Iterative Weakening: Optimal and Near-Optimal Policies for the Selection of Search Bias

Foster John Provost
Department of Computer Science
University of Pittsburgh
foster@cs.pitt.edu

Abstract

Decisions made in setting up and running search programs bias the searches that they perform. Search *bias* refers to the definition of a search space and the definition of the program that navigates the space. This paper addresses the problem of using knowledge regarding the complexity of various syntactic search biases to form a policy for selecting bias. In particular, this paper shows that a simple policy, *iterative weakening*, is optimal or nearly optimal in cases where the biases can be ordered by computational complexity and certain relationships hold between the complexity of the various biases. The results are obtained by viewing bias selection as a (higher-level) search problem. Iterative weakening evaluates the states in order of increasing complexity. An offshoot of this work is the formation of a near-optimal policy for selecting both breadth *and* depth bounds for depth-first search with very large (possibly unbounded) breadth and depth.

Introduction

For the purposes of this paper, search *bias* refers to the definition of a search space and the definition of the program that navigates the space (*cf.*, *inductive bias* in machine learning [Mitchell, 1980], [Utgoff, 1984], [Rendell, 1986], [Provost, 1992]). Bias choices are purely syntactic if they are not based on domain knowledge, otherwise they are semantic. In this work, except where I refer to the incorporation of *knowledge* into the search program (*e.g.*, the addition of heuristics), *bias* refers to syntactic bias choices. The choice of a depth-first search is a coarse-grained choice; the choice of a maximum depth of d is a finer-grained choice. *Search policy* refers to the strategy for making bias choices based on underlying assumptions and knowledge (*cf.*, *inductive policy* [Provost & Buchanan, 1992a]). This paper addresses the problem of selecting from among a set of bias choices, based solely on complexity knowledge. I show that in certain cases, optimal or near-optimal policies can be formed.

The problem is attacked by viewing bias selection as a (higher-level) state-space search problem, where the states are the various biases and the goal is to find a bias which is satisfactory with respect to the underlying search goal (*e.g.*, a search depth sufficient for finding the lower-level goal). For the purposes of the current exposition, let us assume that no knowledge is transferred across biases, *i.e.*, the search with one bias has no effect on the search with another bias. So, the higher-level problem is a search problem where the cost of evaluating the various states is not uniform, and we know (at least asymptotically) the complexity of the evaluation of each state. I will refer to the worst-case time complexity of searching with a given bias as the *complexity* of that bias. Using worst-case time complexity side-steps the problem that some problems may be inherently "easier" than others with a given bias, and allows biases to be ordered independently of the distribution of problems that the search program will encounter.

If we view the strength of a bias to be analogous to the complexity of that bias, we can define the policy *iterative weakening* to be: *evaluate the biases in order of increasing complexity.* (The term *iterative weakening* is borrowed from the *iterative deepening* of [Korf, 1985] and *iterative broadening* of [Ginsberg and Harvey, 1990], which are special cases of the general technique). In cases where the states (biases) can be grouped into equivalence classes based on complexity, where there is an exponential increase in complexity between classes, and where the rate of growth of the cardinality of the classes is not too great (relatively), iterative weakening can be shown to be a near-optimal policy with respect to the complexity of only evaluating the minimum complexity goal state (optimal in simple cases).

Consider an example from machine learning as search, where being able to select from among different search biases is particularly important. The complexity of a non-redundant search of a space of conjunctive concept descriptions with maximum length k is polynomial in the number of features and is exponential in k. Given a fixed set of features, iterative weakening would dictate searching with $k = 1$, $k = 2$, ..., until a satisfactory concept description is found.

However, the size of an ideal feature set might not be manageable. In many chemistry domains the properties and structure of chemicals provide a very large set of features for learning; for example, in the Meta-DENDRAL domain the task is to learn cleavage rules for chemical mass spectrometry [Buchanan & Mitchell, 1978]. In such domains with effectively infinite sets of features, knowledge may be used to order the features by potential relevance. However, it may not be known *a priori* how many of the most relevant features will be necessary for satisfactory learning.

Many existing learning programs represent concept descriptions as sets of rules, each rule being a conjunction of features (*e.g.*, [Quinlan, 1987], [Clark & Niblett, 1989], [Clearwater & Provost, 1990]). The space of conjunctive rules can be organized as a search tree rooted at the rule with no features in the antecedent, where each child is a specialization of its parent created by adding a single conjunct. A restriction on the depth of the search tree restricts the maximum complexity of the description language (the number of conjuncts in a rule's antecedent). A restriction on the breadth of the search restricts the list of features considered. A depth-first search of this space would not only face the classic problem of determining a satisfactory search depth (see Section 3), but also the problem of (simultaneously) determining a satisfactory search breadth. In Section 5 I develop a near-optimal policy for selecting both the depth and the breadth of a depth-first search.

Optimal Policies

The heuristic behind iterative weakening policies is by no means new. As mentioned above, and discussed further below, iterative deepening and iterative broadening are special cases of the general technique. Simon and Kadane [Simon and Kadane, 1975] show that in cases where knowledge is available regarding the cost of a search and the probability of the search being successful, that an "optimal" strategy is to perform the searches in order of increasing probability/cost ratio. In the case where the probability distribution is uniform (or is assumed to be because no probability information is available), this reduces to a cheapest-first strategy. Slagle [Slagle, 1964] also discusses what he calls *ratio-procedures*, where tasks are carried out in order of the ratio of benefit to cost, and shows that these "often serve as the basis of a minimum cost procedure" (p.258).

However, the problem addressed in this paper is a different one from that addressed by Simon and Kadane and Slagle. Their work showed that the cheapest-first strategy was a minimum cost strategy with respect to the other possible orderings of the biases. In this paper, the term *optimal* will be used to denote a policy where the asymptotic complexity is no worse than that of a policy that knows *a priori* the minimum cost bias that is sufficient for finding the (lower-level) goal. To illustrate, given n search procedures, p_1, p_2, ... , p_n, previous work addressed finding an ordering of the p_i's such that finding the goal will be no more expensive than any other ordering of the p_i's. In contrast, I address the problem of ordering the p_i's such that finding the goal will be as inexpensive (or almost as inexpensive) as only using p_{i_m}, the minimum-cost search procedure.

This paper shows that in some cases the cheapest-first strategy is almost as good (asymptotically) as a strategy that *knows* the right bias *a priori*. The implications are that in these cases, it is a better investment to apply knowledge to reduce the complexity of the underlying task (*e.g.*, by introducing heuristics based on the semantics of the domain) than to use it to aid in the selection of (syntactic) search bias (discussed more below).

A Single Dimensional Space

Let us assume the states of our (higher-level) search space can be indexed by their projection onto a single dimension, and that the projection gives us integer values. In a machine learning context this could be the case where the different biases are different types of hypothesis-space search, different degrees of complexity of the description language (*e.g.*, number of terms in the antecedent of a rule), different search depths, etc. From now on, let us refer to the states (biases) by their indices, *i.e.*, i denotes the state that gives value i when projected onto the dimension in question. Without loss of generality, let us assume that $i_1 \leq i_2$ implies that the complexity of evaluating i_1 is less than or equal to the complexity of evaluating i_2. Let $c(i)$ denote the complexity of evaluating i.

Iterative weakening is a rather simple policy in these cases. It specifies that the states should be evaluated in order of increasing i. It may seem that iterative weakening is a very wasteful policy, because a lot of work might be duplicated in evaluating all the states. However, if $c(i)$ is exponential in i, then the arguments of [Korf, 1985] apply. Korf shows that *iterative deepening*, iterative weakening along the search-depth dimension, is an optimal policy with respect to time, space, and cost of solution path. In short, since the cost of evaluating i increases exponentially, the complexity of iterative deepening differs from that of searching with the correct depth by only a constant factor. Thus "knowing" the right bias buys us nothing in the limit. This paper concentrates solely on time complexity.

Theorem: (after [Korf, 1985]) *Iterative weakening is an asymptotically optimal policy, with respect to time complexity, for searching a single-dimensional space where the cost of evaluating state i is $O(b^i)$.*

Iterative Broadening is a similar technique introduced in [Ginsberg and Harvey, 1990], where the dimension in question is the breadth of the search. In this case, the complexity increases only polynomially in i, however the technique is shown to still be useful in many cases (a characterization of when iterative broadening will lead to a computational speedup is given).

Theorem: (after [Ginsberg and Harvey, 1990]) *Iterative weakening is an asymptotically near-optimal policy, with respect to time complexity, for searching a single-dimensional space where the cost of evaluating state i is $O(i^d)$. (It is within a dth-root factor of optimal–see [Provost, 1993].)*

A similar technique is used in [Linial, *et al.*, 1988] for learning with an infinite VC dimension. If a concept class C can be decomposed into a sequence of subclasses $C = C_1 \cup C_2 \cup \ldots$ such that each C_i has VC dimension at most i, then iterative weakening along the VC dimension is shown to be a good strategy (given certain conditions).

Thus, previous work helps us to characterize the usefulness of iterative weakening along a single dimension. However, in specifying a policy for bias selection there may be more than one dimension along which the bias can be selected. The rest of this paper considers multi-dimensional spaces.

Multi-Dimensional Spaces

Consider the general problem of a search where the states have different costs of evaluation (in terms of complexity). We want to find a good policy for searching the space. Let each state be indexed according to its projection onto multiple dimensions, and let us refer to the state by its vector of indices $\vec{i}$ (assume, for the moment, that there is a one-to-one correspondence between states and indices). Let $c(\vec{i})$ be the complexity of evaluating state $\vec{i}$. Iterative weakening specifies that the states (biases) should be evaluated by increasing complexity. Let us consider some particular state complexity functions. (For clarity I will limit the remaining discussion to two dimensions, but mention results for n dimensions. A more detailed treatment can be found in [Provost, 1993]).

Dual Searches

Consider a particular $c(\vec{i})$: $c(i,j) = b^i + b^j$. This is the complexity function for the situation where two (depth-first) searches must be performed, and both subgoals must be discovered before the searcher is sure that either is actually correct.

How well will iterative weakening do on this problem? The following theorem shows that it is nearly optimal–within a log factor. For the rest of the paper, let $\vec{i_g} = (i_g, j_g)$ denote the minimum-complexity goal state, and let $b > 1$.

Proposition: *Given a search problem where the complexity of evaluating state (i,j) is $b^i + b^j$, any asymptotically optimal policy for searching the space must have worst-case time complexity $O(b^m)$, where $m = max(i_g, j_g)$ (the complexity of evaluating the minimum-complexity goal state).*

Theorem: *Given a search problem where the complexity of evaluating state (i,j) is $b^i + b^j$, iterative weakening gives a time complexity of $O(mb^m)$, where $m = max(i_g, j_g)$.*

Proof: In the worst case, iterative weakening evaluates all states $\vec{i}$ such that $c(\vec{i}) \leq c(\vec{i_g})$, where $\vec{i_g} = (i_g, j_g)$ is the (minimum-complexity) goal state. Thus the overall complexity of the policy is:

$$\sum_{\{\vec{i} \mid c(\vec{i}) \leq c(\vec{i_g})\}} c(\vec{i}) = \sum_{\{(i,j) \mid b^i + b^j \leq b^{i_g} + b^{j_g}\}} b^i + b^j.$$

The terms that make up the sum can be grouped into equivalence classes based on complexity. Let a term b^k be in class C_k. Then the overall complexity becomes:

$$\sum_{k=1}^{m} |C_k| b^k,$$

where $|C_k|$ denotes the cardinality of the set of equivalent terms. The question remains as to the number of such terms (complexity of b^k). The answer is the number of vectors (i,j) whose maximum element is k, plus the number of vectors (i,j) whose minimum element is k. The number of such vectors is $2m$, so the overall complexity is: $\sum_{k=1}^{m} 2mb^k$, which is: $O(mb^m)$.♠

Corollary: *Given a search problem where the complexity of evaluating state (i,j) is $b^i + b^j$, iterative weakening is within a log factor of optimal.*

Proof: The optimal complexity for this problem is $O(N) = O(b^m)$; iterative weakening has complexity $O(mb^m) = O(N \log N)$. ♠

For n dimensions, the proximity to being optimal is dependent on n. In general, for such searches iterative weakening is $O(m^{n-1}b^m) = O(N(\log N)^{n-1})$. (See [Provost, 1993].)

If we have more knowledge about the problem than just the complexity of evaluating the various states, we can sometimes come up with a better policy. In this case, the policy that immediately springs to mind is to let $i = j$ and search to depth $i = 1, 2, ...$ in each (lower-level) space. This is, in fact, an optimal policy; the amount of search performed is

$$\sum_{k=1}^{m} 2b^k = O(b^m).$$

We have, in effect, collapsed the problem onto a single dimension. The particular extra knowledge we use in specifying this optimal policy is that a solution found in state $\vec{i}_1$ will also be found in $\vec{i}_2$ if $\vec{i}_1$ is componentwise less than or equal to $\vec{i}_2$. (As is the case for a pair of depth-first searches.)

A Search within a Search

Let us consider a search problem where the complexity of evaluating state (i,j) is b^{i+j}. This complexity function is encountered when evaluating the state involves a search within a search. For example, consider a learning problem where there is a search for an appropriate model, with a search of the space of hypotheses for each model (*e.g.*, to evaluate the model). Iterative weakening is once again competitive with the optimal policy.

Proposition: *Given a search problem where the complexity of evaluating state (i,j) is b^{i+j}, any asymptotically optimal policy for searching the space must have worst-case time complexity $O(b^m)$, where $m = i_g + j_g$ (the complexity of evaluating the minimum-complexity goal state).*

Theorem: *Given a search problem where the complexity of evaluating state (i,j) is b^{i+j}, iterative weakening gives a time complexity of $O(mb^m)$, where $m = i_g + j_g$.*

Proof: Similar to previous proof. Note that in this case, the cardinality of the set of equivalent terms is equal to the number of vectors (i,j) whose components sum to k, which is $k-1$ (given positive components). Thus the overall complexity of the policy is: $\sum_{k=1}^{m}(k-1)b^k$, which is: $O(mb^m)$.♠

Corollary: *Given a search problem where the complexity of evaluating state (i,j) is b^{i+j}, iterative weakening is within a log factor of optimal.*

Proof: The optimal complexity for this problem is $O(N) = O(b^m)$; iterative weakening has complexity $O(mb^m) = O(N \log N)$. ♠

For n dimensions, the proximity to being optimal is dependent on n. In general, for such searches iterative weakening has complexity $O(m^{n-1}b^m) = O(N(\log N)^{n-1})$. (See [Provost, 1993].)

In this case, the policy of collapsing the space and iteratively weakening along the dimension $i = j$ does not produce an optimal policy. If we let $m = i_g + j_g$, in the worst case, as $m \to \infty$, the $i = j$ policy approaches b^m times worse than optimal.

Important: Relative Growth

As we have seen from the preceding examples, in general, the important quantity is the growth of the complexity function relative to the growth of the number of states exhibiting a given complexity. In the cases where there is but one state for each complexity (*e.g.*, iterative deepening, iterative broadening) we have seen that the faster the rate of growth of the complexity function, the closer to optimal. In multidimensional cases, as the dimensionality increases the policy becomes further from optimal because the number of states of a given complexity increases more rapidly.

Let us now consider a multidimensional problem with a (relatively) faster growing complexity function, namely $c(\vec{i}) = b^{ij}$. This is another function where the strategy of choosing $i = j$ and searching $i = 1, 2, ...$ is not optimal, even if we have the extra knowledge outlined above. If we let $m = i_g j_g$, in the worst case, as $m \to \infty$ the ratio of the overall complexity of the $i = j$ policy to the optimal approaches b^{m^2-m} (very much worse than optimal).

However iterative weakening does very well–even better than in the previous case. The following theorem shows that in this case, it is within a root-log factor of being an optimal policy.

Proposition: *Given a search problem where the complexity of evaluating state* (i,j) *is* b^{ij}, *any asymptotically optimal policy for searching the space must have worst-case time complexity* $O(b^m)$, *where* $m = i_g j_g$ *(the complexity of evaluating the minimum complexity goal state).*

Theorem: *Given a search problem where the complexity of evaluating state* (i,j) *is* b^{ij}, *iterative weakening gives a time complexity of* $O(\sqrt{m}b^m)$, *where* $m = i_g j_g$.

Proof: Similar to previous proofs. Note that in this case, the cardinality of the set of equivalent terms is equal to the number of factors of k, which is bounded by $\sqrt{k}$. Thus the overall complexity of the policy is: $\leq \sum_{k=1}^{m} \sqrt{k} b^k$, which is: $O(\sqrt{m}b^m)$.♠

Corollary: *Given a search problem where the complexity of evaluating state* (i,j) *is* b^{ij}, *iterative weakening is within a root-log factor of optimal.*

Proof: The optimal complexity for this problem is $O(N) = O(b^m)$; iterative weakening has complexity $O(\sqrt{m}b^m) = O(N\sqrt{\log N})$. ♠

For n dimensions, the proximity to being optimal is again dependent on n. In general, for such searches iterative weakening can be shown to have complexity $O(m^{\log n} b^m) = O(N(\log N)^{\log n})$. (See [Provost, 1993].) The above results suggest that this bound may not be tight.

The general problem can be illustrated with the following schema: Complexity(IW) =

$$\sum_{\{\vec{i}|c(\vec{i}) \leq c(\vec{i_g})\}} c(\vec{i}) \leq c(i_g) \sum_{i} 1$$

$$\leq c(i_g) \cdot |\{\vec{i}|c(\vec{i}) \leq c(\vec{i_g})\}|$$

which is the complexity of evaluating the goal state, multiplied by the number of states with equal or smaller complexity. This gives slightly looser upper bounds in some cases, but illustrates that there are two competing factors involved: the growth of the complexity and the growth of the number of states. As we have seen, in some cases domain knowledge can be used to reduce the number of states bringing a policy closer to optimal.

Knowledge Can Reduce No. of States: Combining Broadening and Deepening

For rule-space searches such as those defined for the chemical domains mentioned in Section 1, we want to select both a small, but sufficient set of features (search breadth) and a small, but sufficient rule complexity (search depth). Ginsberg and Harvey write, "An attractive feature of iterative broadening is that it can easily be combined with iterative deepening . . . any of (the) fixed depth searches can obviously be performed using iterative broadening instead of the simple depth-first search" ([Ginsberg and Harvey, 1990] p. 220). This is so when the breadth bound is known *a priori*. It will be effective if the breadth bound is small. When neither exact breadth or depth is known *a priori*, and the maxima are very large (or infinite), we are left with the problem of designing a good policy for searching the (high-level) space of combinations of b, the breadth of a given search, and d the depth of a given search.

The complexity of evaluating a state in this space is $O(b^d)$. Strict iterative weakening would specify that we order the states by this complexity, and search all states such that $b^d \leq {b_g}^{d_g}$ (the goal state). We begin to see two things: (i) the analysis is not going to be as neat as in the previous problems, and (ii) as d grows, there will be a *lot* of different values of b to search. The second point makes us question whether the policy is going to be close to optimal; the first makes us want to transform the problem a bit anyway.

In this problem we can use the knowledge that a state $\vec{i}$ is a goal state if $\vec{i}$ is componentwise greater than or equal to $\vec{i_g}$. Since b^d can be written as $2^{d\log(b)}$, our intuition tells us that it might be a good idea to increment b exponentially (in powers of 2). We can then rescale our axes for easier analysis. Let $\hat{b} = \log(b)$, and consider integer values of $\hat{b}$. We now have the problem of searching a space where the complexity of searching state (i,j) is b^{ij}. We know that iterative weakening is a near-optimal policy for such a space. Unfortunately, the overshoot along the b dimension gets us into trouble. Given that the complexity of evaluating the (minimum-complexity) goal state is $O(2^{d_g \log(b_g)})$, the first "sufficient" state reached using our transformed dimensions would be $(\hat{b}_g, d_g)$, where $\hat{b}_g = \lceil \log(b_g) \rceil$. The difference in complexity between evaluating the minimum-complexity goal and the new goal is the difference between $O(2^{d\lceil \log(b) \rceil})$ and $O(2^{d\log(b)})$, which in the worst case approaches a factor of 2^d.

The solution to this problem is to decrease the step size of the increase of $d\hat{b}$. A satisfactory step size is found by collapsing the space onto the (single) dimension $k = d\log(b)$, and only considering integer values of k. Because we are now looking at stepping up a complexity of $O(2^{\lceil d\log(b)\rceil})$ (rather than $O(2^{d\lceil\log(b)\rceil})$), the overshoot of the minimum-complexity goal state is never more than a factor of 2, which does not affect the asymptotic complexity. Using iterative weakening along this new axis brings us to within a log factor of optimal.

Proposition: *Given a depth-first search problem where the complexity of evaluating state (b, d) is b^d (for possibly unbounded b and d), any asymptotically optimal policy for searching the space must have worst-case time complexity $O(2^m)$, where $m = d_g \log(b_g)$ (the complexity of evaluating the minimum-complexity goal state).*

Theorem: *Given a depth-first search problem where the complexity of evaluating state (b, d) is b^d, iterative weakening in integer steps along the dimension $k = d\log(b)$ gives a time complexity of $O(\hat{m}2^{\hat{m}})$, where $\hat{m} = \lceil d_g \log(b_g)\rceil$.*

Proof: In the worst case, iterative weakening evaluates all states k such that k is an integer and $c(k) \leq c(\lceil k_g\rceil)$, where $c(k)$ is the complexity function along the k axis, and k_g is the (minimum-complexity) goal state. Thus the overall complexity of the policy is:

$$\sum_{\{k|c(k)\leq c(\lceil k_g\rceil), k \text{ is an integer}\}} c(k)$$

$$= \sum_{\{k|2^k\leq 2^{\hat{m}}, k \text{ is an integer}\}} 2^k.$$

The states that make up the sum can be grouped into equivalence classes based on complexity. Let a state (b, d) be in class C_k iff $d\log(b) = k$ (for integer k). Then the overall complexity becomes: $\sum_{k=1}^{\hat{m}} |C_k|2^k$, where $|C_k|$ denotes the cardinality of the set of equivalent states. The question remains as to the number of states with complexity of 2^k. The answer is the number of vectors (b, d) where $d\log(b) = k$ (for integer k). Since d is an integer, the number of such vectors is at most k, so the overall complexity is: $\leq \sum_{k=1}^{\hat{m}} k2^k$, which is: $O(\hat{m}2^{\hat{m}})$.♠

Corollary: *Given a depth-first search problem where the complexity of evaluating state (b, d) is b^d, iterative weakening in integer steps along the dimension $k = d\log(b)$ is within a log factor of optimal.*

Proof: The optimal complexity for this problem is $O(N) = O(2^m)$ where $m = d_g \log(b_g)$; iterative weakening has a complexity of $O(\hat{m}2^{\hat{m}})$. Since $\hat{m} = \lceil m\rceil$, $\hat{m} \leq m + 1$. So iterative weakening has a complexity of $O((m + 1)2^{(m+1)}) = O(m2^m) = O(N \log N)$. ♠

When IW is not a Good Policy

Several problem characteristics rule out iterative weakening as a near-optimal policy. The smaller the relative growth of the complexity of the states (wrt. the growth of the number of states with a given complexity), the farther from optimal the policy becomes. For example, in one dimension, if $c(i) = i$ then the optimal policy is $O(i)$ whereas iterative weakening is $O(i^2)$ even when there is only one state per equivalence class. On the other hand, the rate of growth may be large, but so too might the size of the class of states with the same complexity. In the previous sections, we saw equivalence classes of states with cardinalities whose growth was small compared to the growth of the class complexities. If, instead of counting the number of factors of k or the number of pairs that sum to k, we had an exponential or combinatorial growth in the size of the classes, iterative weakening would fail to come close to optimal. (The b^d problem was one where the number of terms grew rapidly.)

One reason for a very large growth in the size of the equivalence classes is a choice of dimensions where there is a many-to-one mapping from states into state vectors. Thus, in the chemical domains, for iterative weakening to be applicable it is essential to be able to order the terms based on prior relevance knowledge. The ordering allows a policy to choose the first b terms, instead of all possible subsets of b terms.

Conclusions

The simple policy of iterative weakening is an asymptotically optimal or near-optimal policy for searching a space where the states can be ordered by evaluation complexity, and they can be grouped into equivalence classes based on complexity, where the growth rate of the complexities is large and the growth rate of the size of the classes is small (relatively).

This has important implications with respect to the study of bias selection. If the bias selection problem that one encounters fits the criteria outlined above, it may not be profitable spending time working out a complicated scheme (*e.g.*, using more domain knowledge to guide bias selection intelligently). The time would be better spent trying to reduce the complexity of the underlying biases (*e.g.*, using more domain knowledge for lower-level search guidance). On the other hand, if the complexity of the biases is such that iterative weakening can not come close to the optimal policy, it might well be profitable to spend time building a policy for more intelligent navigation of the bias space. For example, domain knowledge learned searching with one bias can be used to restrict further the search with the next bias (see [Provost & Buchanan, 1992b]).

This paper assumed that the problem was to choose from a fixed set of biases. Another approach would be to try to find a bias, not in the initial set, that better solves the problem. By reducing the complexity of the underlying biases, as mentioned above, one is creating new (perhaps semantically based) biases with which to search. Even if iterative weakening *is* an optimal policy for selecting from among the given set of biases, a better bias might exist that is missing from the set. (As a boundary case of a semantically based bias, consider this: once you know the answer, it may be easy to prune away most or all of the search space.)

The dual search problem and combining deepening and broadening are examples of when additional knowledge of relationships between the biases can be used to come up with policies closer to optimal than strict iterative weakening. In these cases, knowledge about the subsumption of one bias by another is used to collapse the bias space onto a single dimension. In the former case, iterative weakening along the single dimension was then an optimal policy. In the breadth and depth selection problem, the knowledge about the subsumption of biases is sufficient to give a near-optimal policy. Utilizing more knowledge at the bias-selection level will not help very much unless the complexity of the underlying biases is reduced first.

Acknowledgments

I would like to thank Bruce Buchanan, Haym Hirsh, Kurt Van Lehn, Bob Daley, Rich Korf, Jim Rosenblum, and the paper's anonymous reviewers for helpful comments. This work was supported in part by an IBM graduate fellowship and funds from the W.M. Keck Foundation.

References

Buchanan, B., and T. Mitchell, 1978. Model-directed Learning of Production Rules. In Waterman and Hayes-Roth (eds.), *Pattern Directed Inference Systems*, 297-312. Academic Press.

Clark, P., and T. Niblett, 1989. The CN2 Induction Algorithm. *Machine Learning*, 3: 261-283.

Clearwater, S., and F. Provost, 1990. RL4: A Tool for Knowledge-Based Induction. In *Proceedings of the 2nd Int. IEEE Conf. on Tools for Artificial Intelligence*, 24-30. IEEE C.S. Press.

Ginsberg, M., and W. Harvey, 1990. Iterative Broadening. In *Proc of the Eighth National Conf on Artificial Intelligence*, 216-220. AAAI Press.

Korf, R., 1985. Depth-First Iterative Deepening: An Optimal Admissible Tree Search. *Artificial Intelligence* 27: 97-109.

Linial, N, Y. Mansour, R. Rivest, 1988. Results on learnability and the VC dimension. In *Proc of the 1988 Wkshp on Comp Learning Theory*, 51-60. Morgan Kaufmann.

Mitchell, T., 1980. The Need for Biases in Learning Generalizations (Tech Rept CBM-TR-117). Dept of Comp Sci, Rutgers University.

Provost, F., 1992. Policies for the Selection of Bias for Inductive Machine Learning. Ph.D. Thesis. Dept of Comp Sci, Univ of Pittsburgh.

Provost, F., 1993. Iterative Weakening: Optimal and Near-Optimal Policies for the Selection of Search Bias. Tech Rept ISL-93-2, Intelligent Systems Lab, Dept of Comp Sci, Univ of Pittsburgh.

Provost, F., and B. Buchanan, 1992a. Inductive Policy. In *Proc. of the Tenth National Conf on Artificial Intelligence*, 255-261. AAAI Press.

Provost, F., and B. Buchanan, 1992b. Inductive Strengthening: the effects of a simple heuristic for restricting hypothesis space search. In K. Jantke (Ed.), *Analogical and Inductive Inference (Lecture Notes in Artificial Intelligence* 642: 294-304. Berlin: Springer-Verlag.

Quinlan, J., 1987. Generating Production Rules from Decision Trees. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, 304-307. Morgan Kaufmann.

Rendell, L., 1986. A General Framework for Induction and a Study of Selective Induction. *Machine Learning* 1: 177-226. Boston, MA: Kluwer.

Simon, H., and J. Kadane, 1975. Optimal Problem-Solving Search: All-or-None Solutions. *Artificial Intelligence* 6: 235-247. North-Holland.

Slagle, J., 1964. An Efficient Algorithm for Finding Certain Minimum-Cost Procedures for Making Binary Decisions. *Journal of the ACM* 11 (3), 253-264.

Utgoff, P., 1984. Shift of Bias for Inductive Concept Learning. Ph.D. thesis, Rutgers University.