A Comparison of Analytic and Experimental Goal Regression for Machine Learning

Bruce W. *Porter**
Computer Sciences Department
University of Texas at Austin

Dennis F. *Kibler***
Information and Computer Science Department
University of California at Irvine

ABSTRACT

Recent research demonstrates the use of goal regression as an analytic technique for learning search heuristics. This paper critically examines this research and identifies essential applicability conditions for the technique. The conditions that operators be invertible and that the domain be closed with respect to the inverse operators severely limit the use of analytic goal regression. In those restricted domains which satisfy the applicability conditions, analytic goal regression only discovers required preconditions for operator application. Discovering pragmatic preconditions is beyond the capability of the technique. An alternative, called *experimental goal regression,* is defined which approximates the results of analytic goal regression without suffering from these limitations.

I. Introduction

Goal regression was first used in AI by Waldinger [WALD77] as a technique for detecting and analyzing goal interactions during planning. Given a goal state *G* and an operator *OP,* a *goal regression product* is a description of a sub-goal state 5, such that *OP* applied to *S* achieves *G.* The goal regression product corresponds to Dijkstra's notion of *weakest pre-condition* [DIJK75]. According to Dijkstra, wp(OP,G) is the weakest constraint on a state *S* which guarantees that the application of *OP* to *S* yields a state satisfying *G.****

Recent research in machine learning demonstrates the use of goal regression to improve concept acquisition. In particular, this research focusses on techniques for learning problem solving heuristics. Given an operator sequence for a problem solution, goal regression serves to back-propagate constraints through the sequence to form heuristics for the individual operators. This technique involves formal reasoning with operator semantics and will be called *analytic goal regression.*

This paper is a critical review of the research on analytic goal regression and suggests a change in direction. Section 2 briefly reviews this research. Section 3 discusses the *applicability conditions* for successful application of analytic goal regression to machine learning. These conditions appear to restrict the use of analytic goal regression to a relatively small class of problem domains. Section 4 introduces *experimental goal regression* as an alternative to the analytic technique. Experimental goal regression uses induction from examples to approximate the "correct" result. This removes the constraining applicability conditions and permits goal regression in a wide class of problem domains. Section 5 compares analytic and experimental goal regression.

II. Review of Research

Utgoff [UTG083] demonstrates the use of goal regression to adjust the bias inherent in the concept hierarchy trees used in LEX [M1TC78]. Partial state descriptions are regressed through a solution path to form a composite constraint on initial states in the path. Motivating this work is the observation that the state description vocabulary should be rich enough to describe composite constraints. Typically, this vocabulary is a *priori* domain knowledge. The significance of Utgoff's use of goal regression is that the vocabulary can be dynamically enriched during the learning process.

Porter and Kibler [PORT84A] use an empirical variant of goal regression to improve the rate of learning problem solving heuristics. Their method of episodic learning discovers useful operator sequences [KIBL83B]. The learning is incremental and heuristics which recommend operators applied near the goal state are learned first. These heuristics are regressed through the episode to learn additional heuristic rules. The significance of this use of goal regression is that the rate of episodic learning can be dramatically improved by broadcasting the refinement of one heurbtic through an episode to enable the refinement of other heuristics.

Minton [MINT84] demonstrates the power and potential of goal regression by showing effective learning from a single training instance. With a technique called constraint based generalization, state descriptions are generalized by deducing *why* the training instance is classified as positive. The technique is applied to learning generalized state descriptions for forced wins in two-person games. Given a chain of actions resulting in a forced win, goal re-

* *Support for this research was provided by the Army Research Office, under great number ARO DAAG29-84-K-0060.*
** Partial s*upport for this research was provided by Hughes AI Research Center, Calabasas, CA.*
*** In computer programs OP is a sequence of program statements.

gression is used to back-propagate the important descriptors of the forced win state. The goal regression product is a description of the set of states for which the chain of actions achieves the forced win. The significance of this use of goal regression is that (a limited form of) "one-shot" learning is possible by reasoning with explicit goal descriptions.

III. Applicability Conditions for Analytic Goal Regression

This section defines two applicability conditions for effective use of analytic goal regression.

A. Invertible Operators

The first applicability condition requires that the domain operators be invertible. Given a STRIPS-like declarative operator definition, inverses are easily computed by reversing the roles of the delete-list and add-list. However, procedural representations are more flexible and powerful for defining operator transformations [HEWI72, McDE72]. Unfortunately, analytically inverting a procedurally defined operator appears impossible in general. A similar problem occurs if the goal description is defined procedurally.

LEX-II [MITC83] partially circumvents this problem by providing the learning element with operator inverses. Both the domain operators and their inverses are represented procedurally. As Utgoff discovered [UTG083], the chief shortcoming of this approach is the inability to reason with some operator definitions. For example, consider the substitution operator in symbolic integration which replaces a sub-expression of an integral with a variable. In LEX, this operator is defined as:

$$\int poly(f(x))f'(x)\,dx \rightarrow \int poly(u)\,du, \quad u = f(x)$$

where *poly(f(x))* stands for a polynomial in *x.* The problem discovered by Utgoff is that analytic goal regression with this operator definition fails. The critical constraint that whatever matches /' be the derivative of whatever matches / is not explicit in this operator representation. This constraint is embedded in an opaque representation of the operator. Procedural representations can conceal essential operator constraints and prevent the analytic computation of goal regression products.

B. Relative Closure of Representation Language

The second applicability condition for analytic goal regression requires that the representation language adequately express goal regression products. As we demonstrate, analytic goal regression can produce state descriptions which are unrepresentable in the language used for forward reasoning. In particular this constraint is not satisfied by merely assuming that one has a STRIPS-like representation.

There is no problem with computing a goal regression product if the goal and the operator are fully instantiated. However, regressing expressions through partially instantiated operator definitions can be troublesome [NILS80, pp 288-292]. Consider the operator unstack(x,y) defined with the following STRIPS rule:

Pre and Delete conditions: *handempty, clear(x),on(x, y)*
Add conditions: *holding (x), clear (y)*

The regression of the partial state description *clear (C)* (for some constant *C)* yields the disjunction (y = *C) Vclear (C).* External disjunction is often precluded from concept description languages [MICH83] but commonly occurs in analytic goal regression products.

While a disjunctive clause might be split into separate clauses, thereby eliminating the disjunction, negated clauses can also be troublesome. For example, the regression of *clear(C)* through unstack(x, *B)* yields $\neg(x = C)$ A *clear(C).* These examples demonstrate that the concept description languages must support disjunction and negation if analytic goal regression is used.

The problem of representation language closure arises in Minton's research [MINT84]. As described in section 2, Minton applies goal regression to learn forced move board positions in two person games. In particular, consider Minton's example in the game gomoku. Gomoku is similar to tic-tac-toe except the board is a 19x19 grid and the object is to get 5 X's or O's in a row. An open four-position for X is four X's in a row with an adjacent blank position. An open three-position for X is three X's in a row with two blank positions on one side and one blank position on the other.

We believe that the description language cannot represent all goal regression products in this domain. The description language for board positions is a conjunct of predicates. Minton computes the goal regression product of an open four-position as an open three-position. Assuming the natural gomoku move operator definition:

pre-condition and delete-condition: empty (square)
add-condition: *on (square, X)*

the open three-position is only a subset of the correctly computed goal regression product. An open four can also result from applying the move operator to a

blank - X - blank - X - X - blank

position. Therefore, the correct goal regression product is a disjunctive clause which is outside the expressive power of the description language. This problem might be addressed by extending the description language to include terms which correspond to disjunctive expressions in the original language. But, discovering appropriate shifts in representation language may be more difficult than the problem being addressed.

We believe that these applicability conditions restrict the use of analytic goal regression to a relatively uninteresting set of problem domains. In addition to these necessary limitations, there is an additional pragmatic concern.

The technique forms totally precise rules which may not capture the data at hand. Analytic goal regression could be used in the total absence of experience and might be called learning by reasoning or "no-shot" learning.

The next section proposes an alternative which approximates the results of analytic goal regression but relies on experience, rather than reasoning.

IV. Experimental Goal Regression

Experimental goal regression is an alternative to analytic goal regression in which the applicable conditions are eliminated. The technique yields an approximation of the result of analytic goal regression. This approximation is incrementally refined using standard machine learning techniques.

Experimental goal regression uses induction to approximate a goal regression product. Consider states $s_1, s_2, \ldots, s_n$ in the domain search space which are mapped by operator *OP* into a state satisfying the goal *G*. Experimental goal regression applies standard induction techniques to $s_1, S_2, \ldots, s_n$ to form a partial description of the regression of *G* through *OP*.

Experimental goal regression avoides the limitations of analytic goal regression discussed in the previous section. First, the technique does not preclude procedural operator representations since *OP* is not inverted. Second, goals can be represented procedurally since experimental goal regression uses them only as boolean predicates on state descriptions. Third, experimental goal regression does not require the language to be complete with respect to goal regression products. Only a useful partial characterization of the inverse is generated instead of a completely accurate one as in the case of analytic goal regression.

The practical success of experimental goal regression relies on an efficient construction of multiple examples of goal regression products. One approach to example generation is perturbation [KIBL83A], which relies on inherent regularity and continuity in the search space [LENA83]. Given a single example, perturbation automatically generates and classifies neighboring examples. The selection of the most promising neighbors can be guided with some knowledge of the transformation performed by the operator. Relational operator models [PORT84B] are one technique for approximating operator definitions.

V. An Example

The following example compares analytic and experimental goal regression for learning search heuristics. The task is the algebraic simplification of a pair of simultaneous linear equations labelled *a* and 6. The operator *sub(a,b)* replaces equation 6 with the result of subtracting equation *a* from equation 6. Similarly, *sub(b, a)* replaces equation *a* with the result of subtracting equation *b* from equation a. For simplicity we assume that the *sub* operators are only applicable when the equations have equal x-coefficients. These operators can be defined as:

Operator	Preconditions and *Delete conditions*	Add conditions
sub(a,b)	$a: Ax + B_1y = C_1$ $b: Ax + B_2y = C_2$	$a: Ax + B_1y = C_1$ $b: (B_2 - B_1)y = C_2 - C_1$
sub(b,a)	$a: Ax + B_1y = C_1$ $b: Ax + B_2y = C_2$	$a: (B_1 - B_2)y = C_1 - C_2$ $b: Ax + B_2y = C_2$

As with other systems for learning search heuristics [LANG83, OHLS82], the learning element uses a static evaluation function for credit assignment. Since the overall problem solving goal is to simplify the equations, the function counts the number of non-zero terms in the equations.

Let us examine how analytic goal regression would learn heuristics for the *sub* operator. Consider the following positive training instance for the operator *sub(a,b):*

$$a: \quad 2x + 3y = 5$$
$$b: \quad 2x + 4y = 6$$

Analytic goal regression produces the heuristic rule:

$$\begin{array}{ll} a: & Ax + B_1y = C_1 \\ b: & Ax + B_2y = C_2 \end{array} \rightarrow sub(a,b)$$

This goal regression product is accurate with respect to the operator definition but fails to incorporate the guidance of the static evaluation function. The heuristic rule erroneously recommends *sub*(*a*, *b*) in the state:

$$a: \quad 2x + 3y = 5$$
$$b: \quad 2x + \boxed{0}y = 2$$

The resulting state is:

$$a: \quad 2x + 3y = 5$$
$$b: \quad 0x + \boxed{-3}y = -3$$

The static evaluation function reveals that sub(a, 6) is not effective in reducing the number of nonzero terms in the example.

Experimental goal regression induces a goal regression product from positive examples. The representation of both the *sub* operators and the static evaluation function are irrelevant to the success of experimental goal regression. Examples of effective applications of *sub(a, b)* are generated (perhaps by perturbation of a given example) and classified as positive or negative by the static evaluation function. Induction over the set of positive examples using the climbing hierarchy tree generalization operator yields the heuristic rule:

$$\begin{array}{ll} a: & nonzero_1x + integer_1y = integer_2 \\ b: & nonzero_1x + nonzero_2y = nonzero_3 \end{array} \longrightarrow sub(a, b)$$

where *nonzero* and *integer* are typed variables.

A goal regression product, derived experimentally, incorporates the pragmatic preconditions for effective use of each operator. In the example above, pragmatic preconditions for *sub(a, b)* require that both the y-coefficient and the constant term in equation *b* be non-zero. Pragmatic preconditions are enforced by constraints external to the operator definition. Therefore, they cannot be discovered by analytic goal regression.

We believe that combinations of analytic and experimental goal regression might be more promising than either technique alone. For example, a combination of techniques might mitigate the following problems:

- Analytic goal regression relies on extending the representation language to precisely describe goal regression products (section IIIB).
- Experimental goal regression relies on efficiently generating a set of state descriptions (section IV).

Using a combination of the two techniques, the goal regression product computed by analytic goal regression might be used to constrain the generation of state descriptions used in experimental goal regression. An analytic goal regression product that can be approximated in the representation language can be used as a "seed" for state description generation. The combination of analytic and experimental goal regression is a topic of future research.

VI. Conclusions

Goal regression promises to be a powerful technique in learning about actions. Predominantly, analytic goal regression has been explored. Analytic goal regression is a powerful reasoning technique for "no-shot" learning. That is, learning could proceed given only the definition of the goals and operators. Unfortunately, the necessity to invert operator definitions and to stay within the representation language severely limits the applicability of analytic goal regression. Experimental goal regression is an alternative to the analytic technique which relies on proven machine learning algorithms to approximate the results of analytic goal regression. Coupled with automatic example generation, experimental goal regression is an effective machine learning technique that does not suffer from the limitations of analytic goal regression.

REFERENCES

[DIJK75] Dilkstra, E. Guarded Commands, Non-Determinancy and Formal Derivation of Programs. *Communications of the Association for Computing Machinery 18,* 8, 453-457.

[HEWI72] Hewitt, C. *Description and Theoretical Analysis of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot,* PhD Dissertation, Massachusetts Institutute of Technology, 1972.

[KIBL83A] Kibler, D. and Porter, B. Perturbation: A Means for Guiding Generalization. Appearing in *Proceedings of the International Joint Conference on Artificial Intelligence,* 1983, pp. 415-418.

[KIBL83B] Kibler, D. and Porter, B. Episodic Learning. Appearing in *Proceedings of the National Conference on Artificial Intelligence,* 1983, pp. 191-196.

[LANG83] Langley, P. Learning Effective Search Heuristics. Appearing in *Proceedings of the International Joint Conference on Artificial Intelligence,* 1983, pp. 419-421.

[LENA83] Lenat,D.B. The Role of Heuristics in Learning by Discovery: Three Case Studies. Appearing in Machine Learning, Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (eds.), Tioga Publishing, 1983.

[MCDE72] McDermott, D.V. and Sussman, G.J. The Conniver Reference Manual. AI Memo 259, AI Laboratory, Massachusetts Institute of Technology.

[MICH83] Michalski, R.S. A Theory and Methodology of Inductive Learning. Appearing in *Machine Learning,* Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (eds.), Tioga Publishing, 1983.

[MINT84] Mint on, S. Constraint-Based Generalization: Learning Game-Playing Plans from Single Examples. Appearing in *Proceedings of the National Conference on Artificial Intelligence,* 1984, pp. 251-254.

[M1TC78] Mitchell, T.M. *Version Spaces: An Approach to Concept Learning,* PhD Dissertation, Computer Science Department, Stanford University. (TR STAN-CS-78-711), 1978.

[MITC83] Mitchell, T.M., Utgoff, P.E., Nudel, B. and Banerji, R. Learning by Experimentation: Acquiring and Refining Problem Solving Heuristics. Appearing in *Machine Learning,* Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (eds.), Tioga Publishing, 1983.

[NILS80] Nilsson, N.J. *Principles of Artificial Intelligence,* Palo Alto: Tioga Publishing Co., 1980.

[OHLS82] Ohlsson, S. On the Automated Learning of Problem Solving Rules. Technical Report 10, Uppsala University, Computing Science Department, Uppsala, Sweden.

[PORT84A] Porter, B. and Kibler, D. Learning Operator Transformations. Appearing in *Proceedings of the National Conference on Artificial Intelligence,* 1984, pp. 278-282.

[PORT84B] Porter, B. Learning *Problem Solving,* PhD Dissertation, Computer Science Department, University of California at Irvine, 1984.

[UTG083] UtgofF, P. Adjusting Bias in Concept Learning. *Proceedings of the International Machine Learning Workshop,* June 22-24, 1983, Monticello, Illinois. Sponsored by The University of Illinois at Champaign-Urbana., 1983, pp. 105-109.

[WALD77] Waldinger, R. Achieving Several Goals Simultaneously. Appearing in *Machine Intelligence 8,* Elcock, E.W. and Michie, D. (eds.), New York: Halstead and Wiley, 1977, pp. 94-136.