DOI: 10.22266/ijies2016.0630.03

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Mining Big Data Using Modified Induction Tree Approach

Chintan Bhatt[1]*, C. K. Bhensdadia[2]

[1]*Charotar University of Science And Technology, India*
[2]*Dharmsinh Desai University, India*
*Corresponding author'sEmail: chintanbhatt.ce@charusat.ac.in

Abstract: Data Mining techniques are broadly utilized crosswise over numerous orders to recognize hidden patents, rules or relationships among gigantic volumes of information. Induction Tree, for example, C4.5 is the most favored technique since it functions well under any dataset set being utilized. Exponential ascent in the utilization of the internet because of informal organizations began to get enormous volume of information crosswise over various areas in brief timeframe. These attributes by which the colossal measures of informal organization information are produced make them to order as Big Data. When adapting to huge information (Big Data), the greater part of the current discretization methodologies won't be very productive with respect to implementation. The most effective method to separate significant data from huge information has been a famous open issue. In this paper, we are proposing new algorithm of decision tree in big data. At last, we have shown some result using weka.

Keywords: Big Data Analytics; Classification; Data Mining; Distributed Computing; Induction Tree; Scalability.

1. Introduction

Information is all over the place! The account of how information turned out to be huge begins numerous prior years. Seventy years prior, we run over the main endeavor to measure the increase rate in the volume of data or the term which was known as the "information explosion" (a term initially utilized as a part of 1941). The resulting are the real developments in the historical backdrop of estimating information volumes in the advancement of the possibility of "enormous information" and understanding relating to information or data blast. Different sources [12] of big data are appeared in figure 1.

21st century is the period driven DATA, led by DATA. Present economy is known as DATA economy because of exponential development and digitization of information. According to our thought, Big Data is a gigantic measure of organized/unstructured information gathered from heterogeneous/complex systems. Huge Data is really blend of 7 V's i.e. VOLUME (measure of information), VELOCITY (pace of information era and development), VARIETY (sorts of information), VERACITY (dependability of information), VALUE (estimation of information), Variability (constant changing data) and VISUALIZATION (presentation of data). The term machine learning signifies that, the framework is made to learn by giving important inputs and deliberately analyzing the achieved outputs. Machines can learn under diverse circumstances to be specific, Supervised, Unsupervised furthermore, Reinforcement [1].

Machine Learning algorithms help us to make powerful predictions taking into account enormous information. Different machine learning tasks are Classification, Clustering, Association Rule Mining, Regression, Multivariate querying. Density estimation. Dimension reduction etc. The utilizations of machine learning are as assorted as the utilizations of big data. Bio Informatics, Information Retrieval, fraud detection, Telemedicine [2], Natural Language Processing, Internet of Things are the current applications of machine learning.

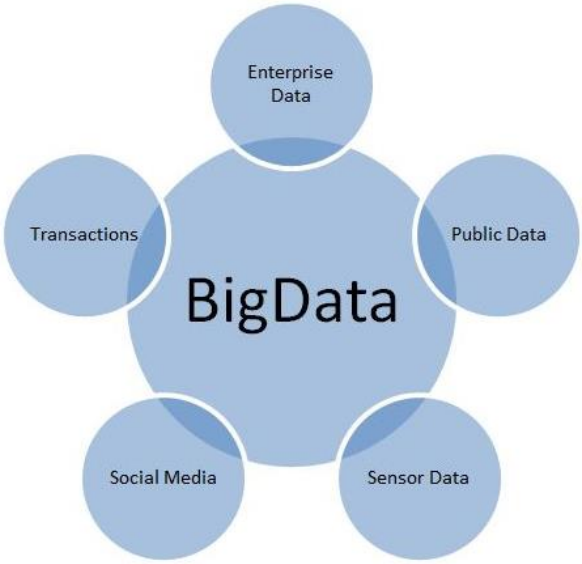

Figure 1 Sources of Big Data

2. Big Data Analytics projects and related softwares and services

Apache Storm: It is a free and open source computation framework. Storm makes it simple to dependably prepare unbounded stream of information, accomplishing for real time processing same as Hadoop which is accomplishing for batch processing. Storm is straightforward, can be utilized with any programming dialect, and is a considerable measure of enjoyable to utilize!

R: It provides open source programming to perform statistical computing on information. It provides package named as "Programming with Big Data in R" i.e. pbdR.

SAS: It provides Analytics by supporting Hadoop and by giving Grid Computing capacities, In-database processing.

Pentaho Business Analytics: It is expanding into huge information simply by ingesting data from the novel sources.

Hortonworks: It is an Open source Hadoop Distribution framework. It is based on top of Hadoop and it permits clients to catch, process and share information with any scale and any configuration in a straightforward and financially savvy way.

Apache Spark: It is a dominant open source worked around data rate, usability and analytics. Yahoo, Baidu, and Tencent, have willingly deployed Spark at huge scale, altogether preparing various petabytes of information on bunches of more than 8,000 nodes. It has rapidly turned into the biggest open source group in enormous information, with more than 750 supporters from 200+ associations.

Caffe: Deep Learning is all about multiple level of representations and abstractions to dig out some meaningful sense/information from raw data. It provides deep learning functionalities with speed, expression, modularity etc. Caffe is known for its architecture, code, speed, community etc.

TensorFlow: It is an open source library for computation using flow graphs. Nodes in the diagram state to scientific operations, while the graph edges state to the multidimensional information arrays (tensors) conveyed between them.

Weka: Mining Big Data using can be done using Weka 3 (Groovy/Jython- java-based scripting language may be used). One of the library In Weka 3.7 also provides access to MOA (Massive Online Analysis).

VowpalWabbit: It covers the rudiments and most regular alternatives, how to utilize the data format for various sorts of issues, for example, Binary Classification, Regression, Multiclass Classification, Cost-Sensitive Multiclass Classification.

Microsoft Azure Machine Learning: It provides facility to build, deploy and share predictive analytics. Azure Machine Learning Studio incorporates several inherent packages and help for custom code.

IBM Watson Analytics: It offers you the advantages of highly advanced analytics without the complexity. It guides data exploration, predictive analytics, and empowers easy dashboard and info graphic creation.

CNTK: Computational Network Toolkit by Microsoft is a deep learning toolkit which is used to explain neural network as a sequence of computational steps using directed graphs. In these graphs, leaf nodes state to system/network parameters, while other nodes state to matrix operations upon their inputs.

DMTK: It handles the problems of various distributed machine learning tasks across many clusters. It is highly scalable, efficient and flexible too.

Veles: It is a distributed platform for development of deep learning applications. It (like TensorFlow) is written in C++ and used Python for automation and coordination among nodes.

Theano [10]: It is a Pyhton library that helps us to define, optimize and evaluate mathematical expressions with multi-dimensional arrays with greater efficiency. So it is also knows ad large-scale computational intensive scientific exploration.

S4: It is an Apache Incubator project. S4 is a broadly useful, conveyed, versatile, fault tolerant

platform that permits software engineers to effortlessly create applications for management of ceaseless unbounded stream of information.

deeplearning4j: It is open-source, distributed learning library for deep-learning, written in Java and Scala. Machine Learning is related to very basic computational task while deep leaning provides computational framework to make sense from complex/huge data. It is an interactive algorithm which is learning at different level of abstractions.

Pylearn2: It is a user friendly machine learning library [11], built on top of Theano. It is a research library (who provides flexibility and extensibility) developed by LISA lab.

3. Machine Learning Frameworks in Big Data

Mahout

Mahout is one of the common framework for Machine Learning. It is known for wide-ranging choice of powerful computation, due to MapReduce engine. Mahout 0.9 was overhauled to 0.10.0 in April 2015. Through this release, the attention is presently on a mathematics domain called Samsara, which incorporates straight variable based mathematics, measurable operations, and information structures. The objective of the Mahout-Samsara is to help clients construct their own conveyed calculations, instead of basically a library of effectively composed usage. Despite everything they offer a suite of calculations for MapReduce and numerous have been advanced for Spark also. Support with H2O and Flink is right now being developed.

Spark MLlib

MLlib covers the same classes as Mahout, furthermore includes relapse models, which Mahout needs. They additionally have calculations for subject demonstrating and regular example mining. Extra devices incorporate dimensionality decrease, highlight extraction and change, advancement, and fundamental insights. By and large, MLlib's dependence on Spark's iterative cluster and gushing methodologies, and additionally its utilization of in-memory calculation, empower employments to run essentially speedier than those utilizing Mahout [3]. In any case, the way that it is attached to Spark may show an issue for the individuals to execute machine learning on different stages.

ML pipelines

As we are talking about all through this paper, building of machine learning pipelines [9] is difficult task, especially working with a blend of different apparatuses. Sparkle ML, an arrangement of identical APIs (for creation and tuning of pipelines)was acquainted in variant 1.2 with such types of issues, making it less demanding to join numerous calculations into one work process. This bundle incorporates devices for dataset changes and consolidating calculations. A simple case of this is to consider about a learner who changes a DataFrame with components into one with forecasts. This is intended to handle all progressions of the learning procedure, beginning with importing information from a source, to removing elements, and preparing and assessing models.

MLbase

In spite of the fact that it is not right now accessible, there has been progressing innovative work at AMP lab called MLbase [6], which cloaks MLlib, Spark, and different tasks to create machine learning on data sets of different sizes. MLlib and Spark (the other segments are MLI) are an API for computation advancement, and ML Optimizer, for robotization of the alteration of hyperparameters. Another part called TuPAQ (Training-upheld Predictive Analytic Query Planner) - one of late presented, which expands on the underlying thought of ML Optimizer. TuPAQ helps as a query interface that permits a client to information abnormal state inquiries in a revelatory dialect and afterward chooses the best model and parameters. Objectives in the advancement of MLbase were to allow machines to learne which are open for the non-master. TuPAQ is an essential stride toward this objective because it pushes hyperparameter tuning; highlight determination and calculation choice down into the framework.

H2O

Out of the majority of the apparatuses examined in this paper, H2O [7] is the one and only that can be viewed as an item, as opposed to an undertaking. While they offer an undertaking release with two levels of bolster, about the greater part of offerings are accessible open source too and can be utilized without the buy of a permit. The most remarkable elements of this item are it gives a graphical client interface (GUI), and various devices for profound neural systems. Profound learning has demonstrated huge guarantee for some zones of machine learning, constructing it a critical component of H2O.

SAMOA

SAMOA [8], a stage for machine gaining from spilling information, was initially created at Yahoo! Labs (Barcelona – 2013) & has been a piece of the Apache since late 2014. It is an adaptable structure that can run locally or on stream processing engines like Storm, S4, and Samza. This is done through a

negligible API intended for a general appropriated stream preparing motor which permits clients to effectively compose ties for SAMOA to new stream processors. In spite of the fact that they as of now offer far less calculations, they jump at the chance to call "Mahout for gushing." SAMOA's calculations are spoken to as coordinated diagrams, alluded to as topologies (word acquired from Storm).

A correlation of the frameworks is appeared in Fig. 2. These are relative evaluations in light of far reaching writing and online documentation audit, not our own test results.

Petuum

Petuum (distributed machine learning framework) provides algorithmic and system interface to large scale by simplifying machine learning algorithms. It can run on clusters like Amazon EC2. Main features of it are Bosen (distributed key-value), Strads (dynamic scheduler), Poseidon (Deep Learning framework), interface for C++ and Java, and YARN & HDFS Support.

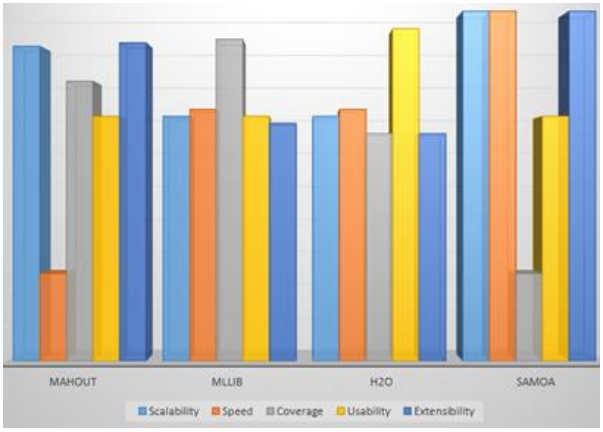

Figure 2 Correlation of machine learning frameworks

4. Decision Tree Induction

The process of knowledge discovery includes three steps: INPUT (Gathering and Preprocessing), DATA ANALYSIS (Data Mining) and OUTPUT (Result and Interpretation of it). In Data Mining, many algorithms are there like supervised algorithms and unsupervised ones. Decision Tree Induction [4] is the successful supervised algorithm broadly utilized for classification and regression. It breaches a dataset into smaller ones while a related decision tree is incrementally created. Decision trees can work with both categorical and numerical information. Traditional algorithms for building a decision tree sort every single persistent variable with a specific end goal to choose where to part the information. This sorting step gets to be time and memory restrictive when managing extensive information.

A decision tree comprises of nodes, branches and leaves. A node comprises of an inquiry in regards to an estimation of a quality, for instance node n in figure 3 [16]. We allude to this sort of node as split-node. Branch is an association between nodes that is built up taking into account the answer of its comparing question. The previously stated node n has two branches: "True (Decision 1)" and "False (Decision 2)". Leaf is an end-point in the tree.

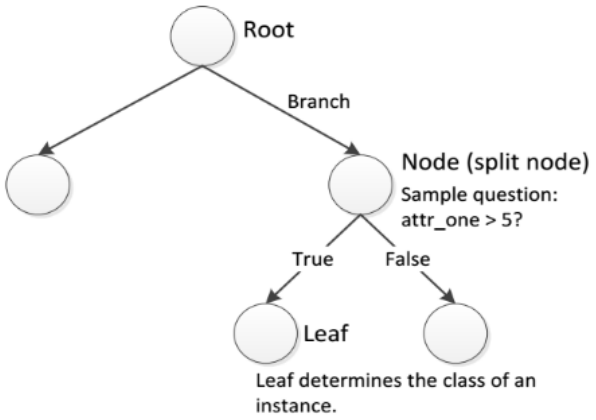

Figure 3 Decision Tree Induction

Algorithm Steps

Give the classes a chance to be signified {C1, C2,…..,Ck}. There are three conceivable outcomes for the substance of the arrangement of preparing tests T:

1. T containing one or more than one samples, all having a place with a solitary class Cj. Decision tree for T is a leaf distinguishing class Cj.

2. T containing no samples.

The decision tree is again a leaf, however the class to be connected with the leaf must be resolved from data other than T, for example, the general greater part class in T. C4.5 calculation utilizes as a measure the most successive class at the guardian of the given node.

3. T containing tests that have a place with a mixture of classes. In this circumstance, the thought is to refine T into subsets of tests that are heading towards single-class accumulations of tests.

A fitting test is picked, taking into account single quality, cap has one or all the more totally unrelated results {O1,O2, … ,On}:

T - Divided into subsets T1, T2, … ,Tn where Ti contains all the samples in T that have result Oi of the picked test. The Decision tree for T comprises of a decision node recognizing the test and one branch

for every conceivable result.

Entropy

In the event that S is set of samples. Here freq (Ci, S) stand for the quantity of tests in S that have a place with class Ci (out of k conceivable classes), and |S| indicates the quantity of samples in S. At that point the entropy of the set S:

$$Info(S) = -\sum_{i=1}^{k} ((freq(Ci,S)/|S|).\log 2\ (freq(Ci, S)/ |S|))$$

After partition of T in accordance with n outcomes of one attribute test x:

$$Info_x(T) = \sum_{i=1}^{n} ((|Ti|/ |T|).Info(Ti))$$

$$Gain(X) = Info(T) - Info_x(T)$$

Here objective is to select an attribute with the highest Gain value.

5. Decision Tree based on MapReduce

Problem with Decision Tree is that too large to scan over a single machine while dealing with big data. So we are proposing Decision Tree with MapReduce!

MapReduce is helpful for batch processing on terabytes or petabytes of information. Some of MapReduce's key advantages are Simplicity, Scalability, Speed, Recovery, Minimal Data Motion etc. Decision Tree based on MapReduce [5] shown in following figure [14]:

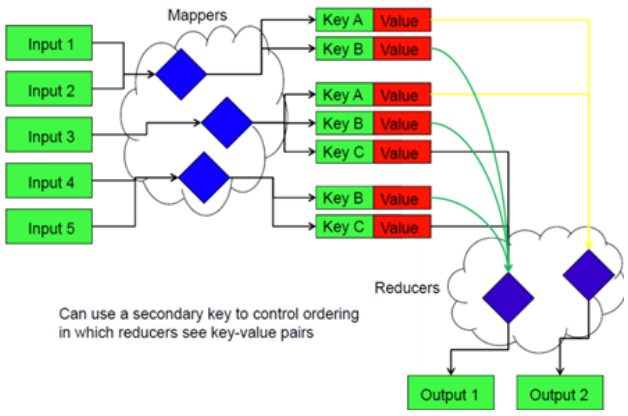

Figure 4 MapReduce Concept

Proposed Algorithm Steps

We fabricate the tree level by level. One MapReduce step fabricates one level of the tree.

Mapper

•Consider different possible parts (Xi,v) on its subset of the data.

•For each split, it stores partial-statistics.

•Partial split-statistics are sent to Reducers.

Reducer

•Collects each split-statistics and chooses best split.

Master builds up the tree for one level.

Mapper loads the model and data about which attribute splits to consider.

Every mapper sees a subset of the information D*

Mapper "drops" each data point to locate the proper leaf node L

For every leaf node L it keeps insights about

(1) Data achieving L

(2) Data in left/right subtree under split S

Reducer totals the statistics (1), (2) and decides the best split for every tree node

The responsibility of map stage is to get the <key, value> type of the thing in $Node0$, and yield the information as $Node1$, $Node2$, $Node3$, … , $Nodem$. Map has the responsibility to get the aggregate line number of preparing. The reduce stage is to get the aggregate number of qualities that has the same key from the yield of guide stage. At that point the <key, sum>s are yield to HDFS. A combiner, who is like reducer, is included before the reducer so as to diminish the measure of the information to be transmitted through system. With the aftereffect of reduce output, it's a straightforward employment for us to get the gain ratio of every attribute in $Nodei$ and get the split attribute that has the maximum gain ratio. Flowchart for the same is depicted in following figure [15]:

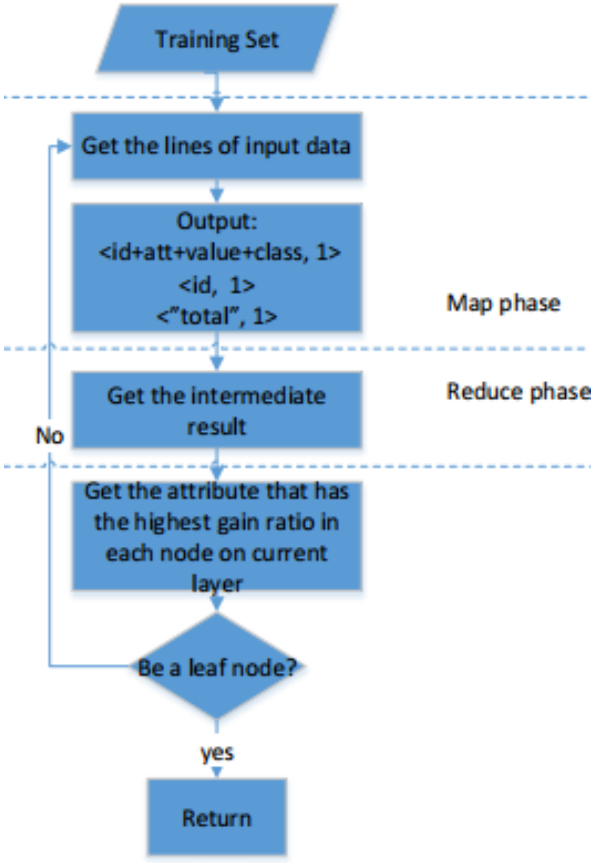

Figure 5 Decision Tree with MapReduce-Flow

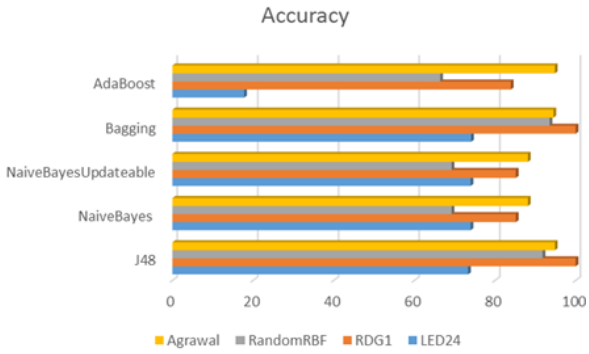

Figure 6 Accuracy of different classifiers in weka

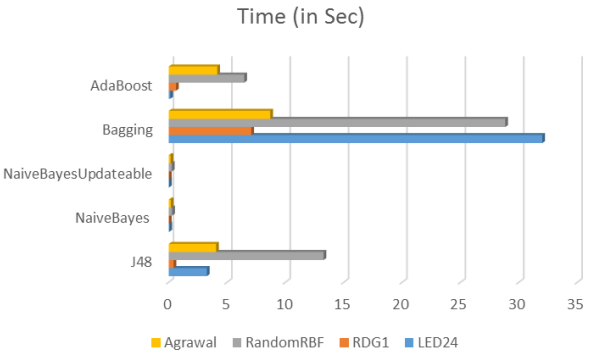

Figure 7 Training time by different classifiers in weka

6. Results

Latest version of weka is also supporting big data. Weka explorer can handle ~1 M instances, 25 attributes. Simple CLI works incrementally whenever it can. We have generated 100000 instances in weka and tested accuracy of various inbuilt datasets in weka with the help of different classifiers like NaiveBayes, Adaboost, Bagging, J48 etc. Results are shown in following figure:

We have also check time required to train each data by different classifiers (shown below).

7. Conclusion

The idea of this paper is to discourse about usefulness of different data mining tools, techniques, platforms, frameworks in big data. Results shows that decision tree can give better accuracy by minimizing training time. This paper has been incited by the reality that filed specialists can utilize the decision tree for big data. Research on big data is not restricted to software engineering field. It can be related to different fields like pharmaceutical services, production, biomedical, transportation (basically by enormous planes like AirBus A380s) and so on. There are many open issues like Security and Privacy Issues; Noise, Outliers and Data Consistency Issues; Data Mining algorithms for Map-Reduce solutions [13] etc.

References

[1] Lloyd Allison, “Types and classes of machine learning and data mining”, Proceedings of the 26th Australasian Computer Science Conference, Vol. 16, Page No. 207-215, 2003.

[2] Chintan Bhatt, "Telemedicine-The New Era of Healthcare" in CSI Communications, Page No 15-16, 2013.

[3] Zheng J, Dagnino A. An initial study of predictive machine learning analytics on large volumes of historical data for power system applications. In: 2014 IEEE International Conference on Big Data; pp. 952–59, 2014.

[4] De Mantaras, R.L., A distance-based attribute selection measure for decision tree induction. Machine Learning 6, 81-92, 1991.

[5] Biswanath Panda, Joshua S. Herbach, SugatoBasu, Roberto J. Bayardo, PLANET: Massively Parallel Learning of Tree Ensembles with MapReduce, VLDB ‘09, August 24-28, 2009.

[6] Tim KraskaAmeet, Talwalkar, John Duchi, Rean Griffith, Michael J. Franklin, Michael Jordan, MLbase: A Distributed Machine-learning System, 6th Biennial Conference on Innovative Data Systems Research (CIDR’13), January 6-9, 2013.

[7] Sara Landset, Taghi M. Khoshgoftaar, Aaron N. Richter and TawfiqHasanin, Journal of Big Data 2015, pp-2:24, 2015.
[8] Gianmarco De Francisci Morales, Albert Bifet, SAMOA: Scalable Advanced Massive Online Analysis, Journal of Machine Learning Research 16 (2015), pp-149-153, 2015.
[9] Joseph K. Bradley, Practical Machine Learning Pipelines with MLlib, Spark Summit East, 2015.
[10] Petuum: A New Platform for Distributed Machine Learning on Big Data Eric P. Xing, Qirong Ho, Wei Dai, JinKyu Kim, Jinliang Wei, Seunghak Lee, Xun Zheng, PengtaoXie, Abhimanu Kumar and Yaoliang Yu, KDD 2015.
[11] Ian J. Goodfellow, David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio, Pylearn2: a machine learning research library, 2013.
[12] http://www.bizruntime.com/bigdata.html.
[13] Zhao W, Ma H, He Q. Parallel k-means clustering based on mapreduce. Proceedings Cloud Comp. 2009.
[14] https://d396qusza40orc.cloudfront.net/mmds/lecture_slides/week6_DT_using_MP.pdf.
[15] Hua Wang, Bin Wu, Shuai Yang, Bai Wang, Yang Liu, Research of Decision Tree on YARN Using MapReduce and Spark, WORLDCOMP ’12, 2012.
[16] Arinto Murdopo, Distributed Decision Tree Learning for Mining Big Data Streams, 2013.