Decision Tree Classifiers
A Comparative Study of Decision Tree Learning Algorithms

Nick Carey

April 20, 2014

Abstract

This is where your abstract goes. An abstract should be a short paragraph telling the reader what you did, who might care, and why. Your abstract is what will be read first; most readers will use the abstract to decide whether or not to read the rest of the paper. It is therefore important to have an abstract that is both clear and concise, as well as accurately describing what the paper is about. 50-100 words is generally a good guideline for length, and it should generally be only a single paragraph.

For instance, for the midterm project paper, you should mention that you used decision trees for classification, and that you used and compared both traditional information-theoretic methods and evolutionary methods for building decision trees. You should also mention how the two techniques performed in comparison to each other. Overall, your abstract should read as a very short (but still grammatically correct) summary of what the rest of the paper is about. It should not contain details about methods or results; you'll get to those later. It should contain a brief summary of your main conclusions and/or contributions to the field. Your abstract should not be any longer than this one; as abstracts go, this is already getting a bit long.

Some publication venues expect you to have "keywords" at the end of your abstract, but you don't need to do that for this course.

1 Introduction

It is incredibly valuable for a program, without human intervention, to make a correct decision given a description of an environment or situation. With traditional imperative programming, a programmer might specify all possible situations along with the correct decision for each situation. However, in most applications this approach would be prohibitively labor-intensive and expensive. Rather, if a program could be given examples of situations paired with correct program behavior, perhaps it could learn how to make the right decision in similar future situations. This is the fundamental machine learning problem of classification; a program, or classifier, must correctly classify, or categorize, a previously unseen object based on past experiences and examples.

Classification is a very difficult task for a program to accomplish. There exist many different techniques and algorithms for classification, and entire textbooks are devoted to different classification algorithms[1]. There is no one best classification algorithm for all applications, as gaurenteed by the "No Free Lunch" theorem[5]. In fact, the best all-purpose classification programs are actually ensembles of classification algorithms where the final classification decision is typically the result of a special voting between many different classifiers running on the same dataset[2]. Yet even ensemble classifiers are not always ideal; it may require prohibitively many resources to be able to run several sub-classifiers at the same time on the same dataset. Also consider that an ensemble classifier will behave differently based on the weighted importance of each sub-classifier[4].

Since there is no one best classification algorithm[5], it is useful to compare classifier performance in different situations. With knowledge of the relative performance of classifiers we can make informed decisions on which classifier to apply to a given application. For example, some classifiers take a very long time to come up with an answer, but the classification is very accurate; other classifiers may give an answer very quickly, but can be less accurate. These properties, along with others, are important to consider when choosing a particular classifier for an application, or when choosing which sub-classifiers to consider in an ensemble classifier[4].

In this paper, we investigate properties of several different decision tree type classifiers. Decision trees are a common and natural method for classifying an object based on that object's attributes[3]. This paper compares the relative performance of different algorithms for generating decision trees.

In Section 2 we describe decision tree classifiers and the algorithms that attempt to compute the optimal decision tree. In Section 3 we outline our experimental set-up for comparing the different decision tree algorithms. Section 4 shows the results of the experiments run, and Section 5 is a discussion of the experimental results.

As indicated by their name, decision tree classifiers generate a decision tree to classify new objects. A decision tree takes a list of attributes belonging to an object and returns a classification for that object[3]. At each node in the tree, a single attribute is examined. Based on the value of the examined attribute, a path to a child node is taken. By testing attributes and following a path through the tree based on the values of those attributes, eventually a leaf node is reached. Each leaf node contains a classification, and arriving at a leaf node is equivalent to deciding a classification for the object under examination.

2 Learning Decision Trees

Here, you should talk in more detail about the specifics of how decision trees are implemented, as a preface to the next subsection.

2.1 Information-Theoretic Methods

Here, talk about traditional decision tree learning. You can give the generic algorithm for decision tree learning, and talk about the need for a method for choosing which variable to add to the tree next, which leads you into the next subsubsection. I've chosen not to number this subsubsection; you don't need numbers everywhere, though you should number section headers. After that, it's up to you. It's mostly a matter of æsthetics.

Entropy and Information Gain

Here, talk about the maximum information gain heuristic. You should both give an intuitive feel for what it does and give the mathematics and theory behind it. You should also discuss why it tends to work well for decision trees, and what strengths and weaknesses it has. This should lead you into a discussion of gain ratio, which you can then define and contrast to straight information gain.

2.2 Genetic Algorithms

Here, talk about how genetic algorithms work. Give the basic algorithm, and discuss what needs to be determined (eg. encoding, fitness function, selection method, population size, etc.).

Genetic Algorithms for Decision Trees

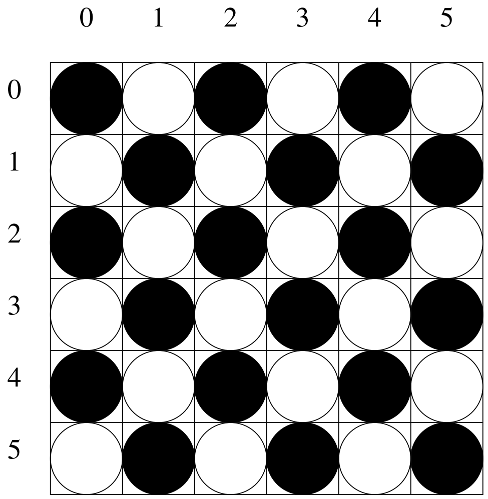
Here, discuss the particulars of your GA. This should definitely include a discussion of your encoding, and how your mutation and crossover functions worked on it. It should also include a discussion of your fitness function, your selection method, and any other choices you made when designing your algorithm. It is very important to note that you should not only talk about what you chose, but also about how and why you chose it. This should not include how you tuned various parameters that were chosen through experimentation; that should come later, and the experiments you used should appear in your results section.

What you do need to talk about is the theoretical reasons behind your choices. For example, what makes your encoding an appropriate one for this domain? How did that impact the choice of mutation and crossover functions? How did it influence your choice of fitness function? How did you come up with your fitness function and why is it a good one for this task?

3 Algorithms and Experimental Methods

In this section, you describe any details of your algorithm that you left out of the more theoretical discussion in the previous section. You don't need to list every single parameter value, but ones that are important to your results should be discussed (eg. don't just list them, talk about why they have that value). Do not, however, discuss the implementation details of your code; the reader is probably not interested in what language you wrote in, or what data structures you used (unless your paper is about languages or datastructures, of course), and she certainly won't care what you named your variables.

You should also describe your experimental methodology; this is where you talk about your data and what you did with it. Talk about what sorts of experiments you performed, and how you validated them. For example, if you used 7-fold cross-validation, you would say that you used it, define what it is, and discuss how you implemented it. It would also be good to discuss



	0	1	2	3	4	5
0	●	○	●	○	●	○
1	○	●	○	●	○	●
2	●	○	●	○	●	○
3	○	●	○	●	○	●
4	●	○	●	○	●	○
5	○	●	○	●	○	●

Figure 1: This is a caption on the figure

the strengths and weaknesses of your chosen validation method; why did you pick the one you did?

Data Sets

Here you should describe the data you used; where it came from, what it represents, what properties it has (eg. binary class? multi class? multi variate? continuous? dimensionality? number of examples? etc.). Talk about all the data sets you used. Be sure to mention and properly cite their source.

Also mention how you pre-processed the data, if you did.

4 Results

The results section should contain your results. It should *not* contain your interpretation of those results. That comes later. This section should be made up primarily of graphs and tables that show your data. You should also have a small amount of text describing what each of the tables and graphs shows, since the caption on the figures should be short. Having text describing the specifics of the experiment that lead to that particular table

	col1	col2	col3
row1	a	b	c
row2	d	e	f
row3	g		h
	i	j	k

Table 1: This is a caption on the table. Try to keep your captions short; don't put multiple paragraphs of text in here. Put the long version in the Results section, and reference the table from there.

would also be good. For example,

> "Table 1 shows the average results of the three algorithms on all the data sets. The parameters used were $N = 7$, and $k = 3.27$; these parameters were found by hand, and little effort was made to tune them optimally. Each algorithm was run three times on each of the seven data sets, and the resulting accuracy scores were averaged."

I'm not going to tell you exactly what tables or graphs you should have here, since it will depend a bit on your results. You should be sure that your results section contains sufficient data to support your conclusions about the relative strengths and weaknesses of the different algorithms. You should also be sure that your data is complete; that is, don't leave data out simply because it doesn't support the point you're trying to make.

You should also be sure that your results are clear and interpretable. Seven pages of raw binary data will do nothing to edify your reader. Similarly, a 1 inch square graph with 12 lines plotted on it will be difficult to extract meaning from, as will a graph with poor (or no) labels on the axes. Your results should be legible both on screen and in hard copy.

You don't want to present results that are just raw data, since that is hard to interpret. But you don't want to over abstract, either, since that leads to results that have little or no meaning (eg. "the average over all different data sets, algorithms, and parameters" is a completely useless statistic for comparing algorithms).

If you are trying do draw comparisons between certain things, try to make sure your results are presented in a way that allows the reader to easily compare them. This might mean having two lines in the same graph, or it might mean having the columns you want to compare be adjacent in a table. You want it to be easy for the reader to follow whatever reasoning you make in your Conclusions section by looking at your Results. You don't want to redundancy (don't put the same data in multiple different tables), but you want the reader to be able to evaluate your conclusions without having to compare things that are three pages apart in your Results.

You should have several pages of results; one or two tables are unlikely to be sufficient to describe your experiments. If they are, you need to do more experiments. On the other hand, if you have more than five pages, you're probably not doing a good job of presenting your data in a clear and concise form.

5 Discussion

The discussion section is where you discuss your interpretation of the data you presented in the results section. This is where you tell the reader how great your algorithm is, and how interesting it is that *this* performed better than *that* on some given data set. You can also speculate about causes for interesting behaviors; for example, if you think you might know why it fails so badly on some particular case, or if you have an insight into why it did well on another case. You don't want to be making wild guesses, but as long as you make it clear that you are not making claims of factual proof, you can go out on a limb a little. For example,

> "In most cases, algorithm A outperforms algorithm B with a significance of 99.8%. However, as can be seen from Figure 1, when applied to the "E. E. Smith" data set, algorithm A does no better than random chance. It seems likely that the failure of algorithm A to learn is due to the extremely sparse distribution of that data set. Because of algorithm A's heavy reliance on data being densely sampled from the true underlying distribution, any sparse data set is likely to show this behavior."

6 Conclusions

The conclusion section should be relatively short, and should not be a summary of your paper. It should, however, bring up what you learned and what impact your results have on the rest of the field (and society as a whole, if applicable, but don't overstate the impact of what you're doing). You should conclude, and bring your paper to an end with any parting thoughts that are appropriate.

Certain types of papers can be ended with a "Summary" section instead of a "Conclusions" section, in which case you would, in fact, summarize the main points of your paper. For this paper, you should write a Conclusions section, not a Summary.

Conclusion also often contain information about what else you would like to do. Sometimes this is a separate subsection, or even a section, entitled "Future Work." The basic idea here is to talk about what the next steps to take would be. This is of benefit to others who are interested in your work and may want to help advance it. It is also a chance for you to acknowledge shortcomings in your work; since we never have infinite time to prepare a paper, there are always more experiments that would have been nice to include. If you list them as future work, then it at least makes it clear that you didn't do those things because you didn't have time, rather than because you didn't realise that they were important to do.

In your paper, you should include a brief discussion of avenues for possible future work in your Conclusions section. It should be tied in with the rest of your conclusion, and should not be an unrelated section tacked on the end (or the middle).

References

[1] Thomas M. Mitchell. *Machine Learning.* McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[2] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[3] Stuart J. Russell, Peter Norvig, John F. Candy, Jitendra M. Malik, and Douglas D. Edwards. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 3 edition, 1996.

[4] Alixandre Santana, Rodrigo G. F. Soares, Anne M. P. Canuto, and Marcílio Carlos Pereira de Souto. A dynamic classifier selection method to build ensembles using accuracy and diversity. In Anne M. P. Canuto, Marcílio Carlos Pereira de Souto, and Antônio C. Roque da Silva, editors, *SBRN*, pages 36–41. IEEE Computer Society, 2006.

[5] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997.