Vol. 1, No. 1 | Jan – June 2017

Optimizing Distributed Machine Learning for Large Scale EEG Data Set

M Bilal Shaikh, M Abdul Rehman, Attaullah Sahito

Department of Computer Science, Sukkur IBA, Pakistan

bilal.shaikh@iba-suk.edu.pk , rehman@iba-suk.edu.pk

Abstract

Distributed Machine Learning (DML) has gained its importance more than ever in this era of Big Data. There are a lot of challenges to scale machine learning techniques on distributed platforms. When it comes to scalability, improving the processor technology for high level computation of data is at its limit, however increasing machine nodes and distributing data along with computation looks as a viable solution. Different frameworks and platforms are available to solve DML problems. These platforms provide automated random data distribution of datasets which miss the power of user defined intelligent data partitioning based on domain knowledge. We have conducted an empirical study which uses an EEG Data Set collected through P300 Speller component of an ERP (Event Related Potential) which is widely used in BCI problems; it helps in translating the intention of subject while performing any cognitive task. EEG data contains noise due to waves generated by other activities in the brain which contaminates true P300Speller. Use of Machine Learning techniques could help in detecting errors made by P300 Speller. We are solving this classification problem by partitioning data into different chunks and preparing distributed models using Elastic CV Classifier. To present a case of optimizing distributed machine learning, we propose an intelligent user defined data partitioning approach that could impact on the accuracy of distributed machine learners on average. Our results show better average AUC as compared to average AUC obtained after applying random data partitioning which gives no control to user over data partitioning. It improves the average accuracy of distributed learner due to the domain specific intelligent partitioning by the user. Our customized approach achieves 0.66 AUC on individual sessions and 0.75 AUC on mixed sessions, whereas random / uncontrolled data distribution records 0.63 AUC.

Keywords: Data Set, Optimizing, Machine Learning

1. Introduction

Machine Learning is a type of artificial intelligence (AI) that provides computers with the ability to learn real time scenarios from observation data. Based on those observations, models are prepared which can predict unknown outcomes. The consistency of that model helps human in making decisions. Models could be categorized as supervised, unsupervised and also semi-supervised. As the data size, speed and its variety have massively increased we have entered in to an era of Big Data. The scalability of tools and techniques used for processing large scale data sets have become an active research direction for researchers in Big Data community. To scale machine learning techniques with large datasets it is a common practice to distribute data on several systems, called data nodes. These distributed nodes contribute their computational power and storage to the overall data intensive task.

During a machine learning task, accumulative measure from different working nodes is calculated. This measure dictates the scale of quality of a machine learning prediction.

A. Research Statement: During any cognitive process in human, brain produces some brain activity. Such an activity could be logged through waves generated in the brain.

These brain waves could be translated to human intentions of what they want to do. EEG [1] is a device to record such wave data. Data collected through EEG is very noisy with low SNR (Signal to Noise Ratio). Due to this noisy nature of collected data, it is challenging to extract event related potential and interpretation of human intention correctly.

During spelling task, the problem is in detecting errors, which is done through analyzing the brain waves of subject. Differentiating between P300’s true and noisy signal is a difficult job. Due to complete paralysis, the patient cannot communicate but it is awake and fully aware. In such a situation, using BCI (Brain Computer Interaction) a patient can establish contact a channel directly from the brain (signals) to the computer. As EEG signals are very noisy so noise could remove after some important feature extraction from the dataset. Other irrelevant information from the dataset could also be identified to remove noise which will help in analyzing the important part of dataset collected. A learner could be prepared to predict the accuracy of spelling error. As the data set is large and could be distributed among various nodes, we have taken this problem in hand to conduct an empirical study. This study will help us in presenting a proof of concept to below research question.

1) Distributed Machine Learning Platforms provide automated random data distribution/partitioning of data set which neglects the advantage of user defined controlled partitioning of datasets. So if we inculcate domain specific intelligence while partitioning the data for different nodes, will this impact on learner’s accuracy?

2. Literature Review

Apache Hadoop is an open source framework for distributed processing and storage of large datasets on commodity hardware. HDFS (Hadoop Distributed File System) is the central technology is designed across low-cost commodity hardware and for the efficient scale out storage. HDFS is responsible for providing reliable and scalable data storage that deals with span of large clusters of commodity servers [3]. Hadoop implements the Map Reduce [4] computational paradigm and using HDFS as its compute node.

HDFS is a distributed file system designed to run on commodity hardware [5]. HDFS key feature is it’s highly fault tolerant behavior. HDFS is designed for deployment on low-cost commodity hardware. HDFS is also good for providing high throughput access to application data and is quite suitable for applications that have large volume of data.

GFS was designed by Google [6] to support the similar goals as previous distributed file system have like HDFS [5] performance, scalability, reliability and availability. Google File System has been driven by observations Google made to meet their storage needs. Google File System presented new extensions to existing distributed file systems keeping various aspects for both micro-benchmarks and real world use.

MapReduce [4] is a paradigm shifting programming model for processing large datasets dealing with parallel distributed algorithm. In map reduce user defines the computations as map and reduce functions and the underlying run time system automatically parallelizes the computation across the large-scale clusters of machines, possible machine failures and schedules other inter-machine communications to make the possible efficient use of the network and disks.

Graph Lab [7] was developed by identifying common pat- terns in ML, it is a parallel abstraction that achieves higher usability, expressiveness and performance. Unlike previous parallel abstractions, Graph Lab offers representation of structured data dependencies, iterative computation, and flexible scheduling. It uses data graph to encrypt the computational structure and data dependencies of problem. It represents local computation as update functions which transform the data on the data graph. Because these update functions can modify overlapping state, the Graph Lab framework provides a set of data consistency models which allow the user to specify the minimum consistency requirements of their application. Spark [8] is a new framework that supports applications that are not focused around acyclic dataflow model retaining the scalability and fault tolerance of MapReduce.

Spark introduces a new layer of abstraction called Resilient Distributed Data Sets (RDDS). An RDD Set is a collection of objects partitioned and read only across a group of machines which reestablishes itself when a partition gets lost. Spark author claims to outperform popular Hadoop by 10 times in iterative machine learning jobs, and highly efficient for interactive query processing to large scale datasets than existing frameworks.

Petuum is a recent framework for distributed Machine Learning [9], the development of Petuum is based on a theoretic ML-centric optimization principle. Petuum formalizes ML algorithms as iterative convergent programs which encompass a larger scope of modern machine learning like MCMC, stochastic gradient to estimate points in latent variable models, coordinate descent, proximal optimization for structured sparsity problems, variation methods for graphical models, among others. Petuum authors claim it to be better than existing ML platform. Petuum displays better performance for being an alternative to single machine algorithms CNN, Caffe and DML [9].

3. Research Methodology

For setting up development environment for the sake of proof of concept, we have used Spyder IDE (Integrated Development Environment) for Python-based development. For using Machine Learning techniques SciKit Learn library [10] was used. ElasticNet API (Application Programmable Interface) provided the implementation of ElasticCV classifier. Numpy [11] was used to partition data into the two-dimension dataset into multiplied is joint 2D datasets. For plotting the ROC graphs of classifiers ggplot [12] is used.

We have a 9.5 GB EEG raw dataset which was selected to conduct empirical experiments. The purpose of collection of data was to predict the error in spelling correction from p300 speller which was used by Perrin et.al [13]. An experiment was carried out over nine different subjects with five sessions each. These five sessions are assumed to be an Epoch window, i.e. a dataset within a time frame which is collected after each stimulus. These Epochs will then processed as training dataset to the classifiers. Perrin [13] has presented an explanation about the configuration of EEG device was used with the subjects.

Dataset contains both training data as well as data for testing of learners. Training dataset consists of 16 subjects while testing dataset comprises 10 subjects; each had attended 5 disjoint sessions on spelling. In master dataset, total trials for training were 5440 and 3400 were test trials. There are two labels of data, (i) Target and (ii) Non-Target.

In preprocessing phase shown in Figure 1.0is regarding EEG signal data, EOG channel was removed implemented in python. EOG (Electro Oculo Gram) channel produces information introduced by the blinking of eye which is a noise in our case. Then, butter-worth filter between 1-40Hz band pass filtered the EEG signals is applied. Butter-worth is also known as maximally flat magnitude filter. Only 1.3 seconds is set for Epochs which is after the occurrence of any possible stimuli or feedback event by the subject. Then feature extraction is applied before the classification. Only preferred electrodes are selected within a recommended time of 1.3 seconds

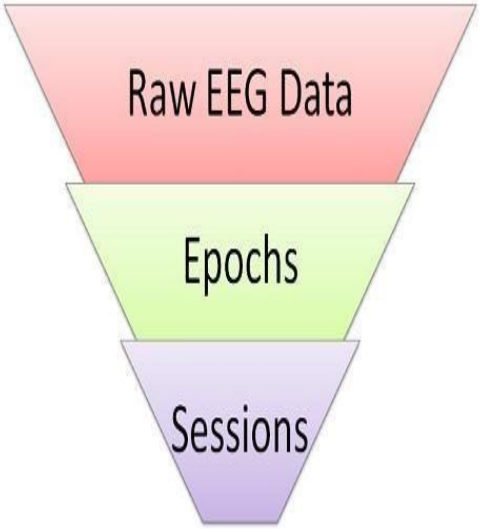

Figure. 1: Pre-processing Workflow from Raw Data Set to Sessions Based Data Partitions

Window later concatenated with Meta data. Data size got reduced. A total master dataset with 5440 instances having 2211 dimensions became available for further processing. The EEG based Feature extraction is done as per the following methods:

1) Dawn Covariance: Two sets of 5 XDAWN spatial filters are estimated, one for each class (Error and Correct). The grand average evoked potential of each class is then filtered by the corresponding set of spatial filters, and concatenated to each epoch. The covariance matrix of each resulting epoch is then used as feature for the next steps [14].

2) Electrode Selection: A channel selection is applied to keep only relevant channels. The procedure consists in a backward elimination with the Riemannian distance between the Riemannian Geometric mean of the covariance of each class as the criterion.

3) Tangent Space: Reduced Covariance matrices are then projected in the tangent space [15]

4) Normalization: Feature Normalization using al1 norm. Epoch windows which was partitioned on the basis of different sessions attended by subjects into five disjoint datasets. This partitioning is totally data dependent, unknown to machine learning learners and underlined infrastructure. This could be called as user defined data partitioning.

A new dimension was added to the dataset to categorically divide it. This dimension labeled each instance with the respective session ID of that instance. All the labeled data was extracted later in order to achieve different sub datasets. These sub datasets could be distributed to different nodes and processed in parallel in case of speeding up the process. Our goal is to focus on optimization in accuracy of learners. Therefore, speeding up the performance is not the important concern here.

4. Classifiers Based on User Defined Intelligent Data Partitioning

Now as the trained dataset is ready after partitioning to train multiple distributed classifiers. So, each learner has its own dataset which has been partitioned as per respective session of the subject. These disjoint partitioned datasets are used for acquiring knowledge about the parameters using

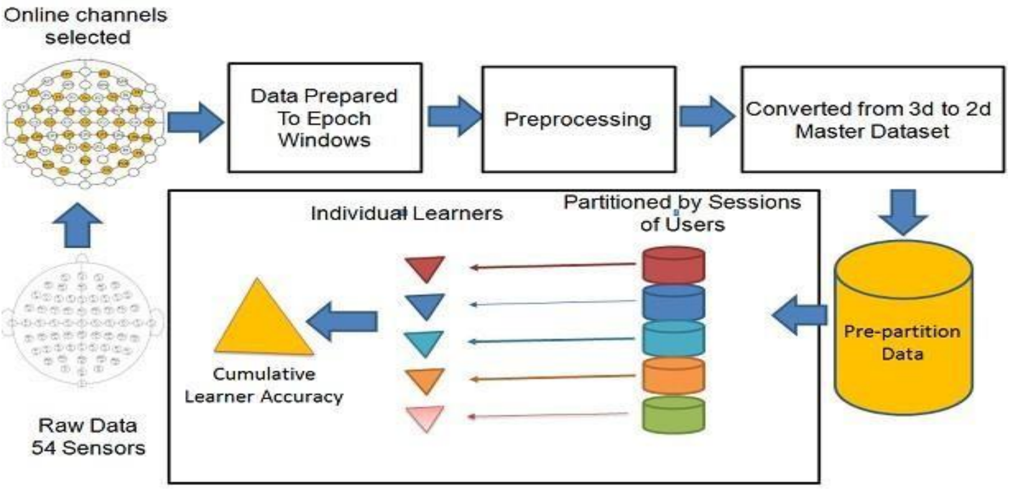
Online channels selected
Data Prepared To Epoch Windows
Preprocessing
Converted from 3d to 2d Master Dataset
Individual Learners
Partitioned by Sessions of Users
Pre-partition Data
Cumulative Learner Accuracy
Raw Data 54 Sensors

Figure. 2: Overall Work Flow: From Pre-processing to Classifier Preparation of the subject.

ElasticNet. ElasticNet overcomes limitations of lasso and ridge regression, it is linear regularized regression algorithm and works well with numerical attributes. Elastic can formulate our problem. Five different classifiers are built individually for each disjoint dataset. As all partitions are based on sessions taken by each subject. This complete workflow is shown in Figure 2.0. Therefore, this controls the partition data as per prior knowledge enabled by the user to define its own data partition for each classifiers training. Each case classifier is not only trained to predict the expected error in user session but it will also help the observer to notice the behavior of user customization which is based on systematic knowledge of the domain. Not incorporated in map reduce paradigm [4].

5. Classifiers Based on Traditional Hdfs like Data Partitioning

On the other side, another five learners are built which are trained on randomly partitioned dataset which is a behavior of HDFS [3] where control over data partitions is not provided. These learners are developed to cross validate against our customized learners which have been injected the user defined domain specific intelligence. Although their data sizes are similar and the instances contained within these disjoint randomly partitioned datasets are different to random distribution.

6. RESULTS

The observed results from the conducted experiments that were described in previous sections are presented here. The observations about the accuracy trends are noted along with the results. First results are shown as per our proposed customized user defined intelligent partitioning and then the results are compared with the platform controlled random like data partitioning used with most distributed machine learning solutions like [17] by NDjuric.

7. Analysis of Results

Experiments were run on all data that after mapping the space from 3-D to 2-D space to make it compatible with ElasticNet classifier. First of all, we are going to analyze how well our approach performs, for this we have used average of areas under ROC curve as measure for accuracy, as it is commonly used in the field of BCI. In ROC [18], [19] curve AUC (Area under curve) determines the credibility of classifier clearer than [4] just scalar metrics. ROC curves along with their AUC for individual sessions based of intelligent data partitioning is shown in Figure 3.

In ROC curve, which is TPR versus FPR, different graph representing different models are showing the impact of data. A variation in ROC Curve can be observed easily. However, every session's data has its own effect towards the creation of respective learner. To get the overall effect of sessions feature towards the accuracies we have average all five accuracies. Our measure of average area under curve for our intelligently partitioned dataset based models is 0.66 for the whole test data set, which shows better performance of models. ROCs could also be combined by aggregating [17] or collecting global sum of accuracies [20].

8. Comparison with Traditional Approach

To explore how well our customized partitioning approach performs, we have compared our approach average model accuracy obtained after random partitioning of data as done HDFS which takes back the control of underlying data distribution on different nodes from the user [5]. Testing data is randomly sampled into 5 sets, each of similar size of trials as it was for our custom partitioned data sets.

A. AUC Comparison: We compared the ROCs obtained from partitioning datasets as per user defined session based intelligence against the ROCs obtained after partitioning the data set in traditional way of HDFS. In both cases, we obtained five learner accuracies which we combined by taking average of each set. After aggregation of accuracy the Average Accuracy in User defined with intelligent partitioning resulted approximately 66 percent obtained from iindividual session based accuracies shown in Figure 3.0. While the average of accuracies obtained from traditional HDFS [5] way of partitioning obtained was around 63 percent. This shows an overall improvement of 3 percent in combined learner's accuracy. If user observes other important features in the data set or empirically test the variation in learner's performance the same data set, Accuracies [21] of Machine Learners could be tweaked at a large span.

9. Conclusion

This research work proposes Intelligent Data Partitioning with test case taken from a BCI P300 speller error detection problem. This approach has shown results that are improving learner's accuracy on even

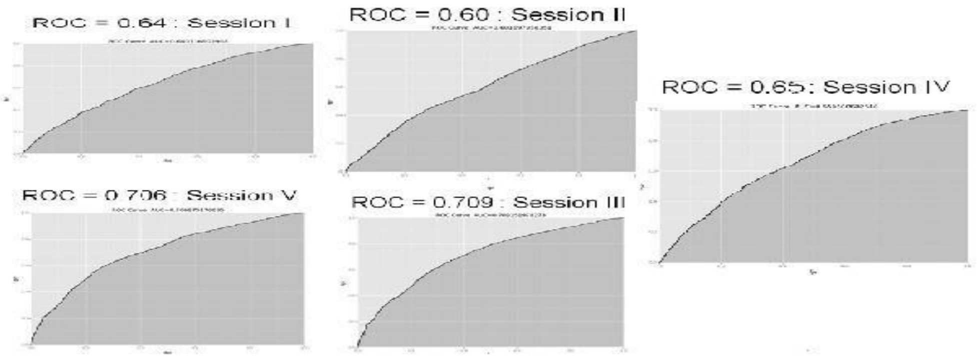

Figure. 3: Area Under ROC Curve of the classifiers from Individual Session

average aggregation. The impact of observers' intelligent data partitioning would increase with higher relevance of partitioning feature. More efficient feature engineering and nature of dataset could also improve the results. Such a type of optimization in distribution machine learning results could also expose other key insights about features of data that are only specific to a domain. This entails that allowing user controlled data partition- in will enable the analyst to dig deeper into the process of efficient machine learning. As per observed results of our proposed approach; the system performs relatively efficient for classification of the selected EEG signals in terms of average AUC in intelligent data partitioning scenario having better results. There is visible evidence for comparison based on average ROC to build a combine decision model while keeping a data attribute under control for AUC. Our proposed approach demonstrates a relatively better AUC in phase of testing supplied with low amount of data for training.

We conclude that our proposed approach will be effective if applied in other machine learning scenarios we could gain even better Average AUC and it could perform better during the other inter-features variability.

Acknowledgment

The authors would like to thank Sukkur IBA ICT Department for their laboratory services which helped us in carrying out empirical experiments.

References

[1] L. A. Farwell and E. Donchin, ―Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials,‖ Electroencephalography and clinical Neurophysiology, vol. 70, no. 6, pp. 510–523, 1988.

[2] T. O. Zander, C. Katha, S. Welke, and M. Ro¨ tting, ―Utilizing secondary input from passive brain-computer interfaces for enhancing human-machine interaction,‖ in Foundations of Augmented Cognition.

Neuroergonomics and Operational Neuroscience. Springer, 2009, pp. 759–771.

[3] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, ―The hadoop distributed file system,‖ in Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010, pp. 1–10. [4] J. Dean and S. Ghemawat, ―Mapreduce: a flexible data processing tool,‖Communications of the ACM, vol. 53, no. 1, pp. 72–77, 2010. [5] D. Borthakur, ―The hadoop distributed file system: Architecture and design,‖ Hadoop Project Website, vol. 11, no. 2007, p. 21, 2007.

[6] S. Ghemawat, H. Gobioff, and S.-T. Leung, ―The google file system,‖ in

ACM SIGOPS operating systems review, vol. 37, no. 5. ACM, 2003, pp. 29–43.

[7] Y. Low, J. E. Gonzalez, A. Kyrola, D. Bickson, C. E. Guestrin, and J. Hellerstein, ―Graphlab: A new framework for parallel machine learning,‖ arXiv preprint arXiv:1408.2041, 2014.

[8] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, ―Spark: Cluster computing with working sets.‖ HotCloud, vol. 10, pp.10– 10, 2010.

[9] E. P. Xing, Q. Ho, W. Dai, J. K. Kim, J. Wei, S. Lee, X. Zheng, P. Xie, A.

Kumar, and Y. Yu, ―Petuum: a new platform for distributed machine learning on big data,‖ Big Data, IEEE Transactions on, vol. 1, no. 2, pp. 49–67, 2015.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., ―Scikitlearn: Machine learning in python,‖ The Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[11] P. P. e. a. Eric Jones, Travis Oliphant. (2016, may) Scipy: Open source scientific tools for python. 2001. [Online]. Available: http://www. scipy.org/ [12] M. C. Sachs and M. M. C. Sachs, ―Package plotroc,‖ 2015.

[13] P. Margaux, M. Emmanuel, D. Se´bastien, B. Olivier, and M. Je´re´mie, ―Objective and subjective evaluation of online error correction during p300-based spelling,‖ Advances in Human-Computer Interaction, vol.2012, p. 4, 2012.

[14] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert, ―xdawn algorithm to enhance evoked potentials: application to brain–computer interface,‖ Biomedical Engineering, IEEE Transactions on, vol. 56, no. 8, pp. 2035– 2043, 2009.

[15] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, ―Classification of covariance matrices using a riemannian-based kernel for bci applications,‖ Neuro computing, vol. 112, pp. 172–178, 2013. [16] H. Zou and T. Hastie, ―Regularization and variable selection via the elastic net,‖ Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 67, no. 2, pp. 301–320, 2005.

[17] N. Djuric, M. Grbovic, and S. Vucetic, ―Distributed confidence-weighted classification on mapreduce,‖ in Big Data, 2013 IEEE International Conference on. IEEE, 2013, pp. 458–466.

[18] S. Wu and P. Flach, ―A scored auc metric for classifier evaluation and selection,‖ in Second Workshop on ROC Analysis in ML, Bonn, Germany,2005.

[19] T. Fawcett, ―An introduction to roc analysis,‖ Pattern recognition letters, vol. 27, no. 8, pp. 861–874, 2006.

[20] A. Priyadarshini et al., ―A map reduce based support vector machine for big data classification,‖ International Journal of Database Theory and Application, vol. 8, no. 5, pp. 77–98, 2015.

[21] P. Simon, Too Big to Ignore: The Business Case for Big Data, ser.Wiley and SAS Business Series. Wiley, 2013. [Online]. Available:
https://books.google.com.pk/books?id=Dn-Gdoh66sgC