International Journal of Engineering Research and General Science Volume 3, Issue 6, November-December, 2015
ISSN 2091-2730

Survey Paper For Real-Time Clustering for Big Data Streams

Bhavik Patel

GTU PG School Ahmedabad, pbhavik6518@gmail.com and +919429660402

Abstract — Big data is a recent term Appeared that has to define the vey large amount of data that surpass the traditional storage and processing requirements. Each and every growing volume of data generation is the reality. Today we are living in Social networks, smart cities, telephone networks, the internet are hand Reviews some of the data in the modern world and much of this information is discarded due to the high storage space. It Would require Relevant data can be extracted from this large amount of information and to be used to build better cities, offers better services, make predictive analysis, group similar information and many more applications. All of this is possible, due to machine learning and data mining can be found where patterns in the ocean of data generated every Second in order to cope with the volume, velocity and variety of data produced a streaming model has-been Studied. Were analysis of data has to uses low memory and process items only once.

Currently we are taking the advance of grid and cloud computing the missing component to help crunch this large amount of data is the power of distributed computing. Stream Processing Engines (SPE) have revealed to be a more flexible and powerful tool for dealing with Big Data.

This project merges the concepts of machine learning, streaming model and distributed computing to build a framework for developing, Testing and applying algorithms on large volume of data streams.

Keywords—

INTRODUCTION

The Internet is a worldwide platform for sharing information and making business. It has become a crucial medium for people and companies to connect and present themselves to the world. In this heterogeneous scenario data is represented in many forms and created by many sources. The large amount of information stored and produced is increasing every day and pushing the limits of storage and processing. Only in 2012 there was a rate of 2.5 quintillion bytes of data (1 followed by 18 zeros) created per day. Social networks, sensor networks, e-commerce and other data producing systems generate massive amounts of bytes per second. All of this information can and should be used for analyzing customer behavior, predicting trend and decision-making.

Map Reduce [2] is a programming model presented by Google for pro cessing large amounts of data in a distributed fashion that can be run on commodity resources, thus scaling the processing power to hundreds or thousands of machines. Map Reduce is and innovative way of parallelizing processing jobs where every problem has to be adapted into a two step process: a mapping phase and a reducing phase. The mappers takes an input key/value pair and produce a set of key/value pairs that are passed to the reducing job, which will merge the values into a smaller set. The open source \twin brother" of Google/Map Reduce and GFS are the Apache Hadoop1 and Hadoop Distributed File System (HDFS) projects [3] started at Yahoo!. A next level of the Map Reduce model are platforms for processing data in streaming where disk I/O operations are reduced for not using _les as its source and storage.

Patterns and relations can be extracted from data using methods from machine learning (ML) and data mining. Machine learning techniques for classifying information and clustering similar data are some of the main goals of these two areas of study. Such techniques are used in many ways and for different purposes. For example, machine learning algorithms can render very accurate predictive models that can be used to predict the price of housing depending on the size or location. Data mining algorithms on the other hand can deduce if a new piece of information is related to information. The literature is vast in machine learning and data mining algorithms and they have been developed in various programming languages. Tools and frameworks are also available as commercial and open source products. This variety of algorithms, tools and frameworks exists because there is no "one-size fits-all" solution, it depends on the problem scope, the volume of data and the complexity of the data. This works will focus on the volume and use some data mining techniques that use one common clustering algorithm - the k-means.

The scope of realtime big data analysis deals with using machine learning algorithms on unbounded streams of data. Data streams can be generated by many di_erent sources such as social networks, sensors, internet traffic, video and many others. To deal with this large volume of possibly unbounded flow of data some distributed stream processing platform has been implemented such as Apache S43 and Twitter Storm 4. These platforms can be considered an evolution of the batch, MapReduce, distributed file system model in the sense that they process owing data instead of always writing and reading from files.

Literature Review

The areas involved in this project are machine learning, data mining, streaming model and distributed systems. In these domains some interesting challenges appear in the present information society. The volume of data produced is enormous and much of the produced information is not used due to the lack of resources to store and process them. It would be too expensive to massively store all the information produced, thus inviable. The processing issue is starting to see a feasible horizon, but still has space to evolve. Therefore the important issue is not to store all the data, but to extract relevant statistics, summaries and models from the produced data.

Big Data

Big data is a recent term that has appeared to define the large amount of data that surpasses the traditional storage and processing requirements. Volume, Velocity and Variety, also called the three Vs, is commonly used to characterize big data. Looking at each of the three Vs independently brings challenges to big data analysis.

Volume

The volume of data implies in scaling the storage and being able to perform distributed querying for processing. Solutions for the volume problem are either by using datawarehousing techniques or using parallel processing architecture systems such as Apache Hadoop.

Velocity

The V for velocity deals with the rate in which data is generated and flows into a system. Everyday sensors devices and applications generate unbounded amount of information that can be used in many ways for predictive purposes and analysis. Velocity not only

deals with the rate of data generation but also with the speed in which an analysis can be returned from this generated data. Having realtime feedback is crucial when dealing with fast evolving information such as stock markets, social networks, sensor networks, mobile information and many others. Aiming to process these streams of unbounded ow of data some frameworks have emerged like the Apache! S4 and the Twitter Storm platforms.

Variety

One problem in big data is the variety of data representations. Data can have many di_erent formats depending of the source, therefore dealing with this variety of formats can be daunting. Distributed key-value stores, commonly referred as NoSQL databases, come in very handy for dealing with variety due to the unstructured way of storing data. This exibility provides and advantage when dealing with big data. Traditional relational databases would imply in restructuring the schemas and remodeling when new formats of data appear.

Algorithm

In the world of machine learning and data mining there are many algorithms in supervised, unsupervised and semi-supervised learning. They are used for different goals such as pattern recognition, prediction, classification, information retrieval and clustering. The practical applications of these algorithms are endless and can permeate any field of study.

A widely used algorithms for clustering is the k-means, which purpose is to split a dataset into k distinct groupings. The goal of k-means is to identify similarities between items in a dataset and group them together based on a similarity function. The most common used function is the euclidean distance function, but any other similarity function can be applied. K-means is an interesting option due to its simplicity and speed.

In essence it is an optimization problem where a local optimal clustering can be achieved, whereas a global optimal is not guaranteed. It is a heuristic algorithm in which the final result will be highly dependent on the initial settings, although a fairly good approximation is possible. For this reason choosing the exact solution for the k-means is an NP-hard problem. K- Means takes as input a number k of desired clusters and data set X 2 Rd.

The goal is to choose k centers to minimize the sum of squared Euclidean distance as presented in the following function.
Objective function,

$$\phi = \sum_{x \in X} (min \|x - c\|^2)$$

Historically k-means was discovered by some researchers from different disciplines. The most famous researcher to be coined the author is Lloyd(1957,1982) [6], along with Forgey(1965) [7], Friedman and Rubin(1967) [8], and McQueen(1967) [9]. Since then it has been widely studied and improved. The following Table 1 shows the pseudo-algorithm for k-means. The algorithm works in and iterative way and alternates between two major steps: reassigning the cluster ID of all points in dataset X and updating the cluster centroid based upon the data points in each cluster.

Algorithm 1 k - means algorithm

Require: Dataset X, number of clusters k

Randomly choose k data points from X

Use the k points as initial set of cluster representatives C

repeat

Reassign points in X to closest cluster mean

Update m such that mi is cluster ID of the ith point in X

Update C such that cj is mean of points in jth cluster

Check for convergence of objective function

until convergence of objective function F(x)

return Set of clusters representatives C, cluster membership vector m

Related Work

Data mining and machine learning have long since been studied and there is a vast literature on these subjects. A recent and more challenging approach for machine learning and data mining is to deal with the large amount of data produced everyday by the Internet and other systems. Finding patterns on data is very relevant for decision-making and having this information fast or on real-time is a bonus. Leaders and decision makers need to respond fast to evolving trends and behaviours.

Distributed Clustering Algorithms

In [20] some distributed clustering approaches are mentioned and they are separated into two categories; multiple communications round algorithms and centralized ensemble-based methods. The multiple communications methods require synchronization steps, which incurs in a large amount of message passing. The centralized approach use asynchronous communication and build local models that are transmitted to a centralized aggregator that build a global model.

Stream Clustering Algorithms

Stream clustering algorithms have been developed as an adaptation of traditional clustering algorithms to _t the streaming model and comply to its constraints. Different techniques were created to deal with evolving data such as one pass processing and summarization. Algorithms that can work on streams and still maintain good results as their batch processing relatives. Additionally to how data is processed, some data structures were developed to deal with the memory usage. Stream clustering can be characterized by two steps: data abstraction (also referred as the online component) and the data clustering (also referred as the offline component).

The online component deals with extracting only the relevant information in specific data structures to be used later on the offline component step. There are four commonly used data structures: feature vector, prototype array, coreset trees and grids

Following is a list of the 13 most relevant data stream clustering algorithms. Each one of them has improvements in performance, memory usage, computational complexity, clustering quality and scalability.

1. BIRCH [Zang et. al 1997]
2. Scalable k-means [Bradley et al. 1998]
3. Stream [Guha et al. 2000]
4. Single pass k-means [Farnstrom et al. 2000]
5. Stream LSearch [O'Callaghan et al. 2002]
6. CluStream [Aggarwal et a;. 2003]
7. DenStream [Cao et al. 2006]
8. D-Stream [Chen and Tu 2007]
9. ODAC [Rodrigues et al. 2006; 2008]
10. SWClustering [Zhou et al. 2008]
11. ClusTree [Kranen et al. 2011]
12. DGClust [Gama et al. 2011]
13. StreamKM++ [Ackermann et al. 2012]

All of these algorithms were designed for a single machine and did not take into account a distributed environment. This work focuses on using one of these algorithms - CluStream - in a distributed environment.

Stream Machine Learning Frameworks

Applying machine learning and data mining algorithms in streams of data has become very popular due to its appliances in various scenarios where streams of data are produced. For example, such algorithms can be used on sensor networks, monitoring of power consumption, telephone logs, internet traffic, social networks and many others. Many tools and frameworks are available as commercial and open source projects. The Massive Online Analysis (MOA) framework is a software environment that contains machine learning algorithms for learning on evolving data streams.

MOA is related to another project, the Waikato Environment for Knowledge Analysis (WEKA), which is a workbench for batch processing of ML algorithms [21]. Stream processing is different from batch in the sense that the dataset is potentially infinite. In order to process data that arrives in high speed some requirements have to be taken into account. Since data is arriving continuously the memory can easily be filled, thus a low memory approach has to be used. Each example has to be processed at a time and at most once; this is known as a one-pass approach. Another requirement is that the algorithms should be able to provide a prediction or summaries at any time, therefore the models have to be constantly updated.

CONCLUSION

Machine learning has become popular and has evolved to become an essential tool for predictive analysis and data mining. This popularity has resulted in the development of many tools for specific and generic uses.

This project presented some of the most important efforts in applying machine learning to clustering problems in different kinds of infrastructure, architecture and data models. The current state of machine learning tools taxonomy and points where SAMOA fits in this current scenario.

REFERENCES:

[1] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, “Zookeeper: wait free coordination for internet-scale systems," in Proceedings of the 2010 USENIX conference on USENIX annual technical conference, USENIXATC'10, (Berkeley, CA, USA), pp. 11{11, USENIX Association, 2010.

[2] J. Dean and S. Ghemawat, “Mapreduce: simpli_ed data processing on large clusters," Commun. ACM, vol. 51, pp. 107{113, Jan. 2008.

[3] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed _le system," in Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and T Technologies (MSST), MSST '10, (Washington, DC, USA), pp. 1{10, IEEE Computer Society, 2010.

[4] K. Tretyakov, “Machine learning techniques in spam filtering," tech. rep., Institute of Computer Science, University of Tartu, 2004.

[5] V. Estivill-Castro, “Why so many clustering algorithms: a position paper," SIGKDD Explor. Newsl., vol. 4, pp. 65{75, June 2002.

[6] S. Lloyd, “Least squares quantization in pcm," Information Theory, IEEE Transactions on, vol. 28, no. 2, pp. 129{137, 1982.

[7] E. Forgy, “Cluster analysis of multivariate data: Efficiency versus in terpretability of classification," Biometrics, vol. 21, no. 3, pp. 768{769, 1965.

[8] H. P. Friedman and J. Rubin, \On Some Invariant Criteria for Grouping Data," Journal of The American Statistical Association, vol. 62, pp. 1159{1178, 1967.

[9] J. MacQueen, “Some methods for classification and analysis of multi-variate observations," in Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., vol. 1, pp. 281{297, Univ. of Calif. Press, 1967.

[10] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “A local search approximation algorithm for k-means clustering," in Proceedings of the eighteenth annual symposium on Computational geometry, SCG '02, (New York, NY, USA), pp. 10{18, ACM, 2002.

[11] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding," in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027{1035, 2007.

[12] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, “The planar k- means problem is np-hard," in WALCOM: Algorithms and Computation (S. Das and R. Uehara, eds.), vol. 5431 of Lecture Notes in Computer Science, pp. 274{285, Springer Berlin Heidelberg, 2009.

[13] M. Inaba, N. Katoh, and H. Imai, “Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract)," in Proceedings of the tenth annual symposium on Computational geometry, SCG '94, (New York, NY, USA), pp. 332{339, ACM, 1994.

[14] D. J. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik, “Aurora: a new model and architecture for data stream management," 2003.

[15] T. S. Group, “Stream: The stanford stream data manager," 2003.

[16] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, and M. Shah, “Telegraphcq: Continuous dataow processing for an uncertan world," 2003.

[17] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. S. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik, “The Design of the Borealis Stream Processing Engine," in Second Biennial Conference on Innovative Data Systems Research (CIDR 2005), (Asilomar, CA), January 2005.

[18] G. Agha, Actors: a model of concurrent computation in distributed systems. Cambridge, MA, USA: MIT Press, 1986.

[19] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “Mining data streams: a review," SIGMOD Rec., vol. 34, pp. 18{26, June 2005.

[20] S. Bandyopadhyay, C. Giannella, U. Maulik, H. Kargupta, K. Liu, and S. Datta, “Clustering distributed data streams in peer-to-peer environments," Information Sciences, vol. 176, no. 14, pp. 1952 { 1985, 2006. }Streaming Data Mining

[21] A. Bifet, G. Holmes, B. Pfahringer, J. Read, P. Kranen, H. Kremer, T. Jansen, and T. Seidl, “Moa: A real-time analytics open source framework," in Machine Learning and Knowledge Discovery in Databases (D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgian nis, eds.), vol. 6913 of Lecture Notes in Computer Science, pp. 617-620, Springer Berlin Heidelberg, 2011.