**Evaluation of Random Forest Performance on Distributed Environment using Harp**

Prashanth Balasubramani

Indiana University  
107 S Indiana Ave, Bloomington, IN 47405   
+1812.318.2839

prasbala@indiana.edu

Shree Harsha Sridharamurthy

Indiana University  
107 S Indiana Ave, Bloomington, IN 47405  
+1812.349.8449

sridhash@iu.edu

**ABSTRACT**

Random forests were introduced by Breiman in 2001. They are versatile wherein they operate on data under two-class or multi-class classification problems. Based on the published paper [1], we formulate the performance for random forests in the parallelization context. We experiment with subsampled data on the on-time airline dataset for a single sequential decision tree and the more advanced parallelized version – the random forest.

**Categories and Subject Descriptors**

D.3.3 [**Programming Languages**]: Language Constructs and Features – *abstract data types, polymorphism, control structures.* This is just an example, please use the correct category and subject descriptors for your submission*.* The ACM Computing Classification Scheme: <http://www.acm.org/class/1998/>

**General Terms**

Your general terms must be any of the following 16 designated terms: Algorithms, Management, Measurement, Documentation, Performance, Design, Economics, Reliability, Experimentation, Security, Human Factors, Standardization, Languages, Theory, Legal Aspects, Verification.

**Keywords**

ID3, Decision Tree, Harp, Random Forest, Machine Learning, Classification, Regression.

# INTRODUCTION

Random forests algorithm is an example of ensemble learning. It can be used in both classification and regression. We first construct a model based on the training data. Then we classify the input data points of the test data using the model.

This approach in classifying the test data assumes the fact that the classes are not ordinal i.e., there is no intrinsic relationship amongst the classes themselves for example, values such as low, medium and high – the only matter of concern would be the features values – numbers present in the columns other than the class.

# PROBLEM DEFINITION

# PROPOSED METHOD

Decision tree is a hierarchical machine learning algorithm used in classification and regression problems. Based on the predetermined dataset called the training dataset, the algorithm builds a data structure consisting of features and classes, using the greedy approach.

The algorithm inputs the training dataset, calculates the information gain (relevance) of the features (fields) based on their entropy (impurity) and proceeds with generating the treenodes representing either the feature or the class associated with its feature (represented by its parent node).

The tree will contain a feature and its child nodes as classes. Based on the input value of the test dataset, the corresponding class will be assigned to the data point.

## Intuition

The basic idea behind comparison of random forests against the single decision tree model is the performance factor. Due to the splitting of test dataset before classification by every decision tree running in parallel, we expect a significant increase in the speed during classification.

## Approach

The parallelized version of the decision tree algorithm implements an ensemble of these individual decision trees. The decision trees are created by smaller sets of randomly chosen data points therefore every decision tree is different form the other. This instills a diverse set of classifiers whose outputs can be aggregated into more accurate resultset in a shorter period of time. The algorithm for parallelizing the random forest model is designed as follows:

**main node:**

reads the file into adjmatrx HashMap of size n

creates m processes

creates global rankValues HashMap initializes each node's rank to 1/n

creates global outBoundNodes HashMap and updates it with dangling nodes

creates global outBoundNodesCount HashMap and updates it with dangling nodes count

creates global inBoundNodes HashMap based on outBoundNodes HashMap

splits the rankValues into n/m chunks and adds remaining nodes to the last process itself

sends the chunks to every process

for every iteration,

waits for all processes to send back the calculated pageRank for their nodes

combines the output of all processes into the global rankValues HashMap

splits the updated rankValues HashMap into n/m chunks again

send the updated chunks to all processes

after iteration completes, prints the rankValues into

**sub nodes:**

receive the chunk of nodes of which the rank values have to be calculated

for every node in the chunk

calculate pageRank based on the global outBoundNodesCount and inBoundNodes HashMaps

sends the new pageRank values chunk back to the main process

# EXPERIMENTS

## Environment

Our testing environment consists of two physical nodes in the Juliet cluster j-082 and j-083.

We first executed our sequential code on both the machines to obtain similar outputs, example for the train dataset of x rows and a test dataset of y rows, the time taken to build the tree was p ms and time taken to classify was q ms.

After splitting the dataset and generating two decision trees ensemble, we obtained

## Questions to be answered

We wanted to know how much would be the increase in performance when the following parameters were changed: the amount of data points randomly selected to build the trees, the number of nodes running the decision tree, the number of records in the training data, and the number of records in the testing data.

## Observations

The title (Helvetica 18-point bold), authors' names (Helvetica 12-point) and affiliations (Helvetica 10-point) run across the full width of the page – one column wide. We also recommend phone number (Helvetica 10-point) and e-mail address (Helvetica 12-point). See the top of this page for three addresses. If only one address is needed, center all address text. For two addresses, use two centered tabs, and so on. For more than three authors, you may have to improvise.

## First Page Copyright Notice

Please leave 3.81 cm (1.5") of blank text box at the bottom of the left column of the first page for the copyright notice.

## Subsequent Pages

For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible.

Table 1. Table captions should be placed above the table

|  |  |  |  |
| --- | --- | --- | --- |
| **Graphics** | **Top** | **In-between** | **Bottom** |
| Tables | End | Last | First |
| Figures | Good | Similar | Very well |

## References and Citations

Footnotes should be Times New Roman 9-point, and justified to the full width of the column.

Use the “ACM Reference format” for references – that is, a numbered list at the end of the article, ordered alphabetically and formatted accordingly. See examples of some typical reference types, in the new “ACM Reference format”, at the end of this document. Within this template, use the style named *references* for the text. Acceptable abbreviations, for journal names, can be found here: <http://library.caltech.edu/reference/abbreviations/>. Word may try to automatically ‘underline’ hotlinks in your references, the correct style is NO underlining.

The references are also in 9 pt., but that section (see Section 7) is ragged right. References should be published materials accessible to the public. Internal technical reports may be cited only if they are easily accessible (i.e. you can give the address to obtain the report within your citation) and may be obtained by any reader. Proprietary information may not be cited. Private communications should be acknowledged, not referenced (e.g., “[Robertson, personal communication]”).

## Page Numbering, Headers and Footers

Do not include headers, footers or page numbers in your submission. These will be added when the publications are assembled.

# FIGURES/CAPTIONS

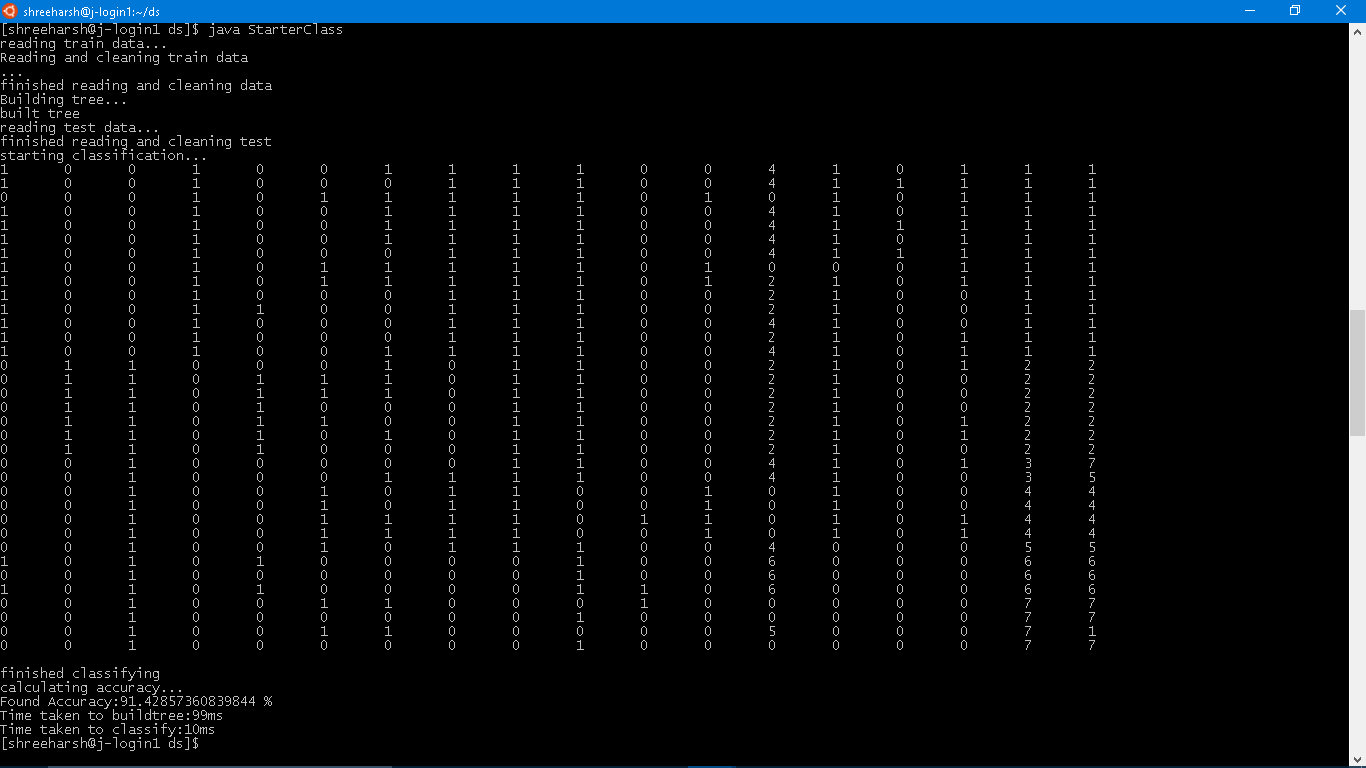


Figure 1: Output of sequential algorithm with 66 train data points and 35 test points.

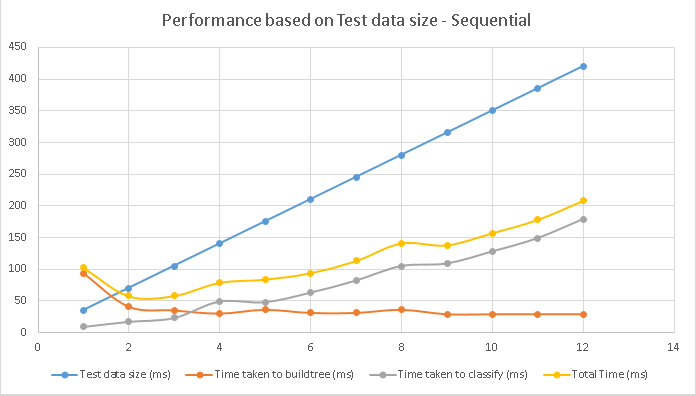


Figure 2: Time taken by the program for varying amounts of test data size.

# SECTIONS

The heading of a section should be in Times New Roman 12-point bold in all-capitals flush left with an additional 6-points of white space above the section head. Sections and subsequent sub- sections should be numbered and flush left. For a section head and a subsection head together (such as Section 3 and subsection 3.1), use no additional space above the subsection head.



Figure 1. Insert caption to place caption below figure.

.

## Subsections

The heading of subsections should be in Times New Roman 12-point bold with only the initial letters capitalized. (Note: For subsections and subsubsections, a word like *the* or *a* is not capitalized unless it is the first word of the header.)

### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized and 6-points of white space above the subsubsection head.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

# ACKNOWLEDGMENTS

Our thanks to Prof. Judy Qiu and the Associate Instructors for the guidance received during the course of learning and development of this project.

# REFERENCES

1. Mitchell, Tom M. (1997) Machine Learning [Online version]. Retrieved from <http://personal.disco.unimib.it/Vanneschi/McGrawHill_-_Machine_Learning_-Tom_Mitchell.pdf>.
2. Genuer, R., Poggi, J.-M., Tuleau-Malot, C., Villa-Vialaneix, N., 2015. Random Forests for Big Data. arXiv preprint arXiv:1511.08327.
3. Wright, M.N., Ziegler, A., 2015. Ranger: A fast implementation of random forests for high dimensional data in C++ and R. arXiv preprint arXiv:1508.04409. Editors, Zhang, C., Ma, Y. Ensemble Machine Learning Methods and Applications. Location: Springer <https://www.transportation.gov/sites/dot.gov/files/docs/2016AprilATCR.pdf>.
4. Statistical Computing and Statistical Graphics. Get the data [Online dataset]. Retrieved from <http://stat-computing.org/dataexpo/2009/the-data.html>
5. Bureau of Transportation Statistics, Airline Service Quality Performance 234. Retrieved from <http://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp>
6. Tan, Steinbach, Kumar. Introduction to Data Mining 2005. Classification: Basic Concepts, Decision Trees, and Model Evaluation https://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

Columns on Last Page Should Be Made As Close As Possible to Equal Length