Introducing New Predicates to Model Scientific Revolution

Charles Ling
Department of Computer Science
University of Western Ontario
London, Ontario, Canada N6A 5B7
Email: ling@csd.uwo.ca
(519)-661-3341 (o), (519)-661-3515 (fax)

Abstract

The notion of necessary new terms (predicates) is proposed. It is shown that necessary new predicates in first-order logic must be directly, recursively defined. We present a first-order inductive learning algorithm that introduces new necessary predicates to model scientific revolution in which a new language is adopted. We demonstrate that our learning system can learn a genetic theory with theoretical terms which, *after* being induced by our system, can be interpreted as either types of genetic properties (dominant or recessive) or genes, depending on the representation of the hypotheses of the same theoretical terms.

To appear in *International Studies in the Philosophy of Science*, 9(2), 1995. (This is not the final version; comments are very welcome.)

1 Introduction

Understanding and automating the processes of scientific discovery is one of the the most ambitious goals of philosophy of science, modern cognitive science, and artificial intelligence. Many aspects of scientific discovery, such as problem solving, the representation and use of large amount of knowledge, wisdom, creativity, and so on, are being explored in recently years (cf. (Langley, Simon, Bradshaw, & Zytkow, 1987; Simon, 1992)). In this paper, we study a computation model of one aspect of scientific discovery — we use machine learning techniques of inferring theory with *new predicates* to model the logical aspect of scientific revolution in which a new language is adopted.

Following Shapiro (1981), Glymour (1985), Osherson and Weinstein (1986), Gaifman, Osherson, and Weinstein (1990), we study the process of scientific investigation in a framework of formal, first-order logic. A scientist is a learning program P in our framework. The domain under investigation by scientists is formalized as mathematical structures, unknown to the scientists and to P. Observations and experiments performed by scientists are ground relations in a first-order structure, available to P. The purpose of scientific discovery by scientists is defined as scientific theorizing, i.e., to *identify the unknown structure* via theory of the structure. That is, P infers, from observations, a set T of first-order logic statements such that T is true in the structure, and T "characterizes" the structure; for example, T derives or "explains" all observations[1].

Assume that a first-order language L used in P contains two disjoint sets, V_o and V_b, of non-logical symbols (constants, functions and predicates). V_o and V_b are called *observable terms*. The structure S to be identified interprets only observable terms — any ground sentence of observable terms corresponds to observations and experiments in the domain of investigation, and therefore, the truth or falsity of it can be observed, obtained, and verified.

[1]The degree of "characterization" of the structure may vary. P, as a "profound" scientist, may produce axioms of the structure that characterize the structure up to isomorphism; or may produce all axioms of the structure (Osherson & Weinstein, 1986, 1989; Gaifman et al., 1990). On the other hand, P, as a "practical" scientist, may just produce enough axioms whose deductive closure contains all observations (ground sentences) received. In this case, T can derive or "explain" all observations. This is the model we study in this paper, since this seems the only feasible criterion for applying machine learning techniques (such as (Shapiro, 1981; Muggleton & Feng, 1990; Ling, 1989)) to model scientific discovery.

The theory T_b of V_b is called the *background knowledge*, and is assumed to be available to P before learning, similar to the knowledge known to scientists before investigation. P receives ground sentences of V_o as new observations that the current background knowledge cannot derive; just as scientists receive new observations and perform new experiments during their investigation. Sentences of V_o true in S form a set E_+ of *positive examples*; while false in S form a set E_- of *negative examples*. During scientific investigation, scientists try to discover a new theory that can "explain" the new observations. In our logic framework, this corresponds to inferring a *target theory* T_o by P that can derive E_+ but not E_-. Therefore, scientific investigation can be modeled in our framework as to find a program P that

> given E_+ and E_- as a (possibly infinite) series of ground sentences in V_o, given V_b and T_b as background knowledge, infer in the limit[2] (Gold, 1967) T_o of V_o and V_b, such that
>
> $T_b \not\models E_+$ (Necessity of Investigation)
> $T_b, T_o \models E_+$ (Logic Sufficiency)
> $T_b, T_o \not\models E_-$ (Logic Consistency)

Assume that scientists revise a theory only if an anomaly appears. In our framework, therefore, P revises T_o only if the current T_o contradicts the incoming positive and negative examples. More specifically, if some incoming $e+ \in E_+$ or $e- \in E_-$ such that

$T_o, T_b \not\models e+$; or
$T_o, T_b \models e-$

then T_o should be revised or reconstructed to be able to prove $e+$ but not $e-$. Several machine learning programs (cf. (Shapiro, 1981; Ling, 1991; Muggleton & Buntine, 1988; Ling, 1989; Muggleton & Feng, 1990; Quinlan, 1990; Aha, Lapointe, Ling, & Matwin, 1994a, 1994b)) have been developed that infer first-order theories from examples based on fixed language. See Section 3.2 for more details.

However, sometimes there exists no T_o, or it is not possible for P to infer T_o for scientific theorizing. In these cases, a certain anomaly (see later) occurs during theory revision, and introducing theoretical terms can be of help. We assume that the set V_t of theoretical

[2] That is, after receiving some finite number of positive and negative examples in V_o, P produces T_o whose deductive closure contains all positive examples but none of the negative examples.

terms is empty at the beginning of learning, just as scientists start querying the domain using observable language only. However, if theorizing with observable terms fails, P may hypothesize theoretical terms V_t and their postulates, and use them to *assist* the theorization of the target theory. Therefore, the modelling of scientific investigation can be expanded to include the introduction of theoretical terms:

infer V_t, T_t, and T_o, such that
$\nexists T'_o$ such that $T_b, T'_o \models E_+$ (Necessity of Theoretical Terms)
$T_b, T_o, T_t \models E_+$ (Sufficiency of Theoretical Terms)
$T_b, T_o, T_t \not\models E_-$ (Logic Consistency)

Note that, just as theoretical terms are not observable to scientists, no training examples of V_t will be available to P.

In this paper, we mainly study computationally feasible modelling of the discovery, rather than focus on theoretical issues (such as (Gaifman et al., 1990)). To simulate a practical scientist using machine learning programs, we require that both T_o and T_t — the result of scientific investigation — be *finite*[3] or *small* according to some empirical criterion. Under these requirements, there are cases where new terms are absolutely necessary. As Hempel (1963) pointed out, additional theoretical terms may open the way to finite axiomatization of scientific theory. That is, a structure S may be finitely axiomatizable only if some additional vocabulary is supplied. In these cases, P fails to identify the structure finitely based on the fixed language (V_o and V_b); the "anomaly" that occurs during theory revision is that the theory T_o would grow *without a bound* when more and more positive and negative examples of V_o are received. If P can hypothesize necessary new terms and their postulates, what was not learnable can become learnable. Therefore, inferring theory with new terms is a very powerful tool. Learning new terms also provides some solutions to the problems of the representation, and change of the representation.

[3]This seems to make our paradigm representation dependent. For example, an infinite but recursive axiomatizable first-order theory may be finitely axiomatizable in second-order or other logic. While we do not have an answer to this philosophical question of what is the appropriate language to model scientific investigation, we maintain that scientists infer theories in a certain representation under a certain manageable scale. First-order logic and finiteness, studied in this paper, is just one such possibilities.

Kuhn (1970) describes scientific discovery as a sequence of *normal science, crisis, revolution,* and *new normal science.* The logical aspects of these four processes can map into our logic framework at a high level. In "normal science", scientists revise theories for resolving anomalies using a fixed paradigm and a fixed language. Thus, normal science corresponds to theory revision based on fixed language (i.e., without new terms), which can be modeled by first-order inductive learning algorithms based on fixed language (see Section 3.2). From time to time, however, "crisis" occurs when scientists fail to cope with the anomalies observed. This is because the language they use is not adequate or powerful enough to construct or revise the theory for the anomalies; only a total rethinking of the domain will suffice. This corresponds to failure in revising the theory based on a fixed language; the theory tends to grow infinitely or out of a reasonable size during theory revision. When crisis occurs, a "revolution" is needed in scientific investigation. The need of a new and more powerful language (e.g., theoretical terms) is one of the main reasons for scientific revolution; at least every scientific revolution involves a paradigm shift, in which a new language is adopted. The new terms introduced can be purely hypothetical — scientists use them to assist the scientific theorizing of observable structures. They can also indicate the hidden features in the domain. Therefore, scientific revolution corresponds to theorizing in which a set V_t of new terms (new constants, new predicates and new functions) is adopted, and a theory T_t of new terms is hypothesized. Since we assume that there is no way to verify the truth or falsity of sentences with theoretical terms in scientific investigation, no positive or negative examples of new terms are available to P in our computational model either. After revolution, a "new normal science" starts in which the scientists are armed with a more powerful language for the never-ending scientific investigation.

The actual scientific discovery and revolution is such a complicated process that we can model only a small fraction of it in our logic framework. More specifically, inferring theory with new terms is not necessarily identical to every actual scientific revolution in the history of science. This is because, first, some of the "new terms" introduced during scientific revolution may actually be observable; they were just not included in the language before revolution. To decide which observable terms are relevant in a domain of investigation, and how to represent them is not trivial at all. For example, are the temperature, location, colour of flowers and so on relevant to the shape of leaves in offspring? We do not study how to choose the appropriate observable language in our framework. Second, some scientific revolution represents a *radical* revision of the theory in the original language. During normal science, the theory usually is revised in small, incremental steps according to some greedy, conservative policy (such as minimal revision). Scientific revolution may correspond to a radical and non-conservative revision of the theory in our logic framework. See Section 5 for more discussions on the limitations of our system. To design feasible models of scientific revolution, we study the inference of first-order Horn-clause theory with new predicates. A *Horn clause* is a first-order logic expression in the form of

$$p \leftarrow true.$$

or simply,

$$p.$$

and

$$p \leftarrow l_1, ..., l_n,$$

where p and l_i are positive literals (atoms). All variables appearing in the clauses are universally quantified. The literal p is often called the *consequence* of the clause, while l_i the *condition* or *antecedent* of the clause. The clause $p \leftarrow true$ or p is interpreted as p is true universally, while the clause $p \leftarrow l_1, ..., l_n$ is interpreted as p is true if all of the $l_1, ..., l_n$ are true. Note that such Horn clauses are in the form of production rules that have been regarded as the way expert knowledge is represented in humans (cf. (Simon, 1992)).

Because of the existence of an efficient theorem prover, i.e., Prolog, learning Horn-clause theory has become a rapidly growing research area. The problem is often called *Inductive Logic Programming* (or ILP[4] for short). See Section 3.2 for more details.

Many aspects of scientific investigation have been studied; see, for example, (Lai, 1991; Glymour & Kelly, 1989; Osherson & Weinstein, 1990; Glymour, 1985). A great deal of previous work on modelling scientific discovery has been done. See a recent book (Langley et al., 1987) for an extensive study and a collection of such systems. BACON, for example, discovers (numerical) laws from data. Some of them (cf. (Langley et al., 1987; Pagallo & Haussler, 1990; Rendell & Seshu, 1990; Matheus & Rendell, 1989; Thagard, 1992; Ling & Marinov, 1993, 1994)) also introduce new terms. For example, BACON (Langley et al., 1987) infers a set of numerical equations with new propositions (called intrinsic new terms) for numerical combinations of the existing ones. Little work has been done on the logical

[4] cf. Proceedings of the first (91), second (92) and third (93) International Workshop on Inductive Logic Programming).

aspects of scientific revolution which involves *necessary* new predicates.

2 An Example: Learning to Predict Colour

Imagine, in a science fiction, a scientist is investigating the genetic properties of certain plants. In particular, he tries to discover the theory that can predict and explain the colour of flowers in offspring. He starts with two initial groups of plants, one with red flowers and the other with yellow flowers. Thus, his language L contains the following[5]:

- constants r and y: for the red and yellow colour respectively;
- constant a and b: for any flower in the red and yellow group respectively;
- predicate $colour(X, Y)$: the flower of X can have colour Y;
- function $ch(X, Y)$: mapping from flowers X and Y to one flower of the next generation via breeding X and Y.

The scientist cross-fertilizes the plants in different ways, and observes the colour of flowers in generations. The experiments and observations are therefore positive examples[6] represented in his language shown in Table 1.

The scientist first attempts to infer the theory of the colour based on the given language. Using inductive learning programs based on fixed language (cf. Section 3.2), he may discover first-order Horn clauses such as:

$$colour(ch(a, X), r).$$
$$colour(ch(X, Y), r) \leftarrow colour(X, r), colour(Y, r).$$
$$colour(ch(b, b), y).$$
$$colour(ch(X, Y), y) \leftarrow colour(X, y), colour(Y, y).$$

The first sentence is a unit clause and can be interpreted as "breeding any plant with the initial red-flower plant can always have red flowers". The last sentence is a Horn clause and

[5]Assume that he has already eliminated other irrelevant factors and observations (such as temperature).

[6]Only positive examples are shown here. Negative examples are the colour of flowers that never occur in those generations. For example, $colour(ch(a, a), y)$ is a negative example.

% The original groups	$10 : colour(ch(b, ch(a, b)), r).$
$1 : colour(a, r).$ %	$11 : colour(ch(b, ch(a, b)), y).$
$2 : colour(b, y).$	$12 : colour(ch(b, ch(b, b)), y).$
% original $\times$ original = 1st generations	% 1st $\times$ 1st
$3 : colour(ch(a, a), r).$	$13 : colour(ch(ch(a, a), ch(a, a)), r).$
$4 : colour(ch(a, b), r).$	$14 : colour(ch(ch(a, a), ch(a, b)), r).$
$5 : colour(ch(b, b), y).$	$15 : colour(ch(ch(a, a), ch(b, b)), r).$
% original $\times$ 1st	$16 : colour(ch(ch(a, b), ch(a, b)), r).$
$6 : colour(ch(a, ch(a, a)), r).$	$17 : colour(ch(ch(a, b), ch(a, b)), y).$
$7 : colour(ch(a, ch(a, b)), r).$	$18 : colour(ch(ch(a, b), ch(a, a)), r).$
$8 : colour(ch(a, ch(b, b)), r).$	$19 : colour(ch(ch(a, b), ch(b, b)), r).$
$9 : colour(ch(b, ch(a, a)), r).$	$20 : colour(ch(ch(a, b), ch(b, b)), y).$

Table 1: Observations represented as (numbered) positive examples.

can be interpreted as an IF-THEN production rule as "if both Y and Y can have yellow flowers then the offsprings of X and Y can have yellow flowers".

However, the scientist fails to discover *all* the rules for predicting the yellow colour. The following clause he might have considered is, in fact, false:

$$colour(ch(X, Y), y) \leftarrow colour(X, r), colour(Y, y).$$

This is because although in some generations red and yellow can produce yellow, but in other generations they produce all red (i.e., no yellow colour)! Thus, the clause is not universally true for all generations. Intuitively, either infinitely many clauses are needed since each of these rules can only predict some specific generations, or a clause with an indefinitely long body is required for tracking the colour of all upper generations. That is, there is no finite set of Horn clauses for predicting the colour of flowers from *upper generations*[7]. A revolution, in which theoretical terms will be introduced, seems necessary

[7]If further experiments are allowed, a finite theory does exist without any new terms, as also shown in (Shen & Simon, 1993). For example, the following theory proves all positive examples and is semantically correct:

$$colour(ch(X, Y), r) \leftarrow colour(ch(X, X), r).$$
$$colour(ch(X, Y), r) \leftarrow colour(ch(Y, Y), r).$$
$$colour(ch(X, Y), y) \leftarrow colour(ch(X, X)y), colour(ch(Y, Y), y).$$

However, determining the colour of any flowers $ch(X, Y)$ depends on further experiments on to him. (Example to be continued.)

3 Learning Theories with Necessary New Predicates

From the machine learning point of view, inductive learning with new terms is much more difficult than learning based on fixed language. The major obstacle is that no positive or negative examples of the new terms are directly available, and no queries about them may be made, either, because new terms are not observable, and they are internal to the learning system. Thus, Shapiro's MIS system with the backtracing algorithm (cf. Section 3.2) can not apply. In addition, the constraint on "correct" new terms is very weak; as long as they serve the purpose of deriving target predicates, they are "correct" in logic and may take any form as hypotheses.

We discuss the general strategy of learning with new terms in the following subsection, and each of its components in detail in the later subsections.

3.1 The General Strategy

The general strategy of scientific theorizing, as we have discussed in the Introduction, is to infer a finite or small theory with possible assistance of new predicates. The inference is failure-driven — new terms will be introduced only when the theory is not learnable without them, and the theory will be revised only when contradiction occurs with incoming data. Therefore, the general strategy can be described at a high level as follows: an algorithm of learning first-order theory based on fixed language is applied first, assuming the theory is finitely learnable without new terms (i.e., normal science). If the language is not powerful enough for theorizing the incoming examples, theory revision and construction would fail (i.e., crisis) as the resulting theory grows indefinitely. In this case, an *Invention Operator* (see Section 3.3) is applied to the current theory, which introduces new predicates (i.e., revolution). A set of positive examples of the new predicates is then obtained by *abduction* — these examples must be true in order to derive the positive examples of the target

$ch(X, X), ch(Y, Y)$. Besides, such a theory does not have much explanatory power.

observations. Learning with fixed language is applied to those examples, inferring the hypotheses of the new predicate (i.e., new normal science). As we will discuss in Section 3.4, the hypotheses of the necessary new predicate must be in the format of direct recursion. The induction of recursive definitions of new predicates is a key generalization step that replaces a potentially infinitely large set of instances of the new predicates by a finite or small number of recursive definitions. The algorithm of learning theory with new predicates is outlined in Figure 1.

```
T = T_b
loop
repeat
   read examples of observable predicates in V_o
   E+ = positive examples; E- = negative examples
   infer theory T based on the current language
                                   % Normal Science
until (T |= E+, T |/= E-) or T grows without a bound
if T grows without a bound
                                   % Crisis (Normal Science failed)
   then apply the Invention Operator on T
                                   % Revolution takes place
         infer by abduction examples of the new predicates
         infer theory based on the expanded language
                                   % New Normal Science
output T                           % Scientific Investigation Succeeds
```

Figure 1: A computational model for scientific discovery.

In the following subsections, we discuss major components of the general strategy in detail.

3.2 Learning Horn-Clause Theory Based on Fixed Language

Several inductive learning algorithms based on fixed language have been implemented. The first of the kind is Shapiro's seminal work on the Model Inference System (MIS) (Shapiro, 1981). More recently, other ILP systems have emerged, such as data-driven (bottom-up) learning system SIM (Ling, 1991), inverse-resolution-based CIGOL (Muggleton & Buntine, 1988; Muggleton, 1990), LGG-based (least general generalization) GOLEM (Muggleton & Feng, 1990), information-based FOIL (Quinlan, 1990), and so on. These systems have their own strengths and limitations in terms of class of learnable theories, completeness, and efficiency. We only overview briefly MIS and GOLEM here since we will use them as programs for learning theory based on fixed language.

MIS (Model Inference System) (Shapiro, 1981) is an incremental[8] algorithm based on Popper's methodology of conjectures and refutations (Popper, 1959). It searches the hypothesis space of first-order clauses from general to specific. MIS starts with the most general theory (a theory with an empty clause) as the current conjecture, and revises individual clauses in the current conjecture whenever contradiction occurs. When the current conjecture T proves a negative example (T is too strong), a *backtracing algorithm* is invoked to remove a faulty clause that is responsible for proving the negative example from T. Removing a clause may overly specialize T as T may fail to prove some positive examples (T is too weak). In this case, the refinements (more specific clauses) of faulty clauses removed are added gradually into T. This process repeats until T is neither too general nor too weak. One of the most important properties of MIS is that it is proved to be complete; it identifies any theory from ground unit clauses in the limit if it is representable without theoretical terms. However, MIS is very inefficient in practice (Kirschenbaum & Sterling, 1991) for larger problems.

GOLEM(Muggleton & Feng, 1990), on the other hand, is one of the most efficient ILP programs, although it infers a restricted class of theories and employs heuristics. The underlying mechanism of GOLEM can be explained briefly as follows. If a clause to be inferred in the target theory is:

$$p \leftarrow l_1, ..., l_n.$$

[8]MIS is incremental in the sense that any false clause removed from the theory will never be introduced into the theory again: only the refinements of the refuted clauses may be introduced.

then there would be a set of clauses, such as

$p' \leftarrow l'_1, ..., l'_n.$
$p'' \leftarrow l''_1, ..., l''_n.$
......

that are instantiations of the target clause. If we generalize these clauses to produce one clause, by applying the least general generalization (LGG) operator (Plotkin, 1970), we should get a clause that will not be more general than the original target clause. The problems are, first, how to obtain the set of the clauses on which LGG should be applied; and second, the length of the resulting clause from LGG is usually very long.

GOLEM solves the first problem by representing all background knowledge and given examples in ground positive literals, and putting *all* of them (say $g_1, ..., g_N$) in the body of the clauses. Therefore, for given positive examples $p_1, p_2, ...$ of the target predicate, we first construct a set of clauses as:

$p_1 \leftarrow g_1, ..., g_N.$
$p_2 \leftarrow g_1, ..., g_N.$
......

But which of those clauses should be chosen for the set on which LGG is applied? Ideally, they should be the instantiations of the *same* target clause. GOLEM uses a heuristic of maximum covering of the positive examples to choose the best clause to join in the set on which LGG is applied. This process repeats until the covering stops growing anymore, and then one such clause is found, and the next one is being constructed. The second problem, the overly long clause from LGG, is solved by putting a restriction (called ij-determination) on the literals to be added in the condition of the resulting clause. GOLEM is shown to infer ij-determination theory with small i and j (e.g., 2) from hundreds of examples quite efficiently (e.g., within a few minutes).

3.3 The Invention Operator: EXTRACT

The motivation behind the invention operator can be described as follows. If there is a clause

$$p \leftarrow l_1, ..., l_n, newp. \qquad (1)$$

which defines the observable term p with the assistance of a theoretical term $newp$ in the target theory, then there should be a set of clauses, such as

$$p' \leftarrow l'_1, ..., l'_n, newp'. \qquad (1_1)$$

$$p'' \leftarrow l''_1, ..., l''_n, newp''. \qquad (1_2)$$

......

that are instantiations of the clause (1). An instantiation of the clause (1) can also be represented as

$$p\theta \leftarrow l_1\theta, ..., l_n\theta, newp\theta.$$

for some substitution θ that substitutes variables to functions or constants. Therefore, (1_1) and (1_2) before *newp* is introduced can be represented as (for some substitutions $\theta_1, \theta_2, ...$):

$$p\theta_1 \leftarrow l_1\theta_1, ..., l_n\theta_1.$$

$$p\theta_2 \leftarrow l_1\theta_2, ..., l_n\theta_2.$$

......

Since they have a common generalization (i.e., $p \leftarrow l_1, ..., l_n.$), we can reconstruct clause (1) with a new term by "extracting" or "factoring out" the generalization of them via the least general generalization operator, and assign the "residues" to a new term.

Therefore, our invention operator, called EXTRACT, is defined as:

> Assume that there exist n substitutions θ_1, ..., θ_n and a set S of n Horn clauses in the form of:
>
> $$p\theta_1 \leftarrow \bar{l}\theta_1.$$
>
> ...
>
> $$p\theta_n \leftarrow \bar{l}\theta_n.$$
>
> where p is a literal, and $\bar{l}$ is a conjunction of literals (e.g., $l_1, ..., l_n$), *The* EXTRACT *operator* transfers S to a logically equivalent set S' of clauses with a newly introduced predicate *newp*. S' consists of
>
> $$p \leftarrow \bar{l}, newp(\bar{x}). \qquad (2)$$
>
> $$newp(\bar{x})\ \theta_1. \qquad (3_1)$$
>
> ...
>
> $$newp(\bar{x})\ \theta_n. \qquad (3_n)$$

Note that EXTRACT not only infers the clause (2) that defines the observable predicate p using a new predicate *newp* in the body of the clause, but also creates a set of ground positive examples (3_1, ..., 3_n) of *newp* by abduction — these clauses are regarded as *internal positive examples* since they are "required" to be true by the positive examples of p and the clause (2).

It is proved (Ling, 1993) that EXTRACT is a logically equivalent operator. That is, $S' \models S$ and $S \models S'$ except instances of *newp*. However, in terms of the number of clauses, S' is actually larger than S. The application of the EXTRACT operator allows for another learning step (in new normal science) in inferring a (small) theory of *newp*.

Some examples of using EXTRACT are shown below. using the same notation as in Prolog, small letters represent constants, while capital letters represent universally quantified variables. (X/T) denotes a substitution of a variable X by a term T.

Example (cf. Section 4.1):
If there are five clauses in the form:

$colour(b, y).$
$colour(ch(b, b), y).$
$colour(ch(b, ch(b, b)), y).$
$colour(ch(ch(a, b), ch(a, b)), y).$
$colour(ch(ch(b, b), ch(b, b)), y).$

These clauses can be rewritten with corresponding substitutions as

$colour(X, y).(X/b)$
$colour(X, y).(X/ch(b, b))$
$colour(X, y).(X/ch(b, ch(b, b)))$
$colour(X, y).(X/ch(ch(a, b), ch(a, b)))$
$colour(X, y).(X/ch(ch(b, b), ch(b, b)))$

EXTRACT introduces a new predicate *newp* and instances of *newp* as:

$colour(X, y) \leftarrow newp(X).$
$newp(b).$
$newp(ch(b, b)).$
$newp(ch(b, ch(b, b))).$
$newp(ch(ch(a, b), ch(a, b))).$
$newp(ch(ch(b, b), ch(b, b))).$

3.4 Conditions for Necessary New Predicates

After EXTRACT infers instances of the new terms by abduction, another inductive learning step is applied to these instances to infer the theory of the new terms as the process of new normal science. If the new terms are necessary for finite theorizing, the hypotheses of the new predicates must satisfy the following condition:

> At least one of the clauses of necessary new predicate p must be defined in direct recursion. That is,
> $newp \leftarrow l_1, ..., l_n, newp.$

The recursive definition of theoretical terms is also discussed in (Shen & Simon, 1993). Note that the arguments of $newp$ in the condition and the consequence of the above clause must be different; otherwise, the clause becomes a tautology. As an example, if $newp1$ is a necessary new predicate, one of its clauses can be:

$$newp1(X, s(Y), Z) \leftarrow newp1(X, Y, Z1), add(Z1, Y, Z).$$

If none of the hypotheses of the new term $newp$ is in the form of direct recursion, then $newp$ in the head of hypotheses of p can be resolved away with p appearing in the body of other clauses. Since these clauses do not have p in the head, the resolvents will not have p in the clause, and therefore, an equivalent theory is obtained in which p is completely eliminated. Thus, if new terms are necessary in finite learning, they must be defined in direct recursion.

Direct recursion in propositional logic is always a tautology, and therefore, is useless in the theory. The possibility of defining new terms in direct recursion *in first-order logic* with different arguments is essential for necessary new terms in terms of finite learning. Most ILP systems, including GOLEM and MIS, can readily infer recursive clauses.

3.5 INNP: An Implemented System

INNP (Introducing Necessary New Predicates), being implemented in Quintus Prolog, is a data-driven inductive learning system that introduces new terms to model the processes of scientific discovery. INNP is an implementation of the general strategy outlined in Section 3.1, in which the invention operator is the EXTRACT operator, and the learning program with fixed language is GOLEM.

Several implementational issues of INNP are discussed briefly below.

- *The decision problem.* Whether or not the theory would grow without a bound is, in fact, undecidable. Therefore, the decision problem is replaced by a finite, constant bound h in INNP. If the theory grows over h during revision, the initial language is regarded as insufficient, and new predicates are introduced.

- *Introducing multiple new predicates.* Sometimes more than one new predicate is introduced simultaneously. In these cases, EXTRACT will apply to several subsets (a partition) of the current theory, with each subset introducing a new and different new predicate. However, searching all possible partitions of T is very expensive. A greedy heuristic is designed, and shown to be quite effective for finding the right partition. Starting from an arbitrary partition as the current partition, the greedy heuristic searches all *revisions* of the current partition — partitions which are modified slightly from the the current one — and chooses the revision from which the smallest theory is produced. Even with this heuristic, the search space is still very large. More domain specific knowledge is needed to constrain the search.

- *Negative examples of the new predicates.* Most learning programs including GOLEM require negative examples; otherwise the hypotheses induced would always be overly general. Since only internal positive examples of the new predicates are available by abduction, the negative examples are obtained by the closed-world assumption (CWA). That is, any instances not produced by EXTRACT are regarded as negative examples.

- *Sparseness of the new predicates.* Sometimes the instances of the new predicates that EXTRACT produces by abduction are too sparse. From these examples alone, it is often impossible for GOLEM (and most other ILP programs such as CIGOL, FOIL, or MIS) to infer a theory of the new predicates that can derive those and *only* those examples. A similar problem in scientific discovery also often occurs; often scientists have to infer a general theory from a few clues. Several methods have recently been studied to solve this problem (cf. (Lapointe, Ling, & Matwin, 1993)).

4 Learning to Predict Colour: the Example Continued

Our scientist, as described in Section 2, fails to come up with a small theory that can explain and predict colour of flowers of any generations. A revolution seems necessary to him, in which a new language with new terms will be adopted.

4.1 Introduction of New Terms

The scientist uses INNP, which, at that point, calls EXTRACT to create new terms. Given an arbitrary initial partition, the size of the resulting theory is usually very large. This is because there are hardly any general Horn clauses to cover a large number of positive examples, so many "exceptions" (positive examples not covered by the clause) are just listed in the resulting theory. For example, the following conjecture is produced from INNP with an arbitrary partition:

$newp7(ch(A, b), B) : -newp7(ch(b, b), B).$

$newp7(b, b).$	$newp8(b, ch(a, b)).$	$newp8(b, ch(a, a)).$
$newp8(b, a).$	$newp8(a, ch(a, b)).$	$newp8(a, ch(a, a)).$
$newp8(a, a).$	$newp9(ch(b, b), ch(a, b)).$	$newp9(ch(b, b), ch(a, a)).$
$newp9(ch(b, b), a).$	$newp9(ch(a, b), ch(a, b)).$	$newp9(ch(a, b), ch(a, a)).$
$newp9(ch(a, b), b).$	$newp9(ch(a, b), a).$	$newp9(ch(a, a), b).$
$newp9(b, a).$	$newp9(a, a).$	

Using the greedy heuristic search described in Section 3.5, INNP looks for partitions that produce smaller theories with new predicates[9]. The hill-climbing heuristic often[10] finds the

[9]Each run of INNP on one partition takes about 5 seconds on a SPACStation. The whole search process of finding the ideal partition takes about 1 hour.

[10]Occasionally the search got caught in local minima.

"ideal partition" with three subsets in the partition as shown below: Set 1 contains all examples of yellow flowers (note that positive examples are numbered in Table 1): 2, 5, 11, 12, 17, and 20; set 2 contains some examples of red flowers 1, 3, 6, 7, 13, 14, 16, and 18; and set 3 contains some examples of red flowers: 4, 7, 8, 9, 10, 14, 15, 16, 18, and 19. When the invention operator EXTRACT is applied on this ideal partition with three subsets, three new predicates, say $n11$, $n10$, and $n10$, are introduced (cf. the example in Section 3.3). Table 2 contains results after applying EXTRACT. The resulting theory after the scientist applies GOLEM on the theory in Table 2 (a new "normal science") is, indeed, very small — INNP produces a theory with only *three* clauses for the new terms, even to our great surprise:

$n00(ch(X,Y)) \leftarrow n00(ch(X,X)), n00(ch(Y,Y)).$
$n10(ch(X,Y)) \leftarrow n11(ch(X,X)), n00(ch(Y,Y)).$
$n11(ch(X,Y)) \leftarrow n11(ch(X,X)), n11(ch(Y,Y)).$

Group 1	Group 2	Group 3
$colour(X,y) \leftarrow n00(X).$	$colour(X,r) \leftarrow n11(X).$	$colour(X,r) \leftarrow n10(X).$
$n00(b).$	$n11(a).$	$n10(ch(a,b)).$
$n00(ch(b,b)).$	$n11(ch(a,a)).$	$n10(ch(a,ch(a,b)).$
$n00(ch(b,ch(a,b))).$	$n11(ch(a,ch(a,a))).$	$n10(ch(a,ch(b,b)).$
$n00(ch(b,ch(b,b))).$	$n11(ch(a,ch(a,b))).$	$n10(ch(b,ch(a,a)).$
$n00(ch(ch(a,b),ch(a,b))).$	$n11(ch(ch(a,a),ch(a,a))).$	$n10(ch(b,ch(a,b)).$
$n00(ch(ch(a,b),ch(b,b))).$	$n11(ch(ch(a,a),ch(a,b))).$	$n10(ch(ch(a,a),ch(a,b))).$
$n00(ch(ch(b,b),ch(b,b))).$	$n11(ch(ch(a,b),ch(a,b))).$	$n10(ch(ch(a,a),ch(b,b))).$
	$n11(ch(ch(a,b),ch(a,a))).$	$n10(ch(ch(a,b),ch(a,b))).$
......	......	......

Table 2: Results of applying EXTRACT on the ideal partition

Because GOLEM constructs hypotheses based on all of the given background knowledge, the above clauses can derive the target examples *if* all background knowledge including information of new terms in the same generations are given. That is, to determine if the colour of a flower Z is yellow, we need to determine if $n00(Z)$ is true or not. To do so, we need to determine, from the first clause, if $n00$ is true for the offsprings of X, X and the offsprings of Y, Y where X and Y are Z's parents. This information is not available for the current generalization, and therefore, these clauses cannot derive anything from the upper generations of flowers. However, it is easy to verify that these hypotheses are *semantically correct*, since these three new terms can be interpreted as follows: $n00$ represents purebred flowers with recessive character, $n11$ purebred flowers with dominant character, and $n10$ hybrid flowers with dominant character. Therefore, the clause

$$colour(X, y) \leftarrow n00(X).$$

says that yellow is a recessive character, and the first clause says that the offsprings of X and Y have recessive character if the offsprings of both X, X and Y, Y have recessive character. This is equivalent to saying that both X and Y have recessive character.

There exists a theory with a small set of Horn clauses[11] as listed in Table 2 that derive colour of flowers from the *upper* generations. Thus proving colour of flowers of any generation is possible, since it is reduced to the first generations. Note that each new predicate is recursively defined (via at least one clause of the new predicate) as we discussed in Section 3.4 as a condition for necessary new terms. See Table 3 for the results.

% Top level...	% About $n10$...
$n11(a)$.	$n10(ch(X,Y)) \leftarrow n11(X), n10(Y)$.
$n00(b)$.	$n10(ch(X,Y)) \leftarrow n11(X), n00(Y)$.
$colour(X, r) \leftarrow n11(X)$.	$n10(ch(X,Y)) \leftarrow n10(X), n10(Y)$.
$colour(X, r) \leftarrow n10(X)$.	$n10(ch(X,Y)) \leftarrow n10(X), n00(Y)$.
$colour(X, y) \leftarrow n00(X)$.	
% About $n11$...	**% About $n00$...**
$n11(ch(X,Y)) \leftarrow n11(X), n11(Y)$.	$n00(ch(X,Y)) \leftarrow n10(X), n10(Y)$.
$n11(ch(X,Y)) \leftarrow n10(X), n11(Y)$.	$n00(ch(X,Y)) \leftarrow n10(X), n00(Y)$.
$n11(ch(X,Y)) \leftarrow n10(X), n10(Y)$.	$n00(ch(X,Y)) \leftarrow n00(X), n00(Y)$.

Table 3: Hypotheses of new predicates

To summarize, the colour of flowers is theorized finitely with the help of necessary theoretical terms; without them, scientific theorizing is impossible unless deriving the colour of flowers from the same generations is allowed.

[11] Several of these clauses were actually produced by INNP with different random seeds. If P is complete (such as MIS), all of these hypotheses can be constructed by P with more examples.

4.2 Interpretations of Theoretical Terms

Theoretical terms are regarded as not observable and internal to the learning system. However, after their postulates are hypothesized, we may want to understand them and interpret their meanings. As we have seen in this example, we may interpret unary predicates $n11, n10$, and $n00$ as "types" of genetic characters. Then, the clauses in Table 3 can be understood as: there are three types of flowers, $n11$, $n10$, and $n00$. Types $n11$ and $n10$ are red, and type $n00$ is yellow. The initial red group is type $n11$, and the initial yellow group is type $n00$. The postulates of new predicates in Table 3 specify exactly how types of character change through generations: type $n11$ and type $n11$ produce type $n11$ only; type $n11$ and type $n10$ produce both type $n11$ and type $n10$; etc. As we now know, we usually call $n00$ recessive character, $n11$ dominant character of purebred, and $n10$ dominant character of hybrid.

As Kleene (1952) proved, any axiomatizable theory without identity can be finitely axiomatized with additional predicates (by three new predicates here); and further, with just one additional predicate. We may make *syntactic changes* to the form of theoretical terms and obtain a logically equivalent theory with only one theoretical term. For example, we can "merge" the three theoretical terms into one theoretical term, called *newp* with an arity of two. The predicate $newp(X, Y)$ simply encodes $n11, n10$ and $n00$ with the second argument Y (Y is a variable with a domain of three *arbitrary* constants or ground terms, say $n11, n10, n00$). To obtain the theory with only one new term, any appearance of $n11(X)$ in the original theory is replaced by $newp(X, n11)$, and similarly for $n10$ and $n00$[12].

Clearly, any finite number of theoretical terms can be merged to one in this fashion. However, there are various ways to encode the original theoretical terms. We may introduce $newp'(X, Y1, Y2)$, and replace $n11$ by $newp'(X, 1, 1)$, $n10$ by $newp'(X, 1, 0)$, and $n00$ by $newp'(X, 0, 0)$. In this case, hypotheses in Table 3 now become:

$newp'(ch(X,Y),1,1) \leftarrow newp'(X,1,1), newp'(Y,1,1).$
$newp'(ch(X,Y),1,1) \leftarrow newp'(X,1,0), newp'(Y,1,1).$
$newp'(ch(X,Y),1,0) \leftarrow newp'(X,1,0), newp'(Y,1,1).$
......

[12]The Extract operator does not introduce new predicates which contain extra arity for different ground terms.

Now we (and the scientist) may interpret $newp'(X, Y1, Y2)$ as "there is a pair of (Y1, Y2) associated with the colour of the flower, and let us call them 'genes' of the flower". The postulates of $newp'$ indicate exactly that the "genes" in parents should segregate and combine in the next generation.

It is very interesting to observe that although "genetic types" and "genes" are two different concepts in Genetics, regarding them as theoretical terms in our logic framework, they are *logically equivalent*. They are just interpretations of *different encodings of the same theoretical terms*. The concepts of both "genetic types" or "genes" are purely hypothetical to our fictional scientist, just as they were to Gregor Mendel, who published his findings in 1865.

5 Discussion and Conclusion

The gene discovery is also discussed in (Shen & Simon, 1989; Darden, 1990). Some aspects of the discovery program LIVE (Shen & Simon, 1989) can be regarded as special cases of the Horn clause learning systems. For example, backtracing history can be represented as Horn clauses in which the predicate in the consequence is defined by predicates with additional "predecessor" functions applied on the arguments. Backtracing one step, several steps, or a finite number of steps in a chain of history can all be represented easily in first-order Horn clauses. The same is true for "looking ahead", i.e., performing further experiments. LIVE has a small number of m-constructors for the discovery of genes in order to improve the efficiency. Properties (eliminability, falsifiability, and identifiability) of recursive theoretical terms have also been studied (Shen & Simon, 1993). The theoretical term (gene) is shown to be eliminable by allowing to test if a given pea is "purebred" or "hybrid". See footnote 7 in Section 2.

As we discussed in the Introduction, scientific discovery and revolution is a very complicated process and we can only model a very small fraction of it in our logic framework. Our model is still too coarse-grained, and cannot cover a wide variety of cases in actual scientific discovery. One of the most critical differences of our system and the actual process of scientific discovery is that INNP is mostly data-driven. Our system has no high-level knowledge and motivation of why a revolution is needed, what new terms should be introduced, and why one set of new terms is better than a logically equivalent other. It is driven by the syntactic criterion of inferring small theory to introduce new terms. On one hand, inferring small theory with a new representation — an intriguing version of the Occam's Razor Principle — has its deep philosophical foundations, psychological importance, as well as learnability properties (Blumer, Ehrenfeucht, Haussler, & Warmuth, 1987). However, applying this strategy *alone* is neither sufficient practically, nor feasible computationally. Many equivalent "revolutions" may be proposed and INNP has no way to distinguish which one is better. In addition, the search space of all possible "revolution" is prohibitively huge. High-level knowledge and other background knowledge is an effective way to reduce the search space. Since the effective use and representation of high-level and common-sense knowledge is one of the most difficult open problems in artificial intelligence at present, to rely on solving the knowledge representation problem for the current goal is outside the scope of this paper. Therefore, we replace high-level and common-sense knowledge by heuristics and search. We hope that we have demonstrated the main point of our paper — certain logical aspects of scientific discovery and revolutions can be modeled computationally via learning first-order theories with new terms.

To summarize, we present an inductive learning algorithm that infers first-order theories with new predicates and use it to model the logical aspect of scientific revolution. We show that our system can introduce new terms which may be interpreted as "genes" or "genetic types".

Acknowledgement

The author gratefully acknowledges Wei-Min Shen who communicated the gene problem to him. Discussions of the related problems with Bill Demopoulos, Scott Weinstein, and people in Machine Learning Group at the University of Ottawa were very helpful.

Reference

Aha, D. W., Lapointe, S., Ling, C. X., & Matwin, S. (1994a). Inverting implication with small training sets. In *Proceedings of the 1994 European Conference on Machine*

Learning, pp. 31 – 48. Springer Verlag.

Aha, D. W., Lapointe, S., Ling, C. X., & Matwin, S. (1994b). Learning recursive relations with randomly selected small training sets. In *Proceedings of the 1994 International Conference on Machine Learning*, pp. 12 – 18. Anonymous ftp: /pub/ling/papers/mlc94-ILP.ps.Z on ftp.csd.uwo.ca.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. (1987). Occam's Razor. *Information Processing Letters*, *24*, 377–380.

Darden, L. (1990). Diagnosing and fixing faults in theories. In Shrager, J., & Langley, P. (Eds.), *Computational Model of Scientific Discovery and Theory Formation*. Morgan Kaufmann: San Mateo, CA.

Gaifman, H., Osherson, D., & Weinstein, S. (1990). A reason for theoretical terms. *Erkenntnis*, *32*(2).

Glymour, C. (1985). Inductive inference in the limit. *Erkenntnis*, *22*, 23 – 31.

Glymour, C., & Kelly, K. (1989). On converging to the truth and nothing but the truth. *Philosophy of Science*, *56*.

Gold, E. (1967). Language identification in the limit. *Information and Control*, *10*, 47–474.

Hempel, C. (1963). Implications of Carnap's work for the philosophy of science. In Schlipp, P. (Ed.), *The Philosophy of Rudolf Carnap*, pp. 685–707. Open Court.

Kirschenbaum, M., & Sterling, L. (1991). Refinement strategies for inductive learning of simple Prolog programs. In *Proceedings of Twelfth International Conference on Artificial Intelligence (IJCAI-91)*, pp. 757–761. Morgan Kaufmann Publishers.

Kleene, S. (1952). Finite axiomatizability of theories in the predicate calculus using additional predicate symbols. *Memoirs of the American Mathematical Society*.

Kuhn, T. (1970). *The Structure of Scientific Revolutions* (2 edition). Chicago University Press.

Lai, T. (1991). Discovery as a problem for the inventor. *British Journal for the Philosophy of Science*, *42*, 327 – 337.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific Discovery – Computational explorations of the creative processes*. The MIS Press.

Lapointe, S., Ling, C. X., & Matwin, S. (1993). Constructive inductive logic programming. In *Proceedings of IJCAI-93 (Thirteenth International Conference on Artificial Intelligence)*, pp. 1130–1136. Morgan Kaufmann Publishers.

Ling, C. X. (1989). *Inventing theoretical terms in inductive learning*. Ph.D. thesis, Department of Computer Science, University of Pennsylvania.

Ling, C. X. (1991). Inductive learning from good examples. In *Proceedings of Twelfth International Conference on Artificial Intelligence (IJCAI-91)*, pp. 751–756. Morgan Kaufmann Publishers. Anonymous ftp: /pub/ling/papers/ijcai91-ge.ps.Z on ftp.csd.uwo.ca.

Ling, C. X. (1993). Inventing necessary theoretical terms for scientific discovery. Tech. rep., Department of Computer Science, University of Western Ontario.

Ling, C. X., & Marinov, M. (1993). Answering the connectionist challenge: a symbolic model of learning the past tense of English verbs. *Cognition*, *49*(3), 235–290. Anonymous ftp: /pub/ling/papers/cognition-verb.ps.Z on ftp.csd.uwo.ca.

Ling, C. X., & Marinov, M. (1994). A symbolic model of the nonconscious acquisition of information. *Cognitive Science*, *18*(4). In press. Anonymous ftp: /pub/ling/papers/cogsci-nonconscious.ps.Z on ftp.csd.uwo.ca.

Matheus, C., & Rendell, L. (1989). Constructive induction on decision trees. In *Proceedings of Eleventh International Conference on Artificial Intelligence (IJCAI-89)*, pp. 645–650. Morgan Kaufmann Publishers.

Muggleton, S. (1990). Inductive logic programming. In *Proceedings of First International Conference on Algorithmic Learning Theory*. OHMSHA Tokyo. Invited paper.

Muggleton, S., & Buntine, W. (1988). Machine invention of first-order predicates by inverting resolution. In *Proceedings of the Fifth International Conference on Machine Learning*.

Muggleton, S., & Feng, C. (1990). Efficient induction of logic programs. In *Proceedings of first International Conference on Algorithmic Learning Theory.* OHMSHA Tokyo.

Osherson, D., & Weinstein, S. (1986). Identification in the limit of first order structures. *Journal of Philosophical Logic*, *15*, 55–81.

Osherson, D., & Weinstein, S. (1989). Paradigms of truth detection. *Journal of Philosophical Logic*, *18*, 1–42.

Osherson, D., & Weinstein, S. (1990). On advancing simple hypotheses. *Philosophy of Science*, *15*.

Pagallo, G., & Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine Learning*, *5*(1), 71–100.

Plotkin, G. (1970). A note on inductive generalization. In Meltzer, B., & Michie, D. (Eds.), *Machine Intelligence 5*, pp. 153–163. Elsevier North-Holland.

Popper, K. (1959). *The Logic of Scientific Discovery.* Basic Books, New York.

Quinlan, J. (1990). Learning logical definitions from relations. *Machine Learning*, *5*, 239–266.

Rendell, L., & Seshu, R. (1990). Learning hard concepts through constructive induction. *Computational Intelligence*, *6*, 247–270.

Shapiro, E. (1981). Inductive inference of theories from facts. Tech. rep. TR-192, Computer Science Department, Yale University.

Shen, W., & Simon, H. (1989). Rule creation and rule learning through environmental exploration. In *Proceedings of Eleventh International Conference on Artificial Intelligence (IJCAI-89).* Morgan Kaufmann Publishers.

Shen, W., & Simon, H. (1993). Fitness requirements for scientific theories containing recursive theoretical terms. *British Journal for the Philosophy of Science*, *44*, 641–652.

Simon, H. (1992). Scientific discovery as problem solving. *International Studies in the Philosophy of Science*, *6*, 3 – 14.

Thagard, P. (1992). *Conceptual revolutions.* Princeton University Press.