

Sequential Inductive Learning

Jonathan Gratch

Beckman Institute, University of Illinois

405 N. Mathews, Urbana, IL 61801

gratch@cs.uiuc.edu

Phone: (217) 244-1503

FAX: (217) 244-8371

Keywords

Decision Trees, Statistical Analysis, Learning Theory

Abstract

In this paper I advocate a new model for inductive learning. Called sequential induction, this model bridges classical fixed-sample learning techniques (which are efficient but *ad hoc*), and worst-case approaches (which provide strong statistical guarantees but are too inefficient for practical use). According to the sequential inductive model, learning is a sequence of decisions which are informed by training data. By analyzing induction at the level of these decisions, and by utilizing the minimum data necessary to make each decision, sequential inductive techniques can provide the strong statistical guarantees of worst-case methods, but with substantially less data than those methods require. The sequential inductive model is also useful as a method for determining a sufficient sample size for inductive learning and as such, is relevant to megainduction, where the preponderance of data introduces problems of scale. The *peephaling* and *decision-theoretic subsampling* approaches of Catlett, Musick, and Russell, which address problems of scale in inductive learning, have informed my work. The technique that I have developed generalizes and extends these prior approaches to megainduction.

This paper has not already been accepted by and is not currently under review for a journal or other conference. Nor will it be submitted for such during IJCAI's review period. This work is supported by the National Science Foundation under Grant NSF-IRI-92-09394.

Sequential Inductive Learning

Abstract

In this paper I advocate a new model for inductive learning. Called sequential induction, this model bridges classical fixed-sample learning techniques (which are efficient but *ad hoc*), and worst-case approaches (which provide strong statistical guarantees but are too inefficient for practical use). According to the sequential inductive model, learning is a sequence of decisions which are informed by training data. By analyzing induction at the level of these decisions, and by utilizing the minimum data necessary to make each decision, sequential inductive techniques can provide the strong statistical guarantees of worst-case methods, but with substantially less data than those methods require. The sequential inductive model is also useful as a method for determining a sufficient sample size for inductive learning and as such, is relevant to megainduction, where the preponderance of data introduces problems of scale. The *peepholing* and *decision-theoretic subsampling* approaches of Catlett, Musick, and Russell, which address problems of scale in inductive learning, have informed my work. The technique that I have developed generalizes and extends these prior approaches to megainduction.

1. INTRODUCTION

For the most part, inductive learning techniques have enjoyed remarkable success in real-world applications. Much of past learning research has focused on tasks involving relatively small amounts of data; in these situations, it is reasonable to use all available information when learning concept descriptions. The explosive growth in our access to electronic information, however, raises the following question: what is the minimum amount of data that can be used without compromising the results of learning? This question is especially relevant for software agents (or softbots) that have access to endless supplies of data on the Internet [Etzioni93, Perkowski94]. So-called megainduction tasks demand techniques to limit or manage this access to information [Catlett91, Harris92]. In other words, we must determine how much data is sufficient to learn, and how to limit the amount of data to that which is sufficient.

Theoretical machine learning research provides some guidelines for answering this question. Unfortunately, these results are generally inappropriate to guide practical learning. Learnability results generally assume that the target concept is a member of some predefined class (such as *k-DNF*), an assumption that is unreasonable in practice. Recent work in *agnostic PAC learning* [Haussler92, Kearns92] relaxes virtually all assumptions about the form of the target concept. The results of these studies are discouraging, however, seem to have little relevance to the machine learning techniques used in practice.

In this paper I introduce an alternative inductive model that bridges the gap between practical and theoretical models of learning. After discussing a definition of sample sufficiency which more readily applies to the learning algorithms used in practice,

I then describe Sequential ID3, a decision-tree algorithm based on this definition. (The algorithm extends and generalizes the decision-theoretic subsampling of Musick, Catlett, and Russell [Musick93], and it can also be seen as an analytic model that illuminates the statistical properties of decision-tree algorithms.) I conclude this paper with a derivation of the algorithm's theoretical properties and an empirical evaluation over several learning problems drawn from the Irvine repository.

2. SUFFICIENCY

Practical learning algorithms are quite general, in that they make few assumptions about the concept to be learned: the data may contain noise; the concept need not be drawn from some pre-specified class; the attributes may even be insufficient to describe the concept. To be of practical interest, a definition of sufficiency must be of equal generality. The standard definition of sufficiency from theoretical machine learning is what I call *accuracy-based*. According to this definition, a learning algorithm must output a concept description with minimal classification error [Kearns92]. Unfortunately, current results suggest that learning in accordance with this definition is intractable, except for extremely simple concept description languages (e.g., even when the concept description is restricted to a simple conjunction of binary attributes, minimizing classification error is NP-Hard [Kearns92]).

To resolve many of the problems associated with the accuracy based definition, I propose a *decision-based* definition of sufficiency that does support efficient learning. According to this definition, the process of learning is treated as a sequence of inductive decisions, or an *inductive decision process*. A sample is deemed sufficient if it ensures some minimum quality constraints on these decisions. As I will discuss decision-tree learning algorithms in particular, it is important to distinguish between decisions made while *leaning* a decision tree, and decisions made while *using* a learned decision tree. Only the former are discussed in this article, which I will refer to as *inductive decisions*.

The accuracy-based and decision-based definitions are distinct. In particular, ensuring high inductive-decision quality does not necessarily ensure low classification error for the induced concept description. And although learning algorithms are ideally designed to minimize classification error, for many learning situations, minimizing inductive decision error is all that can be done tractably.

In addition to these limitations associated with an accuracy-based definition of sufficiency, there are other reasons to favor a decision-based definition. Decision criteria have been proposed to account for factors beyond classification error, such as conciseness of the concept description [deMantaras92]. Unlike an accuracy-based definition, a decision-based definition applies to these criteria as well.

In top-down decision-tree induction, learning is an inductive decision process consisting of two types of inductive decisions: *stopping decisions* determine if a node in the current decision tree should be further partitioned, and *selection decisions* identify attributes with which to partition nodes. Specific algorithms differ in the particular criteria used to guide these inductive decisions. For example, ID3 uses information gain as a selection decision criterion, and class purity as a stopping deci-

sion criterion [Quinlan86]. These inductive decisions are statistical in that the criteria are defined in terms of unknown characteristics of the example distribution. Thus, when ID3 selects an attribute with highest information gain, the attribute is not necessarily the best (in this local sense), but only estimated to be best. Asymptotically (as the sample size goes to infinity), these estimates converge to the true score; when little data is available, however, the estimates and the resulting inductive decisions are essentially random.¹

I declare a sample to be sufficient if it ensures some minimum bound on the randomness of the inductive decision process. By utilizing statistical theory, one can compute how much data is necessary to achieve these bounds. The next section of this paper formalizes this idea and provides an efficient learning algorithm based on this definition.

3. SEQUENTIAL INDUCTION

Traditionally, one provides learning algorithms with a fixed-size sample of training data, all of which is used to induce a concept description. Consistent with this, a simple approach to learning with a sufficient sample is to determine a sufficiently large sample prior to learning, and then use all of this data to induce a concept description. Following Musick *et. al.*, I call this *one-shot induction*. Because inductive decisions are conditional on the data and the outcome of earlier decisions, the sample must be sufficient to learn under the worst possible configuration of inductive decisions.

As an alternative to one-shot induction, I propose an approach that examines each inductive decision in turn and samples just enough data to make that decision. Called *sequential induction* because data is sampled a little at a time throughout the decision process, this approach has two clear advantages to one-shot induction. First, sequential induction requires substantially less data on average than one-shot induction. For example, consider a learning problem in which a single attribute perfectly separates the classes. A sequential algorithm can recognize this property (with high probability) after the first selection decision, whereas a one-shot algorithm cannot make use of this information as it must choose a sample size before learning begins. Second, one-shot induction biases decision quality in an unusual way. In one-shot induction, earlier selection decisions are based on more data and thus have higher quality than later decisions. A sequential algorithm can control decision quality on a decision by decision basis and thus avoid this bias in decision quality.

This section presents a sequential inductive model for top-down decision-tree induction. The approach applies to a wide range of attribute selection criteria, including such measures as entropy [Quinlan86], gini index [Breiman84], and orthogonality [Fayyad92]. For simplicity, I restrict the discussion to learning problems with binary attributes and involving only two classes, although the approach easily generalizes to attributes and classes of arbitrary cardinality. It is not, however, immediately obvious how to extend the approach to problems with continuous attributes.

1. Standard algorithms address this randomness indirectly, by incorporating a separate pruning stage that follows induction [Breiman84, Mingers]. With a sufficient sample, one can dispense with this second stage and “grow the tree correctly the first time.” One will also need a different stopping criterion (see below).

Before describing the technique, I must introduce the statistical machinery needed for determining sufficient samples. First, I discuss how to model the statistical error in selection decisions and to determine a sufficient sample for them. I then present a stopping criterion that is well suited to the sequential inductive model. Next, I discuss how to insure that the overall decision quality of the inductive decision process is above some threshold. Finally, I describe Sequential ID3, a sequential inductive technique for decision-tree learning, and present its formal properties.

3.1 Selection Decisions

Selection decisions choose the most promising attribute with which to partition a decision-tree node. To make such an inductive decision, the learning algorithm must estimate the merit of a set of attributes and choose that which is most likely to be the best. Because perfect selection cannot be achieved with a finite sample of data, I follow the learning theory terminology and propose that each selection decision identify the “probably approximately” best attribute. This means that when given some pre-specified constants α and ϵ , the algorithm must select an attribute that is within ϵ of the best with probability $1-\alpha$, taking as many examples sufficient to insure a decision of this quality. Although this goal is conceptually simple, its satisfaction requires extensive statistical machinery.

A selection criterion is a measure of the quality of each attribute. Because the techniques I propose apply to a wide class of such criteria, I first define this class. In Figure 1, each node in the decision tree denotes some subset of the space of possible examples. This subset can be described by a probability vector that specifies the probability that an example belongs to a particular class. For example, the probability vector associated with the root node in the decision tree summarizes the base-line class distribution. For a given node N in the decision tree, let P_c denote the probability that an example described by the node falls in class c . Then for an attribute A , let $P_{v,c}$ denote the probability that an example belongs to node N , has $A=v$, and belongs to class c . The effect of an attribute can be summarized by the *attribute probabilities*: $P_{T,1}$, $P_{T,2}$, $P_{F,1}$, and $P_{F,2}$. In fact, three probabilities are sufficient as the fourth is determined by the other three: $P_{F,2} = 1 - P_{T,1} - P_{T,2} - P_{F,1}$.

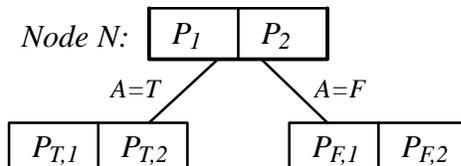


Figure 1. An attribute partitions the set of examples reaching a node into non-overlapping subsets. Within each node, examples are partitioned by the class to which they belong. The effect of an attribute can be summarized by the four probabilities: $P_{T,1}$, $P_{T,2}$, $P_{F,1}$, and $P_{F,2}$.

The techniques I develop apply to selection criteria that are arbitrary, differentiable functions of these three attribute probabilities. For example, expected entropy is an acceptable criterion:

$$e(P_{T,1}, P_{T,2}, P_{F,1}) = -\text{plogp}(P_{T,1}) - \text{plogp}(P_{T,2}) - \text{plogp}(P_{F,1}) - \text{plogp}(P_{F,2}) \\ + \text{plogp}(P_{T,1} + P_{F,1}) + \text{plogp}(P_{T,2} + P_{F,2})$$

Given a selection criterion, the merit of each attribute can be estimated by estimating the attribute probabilities from the data and substituting these estimates into the selection criterion. To determine a sufficient sample and select the best estimate, the learning system must be able to bound the uncertainty in the estimated merit of each attribute. Collectively, the merit estimates form a complex *multivariate* statistic.

Fortunately, a generalization of the central limit theorem, the δ -method, shows that the distribution of the multivariate merit statistic is approximately a multivariate normal distribution, *regardless of the selection criterion's form* (assuming the constraints listed above) [Bishop75 p. 487]. The selection decision thus simplifies to the problem of selecting the ε -max component of a multivariate normal distribution. In the statistics literature this type of problem is referred to as a correlated selection problem and there are known methods for solving it (see [Gratch94b] for a survey of methods). I use a procedure called McSPRT described in [Gratch94b]. STOP1 [Chien95], BRACE [Moore94], and the work of Nelson [Nelson95], are similar to McSPRT and could be used in its place.

McSPRT takes examples one at a time until it determines that a sufficient number have been taken; at this point, it selects the attribute with the highest estimated merit. The procedure reduces the problem of finding the best attribute to a number of pairwise comparisons between attributes. An attribute is selected when, for each pairwise comparison, its merit is significantly greater or indifferent to the alternative. To use the procedure (or the others mentioned) one must assess the variance in the estimated difference-in-merit between two attributes. Figure 2 illustrates how this estimate can be computed.

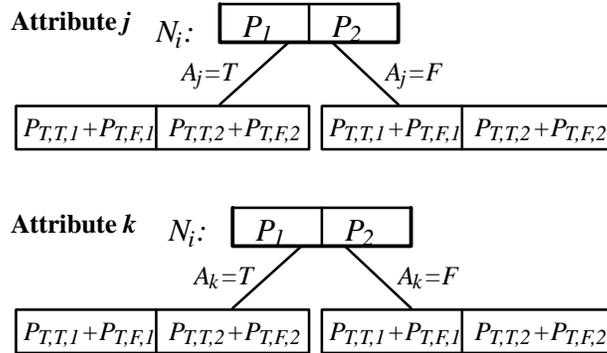


Figure 2. A comparison of two attributes. The attributes partitioned the examples into eight possible bins. The merit of each attribute is a function of these eight probabilities.

The symbol $P_{a,b,c}$ denotes the probability of an example having value a for attribute A_j , value b for attribute A_k , and class c . For a given pair of attributes, the difference-in-merit between the two is a function of seven probabilities (the eighth is determined by the other seven). For example, if the selection criterion is entropy, the difference in entropy between two attributes is

$$\begin{aligned} & \Delta e(P_{T,T,1}, P_{T,T,2}, P_{T,F,1}, P_{T,F,2}, P_{F,T,1}, P_{F,T,2}, P_{F,F,1}, P_{F,F,2}) \\ &= \Delta e(P_1, P_2, P_3, P_4, P_5, P_6, P_7) . \\ &= e(P_1 + P_3, P_2 + P_3, P_5 + P_7) - e(P_1 + P_5, P_2 + P_6, P_3 + P_7) . \end{aligned}$$

This difference can be estimated by substituting estimates for the seven probabilities into the difference equation. The variance of this difference estimate can be derived using the generalized version of the central limit theorem. Using the δ -method, the variance of a difference estimate f is approximately

$$\frac{1}{n} \cdot \left[\sum_{i=1}^7 P_i \left[\frac{\partial f}{\partial P_i} \right]^2 - \left(\sum_{i=1}^7 P_i \left[\frac{\partial f}{\partial P_i} \right] \right)^2 \right], \quad (1)$$

where $\partial f / \partial P_i$ is the partial derivative of the difference equation with respect to the i th probability, and where the seven probabilities, P_i , are estimated from the training data. This can be computed automatically by any standard symbolic mathematical system. (I use Maple™, a system which generates C code to compute the estimate.)

3.2 Stopping Decisions

Stopping decisions determine when the growth of the decision tree should be stopped. In the standard fixed-sample learning paradigm, the stopping criterion serves an almost incidental role because the size of the data set is the true determinant of the number of possible inductive decisions and, therefore, of the maximum size of the decision tree. In sequential induction, however, the algorithm can take as much data as necessary to ensure high quality inductive decisions. Consequently, the stopping criterion directly assumes the role of bounding the number of possible inductive decisions, and thus indirectly determines complexity of the learned concept.

Two sources of complexity motivate the need for a stopping criterion. First, the size of the largest possible decision tree grows exponentially with the number of attributes. Therefore, a tractable algorithm cannot hope to construct a complete tree. Second, as the depth of the tree grows, the algorithm must draw increasingly more data to get reasonable numbers of examples at each decision-tree leaf: if p is the probability of an example reaching a node, the algorithm must on average draw $1/p$ examples for each example that reaches the node. Because the probability of a node can be arbitrarily small, the amount of data needed to obtain a sufficient sample at the node can be arbitrarily large.

I advocate the use of a stopping criterion that addresses both of these sources of complexity. The sequential algorithm should not partition a node if the probability of an example reaching it is less than some threshold parameter γ . This probability can be

estimated from the data and, as in selection decisions, the sequential algorithm need only be probably close to the right stopping decisions. In particular, with probability $1-\alpha$, the algorithm should expand nodes with probability greater than γ , refuse to expand nodes of probability less than $\gamma/2$, and perform arbitrarily for nodes of intermediate probability. A sufficient sample to make this decision can be determined with a statistical procedure called the sequential probability ratio test (SPRT) [Berger80].²

Each leaf node of a tree can be assigned a probability equal to the probability of an example reaching that node, and the probability of all the leaves must sum to one. The stopping criterion implies that the number of the leaves in the learned decision tree will be roughly on the order of $2/\gamma$ and therefore, this stopping criterion determines an upper bound on the complexity of the learned concept.

3.3 Multiplicity Effect

Together, stopping and selection decisions determine the behavior of the inductive decision process, and I have proposed methods for taking sufficient data to approximate each of these inductive decisions. This is not, however, enough to bound the overall quality of the decision process. When one makes multiple inductive decisions, the overall probability of making a mistake is greater than the probability of making a mistake on any individual decision (e.g., on a given roll of a die there is only a 1/6th chance of rolling a five, but after six rolls, there is a 35/35th chance of rolling at least one five). Called the *multiplicity effect* [Hochberg87], this factor must be addressed in order to insure the overall quality of the inductive decision process.

Using a statistical result known as Bonferroni's inequality [Hochberg87 p. 363], the overall decision error is bounded by dividing the acceptable error at each inductive decision by the number of decisions taken. As mentioned previously, I assigned each decision an error level of α . Therefore, if one wishes to bound the overall decision error to below some constant, δ , it suffices to assign $\alpha=\delta/D$ where D is the expected number of inductive decisions. Although I do not know how to compute D directly, it is possible to bound the maximum possible number of decisions, which will suffice. Furthermore – as will be shown – the expected sample complexity of sequential induction depends only on the log of the number of inductive decisions; consequently, the conservative nature of this bound does not unduly increase the sample size.

Space precludes a complete derivation of the maximum possible number of inductive decisions, but this can be found in [Gratch94a]. To derive this number I first find the largest possible decision-tree that satisfies the stopping criterion. This tree has a particular form I refer to as a *fringe tree*, which is a complete binary tree of $\lfloor 2/\gamma \rfloor$ leaves that has been augmented with a “fringe” under each leaf that consumes the remaining attributes. A fringe is a degenerate binary tree with each right-hand branch being a leaf with near-zero probability. A fringe trees is illustrated in Figure 3.

2. I use a one-sided version of SPRT, as it is essential to avoid expanding nodes with probability less than $\gamma/2$.

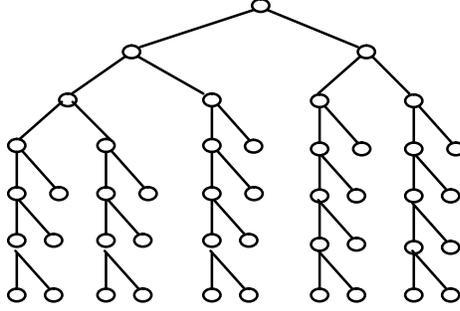


Figure 3. A fringe tree that results from setting γ to 0.40.

The size of a fringe tree depends on the number of attributes, A , and the stopping parameter γ . This size can be shown to be

$$T = F - 1 + (A - d_1)(2^{d_2} - F) + (A - d_2)(2F - 2^{d_2}) \leq (A - d_1 + 1)F - 1 = O(A/\gamma)$$

where $F = \lfloor 2/\gamma \rfloor$, $d_1 = \lfloor \log_2 2/\gamma \rfloor$, $d_2 = \lceil \log_2 2/\gamma \rceil$. Each selection decision consists of a set of pairwise comparisons – one for each attribute that could be split upon. To properly bound the error we must count the number of these pairwise comparisons across every selection decision in the fringe tree, which is

$$S = (2A - 2d_1 + 1)2^{d_1} + \frac{F}{2}(d_1 - A)^2 + \frac{F}{2}(d_1 - A) - A - 2 = O\left(\frac{A^2}{\gamma} + \frac{1}{\gamma} \log^2 \frac{1}{\gamma}\right).$$

The total number of inductive decisions is $T+S$ (dominated in complexity by S).

3.4 Sequential ID3

Sequential ID3 is an algorithm that implements the notions described above. To learn a concept description with Sequential ID3, one must specify a selection criterion (such as entropy) and three constants: a confidence parameter, δ ; an indifference parameter, ϵ ; and a stopping parameter, γ . Given a set of binary attributes of size A and access to classified training data, the algorithm then constructs, with probability $1-\delta$, a decision tree of size $O(A/\gamma)$ in which each partition is made with the ϵ -best attribute. The algorithm does a breadth-first expansion of the decision tree, using McSPRT to choose the ϵ -best attribute at each node while SPRT tests the probability of each decision-tree node.³ If a node is ever shown to have probability less than γ , its descendants are pruned from the decision tree. McSPRT and SPRT require the specification of a minimum sample size on which to base inductive decisions; by default, this size is set to fifteen. Given a selection criterion, Maple™ generates C code to compute the variance estimate. To date Sequential ID3 has been tested with entropy and orthogonality [Fayyad92] as the selection criterion.

Sequential ID3 can be viewed as an elaboration of the decision theoretic subsampling approach of Musick *et. al.*. Sequential ID3 is more general, as it applies to arbitrary selection criteria and relaxes the untenable assumption that attributes are independent. Furthermore, the subsampling approach seems only to have been applied to

3. A node is not partitioned if it (probably) contains examples of only one class, or if all attributes (probably) yield trivial partitions. For simplicity, I ignore these caveats in the following discussion.

inductive decisions at the root node, and does not account for the multiplicity effect. Sequential ID3 addresses both of these limitations. The subsampling approach handles one issue that is not addressed by Sequential ID3: the balance between the size of a sufficient sample and the time needed to determine this size. Sequential ID3 attempts only to minimize the sample size, without regard to the time cost (except to ensure that this cost is polynomial), whereas subsampling approach strikes a balance between these factors.

I have determined a worst-case upper bound on the complexity of Sequential ID3 (the derivations are in [Gratch94a]). Expressed in terms of A , δ , γ , and ϵ , the complexity also depends on B , which denotes the range of the selection criterion (for entropy, $B=\log(2)$).⁴ In the worst case, the amount of data required by Sequential ID3, (i.e., its *sample complexity*) is

$$O\left(\frac{B^2}{\epsilon^2\gamma} \cdot [\log(1/\delta \cdot 1/\gamma \cdot A)]^2\right). \quad (2)$$

The sample complexity grows rapidly with tighter estimates on the selection decisions (quadratic in $1/\epsilon$), and with more liberal stopping criterion ($1/\gamma[\log 1/\gamma]^2$). In this worst case, the algorithm completes in time

$$O\left((A^2 + \log^2 1/\gamma) \cdot \frac{B^2}{\epsilon^2\gamma} \cdot [\log(1/\delta \cdot 1/\gamma \cdot A)]^2\right). \quad (3)$$

For most practical learning problems, Sequential ID3 will take far less data than these bounds suggest. Nevertheless, it is interesting to compare this worst-case sample complexity with the amount of data needed by a one-shot induction technique (which, as noted earlier, determines a sufficient sample size before learning begins). Using Hoeffding's inequality (see [Maron94]), one can show that one-shot induction requires a sample size on the order of

$$O\left(\frac{B^2}{\epsilon^2\gamma} \cdot \log(1/\delta \cdot 1/\gamma \cdot A)\right) \quad (4)$$

which is less by a factor of a log than the amount of data needed by Sequential ID3 (Equation 2). This potential disadvantage to sequential induction highlights the need for empirical evaluations over actual learning problems.

4. EVALUATION

The statistical theory underlying Sequential ID3 provides only limited insight into the expected performance of the algorithm on actual learning problems. One can expect the algorithm to appropriately bound the quality of its inductive decisions, and the worst-case sample and time complexity to be less than the specified bounds. One cannot, however, state how much data is required for a particular learning problem *a priori*. More importantly, one cannot analytically characterize the relationship between decision quality and classification accuracy, because this relationship depends on the structure of specific learning problems. Knowledge of this relationship is es-

4. When comparing selection criteria, it is best to view ϵ not as a fixed number but as a percentage of B .

quential, though, for if Sequential ID3 is to be a useful tool, an increase in decision quality must lead to a decrease in classification error.

I empirically test two central claims. First, decision quality should be closely related to classification accuracy in actual learning problems. More specifically, classification error should decrease as the stopping parameter, γ , decreases or as the indifference parameter, ϵ , decreases. Second, the expected sample complexity of sequential induction should be less than a one-shot method which chooses a fixed-size sufficient sample *a priori*. I first describe the testing methodology and then examine each of these claims in turn. Although Sequential ID3 can incorporate arbitrary selection criteria, this evaluation only considers entropy, the most widely used criterion.

A secondary consideration is how to set the various learning parameters. If the first claim holds, one should expect a monotonic tradeoff between the amount of data taken (as controlled by γ and ϵ) and classification error. The ideal setting will depend on factors specific to the particular application (e.g., the cost of data and the accuracy demands) and the relationship between the parameter settings and classification error, which unfortunately, can only be guessed at. In the evaluation I investigate the performance of Sequential ID3 over several parameter settings to give at least a preliminary idea of how these factors relate.

4.1 Methodology

Sequential ID3 is intended for megainduction tasks involving vast amounts of data. Unfortunately, the current implementation of the algorithm is restricted to two-class problems with categorical attributes, and I do not currently have access to large-sized problems of this type. Nevertheless, by a simple transformation of a smaller-sized data set, I can construct a reasonably realistic evaluation. The idea is to assume that a set of classified examples completely defines the example distribution. Given a set of n training examples, I assume that each example in the set occurs with probability $1/n$, and that each example not in the set occurs with zero probability. An arbitrarily large sample can then be generated according to this example distribution. Furthermore, as the example distribution is now exactly known, I can compute the exact classification error of a given decision tree.

One could criticize this method by noting that the learning algorithm can potentially memorize all of the original examples, allowing perfect accuracy when the original data is noise-free. However, this criticism is mitigated by the fact that the decision trees learned by Sequential ID3 are limited in size. I ensure that the learned decision trees have substantially fewer leaves than the number of original unique examples.

I test Sequential ID3 on nine learning problems. Eight are drawn from the Irvine repository, including the DNA promoter dataset, a two class version of the gene splicing dataset (made by collapsing EI and IE into a single class), the tic-tac-toe dataset, the three monks problems, a chess endgame dataset, and the soybean dataset.⁵ The ninth dataset is a second DNA promoter dataset provide by Hirsh and Noordewier [Hirsh94]. When the problems contain non-binary attributes, they are converted to binary attributes in the obvious manner. In all trials I set the level of decision error,

5. Available via anonymous FTP: <ftp://ics.uci.edu/pub/machine-learning-databases>

δ , to 10%. Both the stopping parameter, γ , and the indifference parameter, ϵ , are varied over a wide range of values. To insure statistical significance, I repeat all learning trials 50 times and report the average result. All of the tests are based on entropy as a selection criterion. Due to space limitations, I consider only the evaluations for the gene splicing dataset and the third monks dataset (monks-3) in detail here.

4.2 Classification Accuracy vs. Decision Quality

Sequential ID3 bases its decision quality on the indifference parameter ϵ and the stopping parameter γ . As ϵ shrinks, the learning algorithm is forced to obey more closely the entropy selection criterion. Assuming that entropy is a good criterion for selecting attributes, classification error should diminish as selection decisions follow more closely the true entropy of the attributes. As γ shrinks, concept descriptions can become more complex, thus allowing a more faithful model of the underlying concept and consequently, lower classification error.⁶ Additionally, an interaction may occur between these parameters: allowing a larger concept description may compensate for a poor selection criterion, as a bad initial split can be rectified lower in the decision-tree (provided the tree is large enough).

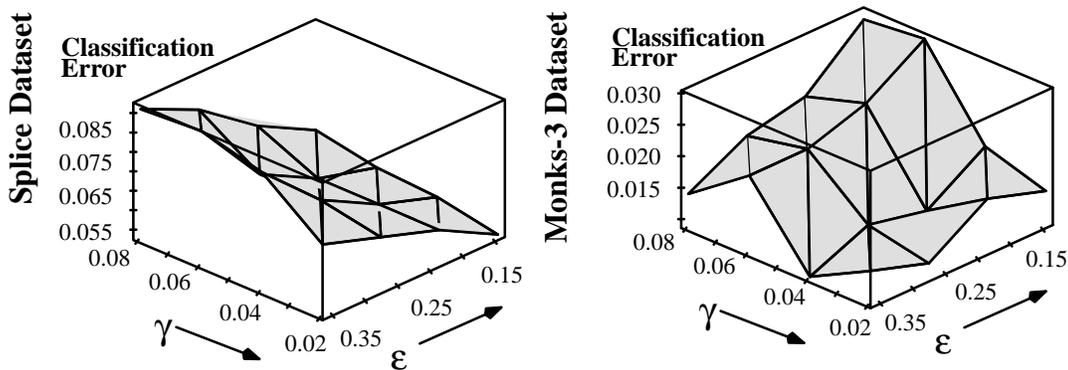


Figure 4. Classification error of Sequential ID3 as a function of γ and ϵ . Decision error is 10%.

Figure 4 summarizes the empirical results for the splicing and monks-3 problems. The results of the splicing evaluation is typical, and supports the claim that classification accuracy and decision quality are linked: classification error diminishes as decision quality increases. The chess, both promoter, and monks-2 datasets all show the same basic trend, lending further support to the claim. (The soybean dataset shows near zero error for all parameter settings) The results of the monks-3 and, to a lesser extent, the monks-1 evaluations raise a note of caution, however: classification error actually increases as the selection decisions become more informed. These later findings suggests that, at least for these two problem sets, entropy is a poor selection criterion. This is perhaps not surprising, as the monks problems are artificial problems designed to cause difficulties for top-down decision-tree algorithms. The tic-tac-toe dataset, interestingly, showed almost no change in classification error as a result of changes in ϵ .

⁶ Note that claim that smaller trees have lower classification error [Breiman84] is made for learning with a fixed amount of data, and thus does not apply to the sequential setting.

4.3 Sample Complexity

Sequential ID3 must draw sufficient data to satisfy the specified level of decision quality. The complex statistical machinery of sequential induction is justified to the extent that it requires less data than simpler one-shot inductive approaches. It is also interesting to consider just how much data is necessary to arrive at statistically sound inductive decisions while inducing decision trees.

In addition to the decision quality parameters, the size of a sufficient sample depends on the number of attributes associated with the induction problem. The splicing problem uses 480 binary attributes, whereas the monks-3 problem uses fifteen. By employing the Hoeffding procedure, I derive the one-shot sample sizes illustrated in Figure 5. Because it has more attributes, the splicing problem requires more data than the monks-3 problem.

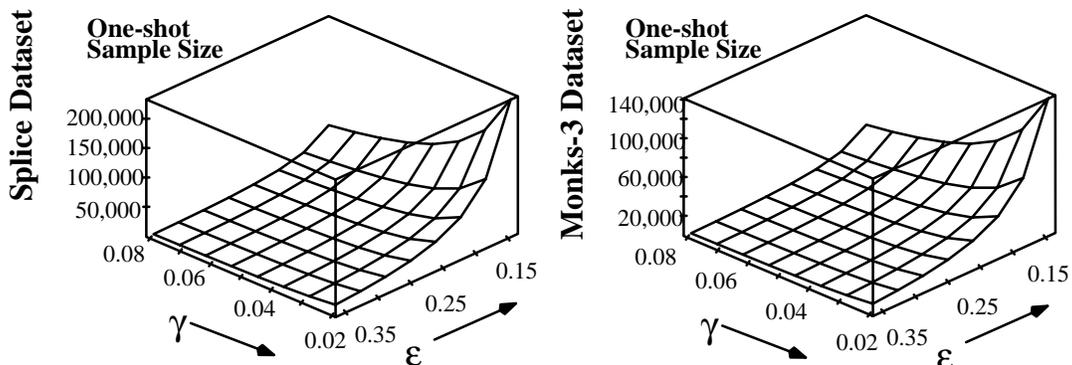


Figure 5. One-shot sufficient sample as a function of γ and ϵ . Decision error is 10%.

Figure 6 illustrates the sample sizes required for sequential induction. The benefit is dramatic for the monks-3 problem: sequential induction requires 1/32th the data needed by one-shot induction. In the case of the splicing data set, smaller but still significant improvement is observed: sequential ID3 used one third the data needed by the one-shot approach. A closer examination of the splicing dataset reveals that many of the selection decisions have several attributes tied for the best. In this situation, McSPRT has difficulty selecting a winner and is forced closer to its worst-case complexity.

Machine learning researchers may be surprised by the large sample sizes required for learning because standard algorithms can acquire comparably accurate decision trees with far less data. This can in part be explained by the fact that the concepts being learned are fairly simple: most of the concepts are deterministic with noise-free data. There is also the fact that “making good decisions” and “being sure one is making good decisions,” are not necessarily equivalent: the later requires more data and, when most inductive decisions lead to good results (as can be the case in simple concepts), “being sure” can be overly conservative. Nevertheless, in many learning applications one can make a strong case for conservatism, especially when the results of our algorithms inform important judgements, and when these judgements are made automatically, without human oversight.

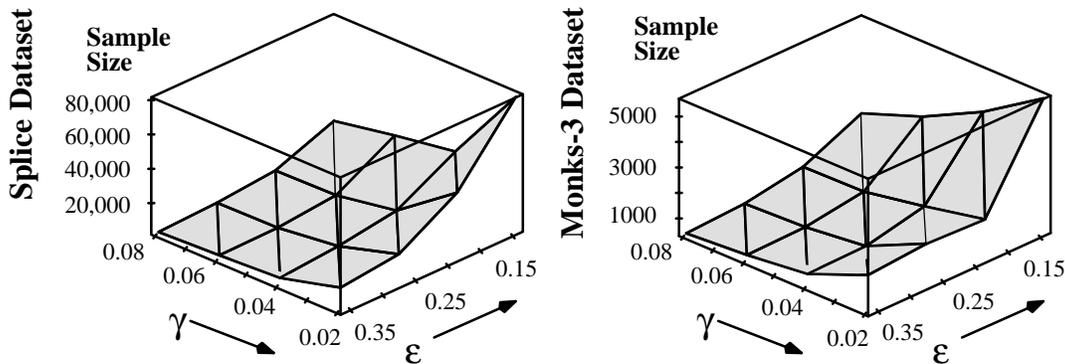


Figure 6. Average sample size of Sequential ID3 as a function of γ and ϵ . Decision error is 10%.

Table 1 summarizes the results of all datasets for two selected values of γ and ϵ (complete graphs can be found in [Gratch94a]). In all but one case, Sequential ID3 required significantly less data than one-shot learning. This advantage should become even more dramatic with smaller settings for the indifference and stopping parameters.

Table 1	$\gamma=0.08, \epsilon=0.36$				$\gamma=0.02, \epsilon=0.09$			
	Prediction Error	Sequential Sample Sz	One-shot Sample Sz	Tree Size	Prediction Error	Sequential Sample Sz	One-shot Sample Sz	Tree Size
Tictactoe	0.174	1032	1729	69	0.081	34,776	118,915	125
Monks-1	0.129	858	1966	46	0.006	12,255	138,319	80
Monks-2	0.317	1136	1966	76	0.171	62,150	138,319	135
Monks-3	0.014	412	1966	28	0.014	5,594	138,319	28
Kr-vs-kp	0.068	1140	2349	23	0.030	68,368	165,451	35
Soybean	0.000	40	2987	3	0.002	455	207,389	3
Promoter1	0.072	3122	3033	20	0.037	36,249	210,320	24
Promoter2	0.274	4554	3045	63	0.119	170,517	211,153	135
Splice	0.085	2592	3310	23	0.053	79,842	228,217	35

5. SUMMARY AND CONCLUSION

Sequential induction is a promising method for efficient learning in problems involving enormous quantities of data. The technique seems especially well-suited for tasks in which a premium must be placed on minimizing sample sizes while still providing statistical guarantees on learning quality. There are several limitations to Sequential ID3 and many areas of future research.

One limitation is that ensuring statistical rigor comes at a significant computational expense. Furthermore, the empirical results suggest that the current statistical model may be too conservative for many problems. For example, in many problems the concept is deterministic and the data noise-free. It is unclear how to incorporate such information into the statistical models. Additionally, the Bonferroni method tends to be an overly conservative method for bounding the overall error level.

There are some practical limits to what kinds of problems can be handled by the sequential model. Whereas one could easily extend the approach to multiple classes and non-binary attributes, it is less clear how to address continuous attributes. Another practical limitation is that although the approach generalizes to arbitrary selection criteria, round-off error in computing selection and variance estimates may be a significant problem for some selection functions. Round-off error contributes to excessive sample sizes on some of my evaluations of the orthogonality criterion. Probably the most significant limitation of Sequential ID3 (and of all standard inductive learning approaches) is the tenuous relationship between decision error and classification error. Improving decision quality can *reduce* classification accuracy due to the hillclimbing nature of decision-tree induction (this was clearly evident in the monks-3 evaluation). In fact, standard accuracy improving techniques exploit the randomness caused by insufficient sampling to break out of local maxima; by generating several trees and selecting one through cross-validation. An advantage of the sequential induction model, however, is that it clarifies the relationship between decision quality and classification accuracy, and suggests more principled methods for improving classification accuracy. For example, the generate-and-cross-validate approach mainly varies the inductive decisions at the leaves of learned trees (because the initial partitions are based on large samples and thus, are less likely to change), whereas it seems more important to vary inductive decisions closer to the root of the tree. A sequential approach could easily make initial inductive decisions more randomly than later ones. Furthermore, the sequential model allows the easy implementation of more complex search strategies, such as multi-step look-ahead. More importantly, the statistical framework enables one to determine easily how these strategies affect the expected sample time. For example, performing k -step look-ahead search requires on the order of k times as much data as a non-look-ahead strategy to maintain the same level of decision quality [Gratch94a]. Therefore, sequential induction is suitable not only as a megainduction approach, but also as an analytic tool for exploring and characterizing alternative methods for induction.

References

- [Berger80] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, Springer Verlag, 1980.
- [Bishop75] Y. M. M. Bishop, S. E. Fienberg and P. W. Holland, *Discrete Multivariate Analysis: Theory and Practice*, The MIT Press, Cambridge, MA, 1975.
- [Breiman84] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, CA, 1984.
- [Catlett91] J. Catlett, "Megainduction: a test flight," *Proceedings of the Eighth International Workshop on Machine Learning*, Evanston, IL, June 1991, pp. 596–599.
- [Chien95] S. Chien, J. Gratch and M. Burl, "On the Efficient Allocation of Resources for Hypothesis Evaluation: A Statistical Approach," *Institute of Electrical and Electronics Engineers IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear)*, 1995.
- [deMantaras92] R. L. deMantaras, "A Distance-Based Attribute Selection Measure for Decision Tree Induction," *Machine Learning* 6, (1992), pp. 81–92.

- [Etzioni93] O. Etzioni, N. Lesh and R. Segal, "Building Softbots for UNIX (Preliminary Reprot)," *Technical Report 93-09-01*, 1993.
- [Fayyad92] U. M. Fayyad and K. B. Irani, "The Attribute Selection Problem in Decision Tree Generation," *Proceedings of the National Conference on Artificial Intelligence*, San Jose, CA, July 1992, pp. 104-110.
- [Gratch94a] J. Gratch, "On-line Addendum to Sequential Inductive Learning," *Available via anonymous ftp to beethoven.cs.uiuc.edu/pub/gratch/sid3-ad.ps*, 1994.
- [Gratch94b] J. Gratch, "An Effective Method for Correlated Selection Problems," Technical Report UIUCDCS-R-94-1898, Urbana, IL, 1994.
- [Harris92] N. L. Harris, L. Hunter and D. J. States, "Mega-Classification: Discovering Motifs in Massive Datastreams," *Proceedings of the National Conference on Artificial Intelligence*, San Jose, CA, July 1992, pp. 837-842.
- [Haussler92] D. Haussler, "Decision Theoretic Generalizations of the PAC Model for Neural Net and Other Applications," *Information and Computation* 100, 1 (1992), pp. 78-150.
- [Hirsh94] H. Hirsh and M. Noordewier, "Using Background Knowledge to Improve Learning of DNA Sequences," *Proceedings of the Tenth Institute of Electrical and Electronics Engineers Conference on Artificial Intelligence for Applications*, Los Alamitos, CA, 1994, pp. 351-357.
- [Hochberg87] Y. Hochberg and A. C. Tamhane, *Multiple Comparison Procedures*, John Wiley and Sons, 1987.
- [Kearns92] M. J. Kearns, R. E. Schapire and L. M. Sellie, "Toward Efficient Agnostic Learning," *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, PA, JULY 1992, pp. 341-352.
- [Maron94] O. Maron and A. W. Moore, "Hoeffding Races: Accelerating Model Selection Search for Classification and Function Approximation," *Advances in Neural Information Processing Systems* 6, 1994.
- [Mingers] J. Mingers, "An Empirical Comparison of Pruning Methods for Decision-tree Induction," *Machine Learning* 3, pp. 319-342.
- [Moore94] A. W. Moore and M. S. Lee, "Efficient Algorithms for Minimizing Cross Validation Error," *Proceedings of the Tenth International Conference on Machine Learning*, New Brunswick, MA, July 1994.
- [Musick93] R. Musick, J. Catlett and S. Russell, "Decision Theoretic Subsampling for Induction on Large Databases," *Proceedings of the Ninth International Conference on Machine Learning*, Amherst, MA, June 1993, pp. 212-219.
- [Nelson95] B. L. Nelson and F. J. Matejcek, "Using Common Random Numbers for Indifference-Zone Selection and Multiple Comparisions in Simulation," *Management Science*, 1995.
- [Perkowitz94] M. Perkowitz and O. Etzioni, "Database Learning for Softward Agents," *Proceedings of the National Conference on Artificial Intelligence*, Seattle, WA, August 1994, pp. 1485.
- [Quinlan86] J. R. Quinlan, "Induction of decision trees," *Machine Learning* 1, 1 (1986), pp. 81-106.