Volume 6, Issue 3, March 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Data Stream Mining Algorithms in Big Data: A Survey

[1]V. Sasidevi, [2]R. Rajadurai

[1] M.Tech Student, [2] Assistant Professor

[1, 2] Department of CSE, Sri Manakula Vinayagar Engineering College,
Puducherry, India

Abstract: The infrastructure build in the big data platform is reliable to challenge the commercial and non-commercial IT development communities of data streams in high dimensional data cluster modeling. The APSO ie., Accelerated Particle Swarm Optimization is a technique which commonly known for data's are sourced to accumulate their continuation in the batch model induction algorithms which is not feasible for the real time applications[8]. In this project, a new technique has been introduced ie., supervised machine learning methods for developing dynamic resource allocation which targets a user defined learning method to identify the workload patterns and also feature selection is used to process the loaded data in the searched space to form the subset of the optimal solution in size to interact their demands in computation. The main theme of this project is to feed up the data in a lightweight feature selection and to designed the streaming data by using APSO, which enables the swarm search layered forms related query dependent performance in the process scheduling and data accuracy in the iterative manner. Thus the Big data in APSO are put under the test of new feature selection algorithm for performance evaluation.

Keywords: Accelerated particle swarm optimization , feature selection, classification And Regression Tree algorithm.

I. INTRODUCTION

Recently a lot of news in the media advocates the hype of Big Data that are manifested in three problematic issues. They are the 3V challenges known as: Velocity problem that gives rise to a huge amount of data to be handled at an escalating high speed; Variety problem that makes data processing and integration difficult because the data come from various sources and they are formatted differently; and Volume problem that makes storing, processing, and analysis over them both computational and archiving challenging. In views of these 3V challenges, the traditional data mining approach which are based on the full batch - mode learning may run short in meeting the demand of analytic efficiency. That is simply because the traditional data mining model construction techniques require loading in the full set of data, and then the data are partitioned according to some divide-and-conquer strategy; two classical algorithms are Classification And Regression Tree algorithm (CART) for decision tree induction and Rough-set discrimination[8]. Each time when fresh data arrive, which is typical in the data collection process that makes the big data inflate to bigger data, the traditional induction method needs to re-run and the model that was built needs to be built again with the inclusion of new data.

II. DATA STREAM MINING METHODS

In contrast, the new breed of algorithms known as data stream mining methods are able to subside these 3V problems of big data, since these 3V challenges are mainly the characteristics of data streams. Data stream algorithm is not stemmed by the huge volume or high speed data collection[2]. The algorithm is capable of inducing a classification or prediction model from bottom-up approach; each pass of data from the data streams triggers the model to incrementally update itself without the need of reloading any previously seen data. This type of algorithms can potentially handle data streams that amount to infinity, and they can run in memory analyzing and mining data streams on the fly. It is regarded as a killer method for big data hype and its related analytics problems[4]. Lately researchers concur data stream mining algorithms are meant to be solutions to tackle big data for now and for the future years to come. In both families of data mining algorithms, stream - based and batch-based, classification has been widely adopted for supporting inferring decisions from big data. In supervised learning, a classification model or classifier is trained by suggesting the relationship between the attributes of the historical records and the class labels which are usually the predictor features of all the data and their predicted classes respectively. Subsequently, the classifier is used to predict appropriate classes given unseen samples.

III. FREQUENT ITEMSET MINING ALGORITHM

A novel approach for mining the frequent itemsets from a data stream has been made. **A**n approach for mining frequent item sets using variable window size by context variation analysis (MFI-VWS-CVA) over data streams. Due to the factor of fixing window size dynamically by concept of variation analysis, the said model is identified as optimal and scalable[3]. A parallel process that determines frequent itemsets from the concept of cached bush structures, it performs frequent item sets removal over data streams. this incremental regular itemset mining algorithm by introducing windowing the streaming transaction with variable window size technique with regard to attain efficient memory usage and execution time. The experiment results confirm that the MFI-VWS-CVA is scalable under different streaming data size and reporting values. In future this model can be extended to perform convenience based frequent item set mining over data streams. Streaming the dataset is high risk and it is a time Consuming.

IV. HADOOP MAP REDUCE FRAMEWORK

In order to have a summarized data results for a particular web application, we have to do log analysis which will help to improve the business strategies as well as to create statistical reports. Hadoop – MR log file analysis tool will provide us graphical reports showing hits for web pages, users activity in which the part of website, the users are interested in traffic sources, etc. From the information business communities can appraise which parts of the website need to be improved, which are the potential customers, from which the geographical, it is getting maximum hits, etc, which will help in designing future marketing plans[7]. Log analysis can be prepared by a collection of methods, but what matters is response time. Hadoop Map Reduce construction provides parallel distributed processing and sturdy data storage for large volumes of log files. Firstly, data get stored in the hierarchy on several nodes in a group, so that the access time required can be reduced which saves much of the processing time. Hadoop's attribute of moving calculation to the data rather moving data to computation helps to improve the response time. Secondly, Map Reduce successfully works for large datasets benevolent the efficient results

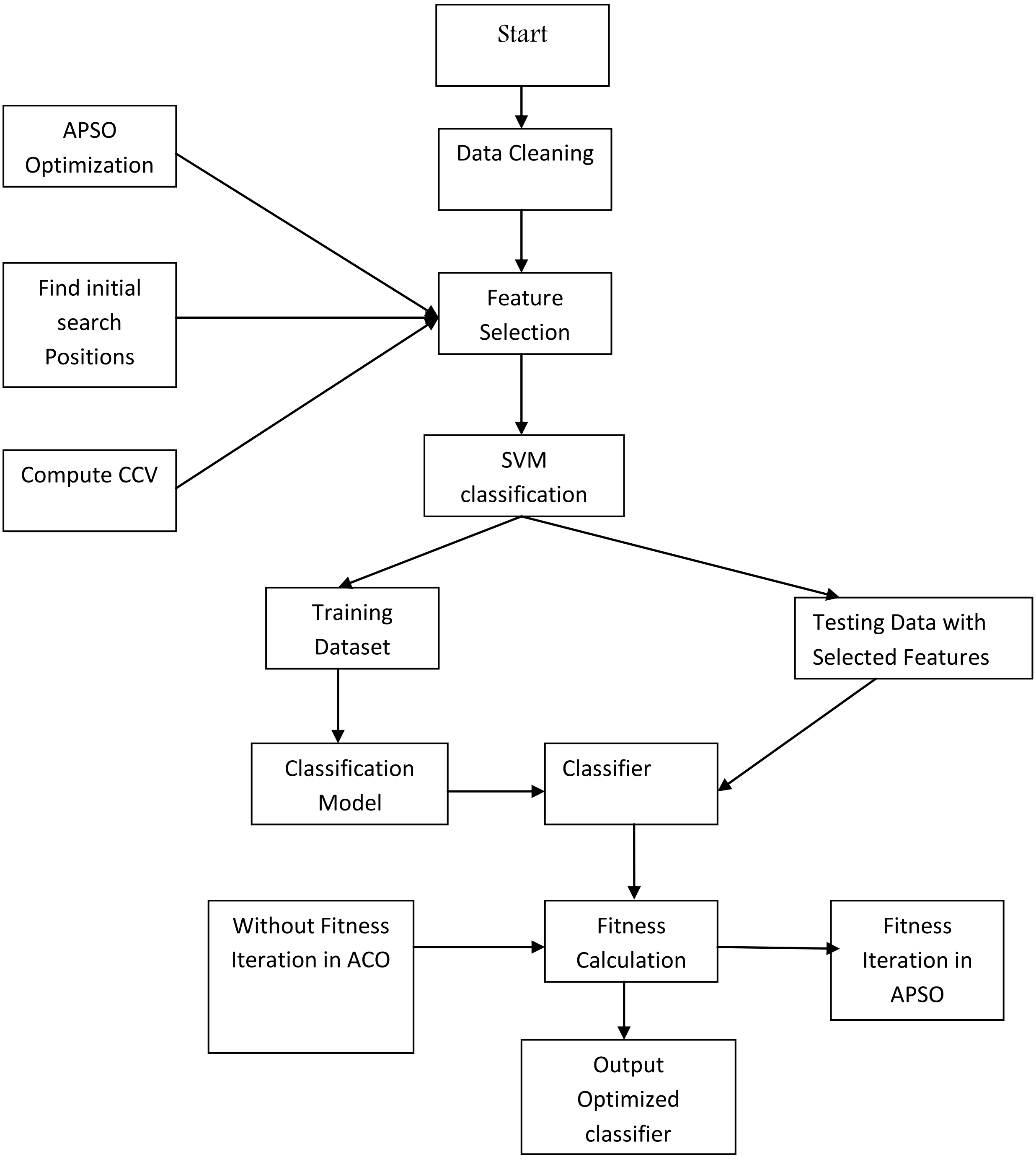
Start
APSO Optimization
Data Cleaning
Find initial search Positions
Feature Selection
Compute CCV
SVM classification
Training Dataset
Testing Data with Selected Features
Classification Model
Classifier
Without Fitness Iteration in ACO
Fitness Calculation
Fitness Iteration in APSO
Output Optimized classifier

Fig1. APSO Architecture

Hadoop framework gives plasticity to the programmer to choose the aspect that he is comfortable to code with. It was executed in business logic on the dataset and could reach a better concert when we have executed it in Hadoop and on a centralized system. In future it will perform the same implementation on different virtual machines and would analyze the behavior of the execution. The database with the tuple data does not be maintained in confidence.

Big data is not only large in volume but they can be structured in many columns giving climb to high dimensionality in feature legroom, which is a well-known problem in data mining. In constructing a categorization model, new feature space is mapped onto a new space of condensed extent[5]. Identification of relevant features is enormously important for organization tasks .Given the high dimensionality in the data, selecting a right detachment of useful features from all the original features is difficult and may be even computationally inflexible[1]. There is no blonde rule of thumbs how this should be done albeit a lot of research labors have been going on, advocating different attribute collection methods. However, to the best of the knowledge, no universal method is being claimed, although newly a high surge in hybrid modes of integrating meta-heuristics into wrapper-based feature selection method .

V. PARTICLE SWARM OPTIMIZATION

It provides the ability to access bulk data with high I/O throughput. As a result, it is suitable for applications that have large I/O data sets. However, the performance of HDFS decreases dramatically when behavior of the operations of interaction-intensive files, i.e., files that have relatively small size but are frequently accessed[8]. The modifications to the HDFS are: (1) changing the single name node structure into an extended name node structure; (2) deploy caches on each rack to recover the I/O performance of accessing interaction-intensive files and (3) using PSO-based algorithms to find a near optimal storage allocation plan for incoming files.Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying need not to advance a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle position and velocity. Each particle's movement is influenced by its local best known spot but, is also guided just before the best known positions in the search-space, which are simplified as better positions are found by other particle. This is expected to move the swarm toward the best solutions. It resolves the economic dispatch problem for considering complex problems to be tackled. It has many complex optimization problems.

Title	Author	Publication	Advantage	Disadvantage
KNOWLEDGE DISCOVERY FROM STATIC DATASETS TO EVOLVING DATA STREAMS AND CHALLENGES	V.SIDDA REDDY, M.NARENDRA AND K.HELINI	INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS	The throughput when dealing with interaction intensive tasks and only cause slight performance degradation for handling large size data accesses	1.Streaming the dataset is high risk one 2.Time Consuming while large amount data is high
SWARM-BASED METAHEURISTIC ALGORITHMS AND NO-FREE-LUNCH THEOREMS	XIN-SHE YANG	BOOK CHAPTER OF THEORY AND NEW APPLICATIONS OF SWARM INTELLIGENCE	The modifications to the HDFS are: (1) changing the single name node structure into an extended name node structure; (2) deploying caches on each rack to improve the I/O performance of accessing interaction-intensive files and (3) using PSO-based algorithms to find a near optimal storage allocation plan for incoming files.	It provides the ability to access bulk data with high I/O throughput. As a result, this system is suitable for applications that have large I/O data sets. However, the performance of HDFS decreases dramatically when handling the operations of interaction-intensive files, i.e., files that have relatively small size but are frequently accessed.
PRITER:A DISTRIBUTED FRAMEWORK FOR PRIORITIZING ITERATIVE COMPUTATIONS	YANFENG ZHANG, QIXIN GAO, LIXIN GAO, AND CUIRONG WANG,	IEEE TRANSACTIONS	In order to address the limitations of existing approaches, an integrated approach needs to be developed for addressing the two major issues related to automated service discovery: 1)semantic-based	Given the large number of web services and the distribution of similar services in multiple categories in the existing UDDI infrastructure, it is

			categorization of web services 2)selection of services based on semantic service description rather than syntactic keyword matching**.**	difficult to find services that satisfy the desired functionality
VDBMR: MAPREDUCE-BASED DISTRIBUTED DATA INTEGRATION USING VIRTUAL DATABASE	YULAI YUAN, YONGWEI WU_, XIAO FENG, JING LI, GUANGWEN YANG, WEIMIN ZHENG	FUTURE GENERATION COMPUTER SYSTEMS	This work is the first to formally define two privacy attacks, namely attribute-correlation attack and inference attack, and propose two countermeasure schemes automaton segmentation and query segment encryption to securely share the routing decision making responsibility among a selected set brokering servers	1.The database with the tuple data does not be maintained confidentially. 2.The existing systems another person to easily access database

VI. CONCLUSION

In Big Data analytics, the high dimensionality and the streaming nature of the incoming data aggravate great computational challenges in data mining. Big Data grows continually with fresh data and are being generated at all times; hence it requires an incremental computation approach which is able to monitor large scale of data dynamically. Lightweight incremental algorithms should be considered that are capable of achieving robustness, high accuracy and minimum pre-processing latency.

REFERENCES

[1] Quinlan, J. R., C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993

[2] Ping-Feng Pai, Tai-Chi Chen, "Rough set theory with discriminant analysis in analyzing electricity loads", Expert Systems with Applications 36 (2009), pp.8799–8806

[3] Mohamed Medhat Gaber, Arkady Zaslavsky, Shonali Krishnaswamy, "Mining data streams: a review", ACM SIGMOD Record, Volume 34 Issue 2, June 2005, pp.18-26

[4] Wei Fan, Albert Bifet, "Mining Big Data: Current Status, and Forecast to the Future", SIGKDD Explorations, Volume 14, Issue 2, pp.1-5

[5] Arinto Murdopo, "Distributed Decision Tree Learning for Mining Big Data Streams", Master of Science Thesis, European Master in Distributed Computing, July 2013

[6] S. Fong, X.S. Yang, S. Deb, Swarm Search for Feature Selection in Classification, The 2nd International Conference on Big Data Science and Engineering (BDSE 2013), 2013, 3-5 Dec. 2013.

[7] [Rokach, Lior, and OdedMaimon. "Top-down induction of decision trees classifiers-a survey." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 35, no. 4 (2005): 476-487.

[8] Aggarwal, Charu C., ed. Data streams: models and algorithms. Vol. 31.Springer, 2007.
Accelerated PSO Swarm Search Feature Selection for Data Stream Mining Big Data Simon Fong, Raymond Wong, and Athanasios V. Vasilakos, IEEE.

[9] D. Borthakur, The hadoop distributed file system: architecture and design,Hadoop Project Website, 2007.

[10] D. Borthakur, Hdfs architecture guide, HADOOP APACHE PROJECT, 2008,http://hadoop.apache.org/common/docs/current/hdfsdesign.pdf.

[11] S. Chandrasekar, R. Dakshinamurthy, P. Seshakumar, B. Prabavathy, C. Babu,A novel indexing scheme for efficient handling of small files in hadoop distributed file system, in: Computer Communication and Informatics (ICCCI),2013 International Conference on, IEEE, 2013.

[12] E. Deelman, G. Singh, M. Livny, B. Berriman, J. Good, The cost of doing scienceon the cloud: the montage example, in: Proceedings of the 2008 ACM/IEEEconference on Supercomputing, IEEE Press, 2008.

[13] D. Fesehaye, R. Malik, K. Nahrstedt, Edfs: a semi-centralized efficient distributed file system, in: Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware, Springer-Verlag New York, Inc,2009

[14] H. Hsiao, H. Chung, H. Shen, Y. Chao, Load rebalancing for distributed filesystems in clouds, 2013.

[15] A. Indrayanto, H.Y. Chan, Application of game theory and fictitious play in data placement, in: Distributed Framework and Applications, 2008. DFmA 2008.First International Conference on, IEEE, 2008.

[16] L. Jiang, B. Li, M. Song, The optimization of hdfs based on small files,in: Broadband Network and Multimedia Technology, IC-BNMT, 2010 3rd IEEEInternational Conference on, IEEE, 2010.

[17] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Neural Networks,1995. Proceedings., IEEE international Conference on, vol. 4, IEEE, 1995.

[18] J. Kennedy, W.M. Spears, Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodalproblem generator, in: Evolutionary Computation Proceedings, 1998. IEEEWorld Congress on Computational Intelligence., The 1998 IEEE InternationalConference on, IEEE, 1998.