To appear in W. Cohen & H. Hirsh (eds.), *Machine Learning: Proceedings of the Eleventh International Conference*. (New Brunswick NJ, July 1994.) Morgan Kaufmann, San Francisco CA.

In Defense of C4.5: Notes on Learning One-Level Decision Trees

Tapio Elomaa
Department of Computer Science
P. O. Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki, Finland
elomaa@cs.helsinki.fi

Abstract

We discuss the implications of Holte's recently-published article, which demonstrated that on the most commonly used data very simple classification rules are almost as accurate as decision trees produced by Quinlan's C4.5. We consider, in particular, what is the significance of Holte's results for the future of top-down induction of decision trees. To an extent, Holte questioned the sense of further research on multilevel decision tree learning. We go in detail through all the parts of Holte's study. We try to put the results into perspective. We argue that the (in absolute terms) small difference in accuracy between 1R and C4.5 that was witnessed by Holte is still significant. We claim that C4.5 possesses additional accuracy-related advantages over 1R. In addition we discuss the representativeness of the databases used by Holte. We compare empirically the optimal accuracies of multilevel and one-level decision trees and observe some significant differences. We point out several deficiencies of limited-complexity classifiers.

1 INTRODUCTION

In his recent article Holte (1993) examined empirically the classification accuracy of one-level decision trees, i.e., concept classifiers that decide an instance's class on the basis of the value of a single attribute. He compared their accuracy to that of the state-of-the-art decision tree learner C4.5 (Quinlan, 1993). The test domains included a large portion of domains which have, over the years, been used as standard test cases for new inductive learning programs. The domains can nowadays be found at the University of California at Irvine repository of machine learning data sets (Murphy and Aha, 1992).

Holte's experiments revealed that the average classification accuracy achieved by C4.5 over 16 commonly used data sets is not much higher than that achieved by his learning algorithm 1R, which produces one-level decision trees. Holte, on the one hand, discusses the implications of his results quite sparingly and, on the other hand, makes quite far reaching conclusions of the significance of his results. The hastiest people have already condemned further research on top-down induction of decision trees (TDIDT) to be futile. We do not believe that to be the case, not at least on the basis of Holte's study. These notes present our grounds.

In this paper we argue that the prediction accuracy differences between 1R and C4.5 witnessed by Holte are more significant than he lets us believe. Furthermore, C4.5 possesses the additional advantage of being stable in classification accuracy, while 1R lacks that facility (Section 2). We also consider, in Section 3, how representative of current and future application domains of inductive learning programs are those data sets that have been collected to the UCI repository. We claim that "naturality" of a data set is often misused criterion. In Section 4, we pay a brief visit to classification accuracy upper bound considerations, as introduced by Holte. We carry out an empirical comparison of the TDIDT and the one-level decision tree learning approaches' optimal classification accuracies. Section 5, then, concentrates on concept representation issues. In particular we argue that there are inherent downsides to using extremely simple classifiers and that the "simplicity first" methodology promoted by Holte does not constitute a good basis for knowledge acquisition. In Section 6 we discuss related work, empirical and theoretical. Finally, Section 7 presents the main conclusions of the current discussion.

2 PREDICTION ACCURACY

The fact that quite simple classification rules perform well on many of the UCI repository data sets really should not come as a great surprise to anyone who has spent some time experimenting with these domains. It is in fact curious that no one (before Holte) has seen it fit to use the accuracy of 1-level decision trees as the benchmark accuracy measure of their empirical experiments instead of the 0-level decision tree (one leaf = the most common class value) accuracy that is often announced. Surely, it has been known that the latter in many cases is not a good reference point. In more recent experiments, one can consider that C4.5's ancestor ID3

(Quinlan, 1986) has often acted in the role of a benchmark accuracy/tree complexity measure.

Anyhow, it is our understanding that it has been known (suspected), even if not articulated, that new induction algorithms do not improve a great amount in classification accuracy from that which could be reached by quite simple rules. Typically these empirical comparisons do not reveal greater differences in classification accuracy between two approaches than those reported by Holte to exist between 1R and C4.5. In that light, such differences cannot be deemed insignificant.

As an example, let us consider hyphenating Finnish text. Hyphenating Finnish is much easier than hyphenating English: the hyphenation points are usually at the boundaries of morphological word units. Native speakers are able to hyphenate Finnish words by ear almost flawlessly. The following very simple CV-rule takes care of over 95% of the cases: there is a hyphenation point before any consonant-vowel pair. In order to be able to find the remaining 5% of the hyphenation points the rule set has to be expanded by four more simple rules and a list of c. 250 exceptions (Karlsson, 1985).

The case of the preceding example is archetype of an artificial intelligence problem. High baseline success is achieved by simple means, but further advances require much more effort. It is these hard-earned final strides that validate the term artificial intelligence. In machine learning, in particular, it is these small improvements beyond baseline accuracy that bring about the potential of surpassing human performance. What we are trying to convey is that it is in accordance with the artificial intelligence paradigm to pay a heavy price for minor improvements when absolute performance limits are approached.

The difference of learning one-level decision trees and TDIDT falls under the aforementioned category. As demonstrated by Holte learning one-level decision trees is inherently limited to the accuracy figures he gives. The fact that C4.5 does not exhibit substantially higher accuracy figures than 1R over the 16 domains under consideration is due to the nature of the data sets. Many of them have the same property as our hyphenation example above. In addition, C4.5 does perform significantly better than one-level decision trees on other data (see Section 3). There is little evidence to support Holte's eventual conjecture that TDIDT should be given up in favor of "simplicity first" methodology. If one of the state-of-the-art algorithms does not impress us over a number of domains, it does not invalidate the whole methodology. Holte should have followed his own suggestion here and have tried to find out the upper bound of classification accuracy of the TDIDT approach in order to evaluate its true potential. We venture to approximate the upper bound by C4.5 in Section 4.

One of the most appealing properties of C4.5 to a machine learning practitioner, in addition to being easy-to-use, is its robustness. Given almost any domain C4.5 will most probably produce a decision tree that is fairly compact and competitive in classification accuracy with the outputs of its equals (c.f. Appendix C in Holte's (1993) paper). There are no cases where C4.5 would fall apart, or if there are, then so will all other similar inductive learning algorithms. Robustness is a property that is often even more important than outright classification accuracy. Robustness is clearly not a property of 1R, not on the "natural domains" used by Holte and even less so on "artificial domains" (see the discussion of the next section). It is the easiest task to design a function that cannot be learned (nearly as) accurately by 1R (as by C4.5).

Sticking to very simple classification rules does not seem to possess the potential of growing into a stable method. To strive after extreme concept simplicity seems to risk consistent accuracy. The robustness of C4.5 may well stem from the small overhead in concept representation. It is the empirical observation of Iba, Wogulis, and Langley (1988) that this really is the case in inductive learning: there is an inherent tradeoff between concept simplicity and coverage. We consider concept representation further in Section 5.

To be honest, there are domains where C4.5 is clearly outperformed, for example, by neural nets (see e.g., Thrun *et al*., 1991). They are domains that would require constructing new features from the attributes that are used to express the data. Even the naturally adaptive connectionist approaches lack adequate capabilities in this respect (Craven and Shavlik, 1993). We leave this aspect of concept learning beyond our current discussion.

3 TEST DOMAINS

Holte believes that the practical significance of his study is largely determined by whether or not the test data is naturally arising. That is, whether the data has in anyway been simplified to make it easier for machine learning approaches. We do not see that question to be so central, not in general and, in particular, not in Holte's study. A more relevant question, with the same intent though, would be: why has somebody chosen to apply machine learning techniques to this particular problem out of the numerous available candidates. The evident answer in many cases, unfortunately, is that the data has (more or less consciously) been conceived as suitable to be handled by machine learning techniques. Thus, even though the data technically has not been "engineered", it has been clear from the outset that the data will handle nicely.

Because of the above-mentioned reason, we do not believe that one can trust the 16 UCI domains used by Holte to be representative of all "real" classification problems except the "hard" ones. The success of 1R proves, e.g., that none of the domains is such that the class of an instance is determined with equal weight by two attributes. One would expect such domains to appear among clinical databases, for instance. It is well known that a single symptom hardly ever determines a disease. A patient's symptoms constitute very natural attributes that typically get recorded. There

are medical domains among the 16 databases used by Holte (e.g., BC and LY); why does the effect then not appear. The answer is that the data sets have been engineered, not to accommodate the learning task, but to conform to the classification task. That is, the attributes are as carefully chosen relevant features as, e.g., in the CH chess domain.

Naturality of test data is a much talked about topic in empirical machine learning research. However, it is very hard to get a hold of what is natural. Holte mentions two criteria: the data set must have been drawn from a real-life application and the examples and the attributes used to describe them arise naturally in the domain. It is very hard to define what is a naturally arising attribute or its range. The naturality of attributes depends on the viewpoint. There may be several equally "natural" ways of characterizing a domain. Even more freedom exists in defining the type of an attribute. Moreover, since examples consist of attribute value recordings, their naturality is inseparably coupled with that of the attributes. The question of attributes and their ranges is a fundamental issue that has a decisive influence on the potential of inductive learning algorithms.

One might approach the naturality of features by considering whether they are the same ones that are used by an expert in the domain. This viewpoint would validate calling natural, e.g., the medical domains of the UCI repository. The shortcomings of the approach, however, are evident. First, the contrast would impose human limits onto the capabilities of machine learning techniques. Second, the learning approaches are essentially intended to relieve situations that lack experts completely.

There is little hope of gathering a collection of test data, such that good results over it would validate drawing as far reaching conclusions as Holte draws based on his experiments. It is essential to test machine learning approaches on data drawn from real-life in order not to lose sight of the real goal of our field. Nevertheless, manufactured data suits the purpose too: it is easier to control self-made data in the sense that monitoring the effects caused by changes in data, e.g., to prediction accuracy is easier. The effects are easier to explain because of the better understanding of the domain.

Different engineering and control applications are among the most promising application areas for, e.g., decision tree learners. The world typically is well-defined in these domains, but there still exists a need for machine learning tools because of the large volumes of data and the complex functions involved, which makes it often practically impossible to manage them by hand. These domains fulfill one of Holte's naturality criteria, namely that the data is drawn from real-life application, but not necessarily the other. The requirements of the application dictate the basic requirements for attributes and their types, but in most cases room is left for engineering the attributes.

For example, we are currently involved in a project that aims to teach an autonomous mobile robot to identify its environment. The project concerns mainly novel ways of sensing and perceiving the world. The sensor input from the robot is certainly real-life data, but the attributes in such a novel approach are very much a matter of experimentation and negotiation. In this natural application it is useful to engineer the attributes to suit learning tools being used. In general, if the knowledge engineer becomes involved with the project in an initial enough phase, there probably exists some possibilities for accommodating the data representation to the tools being used without compromising the needs of the application.

Even in engineering domains we cannot always have the attributes of our choice. There may be a precollected database of examples readily available. In that case we just have to cope with the examples and attributes therein. The data in a database typically conforms to the needs of a certain purpose. Whether the attributes in such a collection are naturally arising is a secondary concern. The attribute values may be result of an arbitrarily complex function applied to the primitive observable features. For instance, in the robot domain six normalized centralized moments of the ultrasonic waveform are used.

A further point to be made of such engineering applications is that they are based on some (to some extent idealized) model of the world. In a mathematical model familiar functions tend to be used. In the simplest form familiar functions are Boolean functions. They have been used in showing that certain functions are guaranteed to lead TDIDT to produce suboptimal decision trees (e.g., Quinlan, 1988). Similar functions serve to prove the converse for 1R: it cannot improve in accuracy from the baseline accuracy for a family of Boolean functions.

Consider, for example, *exclusive-or* functions, which are **true** if an odd number of the attributes (variables) are **true**, otherwise the functions have value **false**. Half of the examples of the domain are positive (have value **true**) and half are negative. Thus the baseline accuracy is 50%. Now, it does not matter by which attribute the data is split, equal numbers of positive and negative examples will always fall at both leaves (assuming equiprobable instances). Thus all labelings of leaves will give the same accuracy — 50%. The problem here of course falls under the category of constructive induction problems.

Applicability of machine learning approaches is currently restricted more by the lack of imagination than inherent limitations of the methodology. Decision tree learning has recently found success in completely new areas that differ from those areas that are covered in the UCI repository. In database mining (e.g., Agrawal *et al*., 1992) the data is organized by attributes and thus appears similar as in traditional machine learning data sets, but the database contents is typically different from that of traditional test data. Manipulating text databases (e.g., Sakakibara, Misue, and Koshiba, 1993) and string learning (e.g., Arikawa *et al*., 1993) are two related applications that will undoubtedly become more and more important. In these applications there are no "naturally arising" attributes at all.

Table 1: The baseline accuracies of the datasets (0Rw) and the accuracies of 1R and C4.5 when trained on the whole data.

	Data set							
	BC	CH	GL	G2	HD	HE	HO	HY
0Rw	70.3	52.2	35.5	53.4	54.5	79.4	63.0	95.2
1Rw	72.7	68.3	62.2	78.5	76.6	84.5	81.5	98.0
IDw	97.2	100.0	100.0	100.0	99.7	98.1	97.8	100.0
	Data set							
	IR	LA	LY	MU	SE	SO	VO	V1
0Rw	33.3	64.9	56.7	51.8	90.7	36.2	61.4	61.4
1Rw	96.0	84.2	75.7	98.5	95.0	87.2	95.6	98.4
IDw	100.0	98.2	100.0	100.0	100.0	100.0	99.8	98.6

4 UPPER BOUND ON CLASSIFICATION ACCURACY

Holte derives an upper bound for the accuracy of 1R type algorithms, denoted by 1R*, by testing the accuracy of all candidate trees on an independent test set and choosing the one with the highest accuracy. This gives the highest classification accuracy that is in principle achievable by optimizing the split attribute selection step of 1R. As Holte correctly points out the value 1R* is a very optimistic upper bound of the accuracy of 1R. We do not see that accuracy 1R* could be reached. If we were able to choose in one step the optimal attribute to split the training set on, then we could employ the selection method also in a recursive procedure and, hence, we would be able to produce optimal multilevel decision trees. That does not seem very probable (see discussion below). Holte compares 1R* to C4.5's accuracy. The comparison shows that in any case the average accuracy of 1-level decision trees produced by 1R cannot reach that of multilevel decision trees produced by C4.5.

A more revealing and well-founded comparison than running C4.5 against the unrealistic improved 1R would have been to contrast optimal multilevel decision trees to optimal one-level decision trees over the test domains. Holte himself advises to compute the upper bound on accuracy achievable by improvements before doing large-scale (empirical) studies, but then neglects to do so for TDIDT.

There are two major optimization points in TDIDT. First, the selection of split attributes. Attributes must be ordered cautiously into the nodes of the evolving decision tree so that maximal accuracy is achieved by suitable labeling of leaves. Holte's experiment concerns this step. In C4.5 and similar algorithms choosing automatically the cut-points of continuous attribute ranges based on the values appearing in the training data presents another target for optimization. A similar test scheme as that used by Holte could be used to experiment with this facility. However, with multilevel decision trees going through all the candidate hypotheses would require much computation.

It is easy to see what is the optimal classification accuracy of any TDIDT algorithm, denoted by ID*. In a perfect domain, where there are no conflicting examples (modulo missing values), it is 100%: an optimal strategy can order the attributes so that all examples become classified correctly by directing each example to its individual leaf, if not otherwise. In a noisy or incomplete domain one is not able to distinguish between colliding examples, i.e., those that have exactly the same attribute values, but differ on classification. Thus the classification accuracy depends on the number of such examples.

We have listed in place of ID* in Table 1 the value IDw: the accuracy of C4.5 on the whole data when trained on the same data.[1] Holte also trained 1R on the whole data and measured the accuracy achieved thus. The value is denoted by 1Rw. The 1Rw values of Table 1 are drawn from Holte's paper.[2] The algorithms are not able to surpass these figures (over infinitely many repetitions) when trained and tested on random portions of the data. Value 0Rw is the baseline accuracy of a domain, i.e., the proportion of those examples in the data that belong to the most frequent class. Over the 16 domains the measures have the following average values: 0Rw 60.0%, 1Rw 84.6%, and IDw 99.3%.

Our approximation of ID* is pessimistic in the sense that there might be possibilities to raise the accuracy to 100% on each of the domains by directing the troublesome few examples to their individual leaves. On the other hand, the approximation is optimistic of C4.5's accuracy even with improvements, because C4.5 now has all possible thresholds of a continuous attribute at its disposal and can choose the optimal out of them. This would not happen if only a portion of the data was used in training. One must bear in mind that we are searching for the *optimal* decision tree here. Our experiment goes to show that a decision tree of the listed accuracy does exist. The *optimal* attribute and threshold selection strategies would find this tree even from a partial sample. The accuracies of these trees averaged over a number of repetitions on a randomly drawn test set would be the same as that on the whole data as listed in Table 1.

[1] With parameters `-c 1 -m 1 -b`. That is, pruning confidence level is set to 1%, the minimum number of objects on either side of a threshold test is set to 1, and windowing is turned off.

[2] On average 1Rw is 0.8 percentage points higher than 1R* ($\overline{\text{1R*}} = 83.8$, $\overline{\text{1Rw}} = 84.6$).

The average accuracy difference of 2.1 percentage points between C4.5 and 1R* over the 16 domains has now changed into an average difference of 14.7 percentage points in favor of IDw. Sure enough, there are many domains where the difference of IDw and 1Rw is less than the average value, but in these domains (e.g., HY, IR, MU, SE, VO) there is not much room for further improvement in absolute terms. For instance, the HY domain has baseline accuracy of 95.2%. Now, any sensible learning scheme is guaranteed to reach at least that accuracy (assuming "fair" training data, i.e., in most unpathological settings). The real challenge of machine learning is rather to conquer as much as possible of the remaining 4.8% of the examples (c.f. the Finnish hyphenation example in Section 2). If we were to tabulate the proportions of examples not covered by the default rule that are correctly classified by 1R and C4.5, the accuracy figures would be quite different from those reported by Holte.

Seemingly another type of situation than that of the HY domain appears, e.g., in the IR domain, which has baseline accuracy of only 33.3%. The three classes of the domain all have equally many representatives. The default rule, however, is only able to mention one of the classes. Once one of the relevant attributes can be used to split the example space and several classes can be mentioned in the leaves, significant improvement in accuracy can be obtained. Only in cases where one class has clear dominance will the default rule achieve high accuracy figures. Top-down induction of decision trees possesses the potential of growing a perfect classifier at least for these data sets. The highest error rate of C4.5 when trained on the whole data is only 2.8% on the BC domain.

The lesson to be learned from the above experiment is that even though C4.5 is an impressive tool and belongs to the élite of contemporary inductive programs further improvements on TDIDT are possible. The average gap of 12.8 percentage points between C4.5's accuracy when trained on a random sample of 50% of the data and the value IDw cannot be bridged: it is only on uninteresting data that induction will produce perfect classifiers from small samples. However, with the cost of slightly reduced induction speed, part of the gap could probably be bridged. For example, the splitting rule selection methods in TDIDT are quite straightforward, more subtle and effective methods could be developed. The threshold selection appears to be a much more complicated problem.

Inspired by the "success" of 1R* Holte proposes to use the value 1Rw as a predictor of the accuracy of more complex rules or as the benchmark accuracy in comparisons. We do not see 1Rw to suit either function.

As pointed out already it is differences of only few percentage points that typically get revealed when contrasting inductive learning algorithms. Since differences of that order exist between the accuracy of C4.5 and 1Rw, the method does not suit the first purpose. Perhaps coarser approximations could be done by 1Rw, but TDIDT approaches are extremely efficient and do not really require such an approximator. Neural programs could do with such an auxiliary function, but we doubt that 1Rw approximates well the accuracy of back-propagation, for instance.

The value 1Rw does not really suit the second suggested purpose because it would be confusing rather than revealing to compare the accuracy of a classifier learned from 2/3, say, of the data to one that has been acquired from the full data. It would, though, be sensible to use 1R* or the accuracy of 1R as an (additional) benchmark accuracy. As our discussion above shows, the information about class distribution that is provided by the baseline accuracy typically reveals more in binary classification tasks than in multiclass ones. In multiclass domains, the default rule is only able to bring out severe distortions in the class distribution. In general, 1R* or the accuracy of 1R would constitute a better benchmark accuracy measure in these cases.

5 THE COMPLEXITY OF A CLASSIFIER

It is generally accepted that simplicity should be favored in the classifiers produced by inductive learning programs. The reasons are many: Occam's Razor tells us so, the average cost of classifying instances by a simple classifier is smaller than that when using a complex classifier (Núñez, 1988; Tan and Schlimmer, 1990), and the relevant things (attributes, cut-points, etc.) will be more clearly visible in a simple rule (comprehensibility). However, because of the inherent tradeoff between simplicity and accuracy (Iba, Wogulis, and Langley, 1988) one is doomed to balance between these desired properties.

Extreme simplicity, 1-level decision trees, for example, does not constitute a good basis for comprehensive representations. Consider, e.g., a medical domain where attribute `fever` becomes chosen as the only attribute into a 1-level decision tree (we are currently involved with a set of *Nephropathia epidemica* patient records where this does happen). Even if the success rate of this tree is quite good it cannot be seen to be a comprehensible one. A medical expert cannot make head or tails out of such a classifier.

Humans by nature are mentally opposed to too simplistic representations of complex relations. In particular, in medical domains, where it is well known that a symptom (an attribute) taken separately seldomly has a high predictive power. Moreover, in a well-managed domain, if a simple and meaningful concept appears, we probably do not need machine learning methods to recognize it. The situation, of course, is quite different in a new or otherwise poorly-managed domain like our mobile robot application.

It does not seem that accurate but flat representations of complex functions would increase the intelligibility, either. On the contrary, humans typically are quite good in disregarding slight excess of information and concentrate on the relevant parts of the presented representation. Moreover, as stressed in the foregoing, it seldomly is the case that we would learn anything new from the simple but quite accurate rule of thumb (one such incident in a medical domain,

though, is recorded in the seminal book of Breiman *et al*. (1984)). More often, we use inductive programs to uncover the tacky details that do not meet the eye immediately. Very large decision trees undoubtedly are hard for humans to grasp, but C4.5 provides an excellent remedy to this: convert the tree into a set of rules (typically with the cost of a slight loss of accuracy) (Quinlan, 1987). The independent rules are much smaller entities and easily comprehended by humans.

Based on the positive experiences on 1R Holte turns to advocate the "simplicity first" methodology as a new research paradigm for machine learning. The methodology is not defined explicitly in the article; only some clues to what it could mean are given. Most probably Holte's idea is to set a syntactic bound for the possible classifiers, i.e., strictly delimit their syntactic complexity. However, the discussion of the preceding sections shows that there will be domains where such bounded-complexity classifiers will not be accurate enough. Then we would have to search through more complex classifiers, which in their turn will not prove accurate enough for all our purposes, and so forth. This is an evil circle that cannot be broken. If we aim to develop general-purpose induction tools that are as accurate as possible, we cannot set rigid syntactic restrictions to the concept representation.

Holte eventually also proposes extending the hypothesis space from one-attribute trees to either two-attribute decision trees or linear decision trees. Decision trees on two attributes do not have essentially more expressive power than one-level trees, even though a slightly larger set of databases is managed by them. Holte does not define exactly what he means by linear decision trees. If he intends trees where each node has at most one nonterminal son, then in the binary case these correspond to decision lists (Rivest, 1987). Decision lists do not have a static complexity bound, i.e., their complexity can be increased dynamically as the complexity of the problem increases. Thus, their expressive power is of different magnitude than that of one-level decision trees. At same time the hypothesis space also grows substantially and the complexity of navigating in it increases considerably. In practical decision list learning there are also other sources of hypothesis space complexity (see e.g., Clark and Niblett, 1989).

Another possibility is that Holte intends maximizing the linearity of the learned decision trees as studied early on by Arbab and Michie (1985). In their approach the decision tree that has maximum degree of linearity is sought for from among the candidate hypotheses. The hypothesis space in this case is the space of all decision trees and the problem of navigating in it still remains. This line of research has not to our knowledge been followed since Arbab and Michie's study.

The "simplicity first" methodology is not exactly new, not even in decision tree learning. For instance, ID3 aims to work by a kind of a simplicity first methodology. Recall that ID3 originally pruned trees in construction time, not in a separate pruning phase (Quinlan, 1986). Thus, ID3 in principle only considers as complex trees as are required to fulfill the confidence determined by its parameter(s). In the first step ID3 checks whether one of the possible 0-level trees would be accurate enough, i.e., is there enough statistical evidence to support not expanding the single leaf further. Then, in the second step all 1-level decision trees are inspected. In successive steps not all possible hypotheses are examined, since part of the constructed tree has already been fixed. Failures in choosing the correct attribute will lead ID3 in practice to produce unnecessarily large trees.

6 RELATED WORK

A relaxed simplicity requirement for decision trees has been proposed by Ehrenfeucht and Haussler (1989). They define the *rank* of a decision tree and propose learning trees of minimum rank. Delimiting the rank of a decision tree is a syntactic restriction, but it is not too stringent: only too bushy decision trees are pruned from the hypothesis space. The minimum rank condition leaves the hypothesis space sparse enough to be exhausted in polynomial time. This then provides the possibility of proving PAC learnability of decision trees of at most a fixed rank. Choosing the data representation from among the decision trees of minimum rank for a given data set guarantees that there is an upper bound on the excess of representation. Unfortunately, the upper bound is quite loose.

A converse research methodology for decision tree learning has been put forward by Catlett (1991). He suggests using very large example sets in growing and pruning decision trees. In practice, the larger the training population is, the larger the tree grown will be. In second step Catlett proposes to "overprune" the constructed tree as to obtain a classifier of the desired complexity. Empirical experiments demonstrate that the pruned trees will stay significantly more accurate than trees built even from large subsets. In this approach the hypothesis search (eventually) proceeds from complex to simple. At any given complexity level the trees constructed thus will be significantly more accurate than those that result from search proceeding to the opposite direction. Furthermore, Catlett's study corroborates the claim that the size of a decision tree, as such, affects its accuracy.

Iba and Langley (1992) have carried out in a simplified setting the average-case analysis of a learning algorithm that produces one-level decision trees, or *decision stumps* as they choose to call them. They derive equations that relate the application domain characteristics to the algorithm's behavior and empirically validate these equations. The study of Iba and Langley was more intended as the first step towards the average-case analysis of existing decision tree learning algorithms than as to promote the idea of learning very simple classification rules.

A study that is highly related to Holte's study and our current discussion has been carried out by Schaffer (1993) who explored the fundamental justification of using pruning tech-

niques in decision tree learning. Schaffer arguments that in principle any overfitting avoidance strategy only amounts to a form of inductive bias and, thus, may well degrade performance instead of improving it. The application domain characteristics determine the success of pruning. Schaffer concludes that progress toward both powerful and general methods is impossible, instead we should explore which domain characteristics make overfitting avoidance a beneficial technique to employ. The underlying idea in Holte's study too is to choose a powerful but limited technique. Holte, however, has taken the idea to such extreme that its application potential has virtually shrunk to nothing.

In their recent study Murphy and Pazzani (1994) have examined empirically the relationship between the size of a decision tree that is consistent with some training data and the accuracy of the tree on independent test data. Their main finding is that, for many of the domains, aiming at the smallest consistent decision tree would not constitute a good learning strategy: the average accuracy of smaller consistent trees is on average less than that of slightly larger trees.

The standard statistical theory of decision rule selection based on empirical data is Vapnik's (1982) *method of structural minimization of risk*, which explains, for example, the results obtained by Murphy and Pazzani. Roughly speaking, the theory determines, in case of inclusive families of decision rules of increasing complexity $S_1 \subset \ldots \subset S_q$, from which family S_d should a decision rule be selected so that the overall risk is minimized. The overall risk consists of the *empirical* and *expected* risk on classifying training examples and unseen instances, respectively. If the decision rule is selected from too small set, there is a high risk of high *approximation error*, i.e., none of the hypotheses is able even to classify the training data with adequate accuracy. If, on the other hand, the decision rule is chosen from too large set of candidates, there is a high risk of high *estimation error*, i.e., some of the hypotheses predict poorly the actual example distribution, even though they have high accuracy on the training data ("overfitting"). Vapnik's theory gives a method of balancing the tradeoff by determining, as a function of the sample size, the correct index d such that an element of S_d, which minimizes the empirical risk, also yields a minimal bound on the expected risk.

7 CONCLUSION

We have meticulously gone through the results of Holte's (1993) recent paper. Most of the conclusions that we drew from his results differ significantly from those put forward by Holte himself. Many of the differences basically stem from the fact that we did not accept the UCI repository data sets to be representative of most typical application domains of decision tree learning, while Holte takes that as a premise of his study. Even more importantly, Holte chose to treat 1R and C4.5's average accuracy difference of few percentage points as negligible, while, as a matter of fact, it is these minor increases in accuracy that are the main objective of the study of classifier learning. Rather than declaring them unimportant, extra attention should be paid to any progress that can be achieved. Furthermore, Holte ignored the fact that even the baseline accuracy of some of his test domains comes to within only few percentage points of C4.5's accuracy.

The main message that we tried to convey is that Holte's study does not prove further research on top-down induction of decision trees to be a lost cause. On the contrary, his results put together with our experiment demonstrate that there is further work to be done and significant improvements to be achieved in this field. Holte has, rather, succeeded in proving that the current collection of standard test data for inductive learning is not up to its function. Moreover, we have tried to demonstrate that outside the data sets inspected in Holte's study neither learning one-level decision trees nor learning any other extremely simple classifiers will constitute a good basis for a successful methodology.

Acknowledgements

This research was financially supported by grants from the Ministry of Education, University of Helsinki, and the Academy of Finland.

Jyrki Kivinen, Henry Tirri, and the anonymous referees provided valuable feedback on earlier versions of this paper.

All test data used was obtained from the University of California at Irvine repository of machine learning databases whose current librarian is Patrick Murphy. The domains BC and LY originate from the Institute of Oncology at the University Medical Center in Ljubljana, Slovenia, where they were gathered by Matjaz Zwitter and Milan Soklič. The origins of the other databases are given in Holte's (1993) paper.

References

Agrawal, R., Ghosh, S., Imielinski, T., Iyer, B., and Swami, A. (1992). An interval classifier for database mining applications. In L.-Y. Yuan (ed.), *Proc. Eighteenth International Conference on Very Large Data Bases* (pp. 560–573). San Mateo, CA: Morgan Kaufmann.

Arbab, B. and Michie, D. (1985). Generating rules from examples. In *Proc. Ninth International Joint Conference on Artificial Intelligence* (pp. 631–633). Los Altos, CA: Morgan Kaufmann.

Arikawa, S., Miyano, S., Shinohara, A., Kuhara, S., Mukouchi, Y., and Shinohara, T. (1993). A machine discovery from amino acid sequences by decision trees over regular patterns. *New Gener. Comput.* **11**: 361–375.

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

Catlett, J. (1991). Overpruning large decision trees. In *Proc. Twelfth International Joint Conference on Artificial Intelligence* (pp. 764–769). San Mateo, CA: Morgan Kaufmann.

Clark, P. and Niblett, T. (1989). The CN2 induction algorithm. *Mach. Learn.* **3**: 261–283.

Craven, M. and Shavlik, J. (1993). Learning to represent codons: a challenge problem for constructive induction. In *Proc. Thirteenth International Joint Conference on Artificial Intelligence* (pp. 1319–1324). San Mateo, CA: Morgan Kaufmann.

Ehrenfeucht, A. and Haussler, D. (1989). Learning decision trees from random examples. *Inf. Comput.* **82**: 231–246.

Holte, R. (1993). Very simple classification rules perform well on most commonly used data sets. *Mach. Learn.* **11**: 63–91.

Iba, W. and Langley, P. (1992). Induction of one-level decision trees. In D. Sleeman and P. Edwards (eds.), *Machine Learning: Proceedings of the Ninth International Workshop* (pp. 233–240). Morgan Kaufmann, San Mateo, CA.

Iba, W., Wogulis, J., and Langley, P. (1988). Trading off simplicity and coverage in incremental concept learning. In J. Laird (ed.), *Proc. Fifth International Conference on Machine Learning* (pp. 73–79). San Mateo, CA: Morgan Kaufmann.

Karlsson, F. (1985). Automatic hyphenation of Finnish. In *Computational Morphosyntax* (pp. 93–136). Publication No. 13. Department of General Linguistics, University of Helsinki.

Murphy, P. and Aha, D. (1992). *UCI Repository of Machine Learning Databases* [machine-readable data repository]. Department of Information and Computer Science, University of California at Irvine.

Murphy, P. and Pazzani, M. (1994). Exploring the decision forest: an empirical investigation of Occam's razor in decision tree induction. *J. Artif. Intell. Res.* **1**: 257–275.

Núñez, M. (1988). Economic induction: a case study. In D. Sleeman (ed.), *Proc. Third European Working Session on Learning* (pp. 139–145). London: Pitman.

Quinlan, R. (1986). Induction of decision trees. *Mach. Learn.* **1**: 81–106.

Quinlan, R. (1987). Simplifying decision trees. *Int. J. Man–Mach. Stud.* **27**: 221–234.

Quinlan, R. (1988). An empirical comparison of genetic and decision-tree classifiers. In J. Laird (ed.), *Proc. Fifth International Conference on Machine Learning* (pp. 135–141). San Mateo, CA: Morgan Kaufmann.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Rivest, R. (1987). Learning decision lists. *Mach. Learn.* **2**: 229–246.

Sakakibara, Y., Misue, K., and Koshiba, T. (1993). Text classification and automatic extraction of keywords by learning decision trees. In *Proc. Ninth Conference on Artificial Intelligence for Applications* (p. 466). Los Alamitos, CA: IEEE Computer Society Press.

Schaffer, C. (1993). Overfitting avoidance as bias. *Mach. Learn.* **10**: 153–178.

Tan, M. and Schlimmer, J. (1990). Two case studies in cost-sensitive concept acquisition. In *Proc. Eighth National Conference on Artificial Intelligence* (pp. 854–860). Cambridge, MA: MIT Press.

Thrun, S. *et al*. (1991). The MONK's problems – a performance comparison of different learning algorithms. Report CMU-CS-91-197. School of Computer Science, Carnegie Mellon University.

Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*. New York, NY: Springer-Verlag.