Machine Learning, 16, 235–240 (1994)

Book Review: *C4.5: Programs for Machine Learning* by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993.

STEVEN L. SALZBERG salzberg@cs.jhu.edu

Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218

Editor: Alberto Segre

1. Overview

Algorithms for constructing decision trees are among the most well known and widely used of all machine learning methods. Among decision tree algorithms, J. Ross Quinlan's ID3 and its successor, C4.5, are probably the most popular in the machine learning community. These algorithms and variations on them have been the subject of numerous research papers since Quinlan introduced ID3. Until recently, most researchers looking for an introduction to decision trees turned to Quinlan's seminal 1986 *Machine Learning* journal article [Quinlan, 1986]. In his new book, *C4.5: Programs for Machine Learning*, Quinlan has put together a definitive, much needed description of his complete system, including the latest developments. As such, this book will be a welcome addition to the library of many researchers and students.

Quinlan discusses a wide range of issues related to decision trees, from the core algorithm for building an initial tree to methods for pruning, converting trees to rules, and handling various problems such as missing attribute values. For each of these issues, he gives a clear description of the problem, usually accompanied by an example, and he describes how C4.5 handles it. The detailed examples are usually drawn from real data sets, and they help greatly to illustrate each problem.

Quinlan also covers a number of limitations of C4.5, such as its bias in favor of rectangular regions, and he discusses some ideas for extending the abilities of the algorithm. Except for a brief survey in the first chapter, he does not discuss competing decision tree construction algorithms (such as Breiman et al.'s CART method [1984]) or competing classification methods, presumably because the book is not intended as a general survey of learning algorithms. The book should be most useful as a tool for machine learning researchers or as a supplementary text in an advanced undergraduate or introductory graduate course. Quinlan does not assume any familiarity on the part of readers with any previous work on decision trees, and he explains all the requisite concepts in the early chapters.

2. Summary of contents

Decision tree algorithms begin with a set of cases, or examples, and create a tree data structure that can be used to classify new cases. Each case is described by a set of attributes (or features) which can have numeric or symbolic values. Associated with each training case is a label representing the name of a class. Each internal node of a decision tree contains a test, the result of which is used to decide what branch to follow from that node. For example, a test might ask "is $x > 4$ for attribute x?" If the test is true, then the case will proceed down the left branch, and if not then it will follow the right branch. The leaf nodes contain class labels instead of tests. In classification mode, when a test case (which has no label) reaches a leaf node, C4.5 classifies it using the label stored there.

Decision trees have been used as classifiers for numerous real-world domains, some of which are mentioned and used as examples by Quinlan; e.g., labor negotiations, hypothyroid diagnosis, soybean disease diagnosis, and credit approval. For many of these domains, the trees produced by C4.5 are both small and accurate, resulting in fast, reliable classifiers. These properties make decision trees a valuable and popular tool for classification. The usefulness of decision trees is so broadly accepted that Quinlan does not spend much time motivating them. Instead, he focuses on details of the C4.5 algorithm and solutions to a set of problems that have arisen over the years among decision tree researchers.

In the first chapter, Quinlan briefly summarizes other models of classification, including rule-based systems, nearest-neighbor (instance-based or case-based) classifiers, neural network learning algorithms, genetic algorithms, and maximum likelihood classifiers. The summary is intentionally very brief, almost telegraphic. Quinlan gives a few references for further reading, and anyone who wants to understand any of these other methods must look elsewhere; the references contained here would only serve to get one started.

Chapter 2 covers the basic divide and conquer algorithm used by all decision tree methods. As Quinlan explains at the beginning of the chapter, this idea has been around since at least the late 1950s. Each internal (non-leaf) node of a tree must contain a test that will divide up the training cases. This presents the first and perhaps most interesting problem in designing a decision tree algorithm: what is the best test to use at a node? Supposing that there are only two classes, then the ideal binary test is one for which an answer of "yes" means the example is always in one class and an answer of "no" means the example is always in the other class. This test would lead to perfect classification accuracy. However, such tests are hard to find, and for many domains they may not exist at all. Thus the problem becomes one of finding a test that comes as close as possible to this ideal, perfect discriminator.

C4.5 and its predecessor, ID3, use formulas based on information theory to evaluate the "goodness" of a test; in particular, they choose the test that extracts the maximum amount of information from a set of cases, given the constraint that only one attribute will be tested. Chapter 2 gives the mathematical definitions of the information gain and gain ratio criteria, and shows how the standard definitions can be extended to handle continuous-valued attributes. What is missing from this chapter, and what would have been interesting to read, is a discussion of alternative "goodness" criteria. Even though C4.5 does not use other criteria, many others have been proposed and used in decision trees (a number are discussed in Breiman et al. [1984]). For example, a much simpler criterion is simply to count the number of examples that would be mis-classified by a test, and to choose the test that minimizes that number.[1]

Chapters 3, 4, and 5 contain short discussions of some common issues that arise in decision tree construction, and show how C4.5 handles them. Many data sets contain cases for which some attribute values are unknown, and Chapter 3 considers this problem. Quinlan has compared different methods for handling unknown attribute values elsewhere [Quinlan, 1989], and he found that while "some approaches are clearly inferior, ... no one approach is uniformly superior." Thus in Chapter 3 he outlines the approach taken by C4.5, which has worked satisfactorily for him, without making any stronger claims about it. The modifications to handle unknown values are straightforward: basically, cases with unknown values are ignored while computing the information content, and the information gain for an attribute X is then multiplied by the fraction of cases for which the value of X is known. Thus if X is unknown for a large fraction of cases, the information gained by testing X at a node will be quite small. This corresponds to the natural intuition about how such attributes should be treated. Quinlan gives a detailed example showing how unknown values may affect the tree construction process.

A more interesting problem is that of overfitting, which is considered in chapter 4. A decision tree that correctly classifies every example in a training set might not be as good a classifier as a smaller tree that does not fit all the training data. In order to avoid this problem, most decision tree algorithms employ a "pruning" method, which means that they grow a large tree and then delete some portion of it. An alternative method is to stop growing the tree once the training set has been sufficiently subdivided (using a "stopping" criterion). Quinlan has experimented with stopping criteria in the past, and in fact some versions of ID3 used this approach to avoid overfitting. But he explains here that the results were uneven, so he has adopted the pruning approach for C4.5. C4.5's pruning method is based on estimating the error rate of every subtree, and replacing the subtree with a leaf node if the estimated error of the leaf is lower. The idea is as follows: suppose that one could estimate the error rate of any node in a decision tree, including leaf nodes. Beginning at the bottom of the tree, if the estimates indicate that the tree will be more accurate when the children of node n are deleted and n is made a leaf node, then C4.5 will delete n's children. If the estimates were perfect, this would always lead to a better decision tree. In practice, although these estimates are very coarse, the method often works quite well. Quinlan illustrates his pruning method with a detailed example from the Congressional voting domain. Before adopting any pruning method though, the reader should look further into the literature, since the efficacy of a pruning method varies in different domains. For example, Schaffer [1992] investigated three different pruning methods and found many conditions in which pruning actually decreases accuracy. The study by Mingers [1989] contains fairly extensive comparisons of different pruning methods in a variety of domains.

For those who want to convert trees into rules, Chapter 5 discusses how to re-write a tree as a set of rules. It also shows how the resulting rules can be simplified, and how some rules can even be eliminated. At first glance, it might seem that pruning the tree should provide all the simplification one could want. But even a pruned tree might contain rules with unnecessarily complicated antecedents, which Quinlan illustrates quite clearly through examples.

The basic method for doing the conversion is very simple, but some of the details for simplifying the rules are much more complicated. This chapter explains both, and as a result is one of the denser chapters in the book. The conversion process takes every leaf of a tree and initially creates one rule. It makes rules by tracing back up the tree and collecting all the tests into a set joined as a conjunction, which becomes the antecedent of the rule. The class label is of course the consequent. To begin simplifying the rules, Quinlan uses a greedy strategy that considers eliminating each of the conditions in turn. He estimates the error rate of the rule when each of the conditions have been deleted, and eliminates the condition that reduces the error rate the most. This continues as long as the error rate can be reduced. If a rule has an unacceptably high estimated error, it might be discarded completely. The problem is that the resulting rule set might contain overlapping rules, and it might not cover all possible cases. Thus one needs to decided which rule to apply when a case is covered by more than one rule, and one also needs to add a default rule that covers cases not otherwise covered. Quinlan's solution is to look for subsets of rules that cover each class C in the data, using the minimum description length (MDL) principle for guidance in finding small subsets. He presents a number of algorithmic ideas for finding these rules, and supplements his descriptions with an example from the hypothyroid diagnosis domain.

Chapter 5 stands out in an odd way, because it leaves the topic of decision trees and ventures into the realm of rule-based systems. Readers who are accustomed to an expert systems framework will find this chapter most useful, as will those who are serving a user community that does not find decision trees appealing. Converting trees to rules gives these people an alternative way of delivering classifiers to domain experts and other users of classification systems.

Chapter 6 discusses the use of windowing, a technique that Quinlan originally developed to overcome the memory limitations of earlier computers. The idea is that a subset of the training data, called a window, is selected randomly, and the decision tree is built from that. This tree is then used to classify the remaining training data, and if it classifies everything correctly, the tree is complete. Otherwise, all the mis-classified examples are added to the window, and the cycle repeats. These memory limitations no longer exist; however, windowing is still available as an option in C4.5. Quinlan explains that he has retained it because in some cases it allows faster construction of trees. In addition, it can also produce more accurate trees, due to the use of randomization. Since the initial window is chosen randomly, the windowing procedure tends to produce a different tree each time it is run. By running C4.5 with windowing several times and picking the tree with the lowest predicted error rate, it is possible to produce a more accurate tree than would be produced without windowing. Another possibility, not mentioned by Quinlan but equally useful in this context, is to generate a number of different trees and allow them to vote on the class of a new case.

In the remaining chapters, Quinlan discusses some details about running C4.5 and performing experiments, which should be useful to anyone who really intends to use the system. In Chapter 9 he includes an example showing how to run the program itself, for those who have the code on disk. Quinlan also discusses a number of shortcomings of C4.5, what might be called built-in biases. Most interesting is the bias towards rectangular regions: since C4.5 only chooses a single attribute to test at each internal node, it splits the attribute space with an axis-parallel hyperplane. As Quinlan explains, this means that whatever the shapes of the class regions, the best that C4.5 can do is to approximate them by hyperrectangles. Another interesting problem is that of grouping attribute values: it may be that some of the values of a given attribute should be grouped together and treated as one value. This is a difficult problem, because the number of such groups is exponential, even if one considers only binary partitions (i.e., partitions that split the examples into two sets). Quinlan gives a nice example that shows another aspect of the problem: suppose the attribute denotes a chemical element. One grouping might include heavy and light elements, while another might include electric conductors and non-conductors. Because these groupings overlap, once a program chose one and divided the training set accordingly, it could no longer use the other. Chapter 7 discusses a greedy algorithm that C4.5 uses to find groupings, but thus far its results have been unsatisfactory, and this remains an open problem. Yet another open problem is that of continuous classes: most work in machine learning on classification has concerned discrete classes. Predicting a real number rather than a discrete class label is quite different, and in some ways much harder. Quinlan's current approach involves putting linear models at the leaves of a decision tree, which in effect means that the tree contains a piecewise linear approximation to a function. This work is still experimental, though, so the code is not included as part of the C4.5 system.

An important warning to potential readers: the main text of the book is very short, ending on page 107, so the thickness of the book can be misleading if one does not thumb through the contents. Nearly two-thirds of the book (pages 109–290) is a listing of the source code, including comments, for the C4.5 system. Including the code was probably unnecessary, especially since it can be purchased on a disk directly from the publisher. Quinlan explains in the introduction that a thorough understanding of the system requires reading the code. While this may be true, it nonetheless would have been more helpful to provide English summaries of the major subroutines. Any interesting algorithmic details will be very hard to find in the raw code. In addition, it is difficult to page through a long source listing examining different, related pieces of code. (Quinlan tried to alleviate this by including a contents pages listing the page numbers of all the files and an index of major functions.) Most programmers are accustomed instead to putting up several windows on their computer screen, where each window contains some code of interest. If a reader wants to look through the code, he would be well-advised to buy the disk.

3. Concluding remarks

This is a very well-written book, and any reader not familiar with Quinlan's writings will quickly discover that Quinlan's style is refreshingly clear. The examples are especially helpful, as they provide concrete illustrations of many of the algorithmic details of C4.5. In part because the writing is so good, it would have been nice to have a longer book in which he surveyed some of the substantial body of literature on decision trees, such as pruning methods, alternative goodness criteria, and experimental results. It would have been especially interesting to read what Quinlan thinks of the numerous experiments comparing C4.5 and its predecessors to the other classification algorithms that are commonly used in the research community. Even at its current length, though, this book will be very useful to researchers and practitioners of machine learning. Anyone intending to use C4.5 in their experiments or on a practical problem will find this book an invaluable resource.

Notes

[1] It might be a useful extension to C4.5 to allow the user to plug in different goodness criteria. On the other hand, the information gain criterion is a defining feature of C4.5, so perhaps the same code with another criterion would not really be C4.5.

References

Breiman, L., and J. Friedman, R. Olshen, and C. Stone (1984). *Classification and Regression Trees*, Belmont, CA: Wadsworth International Group.

Mingers, J. (1989). An empirical comparison of pruning methods for decision tree induction. *Machine Learning* 4, 227–243.

Quinlan, J.R. (1986). Induction of Decision Trees. *Machine Learning* 1:1, 81–106.

Quinlan, J.R. (1989). Unknown attribute values in induction. *Proceedings of the Sixth International Machine Learning Workshop* (pp. 164–168). San Mateo, CA: Morgan Kaufmann.

Quinlan, J.R. (1992). *C4.5 Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann.

Schaffer, C. (1992). Deconstructing the digit recognition problem. *Proceedings of the Ninth International Machine Learning Workshop* (pp. 394–399). San Mateo, CA: Morgan Kaufmann.