From: AAAI-87 Proceedings. Copyright ©1987, AAAI (www.aaai.org). All rights reserved.

A Declarative Approach to Bias in Concept Learning

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

Benjamin N. Grosof
Computer Science Department
Stanford University
Stanford, CA 94305

Abstract

We give a declarative formulation of the biases used in inductive concept learning, particularly the Version-Space approach. We then show how the process of learning a concept from examples can be implemented as a first-order deduction from the bias and the facts describing the instances. This has the following advantages: 1) multiple sources and forms of knowledge can be incorporated into the learning process; 2) the learning system can be more fully integrated with the rest of the beliefs and reasoning of a complete intelligent agent. Without a *semantics* for the bias, we cannot generally and practically build machines that generate inductive biases automatically and hence are able to learn independently. With this in mind, we show how one part of the bias for Meta-DENDRAL, its instance description language, can be represented using first-order axioms called *determinations*, and can be derived from basic background knowledge about chemistry. The second part of the paper shows how bias can be represented as defaults, allowing shift of bias to be accommodated in a non-monotonic framework.

I. Introduction

The standard paradigm for inductive concept learning as hypothesis refinement from positive and negative examples was discussed by John Stuart Mill (1843), and has since become an important part of machine learning research. The currently dominant approach to concept learning is that of a search through a *predefined space* of candidate definitions for one that is consistent with the data so far seen.

The approach that we are proposing is to view the process of learning a concept from examples as an *inference process*, beginning from *declaratively expressed* premises, namely the instances and their descriptions *together with whatever else the system may know*, and leading to a conclusion, namely (if the system is successful) a belief in the correctness of the concept definition arrived at. The premises should provide good reasons, either deductive or inductive, for the conclusions. One part of our project, begun in (Russell, 1986a), is therefore to show how existing knowledge can generate extra constraints on allowable or preferable hypotheses, over and above simple consistency with observed instances. These constraints were grouped by Mitchell (1980) under the term *bias*. This is perhaps an unfortunate term, since it suggests that we have something other than a good reason for applying these constraints. Mitchell himself concludes the paper with:

> It would be wise to make the biases and their use in controlling learning just as explicit as past research has made the observations and their use.

The most important reason for the declarative characterization of bias is that without it, concept learning cannot practically become an integral part of artificially intelligent systems. As long as the process of deciding on a bias is left to the programmer, *concept learning is not something an AI system can do for itself*. And as Rendell (1986) has shown, in typical AI concept learning systems, most of the information is contained in the choice of bias, rather than in the observed instances. We will therefore try to analyze biases to see what they mean as facts or assumptions about the world, i.e. the environment external to the program. We will also need a plausible argument as to how a system could reasonably come to believe the premises of the deductive process; they should be automatically *acquirable*, at least in principle.

We will first describe the *Version Space* method and *candidate elimination* procedure of Mitchell (1978), and will show how the various types of bias present in this method can be represented as first-order statements. We illustrate this by formalizing part of the bias used in the Meta-DENDRAL system (Buchanan and Mitchell 1978), and deriving it from basic knowledge of chemistry.

The second part of the paper deals with the question of *bias shift*: the process of altering a bias in response to observations that contradict or augment an existing bias. We show that this process can be formulated as a nonmonotonic deduction.

This paper is a condensation of two longer papers that are in preparation for publication elsewhere.

II. The Version Space Approach

In this section we describe how the biases used in the Version Space method can be represented as sentences in first-order logic. The following section describes the process of updating the version space as a deduction from the bias and examples.

The Version Space method is the most standard AI approach to concept learning from examples. It equates the space of possible definitions of a *target concept* with the elements of a *concept language*, which is defined on a *predicate vocabulary* that consists of a set of basic predicates that apply to objects in the universe of instances of the concept. The predicates may be arranged into a *predicate hierarchy*, defined by subsumption relations between elements of the vocabulary. This in turn helps to define a *concept hierarchy* on all the possible, *candidate* concept definitions in the concept language, based again on subsumption as a partial ordering. The programmer defines the *initial version space* to be the concept language, in the belief that the correct definition is expressible in the concept language chosen. In addition to the concept language, there is an *instance description language*. The system is also given a classification for each instance: either it is a positive example of the target concept Q, or it is a negative example. At any point in a series of observational updates, some subset (possibly a singleton or the empty set) of the candidate definitions will be consistent with all the observed instances. This subset is called the *current version space*. Further constraints may be used to choose one of the consistent hypotheses as the rule to be "adopted" — the *preference criteria* of Michalski (1983).

The VS approach has the following difficulties:

1. The framework cannot easily accommodate noisy data.
2. It is hard to incorporate arbitrary background knowledge.

3. It is very difficult to come up with a suitable concept language for complex or unfamiliar concepts. Moreover, there is no semantics attached to the choice, and hence no *a priori* generating mechanism.

By casting the updating process as a first-order inference, we hope to overcome the second and third problems; the first can be solved within a more complex, probabilistic model, or by using appropriate default rules (see below).

A. Concept descriptions and instances

The concept language, i.e. the initial version space, is a set $\mathcal{C}$ of *candidate (concept) descriptions* for the concept. The concept hierarchy is a strict partial order defined over $\mathcal{C}$. Each concept description is a unary predicate schema (open formula) $C_j(x)$, where the argument variable is intended to range over instances. Mitchell defines the concept ordering in terms of matching: C_j is less general than C_k if and only if C_j matches a proper subset of the instances matched by C_k. In our formulation, this ordering is a logical relationship between concepts. As in (Subramanian & Feigenbaum 1986), the hierarchy is expressed as a set of facts relating the concepts by implication. The more natural ordering is the non-strict relationship $\leq$, representing quantified implication, where we define

$$(A \leq B) \quad \text{iff} \quad \{\forall x.A(x) \Rightarrow B(x)\}$$
$$(A < B) \quad \text{iff} \quad \{(A \leq B) \wedge \neg(B \leq A)\}$$

This implication relationship between concept descriptions is also Buntine's *generalized subsumption* (1986). Background knowledge, including the predicate hierarchy, that can be used to derive $\leq$ relations between concepts is contained in an *articulation theory* Th_a (so called because it links different levels of description), so that

$$C_j \leq C_k \quad \text{iff} \quad \text{for any } x:, \quad Th_a, C_j(x) \models C_k(x).$$

For example, if we are trying to induce a definition for *SuitablePet*, Th_a might contain $\forall x[BarksALot(x) \Rightarrow Noisy(x)]$, which induces an ordering between

$$C_j = Furry(x) \wedge BarksALot(x) \wedge EatsTooMuch(x)$$

and the more general concept

$$C_k = Noisy(x) \wedge EatsTooMuch(x).$$

Thus the implication relations in the concept hierarchy do not have to be encoded explicitly for every pair of concepts.

An *instance* is just an object a in the universe of discourse. Properties of the instance are represented by sentences involving a. An *instance description* is then a unary predicate schema D, where $D(a)$ holds. 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 $D_i(a_i) \wedge Q(a_i)$. For example, we might have

$$Cat(Felix) \wedge Furry(Felix) \wedge Eats(Felix, 50¢/day) \wedge \ldots$$
$$\ldots \wedge SuitablePet(Felix).$$

A concept description C_j matches an instance a iff $C_j(a)$. This must be derived on the basis of the description D of the instance; the derivation can use facts from the articulation theory Th_a (which thus links instance-level terms to concept-level terms). In order to have complete matching, which is necessary for the VS process to work (Mitchell, 1978), Th_a must entail either $D_i \leq C_j$ or $D_i \leq \neg C_j$ for any instance description D_i and any concept description C_j. When these relationships hold without relying on facts in the articulation theory, we have what is commonly known as the *single-representation trick*.

B. The instance language bias

Our orientation towards the handling of instances is considerably different from that in, say, the LEX system (Mitchell et al. 1983), in which instances are *identified* with syntactic structures, as opposed to being objects which happen to satisfy descriptive predicates. Logically speaking, an instance in Mitchell's system is a *complex term*, rather than a symbol described by sentences. Thus Felix would be represented by, say, $\langle cat; furry; 50¢/day; \ldots \rangle$ instead of a set of sentences about *Felix*. Two instances with the same description become identical (and therefore co-referring) terms; it is therefore logically impossible for them to have different classifications. This is clearly a non-trivial assumption, since it says that the instance description language contains 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 Assumption* (CDA), and note that it may need to be reasoned about extensively. We therefore prefer to make it an explicit domain fact (or set of facts), i.e.

$$(D_i \leq Q) \vee (D_i \leq \neg Q) \quad \text{for every } i.$$

Another way of expressing this fact is to say that D_i determines whether or not Q holds for an object. It therefore corresponds to the *determination* (Davies & Russell, 1987):

$$D_i(x) \succ k\,Q(x)$$

where k is a truth-value variable. The CDA can also be seen as the ability to do *single-instance generalization*:

$$\forall k.\{D_i(a) \wedge kQ(a)\} \Rightarrow \{\forall x.D_i(x) \Rightarrow kQ(x)\}$$

If the instance description language is infinite, then the CDA will be an infinite set of determinations. We can, however, rewrite it in many cases as a small set of axioms by using binary schemata. Take, for example, an instance description language consisting of a feature vector with n components, some of which, e.g. *weight*, *shape*, may have an infinite number of possible values. We can reduce the CDA for this infinite language to a single axiom, which says that the conjunction of the features determines whether or not an instance satisfies Q:

$$\{\bigwedge_{j=1}^{n} F_j(x, y_j)\} \succ k\,Q(x)$$

where $F_j(x, y_j)$ says that x has value y_j for the j^{th} feature. Such a language appears in the ID3 system (Quinlan 1983).

It is clear from its formal expression that the instance language bias *can only be derived from knowledge concerning the target concept itself.* In a later section we will give just such a derivation using the Meta-DENDRAL system as an example.

Another perspective on the CDA is that the determinations underlying it tell us how to recognize and to handle extra or lacking information in observations. If the observational update is $E_i(a_i) \wedge kQ(a_i)$, where E_i is stronger than D_i, the agent uses the determination to deduce the single-instance generalization $\forall D_i(x) \Rightarrow kQ(x)$, not just the weaker $\forall x.E_i(x) \Rightarrow kQ(x)$. If important information is *lacking* in the update, the agent can use its knowledge of relevancy both to exploit the partial information, and to generate a new goal to obtain the missing detail. *Thus the declarative formulation suggests how to generalize the VS method to less structured learning situations.*

C. The concept language bias

The heart of the Version Space method 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*:

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

(Here we abbreviate quantified logical equivalence with "=" in the same way we defined "$\leq$".) This axiom may be very long or even infinite; we can reduce an infinite DDA to a finite set of axioms using determinations, just as for the CDA.

Subramanian and Feigenbaum (1986) introduce the notion of a version space formed from conjunctive *factors*. We can express such a situation with an axiom that says the target concept Q is equivalent to a conjunction of *concept factors* Q_l, with an analogue of the DDA for each factor. If we can express the factor DDA's concisely using determinations, we then have a concise axiomatization of the overall DDA, e.g.:

$$\forall x.Q(x) \equiv \{Q_1(x) \wedge Q_2(x)\}$$
$$F_1(x, y_1) \succ kQ_1(x)$$
$$F_2 x_2 \quad \succ kQ_2\, x$$

III. Version Space Updating

The simple-minded approach to updating the version space is to do forward resolution between the instance observation facts $D_i(a_i) \wedge kQ(a_i)$ and the disjuncts in the DDA, i.e. the candidate concept descriptions. Effectively, each C_j will resolve against an instance that contradicts it (with the help of the articulation theory). Thus, as more instances are observed, the DDA will shrink, retaining only those concept descriptions that are consistent with all the instances. Classification of a new instance a_i using an intermediate version space can be done by a resolution proof for the goals $Q(a_i)$ and $\neg Q(a_i)$ using the current DDA as the database. (Note that these processes are in general only semi-decidable.) We can regard Mitchell's candidate elimination procedure as a special-purpose deductive technique for use with a very limited class of premises. This suggests that we can achieve comparable efficiency properties by implementing Version Space deductions in a theorem-prover with meta-level control (Genesereth 1983), e.g. MRS (Russell 1985).

It is rather clumsy to be working with a huge disjunction of quantified equivalences. We would like to break up the DDA into a set of shorter axioms. This can be done using the boundary-set approach from Mitchell's candidate elimination algorithm, provided we have a finite concept language. We can define the sets $G(\mathcal{C})$ and $S(\mathcal{C})$, the most general and most specific boundaries of the version space; and for each node C_j in the space, the sets $Children(C_j)$ and $Parents(C_j)$ of its children and parents. We now give an alternative set of shorter axioms which together replace the Disjunctive Definability Axiom:

$$\bigvee_{C_g \in G(\mathcal{C})} (Q \leq C_g)$$

$$\bigvee_{C_s \in S(\mathcal{C})} (C_s \leq Q)$$

$$(Q < C_j) \Rightarrow \{\bigvee_{C_c \in Children(C_j)} (Q \leq C_c)\}$$

$$(C_j < Q) \Rightarrow \{\bigvee_{C_p \in Parents(C_j)} (C_p \leq Q)\}$$

For any particular example, the sets *Children* and *Parents* will be enumerated, leading to a finite set of first-order sentences.

Work is currently underway on an MRS implementation of the VS method using this axiomatization. It is important to note that the deductive process needs to be under some higher-level control, in order to handle the case of a collapse of the version space when the observations are inconsistent with the initial concept language bias (see below).

IV. Bias in Meta-DENDRAL

In this section, we show how the instance language bias for an operational learning system can be derived from background knowledge. In the Meta-DENDRAL system (Buchanan and Mitchell 1978; Mitchell 1978) the learning task is to generate rules giving the class of molecules for which a given mass-spectroscopic cleavage process will occur.

The instance description language consists of all possible molecular structures, specified as far as the topological connections between nodes; each node being an atom of a specified chemical element other than hydrogen, with specified numbers of hydrogen neighbors and unsaturated electrons. We thus have the determination

$$StructuralFormula(molecule, structure) \succ kBreaks(molecule, site)$$

where the *structure* argument corresponds to an instantiation of an abstract data type describing the chemical structure in the above manner. To the untrained observer of concept learning systems, an instance language bias such as this, when introduced implicitly, seems practically tautologous. To freshman chemistry students, a molecule *is* a structural formula. But this is to conceal a vast number of background relationships and centuries of scientific research.

Firstly, the instance description language is a deliberate approximation to the known situation, since it ignores the stereochemical properties and energy state of the molecule; the isotope of each atom; and the energy states of the nuclei. These aspects are generally considered irrelevant, on physical grounds, to the behavior of a molecule in a mass spectroscope, though for other purposes such as reaction-rate calculations or NMR they are highly relevant.

Secondly, some properties are ignored for atoms including those, such as identity and history, that we might ascribe to other objects. Few chemists worry about whether the atoms in a sample are known to their friends as Fred.

Thirdly, properties that are determined by aspects already taken into account may also be ignored. For example, the mass, valency, electronegativity, and orbital structure of each of the atoms are relevant to the mass spectroscopy process; yet they are omitted from the instance description because they are determined by the chemical element to which the atom belongs.

The following is a derivation of the instance language bias starting from basic chemical facts. We know on quantum-mechanical grounds that, for any atom a

$$OrbitalStructure(a, o) \succ ChemicalBehaviour(a, ba) \quad (1)$$

$$Element(a, e) \succ OrbitalStructure(a, o) \quad (2)$$

implying:

$$Element(a, e) \succ ChemicalBehaviour(a, ba) \quad (3)$$

since determinations on functional relations are transitive. We also have the following determinations for any molecule m:

$$BondTopology(m, t) \wedge BehaviorOfNodes(m, bn) \succ MolecularChemicalBehaviour(m, bm) \quad (4)$$

$$StructuralFormula(m, s) \succ BondTopology(m, t) \wedge NodeElements(m, n) \quad (5)$$

$$MolecularChemicalBehaviour(m, bm) \succ MassSpectroscopicBehaviour(m, bs) \quad (6)$$

$$MassSpectroscopicBehaviour(m, bs) \succ kBreaks(m, cs) \quad (7)$$

From (3), using the definitions of the predicates *NodeElements* and *BehaviourOfNodes* (omitted here), we can derive

$$NodeElements(m, n) \succ BehaviourOfNodes(m, bn) \quad (8)$$

which we can combine with (4) to give

$$BondTopology(m, t) \wedge NodeElements(m, n) \succ MolecularChemicalBehaviour(m, bm) \quad (9)$$

From (5), (9), (6) and (7) we have, again by transitivity, the instance language bias for Meta-Dendral given earlier:

$$StructuralFormula(molecule, structure) \succ Breaks(molecule, site) \quad (10)$$

The point of this section has not been to elucidate the intricacies of chemistry, but to show how in a "real-world" domain the factual content of part of the VS bias can be arrived at by a deduction from accepted premises representing background knowledge, and in particular to illustrate the use of determinations in expressing these premises. We can now (at least in part) automate the process of *setting up* a VS process.

V. Conclusions

- We showed how to represent in first-order logic the bias in the pure Version Space (VS) method, which is the most standard AI approach to concept learning from examples. The most important part of the bias is implicit in the choice of the instance and concept candidate description languages. A learning system can thus *derive* its own initial version space from its background knowledge. We gave an account of such a derivation for the Meta-DENDRAL system for learning cleavage rules in mass spectroscopy.
- We showed the important role of a form of first-order axiom, *determinations*, in the VS method's bias. We identified a substantive component of the bias in the choice of the *instance* description language.
- We showed how to represent (pure) VS updating as deduction in first-order logic. Using a general theorem-prover, we can therefore incorporate arbitrary first-order background knowledge into the concept learning process.

- Our declarative analysis of VS bias suggests how to extend the VS method to less structured learning situations. The learning agent can use determination-form knowledge to actively identify the relevant aspects of its inputs.

As designers of learning agents, instead of starting with an algorithm and some contrived inputs, we should instead examine what knowledge is typically available about the target concept, and then show how it may be used efficiently to construct plausible rules for concluding concept definitions, given examples. This more first-principles attitude is facilitated by a declarative approach.

We had difficulty declaratively formulating some other kinds of bias which are defined in terms of computational-resource-oriented bounds on data structures or syntactic properties of descriptions, e.g. limits on the sizes of VS boundary sets, and limits on negation or disjunction. The latter seems sometimes to represent real "semantic" knowledge, e.g. the vocabulary choice in LEX (Mitchell et al. 1983); exactly how is unclear. We suspect that possession of a good vocabulary is a *sine qua non* of inductive success.

Shift of Bias as Nonmonotonic Reasoning

Benjamin Grosof and Stuart Russell

I. Introduction

The Version Space method can now be implemented as a deductive process using the instance observations and a declaratively expressed 'bias'. In this part of the paper, we address the issue of *inductive leaps*, and the *shifts* of the biases underlying them, in concept learning. We begin by observing that, viewed declaratively, inductive leaps and shifts of bias are non-monotonic. We develop a perspective on shifts of bias in terms of *preferred beliefs*. We then show how to express several kinds of shifts of "version-space" bias, as deductions in a new, non-monotonic formalism of *prioritized defaults*, based on circumscription. In particular, we show how to express 1) moving to a different, e.g. less restrictive, concept language when confronted by inconsistency with the observations; and 2) the preference for more specific/general descriptions (definitions) of a concept.

II. Inductive Leaps and Shifts of Bias Are Non-Monotonic

If an agent is completely sure of its initial bias, no "inductive leap" is required to reach a definition for the target concept. The potential for retraction is essential to novelty in an inductive learning process. In other words, useful concept learning must be treated as *non-monotonic* inference. When we ascribe a declarative status to bias as something that the agent believes about the external world, then the agent's believed set of sentences in general evolves non-monotonically.

Since we have shown the pure VS method to be monotonic deduction, in what sense is it "inductive", in the sense of making inductive leaps? Our answer would be that in practice, the VS method instantiated with a particular initial version space is used as a sub-program: in advance it is not known whether that initial version space will be expressively adequate. *The potential for shift of bias, especially of concept language bias, is vital to a VS-style learning program's inductive character.* We will use a non-monotonic formalism to study shift of bias in a declarative framework.

Several researchers have identified the automation of the shift of concept language bias, e.g. as in the VS method, as a prime outstanding problem in machine learning.

> Methods by which a program could automatically detect and repair deficiencies in its generalization language would represent a significant advance in this field"(Mitchell 1982, section 6.1)

> Automatic provision or modification of the description space is the most urgent open problem facing automatic learning."(Bundy et al 1985, section 7.3)

One common strategy for a learning agent, e.g. in the STABB system for shifting concept language bias (Utgoff 1984, 1986), and in the Meta-DENDRAL system for learning cleavage rules in mass spectroscopy (Mitchell 1978), is to start with a strong bias, which aids focus and provides a guide to action, and then relax when needful to a weaker bias. This shift is triggered by falling below some acceptability threshold on an evaluation criterion for the working theory. Often the criterion is an unacceptable degree of inconsistency with the observed instances. Note that new information or pragmatic constraints may also lead the agent to *strengthen* its bias.

III. Preferred Beliefs

At bottom of the declarative impulse is the desire to characterize as stably as possible the justifying basis for the agent's beliefs. In this light, to the extent that bias is formulated in such a way that it shifts, then to that extent its formulation fails to be satisfactorily deep. We thus look for a way to formulate deep bias as a set of premises which are highly stable, yet which suffice to justify shifty bias and shifty belief. The notion of a *default* in non-monotonic logical formalisms offers the form of exactly such a stable premise. If we represent the trigger condition for retracting bias as strict logical inconsistency of the bias with the instance observations (as in STABB), then we can neatly use a nonmonotonic formalism.

We can view a default as a *preferred belief*. That is, we prefer to believe the default if it is consistent with our other, non-retractible, beliefs. If the non-retractible beliefs contradict a default, it is retracted. In general, however, defaults may *conflict* with each other. It is useful, therefore, to express preferences, a.k.a. *priorities*, between defaults, as well. In cases of conflict, the agent prefers to believe the default with higher priority. If neither has higher priority, then the agent believes merely that one must be false without saying which. We can regard non-retractible beliefs as having infinite priority.

Our approach to shifts of bias, then, is to express them as the results of retracting different concept language biases, represented as defaults. Stronger and weaker retractible biases co-exist: when both are consistent, the stronger ones hide the weaker. When the stronger become inconsistent before the weaker, we see a dynamic relaxation or weakening of bias.

For now, we will treat instance observations as non-retractible. However, we might make them have less than infinite priority if we wished to deal with noise or inaccuracy in observations, or to tolerate a degree of inconsistency with the observations rather than reject elegant inductive hypotheses.

IV. Prioritized Defaults

Several different nonmonotonic formalisms can express defaults, more or less. Of these, circumscription (McCarthy 1986; Lifschitz 1986) has a number of advantages. It is relatively well-understood mathematically, especially semantically, and can express priorities gracefully. The formalism we employ to describe biases is a meta-language for *specifying* circumscriptive theories.

In our language of *prioritized defaults*, there are four kinds of axioms. A *non-monotonic theory* $\mathcal{NMCLOSURE}(\mathcal{A})$ is defined as the closure under non-monotonic entailment of a set of axioms $\mathcal{A}$.

Base axioms are just non-retractible, first-order axioms:

$bird(Tweety)$ $ostrich(Joe)$ $\neg flies(Hulk)$

$\forall x.ostrich(x) \Rightarrow bird(x)$

Default axioms have the form of labelled first-order formulas. They express preferred, but retractible, beliefs. Default axioms may take the form of open, as well as closed, formulas. An open formula is in effect a schema expressing the collection of defaults corresponding to the instantiations of the schema.

$(d_1 :) \quad :> bird(x) \Rightarrow flies(x)$

$(d_2 :) \quad :> \ ostrich(x) \Rightarrow \neg flies(x)$

Prioritization axioms express priorities between defaults. One default having higher priority than a second means that in case of conflict between the two, the first rather than the second will be entailed by the non-monotonic theory. Thus the following axiom says that the ostrich default is preferred to the bird default.

$$\mathcal{PREFER}(d_2, d_1)$$

This corresponds to inheritance hierarchies, for example, where the slot value (flying) for a more specific class (ostriches) takes precedence over the slot value for a more general class (birds).

Fixture axioms express constraints on the scope of the defaults' non-monotonic effects. They declare that the truth of certain formulas can only be entailed monotonically.

$$\mathcal{FIX}(bird(x))$$

Taking the above set of axioms as $\mathcal{A}$, then the non-monotonic theory $\mathcal{NMCLOSURE}(\mathcal{A})$ contains $flies(Tweety)$, by default. Both default axioms apply to *Joe*, since he is both an ostrich and a bird, but they conflict. The prioritization axiom resolves the conflict. It tells us to prefer the ostrich default. Thus $\mathcal{NMCLOSURE}(\mathcal{A})$ entails $\neg flies(Joe)$. The fixture axiom comes into play by preventing the conclusion that *Hulk* is not a bird, which the consistency of the bird default for the instance *Hulk* seems to tell us to make.

V. Changing Bias Via Non-Monotonic Inference

Now we show how to use our logic of prioritized defaults to describe an agent that starts with a strong concept language bias and shifts so as to weaken it in the face of inconsistency with observations. Space limits us to a simple example; we adapt one from (Mitchell 1982). The agent has an initial bias and two weaker, back-up biases, the weakest being just the instance language bias itself.

The available observations describe each instance as a feature vector of color (red or blue), size (large or small), and shape (circle or triangle). The instance language bias says that the target concept is determined by these three features taken together. The initial concept language bias $\mathcal{CL}_1$ is that the concept is equivalent to a *conjunction* of a *Color* atom and a *Size* atom. A second, fall-back bias $\mathcal{CL}_2$ is that the concept is equivalent to a *conjunction* of a *Color* atom, a *Size* atom, and a *Shape* atom. The instance language bias $\mathcal{IL}$ and the observational updates $\mathcal{OU}^i$ are expressed as base axioms. The concept language biases are expressed as defaults. In addition, we assume the Unique Names Assumption (so $Red \neq Blue$ etc.).

$\mathcal{IL}$:

$$\{ \forall x.\exists!y.Color(x,y) \}$$
$$\{ \forall x.\exists!y.Size(x,y) \}$$
$$\{ \forall x.\exists!y.Shape(x,y) \}$$
$$\{ \forall xy.Color(x,y) \Rightarrow \{(y = Red) \vee (y = Blue)\} \}$$
$$\{ \forall xy.Size(x,y) \Rightarrow \{(y = Large) \vee (y = Small)\} \}$$
$$\{ \forall xy.Shape(x,y) \Rightarrow \{(y = Circle) \vee (y = Triangle)\} \}$$
$$\{Color(x,y_1) \wedge Size(x,y_2) \wedge Shape(x,y_3) \succ kQ(x)\}$$

$\mathcal{CL}_1$:

$$\{ Color(x,y) \succ kQF_1(x) \} \wedge$$
$$\{ Size(x,y) \succ kQF_2(x) \} \wedge$$
$$\{ \forall x.Q(x) \equiv \{QF_1(x) \wedge QF_2(x)\} \}$$

$\mathcal{CL}_2$:

$$\{ Color(x,y) \succ kQFF_1(x) \} \wedge$$
$$\{ Size(x,y) \succ kQFF_2(x) \} \wedge$$
$$\{ Shape(x,y) \succ kQFF_3(x) \} \wedge$$
$$\{ \forall x.Q(x) \equiv \{QFF_1(x) \wedge QFF_2(x) \wedge QFF_3(x) \}$$

$\mathcal{OU}^1$: $\quad Q(a_1) \wedge Q(a_2) \wedge \neg Q(a_3) \wedge$
$Color(a_1, Red) \wedge Size(a_1, Large) \wedge Shape(a_1, Circle) \wedge$
$Color(a_2, Red) \wedge Size(a_2, Small) \wedge Shape(a_2, Circle) \wedge$
$Color(a_3, Blue) \wedge Size(a_3, Small) \wedge Shape(a_3, Triangle)$

$\mathcal{OU}^2$: $\quad \neg Q(a_4) \wedge$
$Color(a_4, Red) \wedge Size(a_4, Large) \wedge Shape(a_4, Triangle)$

$\mathcal{OU}^3$: $\quad Q(a_5) \wedge$
$Color(a_5, Blue) \wedge Size(a_5, Small) \wedge Shape(a_5, Circle)$

$\mathcal{OU}^4$: $\quad Q(a_6) \wedge$
$Color(a_6, Blue) \wedge Size(a_6, Large) \wedge Shape(a_6, Triangle)$

The agent's starting axioms $\mathcal{A}^0$ are:
Base axioms: (the instance language bias) $\mathcal{IL}$.
Default axioms: $(d_3 :) \ :> \mathcal{CL}_1 \quad (d_4 :) \ :> \mathcal{CL}_2$
Prioritization axioms: none.
Fixture axioms:
$\mathcal{FIX}(Color(x))$
$\mathcal{FIX}(Size(x))$
$\mathcal{FIX}(Shape(x))$

Let $\mathcal{A}^m$ denote $\mathcal{A}^0 \wedge \mathcal{OU}^1 \wedge \ldots \wedge \mathcal{OU}^m$, i.e. the agent's axioms after the m^{th} observational update. The agent's *working inductive theory* $\mathcal{WIT}^m$ is then equal to $\mathcal{NMCLOSURE}(\mathcal{A}^m)$.

In $\mathcal{WIT}^1$, i.e. after the first update, the initial concept language bias is uncontradicted, so it holds by default. That is, $\mathcal{CL}_1$ is consistent (and thus so is the weaker $\mathcal{CL}_2$) and thus holds. The corresponding version space has been refined to a single candidate; the agent's working inductive hypothesis is that the concept is that the color is red.

$$\mathcal{WIT}^1 \models \mathcal{CL}_1 \wedge \mathcal{CL}_2 \wedge \{\forall x.Q(x) \equiv Color(x, Red)\}$$

The second update, however, contradicts this hypothesis and the initial concept language bias. Thus in $\mathcal{WIT}^2$, $\mathcal{CL}_1$ is retracted. However, $\mathcal{CL}_2$ is still consistent and thus holds: the agent shifts to the fall-back. The corresponding version space has two members.

$$\mathcal{WIT}^2 \models \neg\mathcal{CL}_1 \wedge \mathcal{CL}_2 \wedge \{ \{\forall x.Q(x) \equiv Shape(x, Circle)\} \vee \{\forall x.Q(x) \equiv \{Color(x, Red) \wedge Shape(x, Circle)\}\} \}$$

After the third update, the version space is again refined to a single candidate:

$$\mathcal{WIT}^3 \models \neg\mathcal{CL}_1 \wedge \mathcal{CL}_2 \wedge \{\forall x.Q(x) \equiv Shape(x, Circle)\}$$

However, the fourth update contradicts this hypothesis, i.e. even the fall-back bias. Thus in $\mathcal{WIT}^4$ the agent retracts the fall-back, i.e. $\mathcal{CL}_2$ as well as $\mathcal{CL}_1$.

$$\mathcal{WIT}^4 \models \neg\mathcal{CL}_1 \wedge \neg\mathcal{CL}_2$$

The agent is then left with a *vacuous* concept language bias which does not go beyond the instance language bias. The version space consists of all subsets of the "describable instances" consistent with the observations. Here, there are four of these, corresponding to the possible combinations of classifications for blue large circles and red small triangles.

In addition to the concept and instance language bias, we can also represent some types of preference bias, including maximal specificity/generality bias, i.e., the preference for the most specific/general among concept descriptions that are consistent with all observations. This corresponds to minimizing/maximizing the extension of the goal concept Q, and hence to the following default axioms:

Maximal Specificity Axiom : $(d_5 :) \ :> \neg Q(x)$
Maximal Generality Axiom : $(d_6 :) \ :> Q(x)$

In order to express the fact that an agent employs (say) maximal generality bias, we just include the Maximal Generality Axiom in the agent's axioms, bearing in mind that maximal generality as a preference may *conflict* with other defaults.

In our example above, intuitively what we would like is to apply (say) maximal generality only *after* attempting to adopt the default concept language biases. To express this formally, we need to ensure that the Maximal Generality Axiom has lower priority than the defaults corresponding to the retractible concept language biases, e.g. by including $\mathcal{PREFER}(d_4, d_6)$ in the agent's axioms. Thus in the above example, after the second update the agent would adopt the more general of the two candidates above as its working hypothesis:

$$\mathcal{WIT}^2_{MG} \models \neg\mathcal{CL}_1 \wedge \mathcal{CL}_2 \wedge \{\, \forall x.Q(x) \equiv Shape(x, Circle) \,\}$$

VI. Conclusions

In this part of the paper we attempted to show how bias shift could be dealt with in our declarative framework.

- We observed that from a declarative point of view, inductive leaps, and shifts of the biases which justify them, are non-monotonic.
- We showed how to declaratively represent shifts of bias, i.e. *"shifty"* bias, using a new language of prioritized defaults, based on circumscription, for "version-space"-type concept language bias.
- We showed that the maximal specificity and maximal generality biases are formulable quite simply: as negative and positive default belief, respectively, about the target concept. Thus we have a logical, semantic formulation for these preference-type biases which Dietterich (1986) listed as "syntactic" and "symbol-level".

Thus we can view inference that is non-deductive at the level of first-order logic, i.e. that is inductive, as deduction in another "knowledge level" associated with non-monotonic beliefs. This allows the use of arbitrary-form non-monotonic "background knowledge". The non-monotonic viewpoint suggests formulating shifts among base-level bias sentences as defeasible "shifty" bias sentences. How to efficiently implement such inference is an open question which we are currently investigating. See (Grosof 1987) for a discussion of implementation issues.

Our declarative formulation also poses the question of the source of the defaults and preferences among beliefs which are the "shifty" premise biases of inductively leaping agents. In our view, the justification of inductive leaps arises not just from probabilistic beliefs, but also from the pressure to decide, i.e. the need to act as if one knows. Because the information about which the agent is quite confident is incomplete, it requires an additional basis to decide how to act. (Since the agent acts some way, we can declaratively ascribe a working hypothesis to its decision principle.) A second reason why bias is needed is that the agent has computational limits on how many inductive hypotheses it can consider, and in what sequence. Thus we expect that the justification for bias is largely decision-theoretic, based both on probabilities and utilities.

We are currently investigating, in addition to implementation issues, how to extend our approach to several other aspects of inductive theory formation, including 1) tolerance for noise and errors; 2) preferences for more likely hypotheses; 3) preferences for simpler hypotheses, as in Occam's Razor; and 4) the decision-theoretic basis for bias preferences.

VII. Acknowledgements

We would particularly like to thank Michael Genesereth and Vladimir Lifschitz for their interest, criticism, and technical help. Thanks also to Devika Subramanian, Haym Hirsh, Thomas Dietterich, Bruce Buchanan, David Wilkins, and the participants in the GRAIL, MUGS, and Nonmonotonic Reasoning seminars at Stanford for valuable discussions.

References

[1] Buchanan, B. G., and Mitchell, T. M., "Model-directed Learning of Production Rules". In Waterman, D. A., and Hayes-Roth, F., (Eds.) *Pattern-directed Inference Systems*. New York: Academic Press, 1978.

[2] Bundy, Alan, Silver, Bernard, and Plummer, Dave, "An Analytical Comparison of Some Rule-Learning Programs". In *AI Journal*, Vol. 27, 1985.

[3] Buntine, W., "Generalized Subsumption and its Application to Induction and Redundancy". In *Proceedings of ECAI-86*, Brighton, UK, 1986.

[4] Davies, Todd. "Analogy". Informal Note CSLI-IN-85-4, CSLI, Stanford, 1985.

[5] Davies, Todd R. and Russell, Stuart J., "A Logical Approach to Reasoning by Analogy". In *Proceedings of IJCAI-87*, Milan, Italy, 1987.

[6] Dietterich, Thomas G., "Learning at the Knowledge Level". In *Machine Learning*, Vol. 1, No. 3, 1986.

[7] Genesereth, M. R., "An Overview of Meta-Level Architecture". In *Proceedings of AAAI-83*, pp. 119-124, 1983.

[8] Grosof, Benjamin N.. *Non-Monotonic Theories: Structure, Inference, and Applications* (working title). Ph. D. thesis (in preparation), Stanford University, 1987.

[9] Lifschitz, Vladimir, "Pointwise Circumscription". In *Proceedings of AAAI-86*, pp. 406-410, 1986.

[10] McCarthy, John, "Applications of Circumscription to Formalizing Common-Sense Knowledge". In *Artificial Intelligence*, Vol. 28, No. 1, pp. 89-116, Feb. 1986.

[11] Michalski, R. S., "A Theory and Methodology of Inductive Learning." *Artificial Intelligence*, Vol. 20, No. 2, 1983.

[12] Mill, J. S., *System of Logic* (first published 1843). Book III Ch XX 'Of Analogy' in Vol. VIII of *Collected Works of John Stuart Mill*. University of Toronto Press, 1973.

[13] Mitchell, Tom M., *Version Spaces: an Approach to Concept Learning*. Ph.D. thesis, Stanford University, 1978.

[14] Mitchell, Tom M. "The Need for Biases in Learning Generalizations". Rutgers University TR CBM-TR-117, 1980.

[15] Mitchell, Tom M., "Generalization as Search". In *Artificial Intelligence*, Vol. 18, No. 2, pp. 203-226, March 1982.

[16] Mitchell, T. M., Utgoff, P., and Banerji, R., "Learning by Experimentation: Acquiring and Refining Problem-Solving Heuristics". In Carbonell, J. G., Michalski, R., and Mitchell T., (eds.) *Machine Learning: an Artificial Intelligence Approach*. Palo Alto, CA: Tioga Press., 1983.

[17] Quinlan, J. R., "Learning Efficient Classification Procedures and their Application to Chess End Games". In Carbonell, J. G., Michalski, R., and Mitchell T., (Eds.) *Machine Learning: an Artificial Intelligence Approach*. Palo Alto, CA: Tioga Press., 1983.

[18] Rendell, Larry, "A General Framework for Induction and a Study of Selective Induction. *Machine Learning, 1*, 1986.

[19] Russell, Stuart J., *The Compleat Guide to MRS*. Technical Report No. STAN-CS-85-1080, Stanford University, 1985.

[20] Russell, Stuart J., "Preliminary Steps Toward the Automation of Induction." In *Proceedings of AAAI-86*, Philadelphia, PA, 1986.

[21] Russell, Stuart J., *Analogical and Inductive Reasoning*. Ph. D. thesis, Stanford University, Dec. 1986.

[22] Subramanian, Devika, and Feigenbaum, Joan, "Factorization in Experiment Generation'. In *Proceedings of AAAI-86*, pp. 518-522, 1986.

[23] Utgoff, P. E., *Shift of Bias for Inductive Concept Learning*. Ph.D. thesis, Rutgers University, 1984.