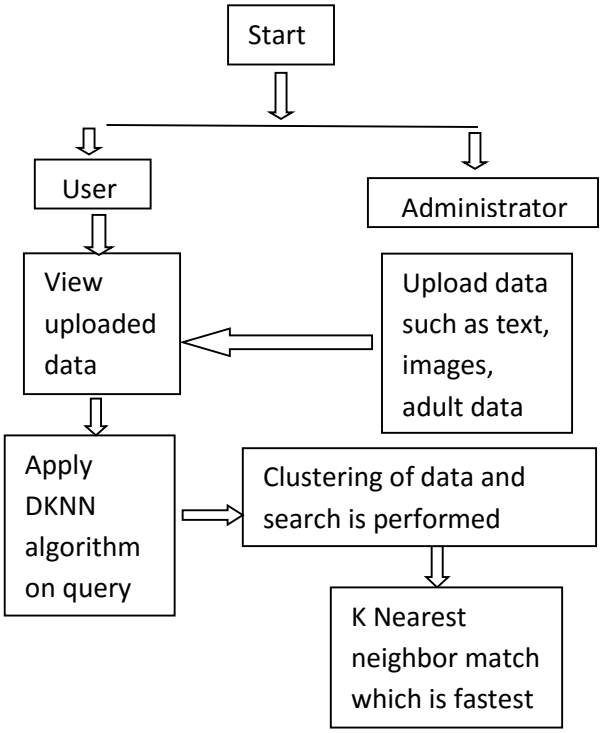

Fig 1. Distributed nearest neighbor matching technique

Fig. 1 shows the overall architecture that reads a data, processes it, and outputs the nearest neighbors. The basic blocks are responsible for determining the query type, clustering the data and matching nearest neighbor of size k within a very short time. To scale very large data sets, there is a need to distribute data in the clusters and perform nearest neighbor search.

The proposed algorithm focuses on matching of nearest neighbors as the query is fired. This operation can be performed on various data such as text, images and adult data i.e. an unordered sequence of data.

Insert data and cluster it.

The data is inserted into the database and cluster it according to the type. Clustering is nothing but the collection of similar contents.

Determine the type of data

A query q is inserted for matching in the data.

Matching nearest neighbor

In this step, the query q is matched with the data within a cluster and gives the nearest neighbor of size k

OUTCOME AND POSSIBLE RESULT

The result of this method focused on finding the distributed nearest neighbor efficiently and fast. The proposed algorithm performed well and efficiently as compared to the existing methods. The purpose here is to work on a very large datasets and within a short time. Here is a proper memory usage and requires less iteration because of clustering. This method is the most accurate one.

VII. COCLUSION

Thus, this method focused on matching nearest neighbors within a distributed data among the clusters. The performance of this method is predictable as compared to the other techniques. The output of this method provide best result as it requires less computation time.

FUTURE SCOPE

In future, this method can be applied on a very large set of data with different dimensions. Researchers are working to extend and enhance the robustness of the method.

REFERENCES

1. Marius Muja, Member, IEEE and David G. Lowe, Member, IEEE, “Scalable Nearest Neighbor Algorithms for High Dimensional Data”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 36, No. 11, November 2014
2. Michael Nolen and King-Ip Lin, “Approximate High-Dimensional Nearest Neighbor Queries Using R-Forests”, IDEAS '13, October 09 - 11 2013, Barcelona, Spain Copyright 2013 ACM.
3. Xiaoxin Tang, Steven Mills, David Eyers, Kai-Cheung Leung, Zhiyi Huang and Minyi Guo, “Data Filtering for Scalable High-dimensional k-NN Search on Multicore Systems”, HPDC’14, June 23–27, 2014, Vancouver, BC, Canada. Copyright 2014 ACM
4. Wolfgang Mulzer and Huy L. Nguyen, Paul Seiferth, “Approximate k-flat Nearest Neighbor Search”, STOC’15, June 14–17, KDD’11, August 21-25, 2011, San Diego, California, USA. Copyright 2011 ACM.
5. Kazuo Aoyama, Kazumi Saito, Hiroshi Sawada and Naonori Ueda, “Fast Approximate Similarity Search Based on Degree-Reduced Neighborhood Graphs”, 2015, Portland, Oregon, USA Copyright ACM.