Tree-Structured Bias*

Stuart J. Russell
Computer Science Division
University of California
Berkeley, CA 94720

Abstract

This paper reports on recent progress in the study of autonomous concept learning systems. In such systems, the initial space of hypotheses is considered as a first-order sentence, the *declarative bias*, and can thus be derived from background knowledge concerning the goal concept. It is easy to show that a simple derivation process generates a concept language corresponding to an unbiased version space defined on a restricted instance description language. However, the structure of a typical derivation corresponds to a stronger restriction still. It is shown that this semantically-motivated, *tree-structured bias* can in fact reduce the size of the concept language from doubly-exponential to singly-exponential in the number of features. This allows effective learning from a small number of examples.

1 Autonomous Concept Learning

The object of concept learning is to come up with predictive rules that an intelligent agent can use to survive and prosper. For example, after being 'presented' with several instances, an agent might decide that it needed to discover a way of predicting when an animal was liable to attack it, and eventually that large animals with long, pointy teeth and sharp claws are carnivorous:

$$\forall x\ Animal(x) \wedge Large(x) \wedge \ldots \Longrightarrow Carnivorous(x)$$

We give this example to emphasize our main concern in this paper: the construction of *autonomous* learning agents. It is now fairly well accepted that the process of learning a concept from examples can be viewed as a search in a *hypothesis space* (or version space) for a concept definition consistent with all examples, both positive and negative (Mitchell, 1982; Angluin & Smith, 1983). Current learning systems are given a hypothesis space and instance descriptions carefully designed *by the programmer* for the purposes of learning the concept that the programmer wants learnt. The job of the learning program under these circumstances is to 'shoot down' inconsistent hypotheses as examples are analysed, rather like a sieve algorithm for finding prime numbers. In practice this task requires some extremely ingenious algorithms, but it is only one aspect of the whole learning problem. We need systems that can construct their own hypothesis spaces and instance descriptions, for their own goals. After all, an agent in the real world may be 'given' its original instance descriptions in terms of pixels, which hardly provide a suitable language in which to describe carnivores. The sentiment of (Bundy et al., 1985) is worth repeating: "Automatic provision ... of the description space is the most urgent open problem facing automatic learning."

Our theoretical project, begun in (Russell, 1986a), has two parts. The first is to analyse what knowledge must be available to the system prior to beginning the learning task and how it can be used to set up a hypothesis space and to choose descriptions for instances. The second part is to analyse the subsequent process of learning a concept from examples as an *inference process*, from instances and background knowledge to the required rule.

The basic approach we have taken (Russell & Grosof, 1987) has been to express the hypothesis space as a first-order sentence, hence the term *declarative bias*. The idea is that, given suitable background knowledge, a system can derive its own hypothesis space, appropriate for a particular goal, by logical reasoning of a particular kind. This paper reports on an important aspect of our research on declarative bias. After giving the basic definitions and theorems pertaining to the automatic derivation of bias, and a brief discussion of the derivation algorithm, I show that the structure of the derivation imposes a strong, and apparently natural, constraint on the hypothesis space. A quantitative analysis of this constraint is given, and then the implications of the results are discussed in a broader context.

1.1 Basic Definitions

First we define the notions used in the logical development below:

- The *concept language*, that is, the initial hypothesis space, is a set C of *candidate (concept) descriptions* for the concept. Each concept description is a unary predicate schema (open formula) $C_j(x)$, where the argument variable is intended to range over instances.
- The *concept hierarchy* is a partial order defined over C. The generality/specificity partial ordering is given by the non-strict ordering $\leq$, representing quantified implication, where we define $(A \leq B)$ iff $\{\forall x.A(x) \Longrightarrow B(x)\}$
- An *instance* is just an object a in the universe of discourse. Properties of the instance are represented by sentences involving a.

*Computer facilities were provided by the Computer Science Division of the University of California at Berkeley, and financial assistance by the AT&T Foundation, Lockheed AI Center and the University of California MICRO program.

- An *instance description* is then a unary predicate schema D, where $D(a)$ holds. The set of allowable instance descriptions forms the instance language $\mathcal{D}$.
- The *classification* of the instance is given by $Q(a)$ or $\neg Q(a)$. Thus the i^{th} observation, say of a positive instance, would consist of the conjunction $D_i(a_i) \wedge Q(a_i)$.
- A concept description C_j *matches* an instance a_i iff $C_j(a_i)$. The latter will be derived, in a logical system, from the description of the instance and the system's background knowledge.

We are now ready to give the definition for the instance language bias. Choosing such a bias corresponds to believing that the instance descriptions in the language contain enough detail to guarantee that no considerations that might possibly affect whether or not an object satisfies the goal concept Q have been omitted from its description. For this reason, we call it the *Complete Description Axiom* (CDA). Its first-order representation is as follows:

Definition 1 (CDA):

$$\bigwedge_{D_i \in \mathcal{D}} (D_i \leq Q) \vee (D_i \leq \neg Q)$$

That is, instances with a given description are either all guaranteed positive or all guaranteed negative.

The heart of any search-based approach to concept learning is the assumption that *the correct target description is a member of the concept language*, i.e. that the concept language bias is in fact *true*. We can represent this assumption in first-order as a single *Disjunctive Definability Axiom* (DDA):

Definition 2 (DDA):

$$\bigvee_{C_j \in \mathcal{C}} (Q = C_j)$$

(Here we abbreviate quantified logical equivalence with "=" in the same way we defined "$\leq$".)

An important notion in concept learning is what Mitchell (1980) calls the *unbiased version space.* This term denotes the hypothesis space consisting of all possible concepts definable *on the instance language.* A concept is extensionally equivalent to the subset of the instances it matches, hence we have

Definition 3 (Unbiased version space):

$$\{C \mid C \text{ matches exactly some element of } 2^{\mathcal{D}}\}$$

As it stands, the extensional formulation of the CDA is inappropriate for automatic derivation from the system's background knowledge. A compact form can be found using a *determination* (Davies & Russell, 1987), a type of first-order axiom that expresses the relevance of one property or schema to another. A determination is a logical statement connecting two relational schemata. The determination of a schema Q by a schema P is written $P \succ Q$, and defined as follows:

Definition 4 (Determination):
$P \succ Q$ iff

$$\forall wx[\exists y[P(w,y) \wedge P(x,y)] \Longrightarrow \forall z[Q(w,z) \Longrightarrow Q(x\ z)]]$$

Determinations involving *unary* schemata (such as "One's age determines *whether or not one requires a measles vaccination in case of an outbreak*") are best expressed using *truth-valued variables* as virtual second arguments. Following Davies and Russell (1987), the truth-valued variable is written as a prefix on the formula it modifies. The letters $ijk\ldots$ are typically used for such variables. Thus the measles determination is written

$$Age(x,y) \succ k\, MeaslesVaccineNeeded(x)$$

The addition of truth-valued variables to the language significantly reduces the length of some formulæ relevant to our purposes, and allows for a uniform treatment.

1.2 Basic Theorems

We now give the basic theorems that establish the possibility of automatic derivation of an initial hypothesis space. Proofs are given in detail in (Russell, forthcoming).

Theorem 1: The disjunctive definability axiom corresponding to an unbiased version space is logically equivalent to the complete description assumption.

Proof: Writing out the CDA as a conjunction, and distributing $\wedge$ over $\vee$, we obtain a disjunction of 2^n disjuncts, where n is the size of the instance description language D. Each disjunct assigns different subsets of the instances to be positive and negative instances, and is thus a concept definition from the unbiased version space. □

Theorem 2: The complete description assumption can be expressed as a single determination of the form

$$D(x,y) \succ k\, Q(x)$$

where $D(x, Y_i) \equiv D_i(x)$.

Proof: Definition 4 for the determination is transformed into Horn form, then we expand the quantification over y and k into a conjunction. Rearrangement of the resulting conjuncts into pairs of disjuncts gives us the CDA. □

From theorem 1 we then obtain

Corollary: The unbiased version space can be expressed as a single determination of the form

$$D(x,y) \succ k\, Q(x).$$

As an example of the power of determinations to express hypothesis spaces, consider the simple case of an instance language with just two boolean predicates G and H. The unbiased version space for this language in figure 1. The corresponding determination is

$$i\, P_1(x) \wedge j\, P_2(x) \succ k\, Q(x).$$

In general, a determination with k Boolean features corresponds to an unbiased version space of 2^{2^k} elements.

2 Deriving an initial bias

In this section I remark briefly on the considerations that apply to the process of deriving a suitable determination to form the initial hypothesis space for a concept learning problem. This will help to put the following section into context.

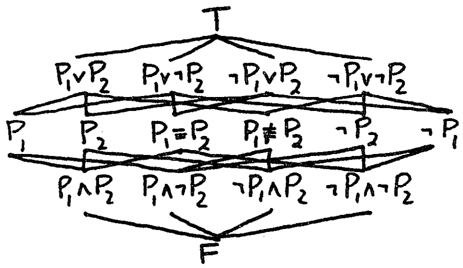

Figure 1: Unbiased version space for two boolean predicates

Although, in principal, the inference of the determination could be performed as a resolution proof, a specialized reasoner is more appropriate. What we want to get out of the inference process is a determination for the goal concept *such that the left-hand side forms a maximally operational schema.* The notion of operationality of a concept definition is central in the literature on explanation-based learning (Mitchell, Keller & Kedar-Cabelli, 1986; Keller, 1987), where it refers to the utility of a concept definition for recognizing instances of a concept. Our use of the term is essentially the same, since the left-hand side of the determination forms the instance language bias. This means that it should be easy to form a description of the instance within the instance language it generates. For example, to learn the *DangerousCarnivore* concept we would like to find a bias that refers to visible features of the animal such as size and teeth, rather than to features, such as diet, whose observation may involve considerable cost to the observer. The particular operationality criteria used will clearly depend on the situation and overall goals and capabilities of the agent. In our implementation we adopt the approach taken by Hirsh (1987), who expresses knowledge about operationality as a set of meta-level sentences. Effectively, these sentences form an 'evaluation function' for biases, and help to guide the search for a suitable instance language bias.

There is also an additional criterion for judging the utility of a particular bias. The success and expected cost of doing the concept learning will depend critically on the size and nature of the bias. A weak bias will mean that a large number of instances must be processed to arrive at a concept definition. Maximizing operationality for our system therefore means minimizing the size of the hypothesis space that is derived from the determination we obtain. The following section describes the computation of the size of the hypothesis space corresponding to a given tree-structured bias.

But what form does the derivation of a bias take? Since we are beginning with a goal concept for which we must find an operational determination, we must be doing some kind of backward chaining. The inference rules used for the chaining will not, however, be standard modus ponens, since we are attempting to establish a universal and the premises used are usually other determinations, as opposed to simple implicative rules. Thus the basic process for deriving a suitable instance language bias is implemented as a backward chaining inference, guided by operationality criteria, and using inference rules appropriate for concluding determinations. These inference rules are given in (Russell, 1986b). The particular rule that is of interest for this paper is the *extended transitivity* rule, valid for functional relations:

$$A \succ B,\ B \wedge C \succ D \ \vdash\ A \wedge C \succ D$$

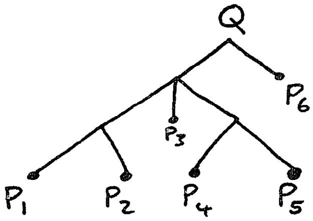

Figure 2: A bias derivation tree

An example of a derivation tree is given in figure 2. The tree corresponds to the derivation of the determination

$$P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5 \wedge P_6 \succ Q$$

If the features P_1 through P_6 are known to be operational, for example if they are easily ascertained through experiment, then the system will have designed an appropriate instance language for the goal concept Q, and hence an initial, 'unbiased' hypothesis space. It is worth noting that there might be a very large number of features potentially applicable to objects in the domain of Q, so this bias represents a considerable restriction.

3 Tree-Structured Bias

It is clear that the unbiased hypothesis space derived by the above procedure will not allow successful inductive learning if used 'as is'. Elsewhere (Russell, forthcoming), I discuss ways in which it can be restricted by the addition of further domain knowledge and the imposition of syntactic restrictions based on computational considerations. I will now show that the determinations used in the derivation of the bias themselves impose a strong additional restriction on the space of possible definitions for the goal concept.

Intuitively, the restriction comes about because the tree structure of the derivation limits the number of ways in which the different features can interact. For example, in figure 2, P_1 and P_2 cannot interact separately with P_3, but only through the function which combines them. Another way to think about it is to consider the value of Q as a function of the variables which are the values of P_1 through P_6. The 'flat' bias determination derived above simply states that

$$q = f(p_1, p_2, p_3, p_4, p_5, p_6)$$

for some boolean function f. The tree-structured derivation in Figure 2 shows that the form of the function is restricted:

$$q = f(g(h(p_1, p_2), p_3, j(p_4, p_5)), p_6)$$

for some functions f, g, h, j. In the following paragraphs the formula for the number of functions allowed by an arbitrary tree structure will be developed. For simplicity of

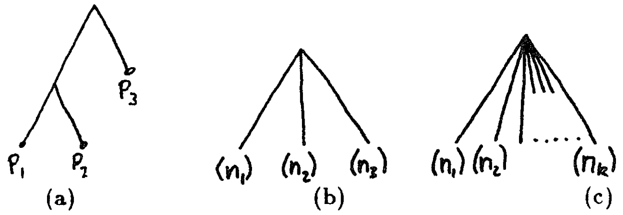

Figure 3: Examples of tree-structured biases

presentation, we will assume that all descriptive features are boolean.

First, , consider the simplest possible nested tree structure, shown in figure 3(a). This corresponds to the functional equation $q = f(g(p_1, p_2), p_3)$. There are $2^{2^2} = 16$ possible functions g, and these are shown in figure 1. Note that the negation of each of the 16 also appears in the set. There are also 16 possible functions f, but this does *not* mean that there are 16 x 16 possible functions q. In four of the functions f, namely $f = \textit{true}$, $f = \textit{false}$, $f = p_3$ and $f = \neg p_3$, the first argument does not appear. The remaining 12 can be divided into 6 pairs which are mirror images under negation of the first argument; e.g., $g \wedge p_3$ and $\neg g \wedge p_3$ form such a pair. Of the 16 possible instantiations of the first argument, i.e. the functions g, two (the true and false functions) generate expressions redundant with one of the four mentioned above. The remaining 14 are divided into 7 pairs, each of which contains an expression and its negation. The combined set of 12 × 14 functions thus consists of 12 × 7 = 84 functions each appearing twice. We thus have 84 + 4 = 88 possible rules for Q instead of $2^{2^3} = 256$ for the flat bias.

In general, if there are n functions in a subtree the number of functions in the supertree will be multiplied by $(n - 2)/2$, for those functions in which the corresponding argument appears.

Consider now case 3(b), in which we have already computed the number of functions in the subtrees to be n_1, n_2, n_3. There are $2^{2^3} = 256$ functions at the top level. We now count the total number of distinct functions generated when these are combined with the functions from the subtrees.

- In 2 of the 256, none of the three arguments appear, giving us 2.
- In 2 of the 256, only the first argument appears, giving us $2((n_1 - 2)/2)$. Similarly, the other branches contribute $2((n_2 - 2)/2)$ and $2((n_3 - 2)/2)$. Hence we get $2((n_1 - 2) + (n_2 - 2) + (n_3 - 2))/2$ in total.
- In 10 of the 256, only the first two arguments appear. Each of these 10 generates $(n_1 - 2)(n_2 - 2)/4$ functions (since we get double redundancy). Thus functions in which only two arguments appear contribute $10((n_1 - 2)(n_2 - 2) + (n_2 - 2)(n_3 - 2) + (n_3 - 2)(n_1 - 2))/4$ in total.
- In 256 − (10 + 10 + 10) − (2 + 2 + 2) − 2 = 218 of the top-level functions all three arguments appear. The redundancy factor is now 8, so we get $218((n_1 - 2)(n_2 - 2)(n_3 - 2))/8$ functions in total.
- The total number of rules consistent with the tree structure is the sum of these four terms.

The general formula for a tree of arbitrary structure can only be given as a recursive relationship between the total number of functions and the number of functions from each immediate subtree (see figure 3(c)) A subtree that is a leaf node contributes 4 functions.

Theorem 3: Let $n_1 \ldots n_k$ be the numbers of functions contributed from the k branches of a tree-structured bias derivation. Then the number of rules consistent with the bias is given by

$$\sum_{j=0}^{k} \frac{A_j}{2^j} S_j(n_1 - 2, \ldots, n_k - 2)$$

where S_j is the sum of products of its arguments taken j at a time, with $S_0 = 1$, and A_j is the number of boolean functions of j variables in which all the variables appear. A_j is computed using the following facts:

$$A_0 = 2$$
$$A_j = 2^{2^j} - \sum_{i=0}^{j-1} \binom{j}{i} A_i$$

Proof: by induction on the structure of the tree. □

These formulæ may be somewhat difficult to interpret. Indeed, it seems surprising that simply organizing the functional expression for Q into a tree would cause a very large reduction in the number of possible functions. But even in a small case the reduction is dramatic: a balanced, four-leaf binary tree structure allows 520 possible rules, as compared to 65536 for the flat bias. In fact, we can state a general result that may be quite important for the possibility of efficient autonomous learning.

Theorem 4: For a tree-structured bias whose degree of branching is bounded by a constant k, the number of rules consistent with the bias is exponential in the number of leaf nodes.

Proof: Any tree with n leaves has at most $n - 1$ internal nodes. Each internal node generates at most 2^{2^k} times the product of the numbers of functions generated by its subtrees. The total number of functions in the tree is thus bounded by $(2^{2^k})^{n-1}$. □

Corollary: Given a tree-structured bias as described above, with probability greater than $1 - \delta$ a concept can be learned that will have error less than ϵ from only m examples, where

$$m = \frac{1}{\epsilon}\left[ln\left(\frac{1}{\delta}\right) + (n-1)2^k\right]$$

Proof: Direct instantiation of Lemma 2.1 in (Haussler, 1988). □

Since the size of the 'unbiased' hypothesis space is doubly exponential in the number of leaves, requiring an exponential number of examples, it seems that the tree structure represents a very strong bias, even beyond that provided by the restriction to a circumscribed set of primitive features. For comparison, a strict conjunctive bias also requires a linear number of examples.

To achieve learnability in the sense of Valiant (1984), we must find a polynomial-time algorithm for generating hypotheses consistent with the tree-structured bias and a set of examples. Such an algorithm has been found for the case in which the functions at each internal node of the tree are restricted to be monotone. The general case seems more difficult. The natural process for identifying the correct rule is simply to identify the correct rule for each subtree in a bottom-up fashion, by generating experiments that vary the features in the subtree, keeping other features constant. Since, by construction, internal nodes of the tree are not easily observable, the induction process is far from trivial.

4 Discussion

Especially given the recent positive results on the learnability of functions in the presence of background knowledge in the form of determinations, due to Mahadevan and Tadepalli (1988), it is tempting to view the above analysis as another class of concepts in the process of being shown to be learnable. It is, however, important to keep in mind that a tree-structured bias is derived from background knowledge of the domain, rather than being a syntactic restriction. In addition, the derivation generates a restricted set of features to be considered, and can thus be seen as providing a solution for the *situation-identification problem* (Charniak & McDermott, 1985). In the theory of learnability, the set of features is considered part of the input, or, for an autonomous agent, to be perhaps the set of all features at the agent's disposal (Genesereth & Nilsson, 1987).

A simple theorem prover for deriving suitable determinations has been implemented, and has been used to automate the derivation of the Meta-DENDRAL bias first shown in (Russell & Grosof, 1987). We are currently in the process of developing suitably broad domain theories so that the system can be used to derive biases for a number of different goal concepts within an area of investigation. The relationship between knowledge-based bias derivation and the intelligent design of scientific experiments is particularly intriguing. A scientist designing an experiment to measure the gravitational acceleration g seems to select exactly the right variables to vary. She does not concern herself with the possible effect of presidential incumbents on the force of gravity; this is a good thing, since otherwise experiments would have to be repeated at four-year intervals. It would be of interest to philosophers of science to be able to model such considerations using a knowledge-based process. There also seem to be strong connections between the idea of tree-structured bias and Hintikka's notion of *interactional depth*, which concerns the degree of nesting and interaction of variables assumed when constructing theories of multi-variable phenomena, such as occur in many-body problems.

On the technical front, there remain questions of how tree-structured bias will interact with other biases such as conjunctive bias and the predicate hierarchy; of how the bias can be used to direct experimentation; and of how we can formally analyse more complex bias derivations, for instance those using other inference rules and those in which the same feature appears several times. In addition, we would like to study the use of other classes of background knowledge. These are all interesting subproblems for a general theory of knowledge-guided induction.

5 Acknowledgements

Particular thanks go to Benjamin Grosof, with whom the early ideas for this work were developed, and to Lise Getoor and an anonymous reviewer (David Haussler) for their contributions. I would like to thank Bruce Buchanan, John Canny, Thomas Dietterich, Haym Hirsh, Marie desJardins, Gerald deJong, Sridhar Mahadevan, Tom Mitchell, Devika Subramanian, Les Valiant, Umesh Vazirani, David Wilkins, and the participants in the machine learning seminar at Berkeley and the GRAIL and MUGS seminars at Stanford for valuable discussion and comments.

References

[1] Angluin, D., and Smith C. H. (1983). Inductive inference: Theory and methods. *Computing Surveys 15*, pp. 237-269.

[2] Bundy, A., Silver, B., and Plummer, D. (1985). An Analytical Comparison of Some Rule-Learning Programs. *Artificial Intelligence, 27.*

[3] Charniak, E., and McDermott, D. (1985) *Introduction to artificial intelligence.* Reading, MA: Addison-Wesley.

[4] Davies, T. R. and Russell, S. J. (1987). A Logical Approach to Reasoning by Analogy. *Proc. Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy.

[5] Haussler, D. (1988). *Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework.* Technical report, Department of Computer Science, University of California, Santa Cruz, CA.

[6] Hirsh, H. (1987). Explanation-based generalization in a logic programming environment. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy.

[7] Keller, R. M. (1987). Defining operationality for explanation-based learning. *Proc. Sixth National Conference on Artificial Intelligence*, Seattle, WA.

[8] Mahadevan, S. and Tadepalli, P. (1988). On the tractability of learning from incomplete theories. *Proc. Fifth International Conference on Machine Learning*, Ann Arbor, MI.

[9] Mitchell, Tom M. (1980). *The Need for Biases in Learning Generalizations.* Technical report CBM-TR-117, Rutgers University, New Brunswick, NJ.

[10] Mitchell, Tom M. (1982). Generalization as search. *Artificial Intelligence*, Vol. 18, No. 2, 203-226.

[11] Russell, S. J. (1986a). Preliminary Steps Toward the Automation of Induction. *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA.

[12] Russell, S. J. (1986b). *Analogical and Inductive Reasoning.* Ph. D. thesis, Stanford University, Stanford, CA.

[13] Russell, S. J. (forthcoming). Autonomous Concept Learning. Unpublished manuscript.

[14] Russell, S. J., and Grosof, B. N. (1987) "A Declarative Approach to Bias in Concept Learning." *Proc. Sixth National Conference on Artificial Intelligence*, Seattle, WA.

[15] Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM, 27*, 1134-1142.