# Quantifying the inductive bias in concept learning (extended abstract)

David Haussler

Department of Mathematics and Computer Science,
University of Denver, Denver, Colorado 80208.

**Abstract**

We show that the notion of bias in inductive concept learning can be quantified in a way that directly relates to learning performance, and that this quantitative theory of bias can provide guidance in the design of effective learning algorithms. We apply this idea by measuring some common language biases, including restriction to conjunctive concepts and conjunctive concepts with internal disjunction, and, guided by these measurements, develop learning algorithms for these classes of concepts that have provably good convergence properties.

**Introduction**

The theme of this paper is that the notion of bias in inductive concept learning *[U86] [R86]* can be quantified in a way that enables us to prove meaningful convergence properties for learning algorithms. We measure bias with a combinatorial parameter defined on classes of concepts known as the Vapnik-Chervonenkis dimension (or simply *dimension* ) *[VC71], [P78], [BEHW86]*. The lower the dimension of the class of concepts considered by the learning algorithm, the stronger the bias. In *[BEHW86]*, this parameter has been shown to be strongly correlated with learning performance, as defined in the learning performance model introduced by Valiant *[V84], [V85]*. This model can be outlined as follows.

A *concept* is defined by its set of instances in some instance space. A *sample* of a concept is sequence of observations, each of which is an instance of the concept ( *positive observation* ) or a non-instance of the concept ( *negative observation* ). Samples are assumed to be created from independent, random observations, chosen according to some fixed probability distribution on the instance space. Given a sample of a target concept to be learned, a learning algorithm forms a *hypothesis*, which is itself a concept. The algorithm is *consistent* if its hypothesis is always consistent with the given sample, i.e. includes all observed positive instances and no observed negative instances. A consistent hypothesis may still disagree with the target concept by failing to include unobserved instances of the target concept or including unobserved non-instances of the target concept. The *error* of a hypothesis is the combined probability of such instances, i.e. the probability that the hypothesis will disagree with a random observation of the target concept, selected from the instance space according to the fixed probability distribution.

Two performance measures are applied to learning algorithms in this setting.
1. The *convergence rate* of the learning algorithm is measured in terms of the sample size that is required for the algorithm to produce, with high probability, a hypothesis that has a small error. The qualification "with high probability" is required because the creation of the sample is a probabilistic event. Even the best learning algorithm cannot succeed in the unlikely event that the sample is not indicative of typical observations. However, while the model is probabilistic, no specific assumptions are made about the probability distribution that governs the observations. This distinguishes this approach from usual statistical methods employed in pattern recognition, where the object of learning is usually reduced to the estimation of certain parameters of a classical distribution. The distribution-free formulation of convergence rate is obtained by upper bounding the worst case convergence rate of the learning algorithm over all probability distributions on the instance space. This provides an extremely robust performance guarantee.
2. The *computational efficiency* of the learning algorithm is measured in terms of the (worst case) computation time required to pass from a sample of a given size to a hypothesis. Our results for conjunctive concepts indicate the possibility of a trade-off between convergence rate and computational efficiency, in which the fastest converging learning methods require significantly more computation time than their slower converging counterparts. In order to optimize this trade-off, applying the general method developed in *[BEHW86]*, we employ heuristic techniques based on the greedy method for finding a small set cover *[N69] [J74]* that trade off a small decrease in the convergence rate for a very large increase in computational efficiency. This general idea forms a secondary theme of the paper.

## 1. Quantifying inductive bias

In the simplest type of inductive concept learning, each instance of a concept is defined by the values of a fixed set of *attributes*, not all of which are necessarily relevant. For example, an instance of the concept "red triangle" might be characterized by the fact that its color is red, its shape is triangular and its size is 5. Following *[MCL83]*, we consider three types of attributes. A *nominal* attribute is one that takes on a finite, unordered set of mutually exclusive values, e.g. the attribute *color*, restricted to the six primary and secondary colors. A *linear* attribute is one with a linearly ordered set of mutually exclusive values, e.g. a real-valued or integer-valued attribute. A *tree−structured* attribute is one with a finite set of hierarchically ordered values, e.g. the attribute *shape* with values *triangle*, *square*, *hexagon*, *circle*, *polygon* and *any_shape*, arranged in the usual "is_a" hierarchy. Only the leaf values *triangle*, *square*, *hexagon* and *circle* are directly observed. Since a nominal attribute can be converted to a tree-structured attribute by addition of the special value *any_value*, we will restrict our discussion to tree-structured and linear attributes.

Equations relating attributes to values will be called *terms*, which are either *elementary* or *compound*. The possible forms of elementary terms are as follows.

For tree-structured attributes: $attribute = value$, e.g. $color = red$, $shape = polygon$.

For linear attributes: $value_1 \leq attribute \leq value_2$ e.g. $5 \leq size \leq 12$. Strict inequalities are also permitted, as well as intervals open on one side. Terms such as $5 \leq size \leq 5$ are abbreviated as $size = 5$.

Compound terms *[MIC83]* can take the following forms. For tree-structured attributes:

$attribute = value_1 \text{ or } value_2 \text{ or } \cdots \text{ or } value_k$

e.g. $shape = square \text{ or } circle$, and for linear attributes: any disjunction of intervals e.g. $0 \leq age \leq 21 \text{ or } age \geq 65$. Disjunctive operators within compound terms are called *internal disjunctions*.

We consider the following types of concepts:
1. pure conjunctive: $term_1 \text{ and } term_2 \text{ and } \cdots \text{ and } term_k$, where each $term_i$ is an elementary term, e.g. $color = red \text{ and } 5 \leq size \leq 12$,

2. pure disjunctive: same as pure conjunctive but terms are connected by "or"
3. internal disjunctive: same as pure conjunctive, but allowing compound terms, e.g.

$$(color = red \text{ or } blue \text{ or } yellow) \text{ and } (5 \leq size \leq 12)$$

These concept types have the following interpretations in the context of rule based knowledge representations.
Pure conjunctive: antecedent of a single, variable-free Horn clause rule (PROLOG rule), e.g.

$$type = pos \leftarrow color = red \text{ and } 5 \leq size \leq 12$$

Pure disjunctive: antecedents of several rules, each with a single term and all with a common consequent.
Internal disjunctive: antecedent of a single rule with pure disjunctive "helper rules" for the compound terms, e.g. for the internal disjunctive concept given above, create a new value "primary" for color and form the rules

$color = primary \leftarrow color = red$
$color = primary \leftarrow color = blue$
$color = primary \leftarrow color = yellow$
$type = pos \leftarrow color = primary \text{ and } 5 \leq size \leq 12$

In Section 2 we will see how collections of rules for internal disjunctive concepts can be generated mechanically from samples. But first, we describe how these and other learning algorithms can be evaluated.

To quantify the inductive bias of a learning algorithm, we use the following notion from [VC71]. Let $X$ be an instance space and let $H$ be a class of concepts defined on $X$, e.g. the class of pure conjunctive concepts over an instance space determined by a fixed set of attributes. For any finite set $S \subseteq X$ of instances, $\Pi_H(S) = \{S \cap h : h \in H\}$, i.e. the set of all subsets of $S$ that can be obtained by intersecting $S$ with a concept in $H$, or equivalently, the set of all ways the instances of $S$ can be divided into positive and negative instances so as to be consistent with some concept in $H$. If $\Pi_H(S)$ is the set of all subsets of $S$ then we say that $S$ is *shattered* by $H$. The *Vapnik-Chervonenkis dimension* of $H$ (or simply the *dimension* of $H$) is the smallest integer $d$ such that no $S \subseteq X$ of cardinality $d+1$ is shattered by $H$. If no such $d$ exists, the dimension of $H$ is infinite.

As an example, suppose $X$ is the instance space defined by one linearly ordered attribute *size* and $H$ is the set of pure conjunctive concepts over $X$. Thus $H$ is just the set of elementary terms involving *size*, i.e. size intervals. For any three distinct instances, i.e. instances where $size = x$, $size = y$ and $size = z$, with $x < y < z$, there is no concept in $H$ for which the first and third instances are positive but the second instance is negative because there is no interval that contains $x$ and $z$ without containing $y$. Hence no set of three instances in $X$ can be shattered by $H$, implying that the dimension of $H$ is at most 2. Since any two out of three distinct instances can be shattered by $H$, this upper bound is tight, at least when *size* has three or more distinct values.

Upper bounds on the dimensions of the more general concept classes introduced above are as follows:
$k$ term pure conjunctive concepts on $n$ attributes, each tree-structured or linear:
(1) $\quad d \leq 4k\log(4k\sqrt{n})$.
For $k$ of size roughly $n/2$ or larger,
(1') $\quad d \leq 2n$
is a better upper bound.
$k$ term pure disjunctive concepts on $n$ attributes:
(2) $\quad d \leq 4k\log(16n)(\log(2k) + \log\log(16n))$.
$k$ term internal disjunctive concepts on $n$ attributes, using a total of $j$ internal disjunctions:
(3) $\quad d \leq 5(k+j)\log(5(k+j)\sqrt{n})$.
Justifications for these bounds are omitted due to lack of space[1].

Let $C$ be a class of target concepts of some type and level of complexity, e.g. $p$-term pure conjunctive concepts over an instance space defined by $n$-attributes. Given a target concept in $C$ and some number $m$ of observations of this concept, a learning algorithm will explore some space of possible hypotheses. This will be called the *effective hypothesis space* of the learning algorithm for target concepts in $C$ and sample size $m$. The *numerical bias* of the learning algorithm is defined as the Vapnik-Chervonenkis dimension of its effective hypothesis space. A lower bias is a stronger bias. For example, in the next section we will present an algorithm (Algorithm 2) for learning pure conjunctive concepts that has the following property: presented with $m$ observations of an unknown $p$-term pure conjunctive target concept over an instance space of $n$ attributes, it always produces a consistent pure conjunctive hypothesis with at most $p\ln m + 1$ terms. Hence the effective hypothesis space of the algorithm for target concepts of this type with sample size $m$ is the class of at most $p\ln m + 1$-term pure conjunctive concepts over an instance space of $n$ attributes. These limitations on the hypothesis space are due to the fact that the algorithm only considers pure conjunctive hypotheses and prefers concepts with fewer terms, two of the informal types of bias identified in [U86]. Using formula (1) with $k = p\ln m$, we can approximately upper bound the numerical bias of the algorithm for $p$-term pure conjunctive target concepts over an instance space of $n$ attributes and sample size $m$ by
(5) $\quad 4p\ln m\log(4p\ln m\sqrt{n})$.
We can now use the following theorem to relate this numerical bias with the convergence rate of the algorithm for these target concepts.

**Theorem 1.** [BEHW86][2] Given any consistent learning algorithm with numerical bias for target concepts in a class $C$ and sample size $m$ of at most $rm^{\alpha}$, where $r \geq 2$ and $0 \leq \alpha < 1$, then for any probability distribution on the instance space, any target concept in $C$ and any $\epsilon$ and $\delta$ between 0 and 1, given a random sample of the target concept of size at least

$$(6) \qquad \max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \left(\frac{8r}{\epsilon(1-\alpha)}\log\frac{8r}{\epsilon(1-\alpha)}\right)^{\frac{1}{1-\alpha}}\right)$$

the algorithm produces, with probability at least $1-\delta$, a hypothesis with error at most $\epsilon$. If the numerical bias is bounded by $r(\log m)^l$, it suffices to have sample size

$$(7) \qquad \max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{2^{l+4}r}{\epsilon}\left(\log\frac{8(2l+2)^{l+1}r}{\epsilon}\right)^{l+1}\right) \quad \square$$

Plugging in formula (5) above into (7) but ignoring the $\log(\ln(m))$ term (i.e. letting $l = 1$ and $r = 4p\log(4p\sqrt{n})$), this theorem shows that given a $p$-term pure conjunctive target concept over $n$ attributes and approximately

$$(8) \ \max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \left(\frac{128p\log(4p\sqrt{n})}{\epsilon}\left(\log\frac{512p\log(4p\sqrt{n})}{\epsilon}\right)^2\right)\right)$$

random observations, Algorithm 2 produces, with probability at least $1-\delta$, a hypothesis with error at most $\epsilon$, independent of the target concept and *independent of the underlying distribution governing the generation of observations*. By a different argument, using bound (1') and (6) with $\alpha = 0$, we can also obtain the upper bound

$$(9) \qquad \max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{16n}{\epsilon}\log\frac{16n}{\epsilon}\right)$$

on the required number of observations, which is considerably better for small $n$.

Both Theorem 1 and related results in [VC71] and [P78] are actually fairly crude upper bounds, especially with regard to the constants, and perhaps some of the log terms as well. Thus it is the asymptotic behavior of the formulas derived from them that is of primary significance. For the special case of Algorithm 2 applied to $p$-term pure conjunctive target concepts over $n$ attributes, what is especially significant about

[1] [VC71], [WD81], [A83] and [BEHW86] give a variety of other examples of concept classes of finite dimension. When H is of finite dimension, Wenocur and Dudley call H a Vapnik-Chervonenkis Class (VCC) [WD81]. The Vapnik-Chervonenkis number of this class, denoted V(H), corresponds to the dimension of H plus one.

[2] This is derived from Theorem 11 of [BEHW86]. We are suppressing some additional measurability assumptions required in the general form of the theorem since they will not be relevant in our intended applications (see appendix of [BEHW86]).

formulas (8) and (9) is that the convergence rate does not depend at all on the size or complexity of the trees that define the values of the tree-structured attributes, nor on the number of values of the linearly ordered attributes. It also shows that the convergence rate depends only logarithmically on the number of attributes and the confidence factor $\delta$. The strongest dependence is on the inverse error $\frac{1}{\epsilon}$ and the number $p$ of terms in the target concept, yet neither of these is much worse than linear.

In fact, the argument used in proving Theorem 1 shows the following stronger result: given any $p$-term pure conjunctive target concept over $n$ attributes and a sample of approximately the size given in (8) or (9), with probability at least $1 - \delta$ *any* consistent hypothesis within the effective hypothesis space of Algorithm 2, i.e. any consistent conjunct with at most $p \ln m + 1$ terms, will have error at most $\epsilon$, independent of the underlying probability distribution that governs the observations. Thus no matter what our method is, if we happen to find a conjunct with at most $p \ln m + 1$ terms consistent with a sample of this size, then we can use this conjunct as our hypothesis and have confidence at least $1 - \delta$ that its error is at most $\epsilon$. This kind of *a posteriori* justification of a hypothesis is what lead Pearl to call the related results of *[V71]* a "Bernoulli theorem for the hindsighted scientist" *[P78]*. From another view, this result shows that with probability at least $1 - \delta$, the version space (see *[MIT82]*) of a random sample of size (8) or (9) of any $p$-term pure conjunctive target concept includes only concepts that are nearly equivalent to it, in the sense that their error is at most $\epsilon$, assuming that the version space is restricted to $p \ln m + 1$ term pure conjunctive concepts. Thus the rapid convergence rate of a learning algorithm with a strong bias can be directly related to the high probability of the rapid shrinking of the version space it works in as the size of the sample grows.

## 2. Application: learning concepts with internal disjunction.

We now illustrate the application of the analytical method outlined above in the stepwise development and analysis of learning algorithms for pure conjunctive, pure disjunctive and finally internal disjunctive concepts.

We will use the single representation trick, as described in *[C82]*: each observation is encoded as a rule, e.g. a positive observation of a red triangle of size 5 becomes: *type = pos ← color = red and size = 5 and shape = triangle.* Let $S$ be a sample encoded in this form and $A$ be an attribute. If $A$ is a tree-structured attribute, for each term $A = v$ that occurs in the sample $S$, mark the leaf of the tree for $A$ that represents the value $v$ with the number of positive observations and the number of negative observations that include the term $A = v$. If $A$ is a linear attribute, build a list of such pairs of numbers, ordered by the values $v$. This data structure will be called the *projection* of the sample onto the attribute $A$.

Given the projection of $S$ onto $A$, we can find the most specific term of the form $A = v$ that implies all of the positive observations, which we call the *minimal dominating term for* $A$[3]. If $A$ is a tree-structured attribute, the minimal dominating term is $A = v$, where $v$ is the value of the node that is the least common ancestor of all the leaves of the tree of $A$ whose values occur in at least one positive observation. The minimal dominating term is found using the climbing tree heuristic of *[MCL83]*. It corresponds to the "lower mark" in the attribute trees of *[BSP85]*. If $A$ is a linear attribute, the minimal dominating term is the term $v_1 \leq A \leq v_2$, where $v_1$ and $v_2$ are the smallest and largest values of $A$ that occur in at least one positive observation, i.e. the result of applying the "closing interval rule" of *[MCL83]*. We can use the minimal dominating terms to find the most specific pure conjunctive concept consistent with a given sample.

*Algorithm 1. (naive algorithm for learning conjunctive concepts)*

1. For each attribute, calculate the projection of the sample onto this attribute and find the minimal dominating term. Let the conjunction of these minimal dominating terms be the expression E.
2. If no negative examples are implied by E then return E,
else report that the sample is not consistent with any pure conjunctive concept.

The effective hypothesis space of this algorithm is the class of all pure conjunctive concepts over some fixed set of attributes and doesn't depend on the sample size or the number of terms in the target concept. Since the dimension of pure conjunctive concepts on $n$ attributes is at most $2n$ by formula (1') above, the convergence rate of this algorithm is given by formula (9) above, i.e. given a random sample of size (9), Algorithm 1 produces, with probability at least $1 - \delta$, a hypothesis with error at most $\epsilon$ for any pure conjunctive target concept and any distribution on the instance space.

While significant in its generality, this upper bound suffers from the fact that the number of observations required grows at least linearly in the number of attributes. In many AI learning situations where conjunctive concepts are used, the task is to learn relatively simple conjuncts from samples over instance spaces with many attributes. In this case a better algorithm would be to find the simplest conjunct (i.e. the conjunct with the least number of terms) that is consistent with the data, rather than the most specific conjunct. With this strategy, given a sample of any $p$-term pure conjunctive concept on $n$ attributes, we always find a consistent pure conjunctive hypothesis that has at most $p$ terms. Thus by the same analysis (i.e. using (6) with $\alpha = 0$) and using formula (1) instead of (1') (with k = p), the upper bound on the sample size required for convergence is reduced to

$$(10) \qquad \max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{32p\log(4p\sqrt{n})}{\epsilon}\log\frac{32p\log(4p\sqrt{n})}{\epsilon}\right)$$

which is logarithmic in the number of attributes. Call this the optimal algorithm. Can it be efficiently implemented? The following shows that it probably cannot.

*Theorem 2.* Given a sample on $n$ attributes, it is NP-hard to find a consistent pure conjunctive concept for this sample with the minimum number of terms.

In proving this theorem, we show that this problem is equivalent to following NP-hard problem *[GJ79]*:

*Minimum Set Cover:* given a collection of sets with union $T$, find a subcollection whose union is $T$ that has the minimum number of sets.

There is, however, an obvious heuristic for approximating the minimum cover of $T$: First choose a largest set. Then remove the elements of this set from $T$ and choose another set that includes the maximum number of the remaining elements, continuing in this manner until $T$ is exhausted. This is called the *greedy method.* Applying it to the problem of finding pure conjunctive concepts, we get the following.

*Algorithm 2. (greedy algorithm for learning pure conjunctive concepts)*

1. For each attribute, calculate the projection of the sample onto this attribute and find the minimal dominating term.
2. Starting with the empty expression $E$, while there are negative observations in the sample do:
   a. Among all attributes, find the minimal dominating term that eliminates the most negative observations and add it to $E$, breaking out of the loop if no minimal dominating term eliminates any negative examples.
   b. Remove from the sample the negative observations that are eliminated and update the projections onto the attributes accordingly.
3. If there are no negative observations left return $E$,
else report that the sample is not consistent with any pure conjunctive concept.

It can be shown that if the set $T$ to be covered has $m$ elements and $p$ is the size of the minimum cover, then the greedy method is guaranteed to find a cover of size at most $p \log m + 1$ *[N69] [J74]*. Hence given a sample of an $p$-term pure conjunctive concept with $m$ negative observations, Algorithm 2 is guaranteed to find a consistent pure conjunctive hypothesis with at most approximately $p \log m$ terms. Using

[3]For simplicity, we will assume that every sample contains at least one positive and one negative observation. This implies (among other things) that a minimal dominating term always exists, and will make our algorithms simpler.

Theorem 1, this gives the approximate upper bound on the convergence rate for Algorithm 2 given by formula (8) in the previous section. Since Algorithm 2 is, like Algorithm 1, a consistent algorithm for arbitrary pure conjunctive concepts, the bound on the convergence rate given in formula (9) holds as well. Note that the bound on the convergence rate for the greedy method is not much worse than the bound (10) for the "ideal" algorithm, yet the greedy method is significantly cheaper computationally.

The compliments of pure conjunctive concepts can be represented as pure disjunctive concepts. Hence this is the *dual form* of pure conjunctive concepts. A variant of Algorithm 2 can be used to learn pure disjunctive concepts. In the dual form, each term must eliminate all negative observations and need only imply some subset of positive observations, and all terms together must imply all positive observations. The dual greedy method is to repeatedly choose the term that implies the most positive observations and add it to the disjunct, removing the positive observations that are implied, until all positive observations are accounted for. This is a variant of the "star" method in *[MCL83]*. Since $k$ term pure disjunctive concepts have a Vapnik-Chervonenkis dimension similar to that of $k$ term pure conjunctive concepts (formula (2)), the analysis of the convergence rate of this algorithm goes through as above.

We now tackle internal disjunctive concepts. The calculation of the Vapnik-Chervonenkis dimension of these concepts given in the previous section indicates that the strongest bias in learning them is to minimize the total number of terms plus internal disjunctions, i.e. to minimize the total *size* of all the terms, where the size of a compound term is defined as the number of internal disjunctions it contains plus one. Let $E$ be an internal disjunctive concept that is consistent with a given sample. As with pure conjunctive concepts, each term in $E$ implies all positive observations and eliminates some set of negative observations. A compound term with this property will be called a *dominating compound term*. We would like to eliminate all the negative observations using a set of terms with the smallest total size. This leads to the following.

*Minimum Set Cover problem with positive integer costs:* given a collection of sets with union $T$, where each set has associated with it a positive integer cost, find a subcollection whose union is $T$ that has the minimum total cost.

Since it generalizes Minimum Set Cover, this problem is clearly NP-hard. However, approximate solutions can found by a generalized greedy method. Let $T'$ be a set of elements remaining to be covered. For each set in the collection, define the *gain/cost ratio* of this set as the number of elements of $T'$ it contains divided by its cost. The generalized greedy method is to always choose the set with the highest gain/cost ratio. As with the basic Minimum Set Cover problem, it can be shown that if the original set $T$ to be covered has $m$ elements and $p$ is the minimum cost of any cover, then the generalized greedy method is guaranteed to find a cover of size at most $p \log m + 1$.

To apply this method in learning internal disjunctions, let the gain/cost ratio of a dominating compound term be the number of negative observations it eliminates divided by its size.

Algorithm 3. (greedy algorithm for learning internal disjunctive concepts)
1. For each attribute, calculate the projection of the sample onto this attribute.
2. Starting with the empty expression $E$, while there are negative observations in the sample do:
   a. Among all attributes, find the dominating compound term $t$ with the highest gain/cost ratio, breaking out of the loop if none have positive gains. If there is no term for the attribute of $t$ already in $E$, add $t$ to $E$. Otherwise replace the old term in $E$ for the attribute of $t$ with $t$.
   b. Remove from the sample the negative observations $t$ eliminates and update the projections onto all attributes accordingly.
3. If there are no negative observations left return $E$,
else report that the sample is not consistent with any internal disjunctive concept.

To implement this algorithm, we need a procedure to find a dominating compound term with the highest gain/cost ratio for a given attribute from the projection of the sample onto that attribute. Since there are in general exponentially many distinct dominating compound terms with respect to the number of leaves of a tree-structured attribute or the number of values of a linear attribute, this cannot be done by exhaustive search. However, there is a reasonably efficient recursive procedure that does this for tree-structured attributes, and a simple iterative procedure for linear attributes. Each of these procedures takes time $O(q^2)$, where $q$ is the number of distinct values of the attribute that appear in the observations. Space limitations preclude a detailed discussion of these procedures.

By formula (3) and the above result on the performance of the generalized greedy method, the numerical bias of Algorithm 3 for $k$-term internal disjunctive target concepts using a total of $j$ internal disjunctions (i.e. of size $k + j$) and sample size $m$ is $\sim$ at most (approx.) $5(k+j)ln(m)\log(5(k+j)\ln(m)\sqrt{n})$ Ignoring the $\log(\ln(m))$ term, formula (7) of Theorem 1 gives an upper bound on the convergence rate similar to that of Algorithm 2 given in equation (8), with $k+j$ substituted for $p$.

## 3. Extensions

There are several possible extensions to these algorithms that would increase their domain of application. We outline two of them here.

1. The ability to handle "don't care" values for some attributes in the sample (see e.g. *[Q86]*, *[V84]*).

A "don't care" value for attribute $A$ corresponds to an observation in rule form having the term $A = any\_value$. In fact, we can go one step further and let observations be arbitrary pure conjunctive expressions, where, for example, the positive observation *shape = polygon and color = blue* means that the concept contains all blue polygons, and the corresponding negative observation means that no blue polygons are contained in the concept. In this form, the problem of learning from examples is seen to be a special case of the more general problem of *knowledge refinement [MIC83]*, wherein we start with a collection of rules that are already known and try to derive from them a simpler, more general (and hopefully more comprehensible) set of rules. This extension can be accomplished by modifying the notion of the projection of the samples onto the attributes to allow terms of the observations to project to internal nodes of the tree-structured attributes or intervals in the linear attributes. Other parts of the algorithm are changed accordingly.

2. Promoting synergy while learning a set of concepts.

So far we have only considered the problem of learning a single concept in isolation. In fact, we would like to build systems that learn many concepts, with higher level concepts being built upon intermediate and low level concepts (see e.g. *[B85]* *[SB86]*). The first step is to extend our notion of concept to include many-valued observations, rather than just positive and negative. In this way we can learn rules that define the values of one attribute in terms of the values of the other attributes. This is essentially knowledge refinement on relational databases *[MIC83]*. Ignoring attributes with many values for the time being, this can be accomplished in a reasonable way by finding a separate concept for each value of the attribute that discriminates this value from all the others.

Once we have learned to recognize the values of the new attribute in terms of the primitive attributes, it can be added to the set of primitive attributes and used later in learning to recognize the values of other attributes. In this scheme new attributes are always nominal. However, they could acquire a tree structure as they are used to define later concepts in the following manner (see also *[U86]* *[BSP85]*): whenever an internal disjunctive concept is formed using a compound term $A = v_1$ *or* $v_2$ *or* $\cdots$ *or* $v_k$, check to see if this same compound term is required by other concepts. If it is required often enough, check the tree for the attribute $A$. If a node for the values $v_1, \ldots, v_k$ can be added without destroying the tree structure, do so. If a new node is added, the compound terms it represents can be replaced by an elementary term using the value of the new node. Thus the collection of rules given in Section 1 for the internal disjunctive concept involv-

ing the primary colors might be created by the "discovery" of the higher level value of *primary* for the attribute *color*. In this way a useful vocabulary of more abstract values for attributes evolves under the pressure to find simple forms for higher level concepts, creating a synergy between learned concepts.

Another type of synergy is achieved by using the algorithm for pure conjunctive concepts along with the dual algorithm for pure disjunctive concepts. If new Boolean attributes are defined for often-used pure conjuncts or disjuncts, then these can allow the recognition of higher level concepts in DNF and CNF respectively by effectively reducing these expressions to pure disjunctive or conjunctive form. Often used internal disjunctive concepts could be used as well. The creation of these new attributes can greatly increase the number of attributes that are considered in later learning tasks, which argues strongly for learning methods whose performance does not degrade badly as the number of attributes grows, such as those we have presented.

## Conclusion.

We have presented a methodology for the quantitative analysis of learning performance based on a relatively simple combinatorial property of the space of hypotheses explored by the learning algorithm. Applications of this methodology have been presented in the development and analysis of learning algorithms for pure conjunctive, pure disjunctive and internal disjunctive concepts. Several open problems remain, in addition to those mentioned above. Some are:
1. Can we develop the proper analytic tools to deal with algorithms that
a. attempt to handle the problem of noisy data *[Q86]* or
b. attempt to learn "fuzzy" concepts that are defined probabilistically with respect to the instance space?
2. What power is gained by allowing the learning algorithm to form queries during the learning process *[SB86] [ANG86]*?
3. Can we find provably efficient incremental learning algorithms (i.e. ones that modify an evolving hypothesis after each observation) to replace the "batch processing" learning algorithms we have given here?
4. To what extent can we extend these results to concepts that involve internal structure, expressed with the use of variables, quantifiers and binary relations (e.g. the c-expressions of *[MCL83]*)?

**Acknowledgements.** I would like to thank Larry Rendell for suggesting the relationship between the Vapnik-Chervonenkis dimension and Utgoff's notion of inductive bias and Ryszard Michalski for suggesting I look at the problem of learning internal disjunctive concepts. I also thank Les Valiant, Leonard Pitt, Phil Laird, Ivan Bratko and Stephan Muggleton and Andrzej Ehrenfeucht for helpful discussions of these ideas, and an anonymous referee for suggestions on improving the presentation.

## References:

[**ANG86**] Angluin, A., "Learning regular sets from queries and counter-examples," Tech. rep. YALEU/DCS/TR-464, Yale University, 1986.
[**A83**] Assouad, P., "Densite et Dimension," *Ann. Inst. Fourier, Grenoble* 33 (3) (1983) 233-282.
[**B85**] Banerji, R., "The logic of learning: a basis for pattern recognition and improvement of performance," in *Advances in Computers,* 24, (1985) 177-216.
[**BEHW86**] Blumer, A., A. Ehrenfeucht, D. Haussler and M. Warmuth, "Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension," *18th ACM Symp. Theor. Comp.,* Berkeley, CA, 1986, to appear.
[**BSP85**] Bundy, A., B. Silver and D. Plummer, "An analytical comparison of some rule-learning programs," *Artif. Intel.* 27 (1985) 137-181.
[**C82**] Cohen, P. and E. Feigenbaum, *Handbook of AI, Vol. 3,* William Kaufmann, 1982, 323-494.
[**GJ79**] Garey, M. and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H.Freeman, 1979.
[**J74**] Johnson, D.S., "Approximation algorithms for combinatorial problems," *J. Comp. Sys. Sci.,* 9, 1974.
[**MCL83**] Michalski, R.S., "A theory and methodology of inductive learning," in *Machine learning: an artificial intelligence approach,* Tioga Press, 1983, 83-134.
[**MIC83**] Michie, D., "Inductive rule generation in the context of the fifth generation," *Proc. Int. Mach. Learning Workshop,* Monticello, Il., (1983) 65-70.
[**MIT82**] Mitchell, T.M., "Generalization as search," *Art. Intell.* 18 (1982) 203-226.
[**N69**] Nigmatullin, R.G., "The Fastest Descent Method for Covering Problems (in Russian)," *Proceedings of a Symposium on Questions of Precision and Efficiency of Computer Algorithms,* Book 5, Kiev, 1969, pp. 116-126.
[**P78**] Pearl, J., "On the connection between the complexity and credibility of inferred models," *Int. J. Gen. Sys.,* 4, 1978, 255-64.
[**Q86**] Quinlan, J.R., "Induction of decision trees," *Machine Learning,* 1 (1) (1986), to appear.
[**R86**] Rendell, L., "A general framework for induction and a study of selective induction," *Machine Learning* 1 (2) (1986), to appear.
[**SB86**] Sammut, C., and R. Banerji, "Learning concepts by asking questions," in *Machine Learning II,* R. Michalski, J. Carbonell and T. Mitchell, eds., Morgan Kaufmann, Los Altos, CA, 1986.
[**U86**] Utgoff, P., "Shift of Bias for inductive Concept Learning," ibid.
[**V84**] Valiant, L.G., "A theory of the learnable," *Comm. ACM,* 27(11), 1984, pp. 1134-42.
[**V85**] Valiant, L.G., "Learning disjunctions of conjunctions," *Proc. 9th IJCAI,* Los Angeles, CA, 1985, 560-6.
[**VC71**] Vapnik, V.N. and A.Ya.Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Th. Prob. and its Appl.,* 16(2), 1971, 264-80.
[**WD81**] Wenocur, R.S. and R.M.Dudley, "Some special Vapnik-Chervonenkis classes," *Discrete Math.,* 33, 1981, 313-8.