KU LEUVEN

Complete decision tree induction functionality in scikit-learn

Ir. Sven Van Hove

Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Engineering and Computer Science

Thesis supervisors:
Prof. dr. Jesse Davis
Prof. dr. ir. Hendrik Blockeel

Assessor:
Dr. ir. Marc Claesen

Mentor:
Elia Van Wolputte

Academic year 2017 – 2018

Preface

I would like to thank everybody who kept me busy the last year, especially my promoter and my assistants. I would also like to thank the jury for reading the text. My sincere gratitude also goes to my wive and the rest of my family.

Ir. Sven Van Hove

Contents

Abstract

The `abstract` environment contains a more extensive overview of the work. But it should be limited to one page. [EHWP16]

Chapter 1

Introduction

Decision tree induction is one of the most well-known tools in the machine learning community. Most of the theoretical groundwork was laid in the last two decades of the previous century. Researchers Leo Breiman and Ross Quinlan have been particularly influential in this space. Contemporary AI researchers focus most of their attentions on neural networks and in particular deep learning — the hype around DeepMind's AlphaGo [SSS+17] victories comes to mind — but decision tree research is not dead. Researchers still continue to propose new or improved algorithms and analyses.

Theory is one thing, but the algorithms need to be implemented as computer programs to actually be useful. Sci-kit learn [PVG+11] is a very popular machine learning library written in Python. As such, it also contains implementations of various decision tree induction algorithms. Before sci-kit learn became popular, a Java library called Weka [EHWP16] (or "Waikato Environment for Knowledge Analysis" in full) was often used instead. Even today, the implementations of decision tree algorithms in Weka are still in many respects superior to those in scikit-learn. Other libraries that implement similar algorithms exist (e.g., Apache Spark [ZXW+16]), but those are beyond the scope of this text.

The goal of this thesis is to alleviate the discrepancies between sci-kit learn and Weka concerning decision tree induction. Mind that decision tree induction tools can never be truly "complete" as stated in the title because the field is immensely broad and still continues to grow. Nevertheless, an effort can be made to improve feature parity between these two popular tools.

1.1 Thesis structure

The structure of the remainder of this text is as follows. First, an overview of the literature study concerning decision tree induction will be presented. In particular the link between an implementation and its underlying algorithm will be clarified, including the effects of that choice on the capabilities of the tool.

Chapter 2

Literature review

The relevant literature for this thesis mostly consists of papers concerning decision tree induction. These go back many decades, but fortunately there are some review and survey papers that make the work easier [Mur98, KZP07]. On top of the classic literature, the source code and accompanying documentation of scikit-learn and Weka has also been a rich source of information.

2.1 Prerequisites

The reader ought to be familiar with basic machine learning concepts such as supervised learning, classification, regression, model validation and ensemble learning. Furthermore, basic knowledge of decision tree induction is expected. The most important basic concepts will be discussed briefly. Topics that are particularly important for the next chapters will be elaborated on.

2.2 Scope

A wide variety of decision tree induction algorithms exists. Here, only the *top down induction of decision trees (TDIDT)* family is considered because it is the most common approach and it is particularly relevant to the software tools under scrutiny.

Furthermore, only classification trees are considered. With little effort, most TDIDT classification algorithms can be converted to regression algorithms. Yet, these are far less popular and better alternatives such as Xgboost [CG16] exist.

Ensemble methods are also out of scope. Recent decision tree algorithms rarely work with a single tree, but rather with an ensemble of trees. Random forests [Bre01] is a very popular example. Regardless, the scope of this thesis concerns the fundamentals of decision trees, and not their derivatives. Implementation improvements suggested in this thesis could still potentially benefit related ensemble methods.

The algorithms in scope are all offline learning methods invented before the big data era. This implies that computation is done locally and that all data has to fit in memory. As such, online learning methods or distributed algorithms are out of scope.

Finally, only univariate tests are in scope. The test performed in each internal node must only evaluate one attribute of the observation. For categorical attributes, this typically implies checking whether or not the input is equal to a fixed category. For numeric attributes, the input value is compared against a fixed threshold using $\leq$ or $>$. Consequently, the input space is partitioned recursively using axis-aligned hyperplanes. This scope limitation precludes well-known but seldom used extensions such as oblique trees.

2.3 Terminology

Throughout the relevant literature, there is a lack of ubiquitous vocabulary shared by all researchers. To avoid confusion, some basic terms are reviewed. A *decision tree* consists of *(internal) nodes* which are connected to other nodes via a one-to-many *parent-child* relation on one hand, and *leaves* which have no children on the other hand. The *root node* is the only node without parent. In a *binary tree*, the number of children per node is either zero or two.

Induction algorithms typically receive a *training set* as input data to construct a decision tree while a *test set* is used afterwards for model validation. These sets are tables of data where each row represents an *observation*. All observation are fully described by a common set of *attributes*. Some attributes are *categorical*, others may be *numeric*. Because decision tree induction is a part of supervised learning, one or more *class labels* are also associated with each observation.

2.4 Advantages and disadvantages

2.5 Comparison to other ML algos

2.6 Math

2.7 Splitting heuristics

2.8 Stopping criteria

2.9 Overfitting

2.10 Extensions

2.11 Conclusion

Chapter 3

Software for decision tree induction

Intro

3.1 Capabilities

3.2 Conclusion

Chapter 4

Results and discussion

Intro

4.1 Pruning

4.2 Categorical attributes

4.3 Conclusion

Chapter 5

Conclusion

Intro

5.1 Contributions

5.2 Retrospective

5.3 Future work

Appendices

Appendix A

The First Appendix

Appendices hold useful data which is not essential to understand the work done in the master's thesis. An example is a (program) source. An appendix can also have sections as well as figures and references.

Bibliography

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[EHWP16] F Eibe, MA Hall, IH Witten, and JC Pal. The weka workbench. *Online appendix for "data mining: practical machine learning tools and techniques.": Fourth Morgan Kaufmann*, 2016.

[KZP07] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[Mur98] Sreerama K Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, 2(4):345–389, 1998.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[SSS+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[ZXW+16] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. Apache spark: A unified engine for big data processing. *Commun. ACM*, 59(11):56–65, October 2016.

Master's thesis filing card

Student: Ir. Sven Van Hove

Title: Complete decision tree induction functionality in scikit-learn

Dutch title: Complete beslissingsboom inductie functionaliteit in scikit-learn

UDC: 681.3*I20

Abstract:
500 word abstract

Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Engineering and Computer Science

Thesis supervisors: Prof. dr. Jesse Davis
Prof. dr. ir. Hendrik Blockeel

Assessor: Dr. ir. Marc Claesen

Mentor: Elia Van Wolputte