A NEW K-NEAREST NEIGHBORS CLASSIFIER FOR N-DIMENSIONAL DATASETS USING MAP REDUCE

Venkata Rajani Katuri
Research Scholar
Shri JJT University
Rajasthan

Dr. M. Sridevi
Associate Professor,
Department of CSE
CVR College of Engineering,
Mangalpally,
Ibrahimpatnam, R R district, T S

Dr. Prasadu Peddi
Professor
Shri JJT University
Rajasthan

Abstract:

The K-nearest neighbors (KNN) machine learning algorithm is a well-known non-parametric classification method. However, like other traditional data mining methods, applying it on big data comes with computational challenges. Indeed, KNN determines the class of a new sample based on the class of its nearest neighbors; however, identifying the neighbors in a large amount of data imposes a large computational cost so that it is no longer applicable by a single computing machine. One of the proposed techniques to make classification methods applicable on large datasets is pruning. LC-KNN is an improved KNN method which first clusters the data into some smaller partitions using the K-means clustering method; and then applies the KNN for each new sample on the partition which its center is the nearest one. However, because the clusters have different shapes and densities, selection of the appropriate cluster is a challenge. In this paper, an approach has been proposed to improve the pruning phase of the LC-KNN method by taking into account these factors. The proposed approach helps to choose a more appropriate cluster of data for looking for the neighbors, thus, increasing the classification accuracy. This topic is known as big data classification, in which standard data mining techniques normally fail to tackle such volume of data. In this contribution we propose a Map Reducebased approach for k-Nearest neighbor classification.

Keywords: *K-nearest neighbors; KNN; classifier, classification.*

Introduction

The k-Nearest Neighbor algorithm (k-NN) is considered one of the ten most influential data mining algorithms. The classification of big data is becoming an essential task in a wide variety of fields such as biomedicine, social media, marketing, etc. The recent advances in data gathering in many of these fields have resulted in an inexorable increment of the data that we have to manage. The volume, diversity and complexity that bring big data may hinder the analysis and knowledge extraction processes. The MapReduce framework highlights as a simple and robust programming paradigm to tackle large-scale datasets within a cluster of nodes. MapReduce is a very popular parallel programming paradigm that was developed to process and/or generate big datasets that do not fit into a physical memory. Characterized by its transparency for programmers, this Framework enables the processing of huge amounts of data on top of a computer cluster regardless the underlying hardware or software. This is based on functional programming and works in two main steps: the map phase and the reduce phase. In a MapReduce program, all map and reduce operations run in parallel. First of all, all map functions are independently run. Meanwhile, reduce operations wait until the map phase has finished. Then,

they process different keys concurrently and independently. Note that inputs and outputs of a MapReduce job are stored in an associated distributed file system that is accessible from any computer of the used cluster.

With the emergence of big data applications, traditional software tools are no longer able to process and manage them in an acceptable time. Exploring large amounts of data and extracting useful information or knowledge is one of the most challenging problems for big data applications. For solving this problem, two general solutions have been proposed; one approach is to distribute the data among different processing machines and do calculations on them simultaneously. By so doing, the high processing capacity of the parallel or distributed system is employed to address the problem of processing large datasets. Another approach is to prune the data and, thus, computation. Indeed, in this way, an attempt is made to reduce the size of the training dataset, which is the input of the learning algorithm in order to be manageable and processable by a single computing machine. One of the popular and useful data mining methods is the KNN classifier. KNN classifies each test sample based on its k nearest neighbors. For finding the k nearest neighbors, the distance between the test samples and all training ones should be calculated. In the case of big data, this requires a vast amount of computational overhead. Some researchers employ distributed frameworks, such as Hadoop, to find the k nearest neighbors overall training samples. These approaches usually lead to finding the exact k nearest neighbors, but at the cost of using a large distributed system. However, some other researchers propose to search for the nearest neighbors in a reduced training dataset. For instance, in the research conducted by Deng et al., k-means clustering algorithm was used to partition data into k clusters. The cluster whose center has the minimum distance from the test sample is selected as the suitable reduced training dataset. The quality of the KNN's classification depends on how well the nearest neighbors are found. The chance of finding the exact k nearest neighbors also depends on how well the large dataset has been pruned.

Literature review

Dawen Xia et al (2016) In big-data-driven traffic flow prediction systems, the robustness of prediction performance depends on accuracy and timeliness. This paper presents a new MapReduce-based nearest neighbor (NN) approach for traffic flow prediction using correlation analysis (TFPC) on a Hadoop platform. In particular, we develop a real-time prediction system including two key modules, i.e., offline distributed training (ODT) and online parallel prediction (OPP). Moreover, we build a parallel k-nearest neighbor optimization classifier, which incorporates correlation information among traffic flows into the classification process. Finally, we propose a novel prediction calculation method, combining the current data observed in OPP and the classification results obtained from large-scale historical data in ODT, to generate traffic flow prediction in real time.

Madhavi Bhamare et al (2017) The k-Nearest Neighbor classifier is one of the most well known methods in data mining because of its effectiveness and simplicity. Due to its way of working, the application of this classifier may be restricted to

problems with a certain number of examples, especially, when the runtime matters. However, the classification of large amounts of data is becoming a necessary task in a great number of real-world applications. This topic is known as big data classification, in which standard data mining techniques normally fail to tackle such volume of data. In this contribution we propose a Map Reducebased approach for k-Nearest neighbor classification.

Hamid Saadatfar (2020) The K-nearest neighbors (KNN) machine learning algorithm is a well-known non-parametric classification method. However, like other traditional data mining methods, applying it on big data comes with computational challenges. Indeed, KNN determines the class of a new sample based on the class of its nearest neighbors; however, identifying the neighbors in a large amount of data imposes a large computational cost so that it is no longer applicable by a single computing machine. One of the proposed techniques to make classification methods applicable on large datasets is pruning. LC-KNN is an improved KNN method which first clusters the data into some smaller partitions using the K-means clustering method; and then applies the KNN for each new sample on the partition which its center is the nearest one. However, because the clusters have different shapes and densities, selection of the appropriate cluster is a challenge.

KNN

In pattern recognition, the k-nearest neighbors algorithm (kNN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression: In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

k-NN Classifier

As similarity-based strategy, k-closest neighbor classifier (k-NN) is connected with our substance gathering. From the earliest starting point organize, classifier figures k for the most part all around that truly matters darken reports (i.e., k closest neighbors) test archive being administered. The closeness of this report to class is figured by social event all around that genuinely matters undefined properties records among k documents, whose classes are crude from such class. The test report is doled command that has most raised closeness to record. Two parameters included are centrality consistency and the number k.

While standard comparability is spread out tf×idf, gathering (0.5+0.5tf/tf max)×idf that performed better in our starter tests, related in this work. The parameter k is energized by evaluations as showed up in going with part.

MapReduce

MapReduce is style setting up that has been executed in couple of structures, including Google's internal use (on extremely essential level called MapReduce) and the standard open-source execution Hadoop which can be gotten,

close HDFS record framework from Apache Foundation. You can utilize utilization of MapReduce to deal with some mammoth scale cements into way that is tolerant gadget insufficiencies. All you have to make are two constrainment, called Map and Reduce, while framework oversees parallel execution, coordination errands that execute Map or Reduce, what's more controls likelihood that one of the available assignments would dismissal for execution. In nutshell, Map_Reduce calculation would execute as compasses for after:

1. About some number of Map assignments each were given in any occasion one projection from coursed record structure. These Map assignments change inconsistency into methodology of key-respect sets. The way wherein where key respect sets are passed on from information is obliged by code made by client for Map work.
2. The key-respect sets from each Map undertaking were gathered by expert controller and filtered through by key. The keys were isolated among all Reduce endeavors, so all key respect sets with relative key breeze up at dubious Reduce task.
3. The Reduce assignments control one key at some unconventional moment, and join lion's offer characteristics related with that key, it were, or another or another. The technique for blend respects is coordinated by the code made by client for Reduce work.

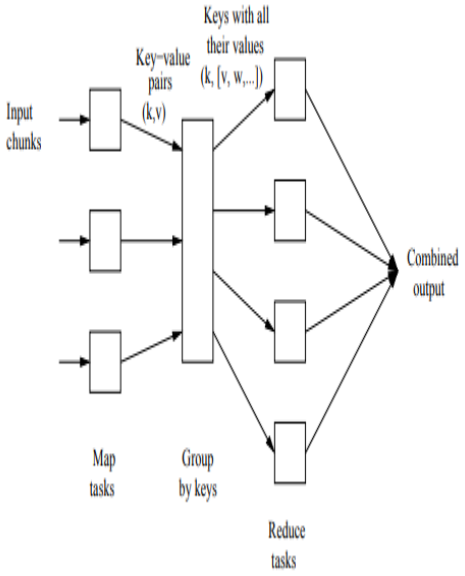


Figure: Schematic MapReduce computation

Classification of MapReduce Algorithms

Srirama, et al contemplated adjusting logical registering to the MapReduce programming model. They sort out considering along with four classes

• The most raised bring up keeps an eye out for checks that may be adjusted as unequivocal accomplishment of MapReduce models; considering whole numbers is case this kind of figurings.

• The underneath normal watches out for estimations that can be balanced as exceptional execution dependable number MapReduce models; Clustering of Large Applications (CLARA) is instance this class of checks.

• The underneath standard class keeps an eye out for iterative estimations; where every segment is tended to as execution specific MapReduce model; gathering figurings, for instance, Dividing Around Medoids (PAM) addresses an occasion of this sort of checks.

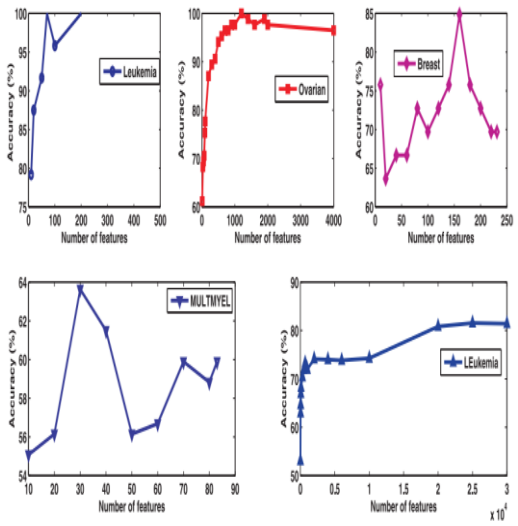
• The fourth class of excludes keeps an eye for complicate iterative checks where the substance of one segment is tended to as

accomplishment of various Map_Reduce models. Conjugate Gradient (CG) is an occasion of this kind of checks.

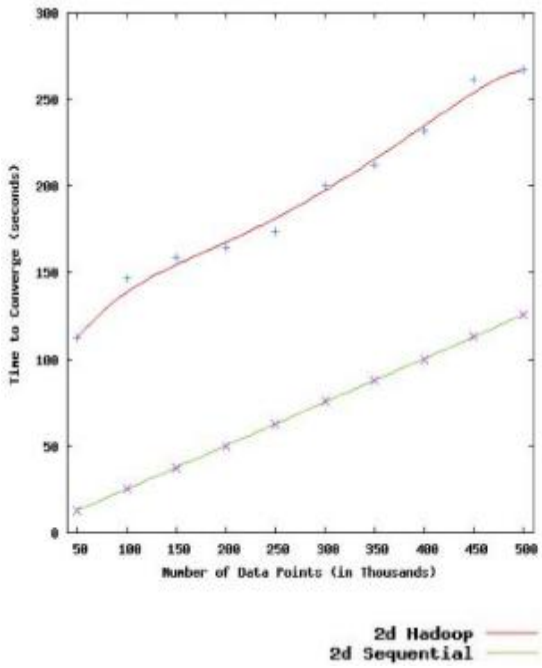
Results and discussion

Analysis of MapReduce based KNN classifier

After territory choice, proposed assembling check MapReduce based K-NN has been connected with referencing datasets. Regardless, with special case in occasion that one causes them to learn dataset, it is hard to pick ideal number of highlights required for depiction. To beat this issue, forward part choice procedure is considered, in which top arranged highlights appearing differently in relation to rising p-values are utilized. Various subsets top arranged highlights are utilized to organize microarray dataset utilizing MapReduce based K-NN and their comparing order correctnesses are figured. At the point when the examples are successively chosen, model planned might be over-prepared or under-prepared. This is on grounds that the examples chose for preparing may contain either just dangerous or just non-harmful examples. To evade this, diminished datasets are isolated for preparing and testing purposes in accompanying manner: each third example is extricated for testing reason and the remainder information tests are utilized for preparing tests.

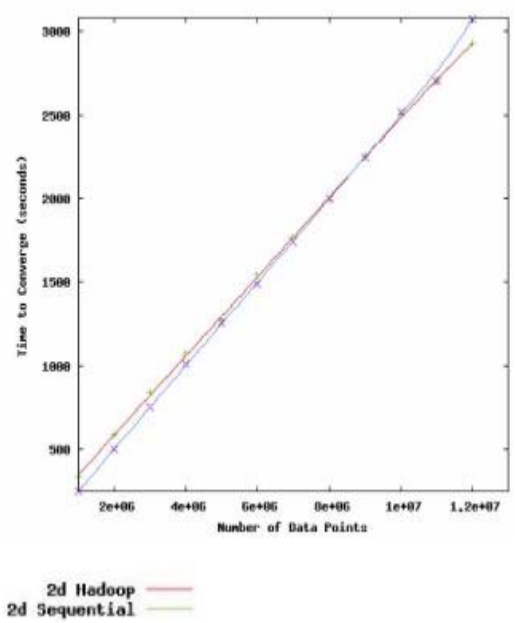


Graph: Testing precision with various arrangement of highlights utilizing diverse microarray dataset



Graph: Consecutive kNN versus Map Red kNN for little measured dataset

The primary perception produced using test results was that straightforward consecutive kNN arrangement took lot lesser time contrasted with MapReduce kNN portrayal for more diminutive datasets. This truth is appeared by diagram in Figure. The chart in Figure shows similar plot time taken for upto 12 million utilizing MapReduce kNN.



Graph: Progressive kNN versus Map Red kNN for huge assessed dataset

It shows that Hadoop MapReduce performed superior to progressive kNN from 8 million server farms which is about 60MB in size. By 12 million and 90MB size, it was discovered that Hadoop finished kNN strategy 150 seconds before back to back kNN.

Conclusion

In this, we have proposed a MapReduce approach to enable the k-Nearest neighbor technique to deal with large-scale datasets. Without a parallelization, the application of the kNN algorithm would be limited to small or medium data, especially when low runtimes are a need. The proposed scheme is an exact parallelization of the k-NN model, so that, the precision remains the same and the efficiency has been largely improved. Features of Hadoop advancement are HDFS and MapReduce where HDFS gives voluminous, get-together information dealing with and MapReduce gives exuberant information managing. The objective of paper is to harden potential usage of MapReduce in the preparing of research information. Guide Reduce customarily underpins pack based high scale parallelization reliable coordinating. The Hadoop MapReduce structure utilizes an appropriated record system to look into make its data. In this way, I/O execution Hadoop MapReduce work unequivocally depends upon HDFS. In this appraisal producers are dissected most explored zone and least inspected region of looking. Outcomes of concentrate can be valuable for aces for their evaluation.

References

1. *Saadatfar, H.; Khosravi, S.; Joloudari, J.H.; Mosavi, A.; Shamshirband, S. A New K-Nearest Neighbors Classifier for Big Data Based on Efficient Data Pruning. Mathematics 2020, 8, 286. https://doi.org/10.3390/math8020286.*
2. *Madhavi Bhamare et al (2017), K-Nearest Neighbor Approach Based On Map Reduce for Big Data Classification, International Journal for Research in Engineering Application & Management, Engineering Application & Management (IJREAM) ISSN: 2454-9150 Vol-03, Issue 04.*
3. *Dawen Xia et al (2016), A MapReduce-Based Nearest Neighbor Approach for Big-Data-Driven Traffic Flow Prediction, IEEE, ISSN NO: 2169-3536, VOLUME 4, Digital Object Identifier 10.1109/ACCESS.2016.2570021.*
4. *29. D. Muti and S. Bourennane. Survey on tensor signal algebraic filtering. Signal Process., 87(2):237–249, 2007*
5. *30. E. Acar, S. A. C¸ amtepe, M. S. Krishnamoorthy, and B. Yener. Modeling and multiway analysis of chatroom tensors. In ISI, pages 256–268, 2005.*
6. *31. E. Acar, T. G. Kolda, and D. M. Dunlavy. An optimization approach for fitting canonical tensor decompositions. Technical Report SAND2009-0857, Sandia National Laboratories, 2009.*
7. *32. E. Acar and B. Yener. Unsupervised multiway data analysis: A literature survey. TKDE, 21(1):6–20, 2009.*

8. 33. *E. Hoke, J. Sun, J. D. Strunk, G. R. Ganger, and C. Faloutsos. Intemon: Continuous mining of sensor data in large-scale self-* infrastructures. ACM SIGOPS Operating Systems Review, 40(3), 2003.*