CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Bradford Scholars

Author Proof

Computational Complexity Analysis of Decision Tree Algorithms

Habiba Muhammad Sani(✉), Ci Lei, and Daniel Neagu

University of Bradford, Bradford, UK
{hmsani,clei1,dneagu}@bradford.ac.uk

Abstract. Decision tree is a simple but powerful learning technique that is considered as one of the famous learning algorithms that have been successfully used in practice for various classification tasks. They have the advantage of producing a comprehensible classification model with satisfactory accuracy levels in several application domains. In recent years, the volume of data available for learning is dramatically increasing. As a result, many application domains are faced with a large amount of data thereby posing a major bottleneck on the computability of learning techniques. There are different implementations of the decision tree using different techniques. In this paper, we theoretically and experimentally study and compare the computational power of the most common classical top-down decision tree algorithms (C4.5 and CART). This work can serve as part of review work to analyse the computational complexity of the existing decision tree classifier algorithm to gain understanding of the operational steps with the aim of optimizing the learning algorithm for large datasets.

Keywords: Classification · Decision trees · Complexity

1 Introduction

Today, the world is overwhelmed with the continuous growth in the amount of data available for learning. Useful information is hidden within this large amount of data. With machine learning techniques, we can analyse these data and extract meaningful information that can aid in decision making [6,19]. As large data are becoming a common norm in many application domains, trying to discover useful pattern and information from these real-world data pose several challenges such as memory and time complexities. Managing and analysing such amount of data requires special and very expensive hardware and software, which often causes various companies and organizations to exploit only a small part of the stored data. According to [3], one of the major challenges for the data mining research community is to develop methods that facilitate the use of learning algorithms for real-world databases. Machine learning is a sub-field of computer science with the goal of training and programming machines to learn

© Springer Nature Switzerland AG 2018
M. Bramer and M. Petridis (Eds.): SGAI-AI 2018, LNAI 11311, pp. 1–7, 2018.
https://doi.org/10.1007/978-3-030-04191-5_17

Author Proof

from experiences to become expert in applying their experiences in new situation [14,17]. There are many classification algorithms available. Decision tree algorithms are among the successful and widely used technique for classification tasks. Decision tree techniques have been around over three decades now and are still actively used in many real-world applications. Given the long history and interest in decision tree algorithms, it is surprising that there has been few work done on the computational complexity of the decision tree algorithm in the literature [11,20]. The goal of this paper is to theoretically and experimentally analyse and compare the complexity of decision tree algorithm for classification task. The decision tree classifiers chosen are the ones with high number of citation and implemented in Scikit-learn python packages for machine learning. The rest of the paper has the following organization: Sect. 2 provides some background of basic concepts in relation to the paper. Theoretical analysis is presented in Sect. 3. Experimental and result analysis were provided in Sect. 4. Finally, Sect. 5 presents the conclusion about this paper as well as a highlight on the future work.

2 Background

2.1 Classification

Classification is a type of supervised learning task in which a training set of labelled examples is given, and the goal is to form a description that can be used to predict previously unseen examples. Formally, the training set is denoted as $S\langle X, y\rangle$. Where X is the set of input attributes $\mathrm{X} = \{x_1, x_2, ..., x_n\}$ and y represents the target or class attribute such that $y = \{c_1, c_2, ..., c_n\}$.

2.2 Decision Tree Algorithm

Decision tree inducers are algorithms that automatically construct a decision tree from a given dataset [13]. Typically, the goal is to find the optimal decision tree by minimizing the generalization error. There are various top-down decision tree algorithms such as IDE [12], C4.5 [16], and CART [8]. Some implementation consists of both the tree growing and tree pruning phase such as (C4.5 and CART) and other algorithms are designed to perform only the tree growing phase [1,10].

2.3 Computational Complexity

Computational resources are crucial in any practical application of learning algorithm. These computational resources are of two basic types: Sample complexity and computational complexity [2,4,7,15]. Algorithm analysis is an important part of a broader computational complexity theory which provides theoretical estimates for resources needed for any given algorithm which solves a given computational problem.

Author Proof

3 Theoretical Analysis

Generally, the actual runtime analysis of an algorithm depends on the specific machine on which the algorithm is being implemented upon. To avoid machine dependence analysis, it is a common approach in literature to analyse the runtime of an algorithm using asymptotic sense which is a standard approach in computational complexity theory [5,18]. In these kind of analysis, it is required that the input size n of any instance to which the algorithm is expected to be applied be clearly defined. However, in the context of machine learning algorithms, there is no clear notion of the input size since learning algorithms are expected to detect some pattern from a dataset and can only access random sample of that data [17]. Therefore, the computational analysis is usually performed to determine the worst-case scenario.

3.1 Analysis of C4.5 Algorithm

Given that in general the decision tree algorithm follows the divide and conquer scheme which is similar to quick sort algorithm. When an algorithm contains a recursive call to itself, its running time can often be described by a recurrence equation which describes the overall running time on a problem of size n in terms of the running time on smaller inputs. In decision tree basically, the computational complexity of building decision tree is mainly concentrated on the criterion function consisting of two basic primary operations the entropy gain calculation of the class attribute and the entropy of the input variables in the training set with respect to the class. The estimated complexity of computing the probability for each class labelled is bounded by the size of the sample. So, the cost is O(n). The computation performed on one input attribute requires O(n log_2 n) and since all attributes are considered then the total cost for this operation will be O(mn log_2 n). Similarly, to analyse the recursive call of the algorithm on the subset of the training set, the estimated complexity for such operation is O(n log_2 n) since at each partition, the algorithm considers the instances and their respective target values. Hence, the total running time complexity for C4.5 algorithm can be estimated by combining the cost for each of the basic operations in decision tree building as:

$$T(S, X, y) = O(n) + O(mnlog_2 n) + O(nlog_2 n) \tag{1}$$

where S is the training set, X is the input attributes and y represents the target. This running time can be simply expressed asymptotically as O(mn log_2 n) which is the dominant factor and it is thus a logarithmic function of n. However, with some approach in which the algorithm repeatedly re-evaluate the dataset each time the procedure is called, the running time can exponentially scale up to $O(m^k n^q)$ where k and q can be any constant c≥2.

3.2 Analysis of CART Algorithm

The complexity of CART can also be estimated using the same notion as in C4.5 since in theory the key operational steps of constructing decision tree generally

Author Proof

follows the same structure. One major difference between the two algorithms lies in the splitting criterion used for the selection of attribute. CART uses gini index splitting criteria. Hence, the process of constructing decision tree using CART algorithm can also be estimated as O(mn log_2 n) where m is the attributes and n is the observations.

4 Experimental Analysis

This section experimentally study the running time of decision tree algorithms. We are basically interested in the behaviour of the algorithms as the number of instances increases.

4.1 Datasets Used

To compare the performance of the classifiers, some frequently used datasets obtained from UCI machine learning repository are used. The characteristics of the datasets is given in Table 1.

Table 1. Characteristics of datasets

Datasets	Sample size (n)	Features	Classes (c)
Breast cancer	699	9	2
Pima diabetes	768	9	2
Banknote	1372	5	2

4.2 Experimental Setup

The experiment was carried out on a 64- bit computer with windows 10 operating system, dual-core Intel i7-6700 (3.4 GHz) desktop with 16 GB RAM running windows 10 operating system. The algorithms used for the experiment were implemented in the popular scikit-learn python library for data analysis and Anaconda-Jupyter Notebook editor 3.6 was used to obtain our results. Even though, our concern is basically on complexity analysis, but the accuracy was also considers since there is usually a trade-off between the two [9].

4.3 Results and Discussion

The goal of the experimental work is to analyse the behaviour of the decision tree algorithms as the size of the input dataset increases. Table 2 shows accuracy score over 10 cross validation runs as well as the running time measured in milliseconds for cancer, diabetes and banknote datasets respectively. Similarly, Figs. 1, 2 and 3 shows the graphical plot of only the running time results presented in Table 2 respectively. Overall, the results show that the running time of both algorithms grows nearly linearly as the size of the input increases as expected. However, CART algorithm outperforms the C4.5 across all the datasets in terms of the running time; but the differences is not significantly pronounced.

Author Proof

Table 2. Accuracy(*Acc.*) and time (t) result for cancer, diabetes and banknote datasets respectively.

Sample size n	Cancer dataset				Diabetes dataset				Banknote dataset			
	C4.5		CART		C4.5		CART		C4.5		CART	
	Acc.	t	*Acc.*	t	*Acc.*	t	*Acc.*	t	*Acc.*	t	*Acc.*	t
50	88.8	16.3	85.0	16.2	42.5	18.5	54.6	17.4	1.0	15.1	1.0	15.0
100	90.0	19.3	84.0	18.9	57.6	23.2	52.6	19.5	1.0	15.6	1.0	15.2
200	93.8	22.8	88.8	22.6	69.8	32.3	65.3	26.9	1.0	15.9	1.0	15.8
300	90.6	26.9	90.8	25.8	60.8	42.3	57.8	32.5	1.0	16.4	1.0	15.9
400	91.7	29.9	89.7	29.0	67.6	54.2	64.8	39.7	1.0	16.5	1.0	16.1
500	92.3	33.7	93.1	32.7	67.5	63.7	67.7	48.5	1.0	16.6	1.0	16.3
600	93.3	36.5	94.3	35.6	69.1	69.8	68.2	54.8	1.0	16.8	1.0	17.0
700	-	-	-	-	68.6	79.2	68.3	60.4	1.0	17.1	1.0	15.9
800	-	-	-	-	-	-	-	-	1.0	44.4	1.0	42.8
900	-	-	-	-	-	-	-	-	1.0	45.5	1.0	45.2
1000	-	-	-	-	-	-	-	-	1.0	62.1	1.0	49.1

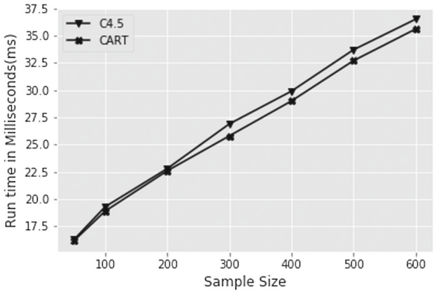

Fig. 1. Run time vs sample size for cancer dataset

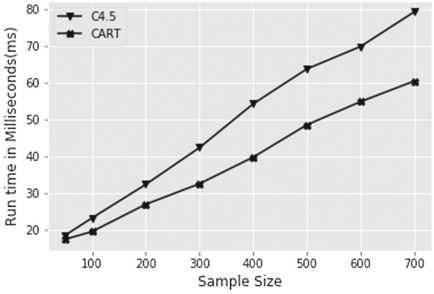

Fig. 2. Run time vs sample size for diabetes dataset

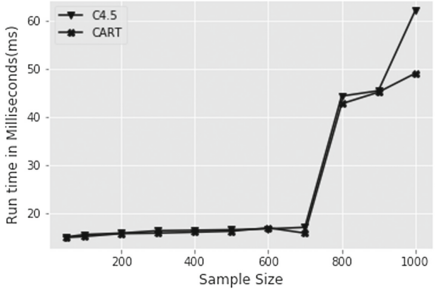

Fig. 3. Run time vs sample size for banknote dataset

Author Proof

5 Conclusion and Future Work

This paper theoretically and experimentally analysed and compared the execution time of the two basic decision tree algorithms implementation in scikit-learn python machine learning library (C4.5 and CART). The deeper investigation into the algorithm behaviour with different problem settings and techniques for possible improvement over the complexity of the algorithm remains the future work to further consider.

References

1. Barros, R.C., De Carvalho, A.C., Freitas, A.A., et al.: Automatic Design of Decision-Tree Induction Algorithms. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-319-14231-9
2. Bovet, D.P., Crescenzi, P., Bovet, D.: Introduction to the Theory of Complexity. Citeseer (1994)
3. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI Mag. **17**(3), 37 (1996)
4. Gács, P., Lovász, L.: Complexity of Algorithms, Lecture Notes. Yale University (1999)
5. Goodrich, M.T., Tamassia, R., Goldwasser, M.H.: Data Structures and Algorithms in Python. Wiley, Hoboken (2013)
6. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier (2011) AQ1
7. Kearns, M.J.: The Computational Complexity of Machine Learning. MIT Press, Cambridge (1990)
8. Leo, B., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group, Belmont (1984)
9. Lim, T.S., Loh, W.Y., Shih, Y.S.: A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Mach. Learn. **40**(3), 203–228 (2000)
10. Maimon, O., Rokach, L.: Introduction to knowledge discovery and data mining. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 1–15. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-09823-4_1
11. Martin, J.K., Hirschberg, D.: On the complexity of learning decision trees. In: International Symposium on Artificial Intelligence and Mathematics, pp. 112–115. Citeseer (1996)
12. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
13. Rokach, L., Maimon, O.: Top-down induction of decision trees classifiers-a survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **35**(4), 476–487 (2005)
14. Rokach, L., Maimon, O.Z.: Data mining with decision trees: theory and applications, vol. 69. World Scientific, Singapore (2008)
15. Roos, M., Rothe, J.: Introduction to computational complexity. Technical report, Institut fur Informatik, Dusseldorf, Germany (2010)
16. Salzberg, S.L.: C4. 5: Programs for Machine Learning by J. Ross Quinlan Morgan Kaufmann Publishers Inc., 1993. Mach. Learn. **16**(3), 235–240 (1994)
17. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge (2014)

Author Proof

18. Sipser, M.: Introduction to the Theory of Computation, vol. 2. Thomson Course Technology, Boston (2006)
19. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2016)
20. Zaki, M.J., Meira Jr., W., Meira, W.: Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press, Cambridge (2014)