BVG Trust

J. Sci. Trans. Environ. Technov. 2018, 11(4) : 183-187

Scientific Transactions in Environment and Technovation

Stream mining with big data architecture by Automated swarm search technique

https://doi.org/10.56343/STET.116.011.004.003
http://stetjournals.com

N.Subhalakshmi*, C.Jasmine and S.Ramya

S.T.E.T Women's college, Sundarakkottai,mannargudi,614016,Tamil Nadu, India.

Abstract

Big data is a leading enabled technology by recent advances in technologies and architecture. However, big data is facing the problem of hardware and processing resources costs, by adoption costs of big data technology prohibitive to small and medium sized businesses. Cloud based big data servers is a set of it services that are provided to a considering the over a network on a leased basis and with the ability to scale up or down their service requirements. because of using cloud as a service process the advantages includes scalability, resilience, flexibility, efficiency and outsourcing non-core activities. The definition, characteristics, and classification of big data along with some discussions on cloud computing are introduced. The feature selection is designed particularly for mining streaming data on the fly, by using accelerated particle swarm optimization (APSO) type of swarm search that achieves enhanced analytical accuracy within reasonable processing time. In this paper, a collection of Big Data with exceptionally large degree of dimensionality are put under test of our new feature selection algorithm for performance evaluation.

Key words: Feature Selection, Swarm Intelligence, Classification, Big Data, Particle Swarm Optimization.

Received : June 2017 Revised and Accepted : March 2018

INTRODUCTION

Big data management is an indispensable activity in the information age. Data migration is the process of transferring data between storage types, formats, or computer systems. It is a key consideration for any system implementation, upgrade, or consolidation. With involvement of information collection mechanisms and advancement of storage and database technology, data sets in today's institutions are often so large, complex and rapidly generated, that they cannot be processed by traditional information and communication technologies. Big data is collections of data sets that are large and complex. According to the IDC forecast, data is expected to grow to 40 zetta bytes (40 trillion gigabytes) in 2020 (IDC 2012) from an estimated 1.8 zetta bytes in 2011.(Quinlan, 1993) With a growth rate of a factor of 300, special attention is required by institutions in capturing, manipulation, storing, searching, retrieval, sharing, transferring, analysis and visualizing big data. In views of these 3V challenges, the traditional data mining approaches 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 (Quinlan, 1993) and Rough-set discrimination (Ping-Feng Pai and Tai-Chi Chen, 2009). 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. In contrast, the new breed of algorithms known as data stream mining methods (Mohamed Medhat Gaber, 2005) 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. 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. 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 (Wei Fan and Albert Bifet, 2014; Arinto Murdopo, 2013). In both families of data mining

*Corresponding Author :
email: *subha.stet@gmail.com*

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 inducing the relationships 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.

Background

The dataset "arcene" is a long sequence of continuous input variables from mass spectrometric data which is captured from cancer patients. There are a large number (10,000) numeric feature extracted from the mass-spectrometric images; the data are used to train a classifier for distinguishing anomalous pattern of cancer from the normal patterns. This is a two class classification problem with continuous input variables. This dataset is one of 5 datasets of the NIPS 2003 feature selection challenge. The "dexter" dataset is a large set of numbers which of each representing certain text words, commonly known as bag-of-word. It is often used for testing feature selection algorithms in text classification. The features are sparse continuous input variables which map to two class labels, there are 20,000 of them. It was used for benchmarking the famous Reuters text categorization problems. The dataset "dorothea" is for drug discovery that has an extremely large number of attributes, 100,000 in total. The numeric attributes which are the structural molecular features certain chemical compounds exist in a particular drug. Based on the molecular features, the data is to be classified as inactive or otherwise which binds to thrombin. The dataset "gisette" is a set of digitized information supposed exist in a 2D matrix that displays whether a digit of 4 or 9. It was used in training a classifier to recognize handwritten numbers. The two types of digits are very loosely structures with lots of confusable information like cap-char challenges. 5000 features represent the on-or-off information per cell in the display matrix. The dataset "madelon" is an artificially generated that consists of a set of numeric data points clustered in thirty-two groups which sit on the vertices of a 5D hypercube, and they are randomly tagged with values of positive 1 or negative 1. The five dimensions contain five informative features, which lead to fifteen linear combinations of features; they combine to form a collection of twenty meaningless informative features. Out of those twenty redundant features a classifier would have to distinguish the examples into two classes (positive or negative labels). Furthermore, redundant features called 'probes' are added for distracting the classifier where those probes carry no predictive power. Then the order of the features in each instance and the order of the instances were randomly shuffled for producing greater challenges. The attributes and characteristics of the five Big Data representative datasets are shown in Table 1. The visualizations of the two datasets, "arcene" and "dexter" are shown in Figure 1 and Figure 2 respectively. Due to very high dimensionality, only the first 20 dimensions from the head and the last 20 dimensions from the tail of the feature vector are shown. Essentially, Figure 1 shows very non-linear relations between the features and the classes, and Figure 2 is a sparse matrix that is made of coexistences of thoughts and ones from the feature values, mapping out again a non-linear features-class relation. The two figures unanimously show the underlying complexity of the feature values pertaining to the predicted classes which needed to be resolved by the classification models.

Table.1. Characteristics of the Big Data Datasets that in Experiments

Dataset	Feature type	Number of features	Number of instances	Number of classes
arcene	Real	10,000	900	Binary: cancerous or normal
dexter	Integer	20,000	2,600	Binary: 2 text categories
dorothea	Integer	100,000	1,950	Binary: active or inactive
gisette	Integer	5,000	13,500	Binary: digit 4 or 9
madelon	Real	500	4,400	Binary: +1 or -1

Traditional and Incremental Model Learning Methods

Data stream mining over Big Data is emerging and it demands for an efficient classification model that is cap able of mining data streams and making a prediction for unseen samples. Traditional classification approach is referred to a method of top-down supervised learning (Rokach *et al*., 2005), where a full set of data is used to construct a classification model, by recursively partitioning the data into forming mapping relations for modelling a concept. Since these models are built based on a stationary dataset, model update needs to repeat the whole training process whenever new samples arrive, adding them to incorporate the changing underlying patterns. The traditional models might have a good performance on a full set of historical data, and the data are relatively stationary without anticipating much new changes. In dynamic stream processing environment, however, data streams are ever evolving and the classification model would have to be frequently updated accordingly. Therefore a new generation of algorithms, generally known as incremental classification algorithms or simply, data stream mining algorithms has been proposed to solve this problem (Aggarwal and Charu, 2007). Hoeffding Tree Domingos and Hulten, 2000).

Traditional methods require the full dataset (newly arrival data and historical data) to update decision model while incremental methods implement a single-pass approach is unnecessary to re-load full dataset. Figure 3 shows the flowchart of classification model induction by using these two families of learning methods. The following figure shows the Comparison of Approaches for Traditional and Incremental Tree-building.

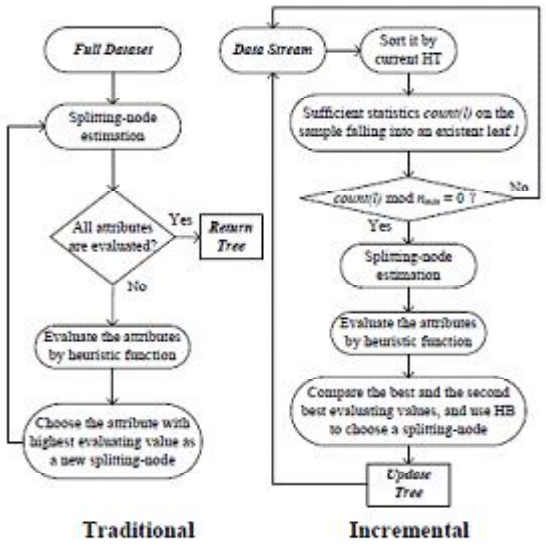

Fig.1.

As a technical drawback in the traditional methods, holding the whole execution process of model-induction in runtime memory is not favorable especially when the input training data is too large. Hence, incremental methods load only a small fragment of the input data stream at a time rather than filling all in one go, for refreshing the classification model incrementally as shown in Figure 1. In incremental learning, Hoeffding bound (HB) is used to decide whether an attribute should be split to establish new nodes provided that sufficient samples for that attribute have appeared in the data stream. The new approach is designed for incremental decision trees, the pioneer of which is Very Fast Decision Tree (VFDT) and sometimes it is more generally called Hoeffding Tree (HT) (Domingos and Hulten, 2000). HT is a classical work using H B in the no desplitting test. This is attributed to the statistical property of HB that controls the node-splitting error rate on the fly.

Incremental Learning model for Data Stream Mining

Two main schools of algorithms were designed for incremental learning: functional-based and decision tree based. The former group concerns about building a model, likely to function as a black-box with numeric weights and coefficients which models the relations between the inputs and the predicted outputs. Two of the most popular functional-based incremental learning algorithms are KStar and Updatable Naïve Bayes. The full name of KStar is "Instance-based Learner Using an Entropic Distance Measure". As the name suggests, it learns incrementally per instance by some similarity function that measures the entropic distance between the test instance and the other instances. Motivated by information theory, the underlying similarity function solves the smoothness problem by summing the probabilities over all possible decision paths for attaining good overall performance. Due to the large amount of summation over all the possible paths, KStar usually required longer processing time than its counterparts. The details of the algorithm and its entropy-based distance function are described in full (John *et al*., 1995). In the same article, KStar w as shown to outperform other rule based and instance based learning algorithms using some empirical datasets. Updatable Naïve Bayes is extended from the famous Naïve Bayes classifiers which embrace a family of simple probabilistic classifiers founded on the principle of Bayes theorem. The algorithm is designed with assumptions of possessing strong independence between the features. An advantage of this assumption is that it only requires a small amount of training data to estimate the means and variances of the features (variables) for computing the probabilities of all the possible outcomes for performing classification. Updatable Naïve Bayes is the online version of Naïve Bayes where the same algorithm continually updates its variables for tuning the hypothesis as it runs; it continually receives a new data instance, predicts its target class based on the current hypothesis; the new instance is used to further update it hypothesis accordingly too. The other major group of algorithms is decision-tree based. By the any-time tree induction principle as discussed in Section 3.2 several research papers have proposed different approaches to improve the accuracy of VFDT in the past decade. Some selected algorithms, together with KStar and Updatable Naïve Bayes will be put into experimental test in this paper. Such incremental decision tree algorithms using HB in node splitting test are so called Hoeffding Tree (HT). HOT (Pfahringer *et al.*, 2007) proposed an algorithm producing some optional tree branches at the same time, replacing those rules with lower accuracy by optional ones. The classification accuracy has been improved significantly while learning speed is slowed because of the construction of optional tree branches.

Feature selection by swarm search and SS-FS

A contemporary type of feature selection algorithm, specially designed for choosing an optimal subset from a huge hyper-space is called Swarm Search-Feature Selection (SS-FS) Model (Simon Fong *et al*., 2014). SS-FS is wrapper-based feature selection model which retains the accuracy of each trial classifier built from a candidate feature subset, picks the highest possible fitness and deems the candidate feature subset as the choice output. The workflow of the SS-FS Model is shown in Figure 4. It can be seen that the operation iterate starting from a random selection of feature subset, continues to refine the accuracy of the classification model by searching for a better feature subset, in stochastic manner. The flow enables the classification model and the chosen feature subset finally converges. The wrapped classifier is used as a fitness evaluator, advising how useful the candidate subset of features is; the optimization function searches for candidate subset of features in stochastic manner. This approach if run by brute-force testing out all the possible subsets, it will take an extremely long time. For there are 10,000 features in ?the "arcene" data, just for example, there are 210,000 103010 possible trials of repeatedly building the?1.9951 wrapped classifier. While the increase in data features goes by O 2 , the high computation costs intensify proportional to the amount of instances; in the case data stream mining, the data feed to the growth of Big Data may amount to infinity!

Evaluation Method

The experiment has two parts: first, we compare two groups of classification learning methods, traditional batch learning and incremental learning on their classification performance such as accuracy, kappa, precision and recall etc. The names of the classification learning algorithms, together with a short description are shown in Table 2. The choices of algorithms for both groups are popular methods that have been used widely in the literature. The data stream mining algorithms which are put under test here are mainly inherited from the Hoeffding principle in growing a decision tree. In addition, two non-decision-tree type of incremental learning such as updatable Naïve Bayes and KStar are tested in the comparison. Secondly the timing performance is evaluated for the two groups of classification, in relation to the cost-benefit of accuracy improvement at the price of extra running time. form of a Dell Precision T7610 PC with Intel Xeon Processor E5-2670 v2 (Ten Core HT, 2.5GH z Turbo, 25 MB) and 128GB RAM. The programming environment is Java Development Kit 1.5. For the algorithms, they are implemented on MOA platform. Default parameter values are set for all experimentation runs. For a fair evaluation over the efficacy of the algorithms, 10-fold cross-validation is used for obtaining an unbiased estimate of the accuracy performance of the classification models. The data is divided into 10 subsets of equal portions; the models by the same algorithm are built 10 rounds, each round sparing out one of the 10 subsets from training the model, as unseen data for performance validation.

RESULTS AND DISCUSSION

The accuracy measure is defined by the number of correctly classified instances over the total instances in the sensor data. In Figure 5 the overall accuracy by the traditional classification algorithms is slightly higher than those by the incremental algorithms: average accuracy 84.6426% for traditional versus 75.5658% for incremental. The top performers are Random Forest and KStar. The performance in general for the pre-processing methods of Original and Cfs is out-performed by FS-PSO and FSPSO. Generally Cfs consistently offered improvement in accuracy for traditional algorithms, though marginally. For incremental algorithms however, Cfs does not always have enhance the accuracy. This may be due to the fact that the calculation of correlation between targets and attributes in the incremental mechanism does not work well with non-stationary data, and vice-versa. The Sw arm Search type of feature selections (FS) unanimously outperformed Cfs. The improvement by FS is most obvious for NB, RHT, HOT, NBup and KStar algorithms. These algorithms have a phenomenon in common as their model structures are loosely represented by a large set of numeric variables. Like HOT and RHT for example, the decision trees are in multiple forms, gathering a pool of possible model candidate during the induction process. NB, NBup and KStar are represented by a large number of conditional probabilities and statistical variables. These models are relatively loosely defined; hence the stochastic search by PSO is appropriate and effective in finding the optimal feature subsets leading to a big leap in performance improvement. The proposed new version of APSO for Sw arm Search, namely FS-APSO nevertheless show s its superior respective to performance improvement over the standard PSO version by FS-PSO. FS-APSO is better than FS-PSO in all cases except NB. This is probably due to the fact that NB is based on applying Bayes theorem with strong hence naïve independence assumptions between the features. So PSO or APSO would have little effect over them. Moreover, for HT, PSO has very poor performance in upholding the accuracy whereas APSO solved the problem. This is because both HT and PSO are very much random-based. Putting them

together would statistically hardly get them both converged. Thereby the results seem random instead of evolving into a best solution. By far, FS-APSO has shown the maximum accuracy improvement compared to original and Cfs, indicating that FS-APSO would be a feasible feature selection scheme for the other family members of Hoeffding Tree. When it comes to performance indicators like Kappa and True Positive rate, the algorithms show similar patterns as described above in Figures 6 and 7 respectively. False positive rate which is also known as false alarm rate is an undesirable feature in machine learning. Figure 8 show s that RHT with Cfs incurred the highest false alarm rate, inferring the unsuitability of correlation - based feature selection for data stream mining especially when many random trees are being generated during runtime.

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 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. In this paper, we investigated the possibility of using a group of incremental classification algorithm for classifying the collected data streams pertaining to Big Data. As a case study empirical data streams were represented by five datasets of different domain that have very large amount of features, from UCI archive. We compared the traditional classification model induction and their counter-part in incremental inductions. In particular we proposed a novel lightweight feature selection method by using Swarm Search and Accelerated PSO, which is supposed to be useful for data stream mining. The evaluation results showed that the incremental method obtained a higher gain in accuracy per second incurred in the reprocessing. The contribution of this paper is a spectrum of experimental insights for anybody who wishes to design data stream mining applications for big data analytics using lightweight feature selection approach such as Swarm Search and APSO.

REFERENCES

Aggarwal and Charu, C. 2007. *Data streams: models and algorithms*. Vol. 31.Springer.

Arinto Murdopo, July, 2013. *Distributed Decision Tree Learning for Mining Big Data Streams*. Master of Science Thesis, European Master in Distributed Computing.

Domingos, P. and Hulten, G. 2000. Mining high-speed data streams. In : *Proc. of 6th ACM SIGKDD International conference on Knowledge discovery and data mining (KDD'00)*, ACM, New York, NY, USA, P. 71- 80. https://doi.org/10.1145/347090.347107

John, G., Cleary, Leonard, E. and Trigg, K. 1995. An Instance based Learner Using an Entropic Distance Measure. In: *12th International Conference on Machine Learning*, P.108-114. https://doi.org/10.1016/B978-1-55860-377-6.50022-0

Mohamed Medhat Gaber, Arkady Zaslavsky, S. and Honali Krishnaswamy, 2005. Mining data streams: A review, *ACM SIGMOD Record*; 34 (2):18-26 https://doi.org/10.1145/1083784.1083789

Pfahringer, B., Holmes, G. and Kirkby, R. 2007. New Options for Hoeffding Trees. In : *Proc. Australian Conference on Artificial Intelligence*, P.90-99. https://doi.org/10.1007/978-3-540-76928-6_11

Ping-Feng Pai and Tai-Chi Chen, 2009. Rough set theory with discriminant analysis in analyzing electricity loads. In : *Proc. Expert Systems with Applications*,P.8799–8806. https://doi.org/10.1016/j.eswa.2008.11.012

Quinlan, J.R. 1993. *Programs for Machine Learning*. Morgan Kaufmann Publishers.

Rokach, Lior and Oded Maimon, 2005. Top-down induction of decision trees classifiers-a survey. Systems, Man and Cybernetics, Part C: Applications and Reviews, *IEEE Transactions* 35 (4) : 476-487. https://doi.org/10.1109/TSMCC.2004.843247

Wei Fan and Albert Bifet, 2014. Mining Big Data: Current Status and Forecast to the Future, *SIGKDD Explorations*, 14 : 1-5. https://doi.org/10.1145/2481244.2481246