ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017

8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)

Organized by

Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

Survey on K-Nearest Neighbour Joins for Big Data on MapReduce

Annapoorna P S[1], K Aafreen[1], Preethi J D[2], Dr Jitendranath Mungara[3]

B.E Student, Dept. of ISE, New Horizon College of Engineering, Bangalore, Karnataka, India[1]

Assistant Professor, Dept. of ISE, New Horizon College of Engineering, Bangalore, Karnataka, India[2]

HoD, Dept. of ISE, New Horizon College of Engineering, Bangalore, Karnataka, India[3]

ABSTRACT: The k-Nearest neighbor classifier is one of the most well know methods in Data mining because of its effectiveness and simplicity.The classification of large amounts of data is becoming a necessary task in a great number of real-world applications. In this contribution we propose a mapreduce-based approach for k-nearest neighbor classification. This model allows us to simultaneously classify large amounts of unseen cases against a big dataset .The map phase will determine the k-nearest neighbor in different splits of the data, the reduce phase stage will compute the definitive neighbors from the list obtained in the mapphase.We introduce a new algorithm for processing queries that define similarity in terms of multiple transformations instead of single one. In this paper, we compare the different existing approaches for computing knn on map reduce, first theoretically and then by performing an extensive experimental evaluation .To be able to compare solutions, we identify three generic steps for knn computations on map reduce data pre-processing, data partitioning and computation ,can be used as a guide to the tackle knn-based practical problems in the context of big data.

KEYWORDS: knn, map reduce, data pre-processing, data partitioning, computation.

I. INTRODUCTION

"Big data" is a term used to describe a collection of data sets with the following three characteristics:

1 .Volume- Large amounts of data generated.

2. Velocity-Frequency and speed of which data are generated, captured and shared 3 .Variety- Diversity of data types and formats from various sources.

The size and complexity of big data makes it difficult to use traditional database management and data processing tools. Data is being created in much shorter cycles from hours to milliseconds. There is also a trend underway to create larger databases by combining smaller data sets so that data correlations can be discovered.

II. BIG DATA ANALYTICS

Big data has become the new frontier of information management given the amount of data today's systems are generating and consuming.

III. BIG DATA COMPUTING

The rising importance of big-data computing stems from advances in many different technologies. Sensors: Digital data are being generated by many different sources, including digital imagers (telescopes, video cameras, MRI machines), chemical and biological sensors), and even the millions of individuals and organizations generating web pages.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017

8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)

Organized by

Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

IV. KEY TECHNOLOGIES FOR EXTRACTING BUSINESS VALUE FORM BIG DATA

Big data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data by enabling high-velocity capture, discovery and/or analysis. Storage and processing technologies have been designed specifically for large data volumes. Computing models such as parallel processing, clustering, virtualization, grid environments and cloud computing, coupled with high-speed connectivity, have redefined what is possible. Here are three key technologies that can help you g1.

Time-series data are of growing importance in many new database applications, such as data mining or data warehousing. A time series is a sequence of real numbers, each number representing a value at a time point. For example, the sequence could represent commodity prices, sales, exchange rates, weather data, biomedical measurements, etc. We are often interested in similarity queries on 1time-series data .For example, we may want to find stocks that behave in approximately the same way (or approximately the opposite way, for hedging);or products that had similar selling patterns during the last year; or years when the temperature patterns in two regions of the world were similar. In queries of this type, approximate, rather than exact, matching is required.et a handle on big dataand even more importantly, extract meaningful business value from it. Information management for big data.[2]

KNN:

The k-nearest neighbour algorithm is considered one of the ten most influential data mining algorithms. This method requires that all of the data instances are stored and unseen cases classified by finding the class labels of k closest instances to them. To determine how close two instances are, several distance or similarity measures can be computed .This operation has to be performed for all the input examples against the whole training dataset. Thus, the response time may become comprised when applying it in the big data context. Given a set of query points R and a set of reference points S, at nearest neighbour join (hereafter kNN) is an operation which, for each point in R, discovers the k nearest neighbours in S. It is frequently used as a classification or clustering method in machine learning or data mining. It can be applied to a large number of fields, such as multimedia, social network time series analysis , bio-information and medical imagery . The basic idea to compute a kNN is to perform a pairwise computation of distance for each element in R and each element in S.

The computational complexity of this pairwise calculation is O(|R|×| S|). Then, finding the k in S for every r in R boils down to sorting the computed distances, and leads to a complexity of at least |S|×log|S|. As the amount of data or their complexity (number of dimensions) increases, this approach becomes impractical. This is why a lot of work has been dedicated to reducing the kNN computational complexity [3]. Data partition step: At the beginning, they partition the points in both R and S into R = {R1,...,RM} and S = {S1,...,SM}, respectively, based on a partitioning method such as random partitioning, grid partitioning, Voronoi diagram partitioning or clustering Another algorithm using grid based partitioning and principal component analysis is developed. Meanwhile, a kNN join algorithm by using SQL queries without modifying a database engine is proposed in [8]. Since it is difficult to provide efficient and scalable distributed indexes in distributed environments, the above algorithms are not suitable to be parallelized.[4]. The pre-processing stage aims either at processing the initial dataset to reduce its complexity or produces extra information about the data. It is an optional step, as some algorithms can directly work on the initial data, as HBkNNJ, H-BNLJ and H-BRJ. The solution consists in mapping high-dimensional data to low-dimensional ones while maintaining the locality relationship between each object with high probability (two elements that are close in a high-dimensional space should remain close in the reduced dimensional space). Computation The general idea to compute a kNN, is to (i) calculate the distance between ri and sj for all i and all j, and (ii) sort these distances in ascending order to find out the first k results. In Map

Reduce, the idea is the same except that what is given to a task must be independent in order to ensure correctness without duplicating the whole dataset [3].

MAPREDUCE: It is a parallel programming model that aims at efficiently processing large datasets. This programming model is based on three concepts:

1 Representing data as key value pairs 2. Defining a map function 3. defining a reduce function [3]. Let TR a training

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017

8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)

Organized by

Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

set and TS a test set of a arbitrary sizes that are sorted in the HDFS as single files. The map phase starts diving the TR set into a given number of disjoint subsets. In Hadoop, files are formed by h HDFS blocks that can be accessed from any computer of the cluster independently of its size. Let m be the number of map processes that will be defined by the end user. Each map task (Map1, Map2 ….Mapm) will create an associated TRj, where 1<j<m, with the sample of chunk in which the training set file is divided. It is noteworthy that this partitioning process is sequentially performed , so that the Mapj corresponds to the j data chunk of h/m blocks. As a results, each map analyse approximately a similar number of training instances. As each map finishes its processing the results are forwarded to a single reduce task .The output key in every map is established according to an identifier value of the mapper. Reduce phase consists of determining which of the tentative k nearest neighbour from the maps are the closest ones for the complete TS. Given that we aim to design a model that can scale the arbitrary size training sets and that it is independent of the selected number of neighbour [1].

[1]. A MAP REDUCE K-NEAREST NEIGHBOR APPROACH FOR BIG DATA CLASSIFICATION

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 reduce–based approach for Knn classification. This model allows us to simultaneously classify large amounts of unseen cases against a big datasets. Reducing the search space for big data mining for interesting patterns from uncertainty. When the number of mappers is increased because of the implementation performed our proposal always provides the same accuracy than the original Knn model experimental framework. Taking the advantage of the setup, reduce and clean up procedures of the reduce task. We will quantify the time spent by MR-KNN in map and reduce phases as well as the global time to classify the whole dataset. As we have claimed before MR-KNN uses one single reduce, since time complexity is more. Due to its way of working, the applications of this classifiers may be restricted to problems with a certain number of example especially, when the runtime matters .simulation on data but not on big data.

[2].ON SIMILARITY-BASED QUERIES FOR TIME SERIES DATA

This set of transformation is rich enough to formulate operation such as moving average and time scaling .we present a new algorithm for processing queries that define similarity in terms of multiple transformation instead of a single one. Results on both synthetic and real data show that new algorithm for simultaneously processing multiple transformations is much faster than sequential scanning. In this paper we propose a fast algorithm to process queries that specify more than one

transformation as the basis for similarity .To achieve this goal we construct minimum bounding rectangle for transformations, can be applied for multidimensional. Itis often interested in similarity queries on time series data for example:-find stocks that have in approximately the same way. similarity queries that such a simple notion of similarity fails to capture.Although itseems if two sequence are similar w.r.t n-day moving average, they should be similar w.r.t(n+1) day moving average ,this is not true in general;a counter example can be found in the extended version.

[3].SOLUTIONS FOR PROCESSING K-NEAREST NEIGHBOR JOINS FOR MASSIVE DATA ON MAPREDUCE

It is a computational intensive task with a large range of applications such as knowledge discovery or data mining .The map reduce programming model because it is suitable for large scale data processing and easily be executed in a distributed environment .It can be applied to a large number of fields such as multimedia ,social network, time series analysis ,bio information and medical imagery .Map reduce is flexible and scalable parallel and distributed programming paradigm which is especially designed for data intensive processing .Map reduce adapted to large scale data processing which can easily be distributed.It seems to be a good target for computing KNN in distributed way. There is no readily available comparison to help users choose the one most appropriate for their needs As the amount of data or their complexity increases, this approach becomes impractical. There are still significant limitations to process KNN on a single machine when the amount of data increases.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 5, June 2017

8th One Day National Conference on Innovation and Research in Information Technology (IRIT- 2017)

Organized by

Departments of ISE, CSE & MCA, New Horizon College of Engineering, Bengaluru, Karnataka 560103, India

[4]. *PARALLEL COMPUTATION OF K-NEAREST NEIGHBOR JOINS USING MAPREDUCE*

The k-nearest neighbor join has recently attracted considerable attention due to its broad applications. The processing kNN joins is very expensive due to the quadratic nature of the join operation .there is an increasing trend of applications to deal with big data ,computing KNN joins becomes more challenging. In order to process such big data to process such big data, parallel and distributed computing using map reduce recently have received a lot of attention. In this paper, we propose the efficient parallel algorithm KNN-MR to process the KNN joins using .To reduce not only the computational cost of KNN joins but also the network cost of communicating across machines, we develop the novel vector projection pruning which enables us to identify non kNN points that are guaranteed not to be included in the result of a kNN join.

V. CONCLUSION

In this paper, we have studied existing solutions to perform the kNN operation in the context of Map Reduce. We have approached this problem from a workflow point of view. We have pointed out that all solutions follow three main steps to compute kNN over Map Reduce, namely pre-processing of data, partitioning and actual computation. We have listed and explained the different algorithms which could be chosen for each step, and developed their pros and cons, in terms of load balancing, accuracy of results, and overall complexity. In a second part, we have performed

extensive experiments to compare the performance, disk usage and accuracy of all these algorithms in the same environment. We have mainly used two real datasets, a geographic coordinates one (2 dimensions) and an image based one (SURF descriptors, 128 dimensions). For all algorithms, it was the first published experiment on such high dimensions. Moreover, we have performed a fine analysis, outlining, for each algorithm, the importance and difficulty of fine tuning some parameters to obtain the best performance. Overall, this work gives a clear and detailed view of the current algorithms for processing kNN on Map Reduce. It also clearly exhibits the limits of each of them in practice and shows precisely the context where they best perform. Above all, this paper can be seen as a guideline to help selecting the most appropriate method to perform the kNN join operation on Map Reduce for a particular use case. After this thorough analysis, we have found a number of limitations on existing solution which could be addressed in future work. First, besides H-BkNNJ, all methods need to replicate the original data to some extend. The number of replications, although necessary to improve precision, has a great impact on disk usage and communication overhead. Finding the optimal parameters to reduce this number is still an open issue. Second, the partitioning methods are all based on properties of R. However, one can expect R to vary as it represents the query set. The cost of repartitioning is currently prohibitive so, for dynamic queries, better approaches might rely on properties of S. Finally, MapReduce, especially through its Hadoop implementation, is well suited for batch processing of static data. The efficiency of these methods on data stream has yet to be investigated.

REFERENCES

[1].D. Li, Q. Chen, and C.-K. Tang, "Motion-aware knnlaplacian for video matting," in ICCV'13,2013.
[2] H.-P.KriegelandT.Seidl,"Approximation-basedsimilaritysearch for 3-D surface segments," Geoinformatica,1998.
[3] X. Bai, R. Guerraoui, A.-M.Kermarrec , and V. Leroy, "Collaborative personalized top-k processing," ACM Trans. Database Syst., 2011.
[4] D. Rafiei and A. Mendelzon, "Similarity-based queries for time series data," SIGMOD Rec., 1997.
[5] R. Agrawal, C. Faloutsos, and A. N. Swami, "Efficient similarity search in sequence databases," in Foundations of Data Organization and Algorithms, 1993.
[6] R.Agrawal, C.Faloutsos, and A.Swami. Efficient similarity search in sequence databases. In Proc. of the 4th Intl. Conf. on Found.of Data Org. and Alg. (FODO '93), pages69–84, Chicago, October 1993.
[7]. R. Agrawal, K.-I. L in, H.S. Sawhney, and K.Shim. Fast similarity search in the presence of noise, scaling ,and translation in time-series databases. In Proc. of the 21st Intl. Conf. on Very Large Data Bases (VLDB '95), pages 490–501,Zurich, September 1995. Morgan Kaufmann Publishers.
[8] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," SIGMOD Rec., vol. 29, no. 2, pp. 427–438, 2000.
[9] C. B¨ohm and F. Krebs, "Supporting KDD applications by the k-nearest neighbor join," in DEXA, 2003.
[10] M. Mineli, M, Chambers, and A.Dhiraj, Big data ,Big Analytics Emerging Business Intelligence and Analytics Trends for Today's Business ,Wiely Publishing ,2013.