Heuristics for Empirical Discovery

Pat Langlcy
Herbert A. Simon
Gary L. Bradshaw

CMU-RI-TR-84-14

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213 USA

June 25 1984

Phis research was supported by Contract N00014-84-K-0345 from the Information Sciences Division, Office >f Naval Research, Approved for public release; distribution unlimited. Reproduction in whole or part is >ermitted for any purpose of the United States Government

fo appear in L. Bole (Ed.), *Knowledge Based Learning Systems.* Berlin: Springer-Verlag, 1984.

Technical Report No. 2

4. TITLE (and Subtitle)

Heuristics for Empirical Discovery

5. TYPE OF REPORT & PERIOD COVERED

Interim Report 1/84-6/84

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Pat Langley
Herbert A. Simon
Gary L. Bradshaw

8. CONTRACT OR GRANT NUMBER(s)

N00014-84-K-0345

9. PERFORMING ORGANIZATION NAME AND ADDRESS

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, PA 15213

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Information Sciences Division
Office of Naval Research
Arlington, Virginia 22217

12. REPORT DATE

25 June, 1984

13. NUMBER OF PAGES

28

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)

unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

To appear in L. Bolc (Ed.), Knowledge Based Learning Systems
Berlin: Springer-Verlag, 1984.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

data-driven discovery
expectation-driven discovery
heuristic search
hill-climbing
empirical laws
intrinsic properties
common divisors
symmetrical laws

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

OVER

Abstract

In this paper, we review our experiences with the BACON project, which has focused on empirical methods for discovering numeric laws. The six successive versions of BACON have employed a variety of discovery methods, some very simple and others quite sophisticated. We examine methods for discovering a functional relation between two numeric terms, including techniques for detecting monotonic trends, finding constant differences, and hill-climbing through a space of parameter values. We also consider methods for discovering complex laws involving many terms, some of which build on techniques for finding two-variable relations. Finally, we introduce the notions of intrinsic properties and common divisors, and examine methods for inferring intrinsic values from symbolic data. In each case, we describe the various techniques in terms of the search required to discover useful laws.

Table of Contents

1. Introduction: An Overview of BACON

Science is a multi-faceted endeavor, encompassing such diverse activities as designing experiments, discovering empirical laws, constructing new measuring devices, formulating theories, and testing hypotheses. Yet despite its complexity, the scientific process appears amenable to analysis in terms of the same concepts that have been successfully applied to other aspects of intelligence — the notion of *search* through a problem space, and the notion of *heuristics* for directing that search. In this paper we examine one facet of science — the empirical discovery of numeric laws — and describe a set of AI systems that are capable of such discovery. The programs are successive versions of the BACON system, named after Sir Francis Bacon, since their data-driven heuristics are similar in spirit to those proposed by the early philosopher of science in the Sixteenth Century.

Before examining the BACON heuristics in detail, let us briefly review the concerns and capabilities of different incarnations of the system. BACON.1 [1] was concerned mainly with the discovery of simple numeric laws relating two variables, and employed heuristics for noting trends and constants to this end.[1] BACON.2 [2] was concerned with the same task, but employed a simple differencing technique for finding such relations. We examine methods for discovering two-term laws in the first section below. BACON.3 [3] returned to the trend and constant detectors used in the first version, but combined these with a method for recursing to higher levels of description. This allowed the system to discover complex laws involving multiple variables, and this is the focus of section 3 below.

The BACON.4 system [4, 5] employed all of the heuristics used in BACON.3, but also included methods for postulating intrinsic properties and noting common divisors in cases where symbolic terms were involved. We discuss these methods in some detail in sections 4 and 5, using examples from the history of physics and chemistry. The fifth version of the system, BACON.5 [6, 7], was identical to BACON.4, except that it incorporated a differencing technique for finding simple numeric relations (this was more general than the one used in BACON.2), as well as expectation-based methods for reducing search through the space of laws. Finally, BACON.6 [8, 9] differed from its predecessor in that it replaced the differencing method with a hill-climbing technique that was more capable of handling noisy data. Table 1 summarizes the methods used in successive versions of BACON. In this paper, we examine the various heuristics for empirical discovery, focusing on their capabilities and their requirements. We will not spend much space on the individual BACON systems, since the interested reader may find descriptions in earlier papers.

2. Discovering Simple Numerical Laws

Let us begin with an apparently simple problem — determining the functional relation between two numeric terms. To be more specific, given two terms X and Y, along with a set of paired observations (x1, y1), (x2, y2), and so forth, we would like to find some function F such that Y = F(X) predicts the observed data as closely as possible. Analytic solutions to this problem, such as the methods of regression and correlation, have been developed in the field of statistics. Although quite robust, these methods require one to assume a linear function (or some other simple form), thus ruling out many plausible relations. A few Artificial Intelligence researchers [10, 11] have tentatively explored search-based approaches to this problem, but no systematic treatment has been carried out. Below we examine four different heuristic curve-fitting methods that vary in terms of the size of the space they search, and in their ability to deal with noisy data.

[1] BACON.1 also included heuristics for finding conditions on laws and for noting periodicity, but since these were not concerned with numeric law discovery, we will not focus on them here. BACON.2 and BACON.3 also included additional methods that we will ignore in the present paper.

Table 1. Components of successive BACON systems.

- BACON.1
    - Trend and constancy detectors
- BACON.2
    - Specialized method for finding constant differences
- BACON.3
    - Trend and constancy detectors
    - Recursing to higher levels of description
- BACON.4
    - Trend and constancy detectors
    - Recursing to higher levels of description
    - Intrinsic property method
    - Common divisor method
- BACON.5
    - General method for finding constant differences
    - Recursing to higher levels of description
    - Intrinsic property method
    - Common divisor method
    - Expectation-based methods
- BACON.6
    - Hill-climbing method for dealing with noise
    - Recursing to higher levels of description
    - Intrinsic property method
    - Common divisor method
    - Expectation-based methods

2.1. Detecting Trends and Constants

Some early versions of BACON (1, 3, and 4) included four simple heuristics for finding two-term numeric laws. The first two of the rules are responsible for noting constant values and linear relations, both of which lead directly to the formulation of a law. The second two rules come into play when neither of the first pair can be used; these define some *theoretical term* as the product or ratio of existing terms. Once defined, BACON recursively applies its rules to these theoretical terms, looking for constant values, linear relations, or other trends, continuing until some law is discovered. These heuristics can be summarized as follows:

1. If Y has the value V in a number of cases, then hypothesize that Y always has that value.
2. If X and Y are linearly related with slope S and intercept I in a number of cases, then hypothesize that this relation always holds.
3. If X increases as Y decreases, and X and Y are not linearly related, then define a new term T as the *product* of X and Y.[2]
4. If X increases as Y increases, and X and Y are not linearly related, then define a new term T as the *ratio* of X and Y.

The last two rules can be viewed as directing BACON's search through the space of theoretical terms, where

[2] Actually, the third and fourth rules apply only when the values of X and Y have the same sign. In cases where X and Y have opposite signs, two analogous rules propose the opposite actions; for example, when the values of X and Y increase together, and X and Y have different signs, the product XY is defined.

each new term is defined as an arithmetic combination of directly observable variables. They focus attention on terms that show potential for leading to constant values or linear relations, while the first two rules are responsible for detecting such laws once the appropriate terms have been defined.

The operation of these heuristics is best understood through an example. Consider the distance D of the planets from the sun (measured in astronomical units) along with the period of those same planets (measured in years). As one can see in Table 2, neither the values of D or P are constant, nor are they linearly related. However, the values of D increase along with those of P, so our fourth heuristic tells us to define the new term D/P as the ratio of these variables. Upon computing the values of this term, we find that they are neither constant nor linearly related to any other term, but that they do increase as those of both D and P increase. At this point we might define either the product D^2/P or the product PD/P. However, the latter of these is equivalent to the distance, and so should be abandoned. When the values of D^2/P are calculated, we find that they increase as those of D/P decrease. As a result, we would define the product D^3/P^2 and compute its values. Since these have the constant value 1.0, the first heuristic would apply, hypothesizing that this holds for all planets. Acting together, our heuristics have directed us through the space of theoretical terms, arriving at the functional equivalent of Kepler's third law of planetary motion.

Although the linear relation detector was not used in the above example, it is useful in other cases, such as the discovery of Ohm's law for electric circuits. In this situation, one varies the length L of a wire and observes the resulting current I. The values of I increase as those of L decrease, but since they are not linearly related, the product IL would be defined. Upon calculating the values of this term, one finds that IL and I are linearly related by the equation $IL = bI + v$ (where the particular values of b and v depend on the battery used). This equation is equivalent to Ohm's law, with the slope b representing the internal resistance of the battery and the intercept v its voltage.

Table 2. Data obeying Kepler's third law.

PLANET	D	P	D/P	D^2/P	D^3/P^2
MERCURY	0.382	0.241	1.607	0.622	1.0
VENUS	0.724	0.616	1.175	0.851	1.0
EARTH	1.0	1.0	1.0	1.0	1.0
MARS	1.524	1.881	0.810	1.234	1.0
JUPITER	5.199	11.855	0.439	2.280	1.0
SATURN	9.539	29.459	0.324	3.088	1.0

The above heuristics do have limitations. For example, they cannot discover polynomial functions of degree two or higher, including such simple laws as $Y = aX^2 + b$. In such cases, the heuristics actually lead the system to ignore theoretical terms that are required to state the laws. However, they have been able to discover a number of nontrivial laws from the history of physics, including Galileo's laws for the pendulum and constant acceleration, as well as Ohm's law and Kepler's law. In addition, when combined with a heuristic for recursing to higher levels of description (discussed in the section 3), these methods can induce much more complex relations, such as Coulomb's law of electrical attraction. What is amazing, then, is not the absolute power of these heuristics, but the fact that such *simple* rules are so useful in directing search through the space of possible laws. For instance, in finding Kepler's law, BACON.1 examined only 4 theoretical terms, compared to some 16 terms (a conservative estimate) that would be examined by a straightforward generate-and-test approach that considers simpler terms before more complex ones. One is tempted to infer that the early scientists also employed such simple heuristics to search for regularities, but that would lead us into historical discussions for which we do not have space.

As implemented in the early versions of BACON, this method had only modest capabilities for dealing with noisy data. The system required some ability on this dimension simply to deal with round-off errors in computing the constant values of higher level terms such as D^3/P^2. However, we did not explore the effects of significant noise in our experiments with BACON.1 through BACON.4. In section 3, we discuss an extension of this basic approach that has excellent noise-handling capabilities, as well as being able to discover laws relating multiple terms. First, though, let us examine some other approaches to finding simple numeric laws.

2.2. Finding Constant Differences

Despite the attraction of the trend and constancy detectors, their limitations led us to explore more robust function-finding methods. As a result, BACON.2 incorporated a heuristic that searched for constant differences, and BACON.5 included a more general version of this same method; in this paper we will describe only the more general scheme. This technique searches for polynomial relations between two variables, and with a simple extension, can discover all of the laws attainable by the earlier method (though in a different form), along with many others as well.

Table 3. Determining the coefficient of a quadratic term.

X	Y	Y'	Y"
1	6		
		14	
3	34		3
		29	
6	121		3
		50	
10	321		3
		77	
15	706		

Again, the method is best explained with an example. Consider the law $Y = 3X^2 + 2X + 1$, where X is the independent term and Y is the dependent variable. Table 3 presents some values obeying this law, along with some differences computed by the method. The first step involves examining the first differentials of Y with respect to X for successive values. This term is computed just as one would compute the slope of a line. For instance, the first value would be $(34 - 6)/(3 - 1) = 14$, the second would be $(121 - 34)/(6 - 3) = 29$, and so on, until all successive values of X and Y have been combined in this manner. Since the resulting differentials are not constant, the process is repeated. The second differentials are computed using the first differentials in the numerator and the X values in the denominator; however, differences are taken between every other X value instead of directly adjacent ones. Thus, the first of the second differentials would be $(29 - 14)/(6 - 1) = 3$, the second would be $(50 - 29)/(10 - 3) = 3$, and so on. These values *are* constant, so the system infers that the function relating the two terms includes an X^2 term with a coefficient of 3. (This method works equally well for real-valued coefficients.)

Given this knowledge, BACON subtracts out the variance accounted for by the X^2 term, and repeats the process on the values of $Y - 3X^2$. Table 4 presents the resulting values, along with the first differentials of this term. Since the values of this differential have the constant value 2, the program infers that an X term with coefficient 2 is present in the final equation. Again, this component is subtracted and the resulting values of $Y - 3X^2 - 2X$ are examined, as shown in Table 5. The values of this term are the constant 1, so BACON infers that the final form of the equation is $Y = 3X^2 + 2X + 1$. This differencing technique lets the system discover any polynomial relation, provided enough data have been observed to compute the necessary differentials. Thus, four observations would be required if the equation is of the third degree, three would be necessary if an X^2 term is involved, and so on. However, to ensure that spurious relations are not found in this manner, BACON insists that the functions it discovers be overidentified by the data, though the exact number of additional observations is controlled by the user.

Table 4. Determining the coefficient of a linear term.

X	$Y - 3X^2$	$(Y - 3X^2)'$
1	3	
		2
3	7	
		2
6	13	
		2
10	21	
		2
15	31	

The differencing method can be viewed as carrying out a heuristic search through the space of polynomial functions. This search begins by considering candidates for the largest terms in the relationship, examining constant, linear, and quadratic terms in turn. Terms that fail the test for constant differentials are rejected, along with all branches occurring below them in the search tree. Terms that pass the constant difference test are retained, and the more fully specified functions occurring below them in the tree are considered by entertaining various possibilities for the next highest term in the function. Some of these are also rejected, but some are retained, and the process continues until one or more functions with zero residuals have been generated. By applying the test for constant differentials at each level of the search tree, this technique examines only a few of the many functional relations that it would otherwise have to consider.

Table 5. Determining the constant term in an equation.

X	$Y - 3X^2 - 2X$
1	1
3	1
6	1
10	1
15	1

A simple extension of this method lets BACON deal with more complex laws as well. In addition to considering polynomial relations between X and Y (or between any two terms), the system also considers relations between X and Y^2, X and Y^3, and so forth. This enables the discovery of Kepler's third law (in the form $P^2 = aD^3$), as well as many other numeric relations. Finally, the system can also consider transformations of both the independent and dependent terms, such as inverse(Y), sine(Y), log(Y), and so on. The first of these lets the system discover Ohm's law using the differencing method; in this case, the law is stated as $I^{-1} = aL + b$, where $a = \text{voltage}^{-1}$ and $b = \text{resistance/voltage}$. We have not yet uncovered any heuristics to direct search through the space of possible transformations, so BACON.5 considers each transformation in turn, using a simple-minded generate and test strategy. However, the user can direct the system's attention by including some transformations and excluding others.

The differencing heuristic as implemented in BACON.5 can also deal with some degree of noise, though the method has certain limitations. Rather than requiring a constant differential to be found, the system is satisfied with *near* constant values. The meaning of *near* is determined by two parameters — the *relative* error R associated with a dependent term, and the *absolute* error A associated with that term. These parameters are used to construct an interval around the observed mean of a given differential. If all values fall within that interval, BACON infers that any divergence from the mean is due to noise of one form or another.

For instance, consider the values of X and Y in Table 6. The values of Y are identical to those in Table 2, except that some variation from the law $Y = 3X^2 + 2X + 1$ has been included. Suppose that BACON is given a relative error of 0.07 for Y, along with an absolute error of 0.001. Upon examining the values of Y, the system finds a mean of 240. Multiplying this mean by the relative error gives the product 16.8, from which BACON creates the interval (223.2, 256.8) by adding and subtracting this amount from the mean. (In this case, the absolute error has negligible influence, although it becomes important when values approach zero.) Since all of the Y values fall outside this interval, the program infers that Y is not constant. When the values of the first differential are computed, the mean 43.29 results. Since these values are based on pairs of the original data points, noise can be confounded, so both the relative and absolute error terms are doubled,[3] giving the interval (37.23, 49.35). Again, most results lie outside this interval, so the process is repeated. When the second differentials are calculated, the mean 3.22 is obtained. The error terms are again doubled, since the second differentials further compound the potential noise, leading to the interval (2.31, 4.13). All of the values fall within this interval, so BACON concludes that the second differential has the near-constant value 3.22. This process is repeated after the X^2 term has been subtracted, to estimate the other coefficients of the equation.

Table 6. Noisy data obeying the law $Y = 3X^2 + 2X + 1$.

X	Y	Y'	Y''
1	6.15	13.50	
3	33.15	30.29	3.36
6	124.03	47.24	2.42
10	312.98	82.14	3.88
15	723.65		

When BACON transforms the dependent values to consider more complex relations such as $\log(Y) = f(X)$ and $Y^{-2} = g(X)$, the absolute error term A must be transformed as well. If we let M represent the mean of the observed dependent values and M_t stand for the mean of the transformed dependent values, then the transformed absolute error term $A_t = AM_t/M$. This formula guarantees that the ratio A/M will remain constant across transformations. To see why this is necessary, consider the dependent term Y with mean 100, and the transformed variable Y^{-1} with mean 0.01. If the absolute error for Y is 0.1, we certainly do not want to use the same error term for the much smaller values of Y^{-1}. In this case, the transformed error term would be $(0.1)(0.01)/100 = 0.0001$, which seems much more reasonable. Fortunately, there is no need to alter the relative error term R when transformations are employed.

How robust is the differencing method with respect to noise? As long as polynomials with only a single term are involved, such as $Y = aX^3$ or $Y = bX$, it performs quite well. However, recall that in order to estimate the parameters for more complex laws, BACON must first estimate one parameter, subtract out some of the variance, then estimate the second parameter, and so forth. Thus, in estimating the parameters for the

[3] This is a conservative estimate. Since each value of Y may be either $RY + A$ too high or too low, it is possible that the difference of two Y values will be $2(RY + A)$ too high or two low.

law Y = aX^2 + bX -4- c, the system would first determine a, then use this estimate in determining b, and employ both estimates in finding the value of c. Unfortunately, small errors in the estimation of the quadratic term a can seriously affect the estimate of the linear term 6, leading to even greater effects for the estimate of the constant term *c.* Thus, while the differencing technique has some abilities for handling noise, a better method would be desirable, and we examine such an approach below.

2.3. Hill-Climbing through the Space of Parameters

Both of the above methods carry out search through a space of functional relations, though they explore somewhat different spaces and certainly employ different operators for generating new problem states. However, in both cases, these operators actually use the data in generating the new state. Thus, the two methods may be characterized as *data-driven.* Unfortunately, this reliance on the data leads to difficulties when significant noise is present, and to deal with such situations, we have explored (in the BACON.6 program) a more enumerativc approach to finding numeric laws.

In this method, the user provides BACON with one or *move forms* of law that it should consider in attempting to summarize the data. For instance, the system might be told to examine laws of the form Y = aX^2 + bX + c, as well as those having the form sin(Y) = aX + b (where X is independent and Y is dependent). These forms define the space of laws that BACON should consider in its search for numeric relations. Based on each of the forms, the system generates a set of initial states from which to begin the search. These states are simply instantiated versions of the abstract forms with 1, 0, or —1 inserted for the parameters. For instance, given the form Y = aX^2 + bX + c, BACON generates nine initial states: [a=l, b=l], [a=l, b=0]f [a=l, b= -1], [a=0,"b=l], [a=0, b=0], [a=0, b= -1], [a= - 1 , b=l], [a=-l, b=0], and [a= —1, b=—1]. These values are chosen because they are well-distributed throughout the space of parameters, so that the optimum point should lie near one of them. The constant term c is not included, since its value can be computed once the quadratic and linear terms have been estimated.

Starting from these points, BACON carries out a form of hill-climbing through a k-dimensional space (in this case k = 2). At the outset, the system evaluates each of the initial points in terms of its ability to summarize die observed data. This is accomplished by substituting each combination of parameter values into the form, and predicting the value of the dependent term Y for each value of the independent term X. The system then computes the *correlation* of the predicted and observed Y values; this is required since we are concerned with relative parameter values rather than absolute ones. This process is repeated for each of the initial points in the space, and the N highest scoring parameter sets are selected for further exploration.

For each of the N points that are retained, BACON generates a new set of points by adding 0.5 to each of the parameter values. In addition, a second set of points is generated by subtracting 0.5 from the same points. For example, if the initial points [a = 1, b = l]and[a=0, b= -1] had been retained (with N = 2), then twelve additional points would be produced: [a=1.5, b = l], [a = l, b = 1.5], [a = 1.5, b = 1.5], [a = 0.5, b= -1], [a=0, b=-0.5], and [a=0.5, b=-0.5] by adding 6.5; as well as [a = 0.5, b = l], [a = l, b = 0.5], [a=0.5, b = 0.5], [a= -0.5, b= -1], [a = 0, b= -1.5], and [a= -0.5, b= -1.5] by subtracting 0.5. Each of these points is evaluated in turn, and compared to the points from which they were generated. As before, the N best parameter combinations are selected, based on their ability to predict the observed Y values in terms of the X values.

Using the N new points, the process is repeated, this time adding and subtracting the value 0.25 from each of the parameter values. Again each of the resulting points is evaluated, with the N best combinations being retained for use on the next cycle. In this way, BACON continues to improve its ability to predict the observed data, using a method of successive approximation. This beam-search version of hill-climbing proceeds, with the size of the step being halved on each cycle, until the step size falls below a user-specified level. At this point, the system stops its search, and for each of the N best parameter sets, the program generates a specific equation that can be used to predict Y in terms of the X values.

Table 7 summarizes the path taken in summarizing noise-free data obeying the polynomial $Y = 3.1X^2 + 2.35X + 1.0$, using a beam size of one. At the outset, the system generated the nine initial states given above, finding that the pair [a = 1, b = 1] best predicted the observed data. On the next cycle, BACON used a step size of 0.5 to produce six new states, this time finding the pair [a = 1.5, a = 1.0] to have the highest score. This process continued until the system reached a step size of 0.03125, the user-specified condition for halting. At this point, the best set of parameters was [a = 1.03125, b = 0.78125], which accounted for 0.9999999999 of the variance. In other words, the Y values predicted by these parameters were highly correlated with the observed Y values. However, BACON still had to compute the actual values for these parameters, along with the value for the constant term in the equation. This was easily done using the coefficients in the regression equation relating the observed and predicted values of Y, and BACON generated the final law $Y = 3.1001\ X^2 + 2.3485\ X + 1.0031$. Although the estimated coefficients are not identical to those from which the data were generated, they are remarkably close.

Table 7. Beam search discovery of the law $Y = 3.1001\ X^2 + 2.3485\ X + 1.0031$.

Step size	Quadratic term a	Linear term b	Correlation
1.00000	1.0000	1.00000	0.9999827810
0.50000	1.5000	1.00000	0.9999974065
0.25000	1.2500	1.00000	0.9999994657
0.12500	1.1250	0.87500	0.9999998815
0.06250	1.06250	0.81250	0.9999999865
0.03125	1.03125	0.78125	0.9999999999

This method is quite robust with respect to noise, since it uses the data only to test hypotheses rather than to generate them. When noisy data are involved, the path taken is very similar to the noise-free case we have just examined. The main difference is that the correlations used in evaluating parameter combinations never reach the same heights, since the observed Y values can never be completely predicted by the X values. The greater the noise in the data, the lower the final score that BACON must accept. Still, the evaluation function generally leads the system to parameter estimates that closely approximate the correct values. Unlike the differencing technique, this method does not require the user to provide an estimate of the amount of noise in the data; it arrives at the best parameters for summarizing the data, whether these are very good or very poor predictors.

However, note that if the user has provided a number of possible forms, the system must still select between these competitors. To this end, BACON employs a second evaluation function. First, the system computes the complexity C of each hypothesis (measured by the number of non-zero terms in the expression) and the variance V explained by the rule. It then combines these values into the ratio V/C, a function that improves with better fits to the data, and decreases as hypotheses become more complex. Once the scores have been computed, BACON finds the best score and multiplies it by a user-specified system parameter; the values of this parameter must fall between zero and one (we have used 0.8 in our runs). The resulting score is treated as a threshold; all hypotheses having scores below this limit are rejected, while those with scores exceeding it are retained. This strategy has a useful property. If one or a few hypotheses are clearly superior to the others, only these are retained; however, if many of the hypotheses are basically equivalent, then all of these are kept. In our experiments with noisy data, this approach has proven much more robust than the earlier data-driven methods.

2.4. Expecting Similar Relations

The heuristics we have examined so far all rely heavily on the observed data to direct their search through the space of functional relations. However, once the system has discovered a law in one context, it makes sense to use that information to direct the search process in related contexts. To this end, we have introduced certain *expectation-driven* heuristics (in BACON.5 and BACON.6) that take advantage of early findings to aid the discovery process at later points. In this way, BACON can reduce its search without any loss in generality, since the particular class of hypotheses that the system considers depends on those found earlier by the data-dependent methods. This contrasts with the type of domain-specific expectations found in most experts systems, which greatly constrain these systems' range of application.

The simplest of BACON's expectation-driven heuristics proposes that if the system has found a law in one context (i.e., when the independent terms not included in the law are held constant), it should expect a similar *form* of law to hold in a new context (i.e., when those terms take on different values). For example, once the system has discovered that Kepler's third law holds for the planets orbiting the sun, it could employ this *similar relations* heuristic to predict an analogous law for the moons of Jupiter. Specifically, if the law $D^3 = 1.0P^2$ were found in the first situation (when the sun was held constant), BACON would expect that a law of the form $D^3 = kP^2$ would hold in the new case (when the sun was replaced by Jupiter), though it would not yet know the value of the parameter k. In the case of BACON.5's differencing method, such a prediction lets the system immediately consider polynomial functions of D^3, rather than considering functions of D, D^{-1}, D^2, and D^{-2}, which would normally be tried first. In the case of BACON.6's hill-climbing method, an additional savings occurs. Rather than searching the three-dimensional parameter space associated with the form $D^3 = aP^2 + bP + c$, the system can search the much simpler one-dimensional space to estimate the quadratic coefficient *a*, since it expects both the linear and constant terms to be zero.

The reader will recall that BACON.5's differencing method requires different numbers of observations to estimate the parameters for polynomials of different complexity. This leads to a second expectation-driven method that we will call the *data-reduction* heuristic. Initially, BACON.5 gathers more than the necessary number of observations to ensure that a law is correct. However, once the system expects a particular form of a law to hold, it can determine the number of observations necessary to estimate the desired parameters, and collects only the minimum number of observations necessary to complete its description of the current law. Thus, in the above example, BACON would need only three data points to determine the value of *k* for the Jovian moons. Of course, additional observations would be required if significant noise were present, but the principle of reduced data would remain.

This method leads to only minor computational savings for two-term laws, but for more complex multi-term relations (discussed in the following section), the savings can be quite significant. Thus, in discovering the ideal gas law, the standard version of BACON.5 (without the data-reduction heuristic) ran for some 35 CPU seconds. Using the data-reduction method, the program arrived at the same law in only 21 CPU seconds. Very similar results emerged with runs on Coulomb's law, another four-term relation with a somewhat different form.

3. Discovering Complex Numeric Laws

The methods described in the previous section can discover numeric relations between *two* variables, but more complex relations lie beyond their scope. For instance, one would like methods for discovering functions involving many variables, such as the ideal gas law and Coulomb's law of electric attraction. Upon closer examination, one finds that there exist two quite different situations in which one can attempt to discover complex laws. In the first case, one has experimental control over all but one of the terms, so that the traditional method of "varying one term at a time" can be used to separate the effects of each independent term on the dependent variable. This is the approach we explored in BACON.3 and successive versions of the system, and we discuss the basic method below. After this, we examine an extension of the method that employs knowledge of symmetry to reduce search through the space of laws. In the second situation, there is no experimental control over any of die observable terms, and one can only observe co-occurring values. Later in the section, we examine a method for dealing with such cases, based on a generalization of the heuristics used in BACON. 1. The distinction between experimental science and observational science seems a major one, and there is no a priori reason to expect that identical methods will prove useful in both contexts.

3.1. Recursing to Higher Levels of Description

In order to let BACON discover laws relating many numeric terms, we introduced another heuristic that let it summarize regularities at different *levels ofdescription*. This method operates when the system is given a number of terms over which it has experimental control BACON begins by holding all but one of the terms constant, and discovering a specific law in that context. The constant values found in this situation are stored along with the independent values for which they occurred. Different constants are found for different contexts, and when enough values have been found, the system treats them as dependent values at a higher level of description, and attempts to find a higher level relation. The system employs the same method to find the second level law as it did at the lower level. After a law at the second level has been found, the program recurses to still higher levels, until all of the independent terms have been incorporated into a unified law, and all of the data have been summarized.

BACON's discovery of the ideal gas law provides a useful example of this strategy. This law may be stated as PV = 8.32N(T — 273), where P is the pressure on a quantity of gas, the dependent term V is the volume of the gas, T is the temperature of the gas in degrees Celsius, and N is the quantity of gas in moles. In order to run an experiment, BACON must be provided with values for each independent term; let us suppose the system is told to examine N = 1, 2, and 3, T = 10,20, and 30 (Celsius), and P = 1000, 2000, and 3000. In discovering this law, BACON begins by holding N at 1 and T at 10, and varying the values of the pressure P, examining the resulting values of V in each case. Suppose that for P = 1000, 2000, and 3000, the program observes V = 236,1.18, and 0.78. Using one of the methods for finding simple laws we described in the last section, BACON arrives at the relation V"*^{1} = 0.000425 P. In addition, it introduces the theoretical term a as the coefficient of P in this equation. The constant value a = 0.000425 is stored with the values N = 1 and T = 10 for later use, and system moves on to a new experimental combination.

BACON'S next step is to continue holding N at 1, but to examine a different value of T, say 20, and to find a new relation between P and V in this new context. Suppose in this situation, for P = 1000, 2000, and 3000, the system finds that V = 2.44,1,22, and 0.81. Although the same form V"^{1} = aP continues to hold, in this case it estimates that the parameter a = 0.000410, instead of the earlier value. This new value is stored with N =1 and T = 20 for future use. Analogous events occur when BACON considers T = 30, with the system finding that a = 0.000396. At this point, the program has three values for the term a, each associated with N = 1 and with a different value for T. Accordingly, it attempts to find a relation between the parameter a and the temperature T. Using the same method it used at the lower level, BACON finds the linear relation

a^{-i} = 8.32 T + 2271.4. It also defines b and c, two theoretical terms that correspond to the two parameters in this equation. 'The values b = 8.32 and c = 2271.4 are stored with N = 1 for later use, and the system continues.

This process is repeated for N = 2, with BACON finding different estimates of the parameter a, and relating them to the values of"T. In this case, it finds the relation a"^{1} = 16.64 T + 4542.7, and stores the values c = 16.64 and 4542.7 with N =2. Analogous events occur for N = 3, giving the value 24.96 for b and 6814.1 for c. At this stage, BACON has enough values of N and b to search for a relation between the two terms, and it arrives at the law b = 8.32 N. Similarly, it looks for a law relating N and the parameter c, this time finding c = 2271.4 N. In addition, it defines two terms d and c, having the values 8.32 and 2271.4, respectively. 'These two values arc not conditional upon the values of any other term (at least as far as BACON knows), and so the system halts, having summarized all of its data, and relating the terms V, P, T, and N to one another.

Let us examine the mapping between BACON'S laws and the ideal gas law. 'The first level law has the form V""'$^!$ = a P, while the second level law has the form a^{-1} = b T + c. Substituting (b T + c)"~^{1} for a in the first law, we get V""^{1} = (b T + c)~$^]$ P. From the two third level laws, we know that b = 8.32 N and c = 2271.4 N, and substituting for b and c, we obtain V"^{1} = (8.32 NT •+• 2271.4 N)""^{1} P. Dividing through by P, inverting both sides of the equation, and factoring out 8.32 N, we get PV = 8.32 N (T + 273). The standard version of the law is PV = 8.32 NT, where T is measured in degrees Kelvin. Since adding 273 degrees Celsius converts the Celsius scale into the Kelvin scale, the above equation is equivalent to the standard form of the ideal gas law. In a sense, BACON has generated its own measurement scale for the temperature, in order to state the relation succinctly. Table 8 summarizes the steps taken in this discovery, comparing BACON'S version of the law with the standard version, and showing the independent terms held constant at each level of description.

Table 8. Summary of ideal gas law discovery.

BACON's version	Standard Version	Constant Terms
V^aaP	PV = k	T,N
V""^{1} a P/(bT+c)	PV = k(T-273)	N
V-^{1} = P/(dNT + eN)	PV = 8.32N(T-273)	

Taken together, the heuristics for finding simple numeric relations and recursing to higher levels give BACON considerable power. Using these two strategies, the system has successfully rediscovered versions of Coulomb's law of electrical attraction, a complex version of Ohm's law, and the ideal gas law. The recursion heuristic has been somewhat modified in BAC0N.6 to aid the system in dealing with noise. Since more than one functional relation may be found acceptable by the hill-climbing method, the system must be able to store each of the resulting sets of parameters at higher levels of description. However, if a hypothesis is found to be acceptable in one context, but in a later context no analogous hypothesis is found to fit, the hypothesis is rejected and its associated parameters are removed from the higher level store. For example, suppose the laws Y = 3.0X^2 + 2.0 and log(Y) = 0.5X + 4.0 both fit the data fairly well when Z = 1.0. As a result, the parameter values 3.0, 2.0, 0.5, and 4.0 would be stored at the second level of description for later use. Now suppose that when Z = 2.0, only the law log(Y) = 0.75X + 6.0 obeys the data well enough to be accepted. Since it does not appear to be general, the first law relating Y to X would be rejected as a summary of the initial data, and the parameter values 3.0 and 2.0 would be removed. One can imagine more lenient versions of this strategy in which a few exceptions were allowed, but some approach of this type that tests the generality of functional forms would appear to be very useful.

3.2. Discovering Symmetrical Laws

Although the method of varying one term at a time considerably simplifies the task of discovering complex laws, it can require considerable amounts of data to be gathered. One way to avoid this is to employ expectation-driven heuristics to reduce the amount of data required to identify a complex law. For example, the notion of symmetry has played an important role in the history of physics, and one might well use expectations of symmetry to constrain search through the space of laws. Table 9 presents three well-known laws that exhibit symmetry — Snell's law of refraction, conservation of momentum, and Black's heat law.

Although BACON.3 and its successors can discover these laws with just the heuristics we have already described, the inclusion of a new heuristic that postulates symmetry significantly reduces the search required to find the relations. This heuristic applies whenever BACON is asked to run an experiment involving two objects that have the same set of associated variables. The method first varies in turn all terms associated with the first object, and finds an Nth level description of the relationship between these terms. Once the form of this law is known, the heuristic *assumes* that the same function will relate the terms associated with the second object. It then computes the values of these two higher level functions for a number of situations, and checks to see if they are linearly related. If so, the symmetry assumption is verified, and the two terms are combined into the final law.

Table 9. Symmetrical laws discovered by BACON.

Snell's law of refraction	$\text{sine } \theta_1/n_1 = \text{sine } \theta_2/n_2$
Conservation of momentum	$m_1(V_1 - U_1) = -m_2(V_2 - U_2)$
Black's specific heat law	$c_1M_1(T_1 - F_1) = -c_2M_2(T_2 - F_2)$

As an example, consider BACON's discovery of Snell's law of refraction, as summarized in Table 10. The program starts with two objects and two variables associated with each object — the medium through which light passes, and the sine of the angle the light takes. Varying medium$_2$ and holding medium$_1$ and sineθ_1 constant, the system postulates an intrinsic property, n_2, whose values are associated with different media. Of course, the ratio sine θ_2/n_2 has the constant value 1.0. At this point, BACON relates the terms associated with the second object, and decides that it should examine the values of sine θ_1/n_1 and relate them to the former ratio. Upon gathering additional data, the program discovers that the two ratios are identical, or that sine θ_1/n_1 = sine θ_2/n_2, which is one statement of Snell's law.

The BACON system has discovered two other symmetrical relations — conservation of momentum and Black's specific heat law — following very similar paths. Table 9 presents the full form of the laws; directly observable terms are shown in upper case, while intrinsic properties (discussed in the following section) are shown in lower case. We may have given the impression that the symmetry heuristic eliminates search entirely, but this is not the case. Although the set of hypotheses considered is drastically reduced, it is not always narrowed to a single function. Only one symmetrical hypothesis emerges in Snell's law, since the relation that is found combining the first set of terms involves only one parameter. In discovering both Black's law and conservation of momentum, two parameters occur, so the enhanced version of BACON considers two possible symmetries. For both laws, one of these symmetries is found to satisfy the observed data, while the other is rejected. However, because the system has strong expectations, it can test its hypotheses against much less data than would be necessary using the purely data-driven approach.

In the case of symmetrical laws, we have another example of expectation-driven heuristics and their use in reducing search through the space of possible laws. In rediscovering Snell's law, the data-driven version of

BACON.5 required some 40 CPU seconds. However, when the symmetry heuristic was included (along with the similar relations and data reduction heuristics), the system found the same law in only 5 CPU seconds. Even greater savings occurred for Black's heat law, since this involved a total of eight terms. In this case, the data-driven version took 8433 CPU seconds, while the expectation-driven version (using the symmetry heuristic) required only 23 CPU seconds, an improvement of more than two orders of magnitude. Moreover, the symmetry heuristic accomplished this reduction with little loss in generality, since symmetry relations can be found in a wide variety of scientific domains.

Table 10. Discovering Snell's law of refraction.

MEDIUM_1	$\text{SIN } \theta_1$	MEDIUM_2	$\text{SIN } \theta_2$	n_2	$\text{SIN } \theta_2/n_2$
VACUUM	0.25	WATER	0.33	0.33	1.0
VACUUM	0.25	OIL	0.37	0.37	1.0
VACUUM	0.25	GLASS	0.42	0.42	1.0

3.3. Finding Observational Laws

The numeric heuristics we have considered, together with the method of recursing to higher levels of description, are very useful for finding laws when some of the variables are under experimental control. However, since this approach relies on the ability to vary the values of one term while holding the others constant, it cannot be used to discover relations in purely observational data. Interestingly, a modified version of BACON's first numeric heuristics, which we described in section 2.1, can be be applied in such cases. The modification is a simple one: rather than looking for monotonic trends, one looks at *correlations* between terms. Thus if one found X and Y to be positively correlated, their ratio would be defined as a new term, while if they were negatively correlated, their product would be considered. Since correlations are used, one can apply this revised method to observational data involving many covarying terms. Of course, any pair of terms has some correlation, so one must have some means for directing search through the space of new terms. Our solution is to carry out a beam search, in which only the N highest correlations are used to define new terms. The values of these terms are computed, as well as their correlations with other terms, and the process is repeated. As more complex terms are defined (redundant terms are eliminated), the best correlations approach 1 or -1, until eventually a near linear relation is found and a law is formulated. The system continues until all terms have been incorporated into a law, or until the terms become too complex.

We have implemented this approach to observational discovery, and Table 11 presents a trace of the system discovering the law $XY/WZ = 1$, using a beam size of two. The program begins by finding the pairwise correlations between the four observable terms X, Y, Z, and W. Since the two highest correlations (in terms of absolute values) occur between W and Z (-0.5617) and between Y and X (-0.5099), these are used as the basis for two new terms. Since both correlations are negative, the two products XY and WZ are defined, and the system computes their values. Next, the correlations between the new terms and the directly observable terms are calculated, and the highest scores are noted. The two highest correlations are between W and XY, and between W and WZ. Since both of these are positive, the products XY/W and WZ/W are defined. However, since the second of these is equivalent to the observable term Z, it is dropped from consideration and the next highest correlation is used. Since the pair Y and XY leads to XY/Y, which equals X, this ratio is rejected and a fourth pair is considered. This time the pair Y and WZ leads to the ratio WZ/Y, which is a genuinely new term. When the third round of correlations is computed using XY/W and WZ/Y, the system finds two relations that exceed its threshold for actual laws. First, it finds the correlation between XY/W and Z to be 1.0, leading to the law $XY/W = 1.0Z$, or $XY/ZW = 1.0$. Second, it finds an identical correlation between WZ/Y and X, producing the law WZ/Y = 1.0X, which can also be transformed into XY/ZW = 1.0. The current version of the system does not realize that these laws are identical, but it does know that once all known terms have been incorporated into a law, it can halt its search through the space of products and ratios.

Table 11. Discovering the law XY/WZ = 1 using observational data.

```
The correlation between W and Z is -0.5617
The correlation between W and Y is  0.4704
The correlation between W and X is  0.3161
The correlation between Z and Y is -0.0157
The correlation between Z and X is -0.1234
The correlation between Y and X is -0.5099

Defining XY as the product of X and Y
Defining WZ as the product of W and Z

The correlation between W and XY is  0.8907
The correlation between W and WZ is  0.8907
The correlation between Z and XY is -0.2343
The correlation between Z and WZ is -0.2343
The correlation between Y and XY is  0.6484
The correlation between Y and WZ is  0.6484
The correlation between X and XY is  0.2457
The correlation between X and WZ is  0.2457

Defining XY/W as the ratio of XY and W
Defining WZ/Y as the ratio of WZ and Y

The correlation between W and WZ/Y is  0.3161
The correlation between W and XY/W is -0.5617
The correlation between Z and WZ/Y is -0.1234
The correlation between Y and WZ/Y is -0.5099
The correlation between Y and XY/W is -0.0157
The correlation between X and XY/W is -0.1234
I've found a law: XY/W = 1.0Z
I've found a law: WZ/Y = 1.0X
```

We have not incorporated this correlational method into BACON, since that system is so oriented toward dealing with experimental data. However, the method can discover many of the laws found by BACON even without experimental control over the variables involved. For instance, the method discovers Kepler's third law in much the same way as BACON, and it can find the ideal gas law, though it does so at a single level of description. In addition, since the technique uses correlations to direct its search through the space of terms, it is quite robust with respect to with noise. Although this approach cannot arrive at the polynomial relations found by the differencing method, it is a very promising approach to observational discovery that we plan to explore further.

4. Postulating Intrinsic Properties

The heuristics we have examined are appropriate for finding relations between numeric variables, such as distance and current, but they cannot by themselves deal with situations involving *nominal* or symbolic terms. In particular, suppose one's dependent variables are numeric, while the independent terms take on only symbolic values. In this section, we describe a discovery method, first introduced in BACON.4, that deals with such cases. The approach involves postulating a new term, which we call an *intrinsic property*. It also involves inferring a set of numeric values for the new property, associating these values with the observed symbolic values, and retrieving these inferred values when appropriate. Once BACON has associated a set of numeric intrinsic values with a set of nominal values, it can apply its numeric heuristics to discover new laws. We begin with a simple example in which intrinsic properties are required, and then examine a second case that led us to introduce a more conservative strategy for retrieving intrinsic values. After this, we consider the role of symmetry in situations involving intrinsic terms, and propose extensions to the method that handle cases in which multiple intrinsic properties occur.

4.1. Postulating Properties and Inferring Values

We saw earlier that BACON could rediscover Ohm's law when given numeric measures for external resistance (the length of the wire) and current. However, suppose we assume a slightly different situation in which the dependent variable I (the current) is still numeric, but in which the two independent terms – the battery and the wire used in the current – take on only nominal values. For instance, let us take three batteries – A, B, and C – and three wires – X, Y, and Z. These can be combined in different ways to generate different currents, but nothing is known directly about the batteries and wires except their identities. In this case, BACON could vary the battery and wire and observe the resulting current, but since the independent values are nominal, it would not be able to find a numeric law. Our solution is to let the system "invent" numeric terms that are associated with the batteries and the wires, such as voltage and resistance. We will call such terms *intrinsic properties*, since their values are associated with particular objects or sets of objects.

Let us examine the process of postulating intrinsic properties for our modified electrical experiment. Table 12 presents the currents for nine combinations of batteries and wires. These currents were computed by assuming the voltages $V_A = 4.613$, $V_B = 5.279$, $V_C = 7.382$ for the batteries, and the resistances $R_X = 1.327$, $R_Y = 0.946$, and $V_Z = 1.508$ for the wires. In addition, we assumed that the internal resistance for each battery was negligible. Upon examining the first three rows of the table (when the battery is A), BACON notes that the current I varies as the wire is varied. Since it cannot relate a numeric term to a nominal one, it postulates a new term (let us call it the *conductance c*), and bases its values on those of the current. Given the two numeric terms, any of the numeric discovery methods we have described would immediately find that they are linearly related with a slope of 1.0 and an intercept of zero. Of course, this is hardly surprising, since the values of c were defined to be those of the current I. However, the tautology disappears when BACON considers the next three rows (in which the battery is B). In this case, the system has already encountered the wires X, Y, and Z, so it retrieves their associated intrinsic values and compares them to the observed currents. This time BACON's numeric method also finds a linear relation with a zero intercept, but here the slope is 1.1444 rather than 1.0. An analogous law is discovered when the final three rows are examined, this time with 1.6003 as the slope.

Table 12. Postulating the property of conductance.

BATTERY	WIRE	CURRENT I	conductance c	voltage v
A	X	3.4763	3.4763	1.0
A	Y	4.8763	4.8763	1.0
A	Z	3.0590	3.0590	1.0
B	X	3.9781	3.4763	1.1444
B	Y	5.5803	4.8763	1.1444
B	Z	3.5007	3.0590	1.1444
C	X	5.5629	3.4763	1.6003
C	Y	7.8034	4.8763	1.6003
C	Z	4.8952	3.0590	1.6003

Once these three relations have been found, BACON uses the slopes of these lines to search for a relationship at the second level of description. However, the system again finds that it cannot relate a nominal variable (the battery) to a numeric term (the slope). Accordingly, a new, higher level intrinsic property (let us call it the *voltage v*)is created, with its values based on the slope values and associated with the different batteries. Again a tautological linear relation is found, but since no other independent terms exist to be varied, the system cannot move beyond this stage to discover empirically meaningful laws. At this point, BACON halts, having arrived at two intrinsic properties and their values for different objects. The values of the conductance c are associated with individual wires, while the values of the voltage v are associated with particular batteries. In addition, these terms are related to the current by the law $I/vc = 1$. Since the conductance c is the inverse of the resistance r, we can restate this relation as $I = v/r$, which is one form of Ohm's law for electric circuits. Moreover, the values obtained for v and r differ only by a constant factor from the values we used to compute the currents, meaning that BACON has effectively regenerated these values using only nominal values and their associated currents. The constant factor was introduced when BACON used the first set of currents as its values for the conductance, since the introduction of intrinsic properties involves the selection of a measurement scale along which nominal values can be ordered.

Note that intrinsic properties are useful only in cases involving at least *two* independent nominal terms. This is because the first set of dependent values must be used in defining the intrinsic values. Unless one or more additional sets of dependent values are observed, the law incorporating the new property will be tautological and have no predictive power. Thus, this method differs from the numeric techniques we have considered, since the latter can be used to find simple laws relating two variables, while the intrinsic property method applies only to the discovery of complex laws involving three or more terms, and multiple levels of description. Also note that upon achieving predictive power, any law involving an intrinsic property also acquires the ability to make *incorrect* predictions. This suggests a more conservative version of the method, to which we now turn.

4.2. Generalizing Conditions for Retrieval

The strategy described above works well for cases in which intrinsic properties are associated with *single* nominal variables, as conductance is associated with the wire and voltage with the battery. However, one can find cases in which an intrinsic property is instead associated with multiple nominal terms, and to deal with these situations BACON must use a more cautious strategy. As an example, let us consider the friction between two surfaces. Here we have two independent nominal terms, the composition of the first surface and the composition of the second, and one dependent numerical term, the friction observed when the two surfaces are placed in contact. Superficially, this arrangement is very similar to the battery-wire case, and one might expect to be able to postulate an intrinsic property associated with individual surfaces, and to use these values to predict the observed frictions.

Let us step through the strategy described earlier and see how it fares. As before, the system would begin by holding the first surface constant and varying the values of the second surface. Upon noting that the friction is different in each case, BACON would postulate an intrinsic term (let us call it F), and base its values on those of the observed friction. The program would also discover the tautological relationship between F and the friction, and store this information for later use at a higher level of description. BACON would then consider the same values for the second surface, this time using a different value for the first independent term. After observing the friction values, the system would retrieve the intrinsic values that it had associated with the three values of the second surface, and attempt to relate these values of F to the observed frictions. However, in this case, no relation can be discovered, nor can one be found when the first surface is again varied. In this case, associating the intrinsic values with the second surface alone was inappropriate. Instead, these values should be retrieved only when a particular *pair* of surfaces are involved.

In order to deal effectively with cases in which intrinsic values should be associated with sets of terms, BACON employs a more conservative strategy for retrieving intrinsic values. When a property is first postulated, the system assumes that all independent nominal terms are relevant, and so associates the various intrinsic values with a *conjunction* of the nominal values. For instance, given the first three rows of Table 12, BACON associates 3.4763 with both wire X and battery A, 4.8763 with wire Y and battery A, and 3.0590 with wire Z and battery A. When the battery is varied and the wires are reexamined, the system does *not* immediately retrieve the various conductances. However, it *does* attempt to relate the newly observed currents to the original conductances. Upon finding a linear relation, BACON infers that the battery does not affect the conductance, and removes it as a condition for retrieval. When the battery C is considered, the three conductances (now associated only with the wires) are immediately retrieved and related to the observed currents. In the friction example, no linear relation is found when the second set of values is examined, so the values of the first surface are retained as conditions for retrieving the values of F. New conditions are associated with the second set of F values, and still another set with the third. More complex examples are possible in which some nominal terms are relevant while others are not, and BACON's intrinsic property heuristics are general enough to deal with such cases.

The retrieval of intrinsic values under certain conditions can be viewed as a form of expectation-driven discovery, since the system uses knowledge it has gained in one context to aid discovery in a similar yet different context. The main difference is that with the numeric expectation-driven techniques, the *form* of some law is retrieved and used to reduce search, while in the intrinsic property method, a set of *values* are retrieved and used in the discovery process. Also, the numeric methods BACON employs do not require it to generalize the retrieval conditions on the forms before they are used. However, one can imagine a more conservative version of the system that required a particular form of law to prove itself useful in a number of contexts before being used with confidence. This leads us to return our attention to another form of expectation-driven discovery that can also be adapted to laws involving intrinsic properties.

4.3. Symmetry and Intrinsic Properties

We have seen how BACON can use the assumption of symmetry to drastically reduce both its search through the space of laws, and the amount of data it must gather. The notion of symmetry can also be applied to the intrinsic property method, leading to additional computational savings. Let us examine a case where both symmetry and intrinsic terms occur, such as Black's specific heat law. This may be stated as

$$T_f = (c_1M_1T_1 + c_2M_2T_2)/(c_1M_1 + c_2M_2),$$

where T_1 and T_2 are the initial temperatures of two liquids, M_1 and M_2 are the respective masses, c_1 and c_2 are the specific heats, and T_f is the final temperature after the two liquids have reached equilibrium. Specific heat is an intrinsic property associated with the particular (symbolic) type of liquid used, and must be inferred from the directly observable temperatures and masses.

In rediscovering Black's law, BACON varies the initial temperatures and masses, and relates these terms to the resulting final temperature. When it begins to vary the substances used for the first liquid in the experiment, it finds that different substances give laws with the same form but with different coefficients. Accordingly, the system postulates an intrinsic property whose values are based on these differences. When the second liquid is varied and the values of the first liquid are reexamined, BACON finds a linear relation between its first set of intrinsic values and the coefficients found in the new context. As a result, it generalizes the conditions for the retrieval of these intrinsic values, associating them only with the values of the first liquid.

This is the point at which the symmetry assumption comes into play. It is natural to assume that a substance will have the same specific heat, whether it is used as the first liquid in the experiment or as the second liquid. Thus, if the same symbolic values are used for analogous variables, and the program has associated intrinsic values with those symbols in one context, it retrieves them immediately should it require them in the other context. This strategy lets the system avoid postulating an entirely new property in such cases, which would lead it to tautological rather than empirical laws at the higher levels of description. This method works equally well for other symmetric laws, such as Snell's law of refraction, in which the intrinsic term *index of refraction* is inferred, and conservation of momentum, in which the concept of *inertial mass* is generated.

Table 13. Inferring multiple intrinsic properties.

S	T	D	j	m	b
X	A	1	1	1	0
X	B	2	2	1	0
X	C	3	3	1	0
Y	A	4	1	2	2
Y	B	6	2	2	2
Y	C	8	3	2	2
Z	A	4	1	3	1
Z	B	7	2	3	1
Z	C	10	3	3	1

Symmetry plays a different role in the intrinsic property method than in the purely numeric techniques. In the latter, the symmetry assumption lets BACON determine the actual *form* of the law with less search than it would otherwise require. With respect to intrinsic properties, the symmetry assumption simply allows the system to sidestep the generalization process, letting it retrieve intrinsic values in new contexts that would normally require additional observations. Of course, these two applications of symmetry can be used in conjunction, and this is precisely the course BACON follows when it encounters nominal terms in symmetrical situations.

4.4. Extending the Intrinsic Property Method

BACON's intrinsic property heuristics appear to be quite general, and have been used to rediscover Proust's law of definite proportions and a version of Archimedes' law of displacement, as well as the modified version of Ohm's law we have already considered. In addition, we saw that the notion of symmetry could be easily incorporated into the approach, leading to a number of other laws. However, the existing method does have its limitations. For example, it seems odd that BACON'S numeric heuristics arc able to discover complex functional relations, while the intrinsic property heuristic considers only linear relations with zero intercepts. In fact, as implemented, the intrinsic property method does not require *any* search; it entertains a single hypothesis, and if this fails, the system docs not generalize (as in the friction example). Fortunately, the method can be extended to deal with more complex circumstances, and the numeric techniques can be used to direct search through the space of possibilities.

Let us begin with an example that is only slightly more complicated than those we have already considered. Suppose we have two nominal independent terms S and T, along with the single dependent term D, and suppose we observe the values of D shown in Table 13. Since the values in the first three rows differ (when T is varied and S is held constant), we would posit an intrinsic property (say j), and base our initial set of intrinsic values on the observed values of D. When the second set of D values are observed, we find a linear relation between these values and the original set, but this case differs from those we have seen before in having a nonzero intercept. The natural way to deal with this situation is to create a property for the intercept (b) as well as for the slope (m), and to consider both terms when one searches for second level laws. As a result, one would postulate *two* intrinsic properties at the second level of description, basing their values on the values of m and b. This extension of the basic method should work equally well in cases involving quadratic and higher relations, though even more second level intrinsic properties could result in such cases.

In fact, relations involving nonzero intercepts and other complex relations will be found precisely when multiple intrinsic properties are involved. For instance, the data in Table 13 were generated by the law D = mj + b, where j is an intrinsic property associated with T, and m and b are independent intrinsic properties associated with S. Since the ability to infer multiple intrinsic properties would seem to be very useful, it is appropriate to consider the circumstances under which this extended method will succeed. The approach appears to work if, at a given level of description at which the nominal term Y is varied first and the nominal term X is varied second, and the dependent term D is affected, there exists a polynomial relation D = P(i), where i is a single intrinsic property associated with Y, and for which the coefficients of the polynomial are intrinsic properties associated with X. For instance, if m and n are intrinsic properties associated with X, and p and q are associated with Y, the method will work for the relations D = mp, D = mp + n, and even D = mp^2 + np. However, the method will fail if the relation D = mp + nq is involved, since it cannot handle interacting sets of intrinsic properties. Also note that the order in which terms are varied can be significant. If the data in Table 13 are rearranged so that S is varied before T, the method will also fail. Given this constraint, future versions of BACON may be forced to examine their data in different orders if they hope to uncover multiple intrinsic properties.

To summarize, we see that BACON'S ability to find complex numeric relations can be employed to discover multiple intrinsic properties. In general, any form of numeric law that the system can discover **can** also be used in assigning intrinsic values. Thus, the differencing method and the hill-climbing method **could** lead to intrinsic properties based on *transformations* of observable terms, since both can discover laws such as Y = a $\sin^2(X)$ 4- b sin(X) + c. For example, consider the data in Table 14, in which we again have two nominal terms (S and T) and a single dependent term (D). Upon looking for a relation between the first set of D values and the second set, no polynomial relation is apparent. Since no relation can be found between **the** directly observable terms, the next natural step would be to examine transformations of the two sets of values, and attempt to relate them.

In this case, a linear relation is found if we examine the *logarithms* of both terms, and a similar relation holds between the first and third sets of values. Based on this success, we would postulate an intrinsic property (i) and base its values on those of D. However, since a transformation of the second set of values was required to discover the relation, our final law will be simplified if we use log(D) for our intrinsic values, rather than the values of D themselves. This leads to a single higher level parameter, which can be expressed as log(D)/i, since we have a zero intercept. This term takes on different values for different values of S, and so leads to a single intrinsic value at the second level of description. The data in Table 14 were computed from the law $D = p^q$, where p is an intrinsic term associated with T, and q is similarly associated with S. The intrinsic values $p_a = 1$, $p_b = 2$, $p_c = 3$ and $q_x = 1$, $q_y = 2$, $q_z = 3$ were employed. Thus we see that by considering transformations, an extended version of BACON would be able to discover intrinsic properties involved in laws of a very different form than we have seen before. Now that we have introduced the notion of intrinsic properties and explored its implications in some detail, let us turn to another discovery method that builds on this concept.

Table 14. Basing intrinsic properties upon transformations.

S	T	D	log(D)	i	log(D)/i
X	A	2	0.301	0.301	1
X	B	3	0.477	0.477	1
X	C	4	0.602	0.602	1
Y	A	4	0.602	0.301	2
Y	B	9	0.954	0.477	2
Y	C	16	1.204	0.602	2
Z	A	8	0.903	0.301	3
Z	B	27	1.431	0.477	3
Z	C	64	1.806	0.602	3

5. Finding Common Divisors

The early versions of BACON were designed with laws from physics in mind, but after BACON.4 had been successfully tested on a number of such laws, we began to look for other applications. In examining the history of early chemistry, we found that BACON's heuristics for finding numeric relations and inferring intrinsic properties were necessary, but not sufficient, to discover many of the empirical laws in this domain. Closer examination revealed that early chemists also employed the notion of *common divisors* for a set of data, leading them to a number of laws that could not be found using the other heuristics in isolation.

For instance, John Dalton's law of multiple proportions (1808) states that if two elements can combine in different ways (leading to different compounds), then the combining weights will always occur in small integer proportions. Thus, 1.3 grams of oxygen combines with 1.0 gram of carbon to form the gas carbon monoxide, while 2.6 grams of oxygen combine with the same amount of carbon to generate carbon dioxide. In 1808, Joseph Gay-Lussac proposed a similar law for volumes, stating that the combining volumes of gases always occur in small integer multiples of one another. In 1815, William Prout hypothesized that the atomic weights of the elements were all multiples of the weight for hydrogen, suggesting that this was the basic building block of nature. Nearly fifty years later, integral values also figured prominently in Stanislao Cannizzaro's redetermination of the atomic weights. At least in the early days of quantitative chemistry, the notion of common divisors played a central role in the search for regularity.

5.1. Detecting Integer Relations

In order to account for these discoveries, we introduced a method (first used in BACON.4 [4, 5]) for noting common divisors in a set of data. This heuristic may be viewed as a special case of the intrinsic property heuristic, since it is applied whenever the latter is applied, but produces useful results only in some cases. Let us consider how BACON uses this method to rediscover Dalton's law of multiple proportions. Table 15 presents some data on the manner in which nitrogen combines with oxygen. Three of the independent terms – the first element (in this table always oxygen), the second element (in this table always nitrogen), and the resulting compound – take on nominal values, while the fourth independent term – the weight of the first element used in the reaction – and the single dependent term – the weight of the second element in the reaction – take on numeric values. Upon varying the first weight and observing its effect on the second weight, BACON finds linear relations and defines the ratio W_2/W_1, though different slopes occur for different values of the second element and the compound.

Up to this point, the system has used only its heuristics for finding numeric relations, and the relations it finds are equivalent to those first stated by J. L. Proust (1797) in his law of definite proportions. The constants for the various oxygen-nitrogen reactions are shown in Table 16, along with those for two oxygen-carbon reactions; this table shows BACON's second level summary of the original data. As we have seen, BACON treats such summaries as if they were data, and applies its heuristics to see if any higher level relations can be uncovered. In this case, since the independent terms are nominal and the dependent term is numeric, and since the values of W_2/W_1 are not constant, the program postulates an intrinsic property (let us call it i), and this is the point at which the common divisor detector comes into play. It examines the dependent values of the first three rows and notes that they have the common divisor 0.57. A similar discovery is made for the carbon reactions, though the divisor is 1.33 in this case.

Table 15. Determining the combining weights for reactions.

ELEMENT_1	ELEMENT_2	COMPOUND	W_1	W_2	W_2/W_1
OXYGEN	NITROGEN	NITRIC OXIDE	1.0	1.14	1.14
OXYGEN	NITROGEN	NITRIC OXIDE	2.0	2.28	1.14
OXYGEN	NITROGEN	NITRIC OXIDE	3.0	3.42	1.14
OXYGEN	NITROGEN	NITROUS OXIDE	1.0	0.57	0.57
OXYGEN	NITROGEN	NITROUS OXIDE	2.0	1.14	0.57
OXYGEN	NITROGEN	NITROUS OXIDE	3.0	1.71	0.57
OXYGEN	NITROGEN	NITROGEN DIOXIDE	1.0	2.28	2.28
OXYGEN	NITROGEN	NITROGEN DIOXIDE	3.0	4.56	2.28
OXYGEN	NITROGEN	NITROGEN DIOXIDE	2.0	6.84	2.28

At this point, BACON diverges somewhat from the course it would normally take in storing intrinsic values. Rather than basing these intrinsic values directly on the dependent values, it divides the latter by the common divisor, giving a set of integers. Thus, the intrinsic value stored for the oxygen-nitrogen-nitric oxide triple would be 2, the value for oxygen-nitrogen-nitrous oxide would be 1, and that for oxygen-nitrogen-nitrogen dioxide would be 4. As a result, the values of the ratio W_2/W_1i become simply the observed common divisors, which are 0.57 for the oxygen-nitrogen pair and 1.33 for the oxygen-carbon pair. Otherwise, events proceed as they normally would. The system initially stores all independent values as conditions for retrieval of the integral intrinsic values, and checks to see if these conditions should be generalized. In this case, no generalization can be made, since the values of the compound are never repeated for different values of the elements. Although this makes non-tautological laws impossible at the current level of description, the fact that *different* values occur for the ratio W_2/Wji make it possible for further discoveries to be made at higher levels. This approach to breaking out of the tautological loop is best illustrated with another example from the history of chemistry.

'Fable 16. Datum's law of multiple proportions.

ELEMENT^	ELEMENT$_2$	COMPOUND	W_2/W_I	i	W_2/W_xi
OXYGEN	NITROGEN	NITRIC OXIDE	1.14	2.0	0.57
OXYGHN	NITROGEN	NITROUS OXIDE	0.57	1.0	0.57
OXYGEN	NITROGEN	NITROGEN DIOXIDE	2.28	4.0	0.57
OXYGEN	CARBON	CARBON MONOXIDE	1.33	1.0	1.33
OXYGEN	CARBON	CARBON DIOXIDE	2.66	2.0	1.33

5.2. Complex Laws Involving Common Divisors

Table 17 presents a slightly different formulation of the standard chemical experiment. In this case we told BACON to vary only one of the elements entering the reaction, along with the resulting compound and the weight of the element used in the reaction The first two of these terms are nominal, while the third is numeric, as is the single[4] dependent term — the volume V_c of the resulting compound. After gathering these data, BACON employed its numeric heuristics to find linear relations between W_e and V_c. Since these lines always had zero intercepts, we will focus on the slopes, which can be represented as W_e/V_c. Table 17 presents the slopes discovered for a number of element-compound pairs, which were then stored by BACON as second level summaries.

Table 17. BACON's rediscovery of Cannizzaro's law.

ELEMENT	COMPOUND	w/v_c	integer j	divisor W_e/VJ
HYDROGEN	WATER	0.0892	2.0	0.0446
HYDROGEN	AMMONIA	0.1338	3.0	0.0446
HYDROGEN	ETHYLENE	0.0892	2.0	0.0446
OXYGEN	NITROUS OXIDE	0.715	1.0	0.715
OXYGEN	SULFUR DIOXIDE	1.430	2.0	0.715
OXYGEN	CARBON DIOXIDE	1.430	2.0	0.715
NITROGEN	NITROUS OXIDE	**1.250**	2.0	0.625
NITROGEN	AMMONIA	0.625	1.0	0.625
NITROGEN	NITRIC OXIDE	0.625	1.0	0.625

Upon examining these higher level data, the system postulates an intrinsic property (let us call it j), and finds the dependent values to have common divisors. This leads to integers being associated with the various element-compound pairs, such as 2 for hydrogen-water, 3 for hydrogen-ammonia, and 2 for hydrogen-ethylene. These values correspond to the coefficients for the elements in the balanced equation for each

[4] In the actual run, BACON was also told to examine two other dependent terms - the weight of the compound and the volume of the element. The program found a number of additional relations involving these variables, including Gay-Lussac's law of combining volumes, but for the sake of brevity we will not focus on them here.

compound, though BACON does not interpret them in this fashion. In addition, the term W_e/V(j takes on the values of the common divisors that were found. As we noted before, these values are different, so that even though BACON cannot generalize the retrieval conditions on its intrinsic values (again because the compounds are never repeated), the potential for higher level discoveries remains. In this case, that potential is fulfilled, as shown in Table 18, which lists the third level summaries that result from BACON's endeavors. Given different values for W^V^j, the program postulates an intrinsic property (k), but a common divisor is again found in these higher level values. Integers are computed and associated with the elements; these integers are equivalent to the relative atomic weights found by Cannizy.aro in 1860. Moreover, BACON'S statement that these values have a common divisor is equivalent to Prout's hypothesis that all atomic weights are divisible by the weight of hydrogen (though again, it docs not interpret its finding in this manner).

Table 18. BACON's determination of relative atomic weights.

ELEMENT	divisor	k	divisor/k
HYDROGEN	0.0446	1.0	0.0446
OXYGEN	0.715	16.0	0.0446
NITROGEN	0.625	14.0	0.0446

5.3. Extending the Common Divisor Method

As with the intrinsic property method, the current version of the common divisor heuristic carries out very little search. It does consider the possibility that the inverses of a set of values will have a common divisor, rather than die values themselves, but one can imagine more sophisticated strategies. At first glance, it seems reasonable to extend the method to handle multiple properties in the same way we proposed for the basic intrinsic property method. However, this extension works only in cases where some relation can be found between two sets of dependent values, and these are precisely the situations where there is no need to look for common divisors. This is because one can generalize the retrieval conditions, and thus find non-tautological laws without resorting to postulating integer values for the intrinsic properties. The two methods are complementary, since the common divisor heuristic may lead to useful results even if the more basic approach fails.

However, the two other extensions we proposed earlier apply equally well to searching for integral relations. If neither the observed values of a dependent term D nor its inverse D""^{1} have a common divisor, then BACON could examine transformations of the term, such as log (D), sin (D), and D^2. Similarly, if multiple dependent terms X and Y are present, the system could consider combinations of these terms, such as XY, X/Y, X^2Y, and so forth. If no divisor was forthcoming, the program could even examine transformations of these terms, such as sin (XY), or even combinations of transformations, such as sin (X) log (Y). Of course, this would lead to vast search spaces, and unless we can find heuristics to direct search through these spaces, the chance of finding useful laws would be small. Fortunately, once a transformation or combination has been proved worthwhile in one context, BACON could immediately try it in analogous contexts, and so reduce search considerably. This is another instantiation of the expectation-driven approach to discovery that we first introduced in BACON.5.

It is interesting to note that, as far as we know, no truly complex common divisors have been found in the history of science. There are two possible explanations for this absence: (1) scientists are very good at selecting useful variables, so that such transformations and combinations are seldom necessary; or (2) such laws are so complex that scientists have simply never discovered them. In either case, we feel that an extended version of the common divisor method should be included in future versions of BACON, and tested on its ability to discover useful concepts and interesting empirical laws.

6. Discussion

In the previous pages we have examined the process of empirical discovery, focusing on a number of heuristics for this domain. We have implemented and tested these heuristics in successive versions of the BACON system, and we have discussed their capabilities and limitations in earlier sections of the paper We have also suggested some extensions to BACON'S discovery methods, such as altering the intrinsic property heuristics so that they consider multiple terms and transformations of directly observable terms. In this section we will also propose some directions for future research, though here we will focus on more global issues that transcend the particular methods that BACON employs. We will address three issues — the role of structural knowledge, the importance of qualitative laws, and die relation between quantitative empirical laws and theoretical explanations. In closing, we consider whether BACON should be classified as an expert system, despite the simplicity and generality of its discovery methods.

6.1. The Role of Structural Knowledge

In each of the versions of BACON we have described, the system relied on the programmer to suggest a set of independent and dependent terms that it should examine. Thus, extending the program to select potentially relevant variables is an obvious direction for future research. There is little doubt that research scientists employ domain-specific knowledge in deciding which variables to examine and which experiments to run, and if we hope to extend BACON in this direction, it will also have to accept and manipulate such domain knowledge. However, it would be very desirable to implement these components in a general manner, using a few simple reasoning methods that could operate on many different instances of domain knowledge.

The notion of structural knowledge suggests such a potentially general approach. Given a description of some physical or social situation, one can often reason about potential causes and effects, while having no knowledge of the particular equations governing the situation's behavior. For instance, suppose we have the description of a bridge in terms of the connections between various components. If we believe that forces can only be transmitted through adjacent objects, then we can make immediate inferences about which variables are directly related, and which are unrelated or indirectly related to one another. In addition to simplifying the search through the space of possible laws governing stable entities such as bridges, these inferences will also suggest which experiments to run, since for a given dependent variable, we expect certain terms to be relevant and others to be irrelevant.

The same type of reasoning occurs in economics and econometrics, where scientists begin with certain beliefs about direct connections between socio-economic variables, and use these beliefs to simplify the task of modeling large-scale human behavior. In this case, assuming that there is no direct connection between two variables (such as the unemployment rate and the price of wheat) is equivalent to assuming a zero coefficient in a set of complex simultaneous equations. Once enough such assumptions have been made, the set of equations can be solved and the parameters of the system can be estimated. In some cases, the causal assumptions can be quite general. For example, one naturally believes that a later event can never cause an earlier event, allowing one to rule out entire classes of potential relations. Since economics is an observational science, these inferences cannot be used to aid experimental design, but they can be very useful in directing search through the space of quantitative empirical laws.

BACON has already shown an ability to employ certain kinds of structural knowledge, for the symmetry assumption used in discovering conservation of momentum and other laws can be viewed as an example of such knowledge. The reader will recall that the assumption of symmetry was made in cases involving two analogous objects. The "structure" implicit in this assumption was that no direct causal connections occurred between the observable terms associated with each object. Rather, a single causal connection occurred between two instances of an inferred theoretical term, one associated with each object; moreover, this theoretical term could be expressed as some combination of an object's associated observable terms. Thus, the task of empirical discovery was reduced to finding the appropriate combination of observable terms, and determining which of the few possible symmetries actually summarized the data. However, the symmetry assumption did more than simply reduce BACON's search through the space of empirical laws. In addition, it led the program to alter its experimental designs so that it gathered much less data, since fewer observations were required to arrive at an acceptable law.

Unfortunately, BACON's use of symmetry was implemented procedurally rather than declaratively, so that it must be restated before we can begin to explore general methods for using structural knowledge to aid the discovery process. Still, the symmetry heuristic will act as a useful example in our attempts to implement more general methods, and it is encouraging to know that the BACON framework has the potential to incorporate such domain knowledge with its data-driven methods. We hope that the combination of data-driven and knowledge-driven discovery methods will lead to a more robust system than would be possible using either method in isolation.

6.2. The Importance of Qualitative Laws

Another approach to determining potentially relevant terms involves the notion of *qualitative* laws. Since qualitative laws are generally formulated before their quantitative counterparts, they are a likely source of knowledge for determining which variables to examine. Let us return to an example from the history of chemistry, and explore the relation between these two types of empirical laws. In an earlier section, we described BACON's rediscovery of Dalton's law of multiple proportions. During its data-gathering process, BACON varied the values of three nominal terms – the two elements entering a reaction, and the compound resulting from that reaction. In the run we described, the system was provided not only with the independent terms it should examine, but with their values as well.

Let us examine the sort of knowledge a discovery system might require to design this experiment on its own. Suppose the system had qualitative descriptions of various chemical reactions, such as (reacts inputs (oxygen nitrogen) outputs (nitric-oxide)) and (reacts inputs (oxygen nitrogen) outputs (nitrous-oxide)). Since the arguments of the predicate *reacts* can differ, these suggest obvious independent terms that the system can vary in an experiment. Now suppose that the system knew that only certain elements reacted with one another; using this information, it could limit itself to certain combinations that it knows will give results. Finally, suppose the system had placed certain substances (such as nitric-oxide and nitrous-oxide) into the same *class*, based on similar features (e.g., both result from reactions involving oxygen and nitrogen). Such a classification scheme, together with knowledge of potential variables and useful combinations of their values, could be used to generate an experimental design like that shown in Table 16.

Of course, if we must provide such domain-specific knowledge to the discovery system, we have not done much better than providing a complete experimental design. However, if the system could discover such qualitative knowledge on its own, and use this information in designing experiments, then this would be significant progress. Since BACON is designed for discovering quantitative empirical laws, one might need an entirely different system that could discover qualitative laws from facts such as the reactions shown above. However, the interaction between BACON and the proposed system would be quite direct, with the new program providing BACON with a basic plan for collecting data. Although the task of finding qualitative laws is interesting in its own right, we are much more interested in the potential for interaction between qualitative and quantitative discovery systems.

6.3. Empirical Laws and Explanations

A third avenue to constraining the search for empirical laws relies on the use of theoretical knowledge or explanations. Given some theory that accounts for a class of phenomena, one can often use this theory to predict those independent terms that will affect a given dependent variable, and in some cases, even predict the *form* of the relation. These predictions can then be tested empirically, providing evidence for the theory if they are borne out. For instance, Dalton's atomic theory can be used to predict and explain both the law of multiple proportions and Gay-Lussac's law of combining volumes, while Newton's theory of gravitation explains both Galileo's law for falling bodies and Kepler's three laws of planetary motion.

However, we encounter the same difficulty using theories to direct the search for empirical laws as we did using qualitative laws for this purpose. If we gave our discovery system detailed knowledge of some domain, we would be effectively building in its discoveries. This would be especially true for the examples given above, since these empirical laws were proposed *before* the theories that were eventually formulated to explain them. In many cases in the history of science, empirical laws were discovered first, and provided the raw material from which theories were constructed. Thus, an obvious direction for future research would be to develop a discovery system that generates such theories; this system would accept BACON's output – empirical laws – as its inputs, and search a space of theories – either structural (like the atomic theory) or mechanistic (like the kinetic theory) – which might explain these laws.

The details of such a theory-building system are far from clear, though some of BACON's current heuristics suggest interesting possibilities. For instance, the notion of common divisors leads naturally to structural models involving component particles, such as the atomic theory. Similarly, the notion of symmetry often seems associated with the conservation of some theoretical quantity, such as heat or momentum. The proposed system might have a small repertoire of theory *types*, each associated with some cue such as the discovery of common divisors or the discovery of a symmetrical law. The particular laws that were found could then be used to instantiate the prototypical theory, leading to a specific theory capable of explaining the empirical laws.

Of course, once such a theory has been forwarded, there is nothing to prevent a BACON-like system from using this knowledge to direct its search for new empirical laws. This would be very similar to BACON's existing expectation-based discovery methods, although the system's expectations would be based on rather more sophisticated grounds in this case. In other words, it may be possible to establish a feedback loop in the discovery process, with BACON finding an initial set of empirical laws using the techniques we have discussed, followed by a theory formation system using these laws to produce explanations, followed in turn by BACON using the resulting theories to find new empirical laws, and so forth. This approach is attractive because it potentially provides the search-reducing power of theory-based discovery without requiring the programmer to build in theoretical knowledge. When combined with the system for finding qualitative laws proposed above, we will have the beginnings of a truly integrated model for the process of scientific discovery.

6.4. Evaluating BACON

There is some question about how to evaluate the BACON systems. Our research goal has never been to model the historical discovery process in detail, though we have turned to the history of science for ideas on discovery methods and for tests of those methods. Neither have we focused on constructing a tool for scientific data analysis that could be used by present-day researchers, though one can imagine extensions of BACON that would be used in this manner. Rather, we have attempted to understand the general principles underlying scientific discovery, in particular the discovery of quantitative empirical laws. With respect to this goal, we feel that we have been quite successful, since we understand considerably more about this process than we did at the outset of our research some years ago. Moreover, the principles and methods we have uncovered appear both simple and general, criteria usually considered desirable for scientific theories.

Some colleagues have Suggested that BACON may be viewed as an expert system for the domain of empirical discovery. In fact, our concern with generality and simplicity was largely a reaction against the traditional expert system approach of building in considerable domain knowledge of great specificity. Still, IJACON does share certain characteristics with expert systems, and we should examine this relationship, however briefly. In particular, expert systems can be viewed as moving through potentially very large search spaces; however, their motion through these spaces is constrained by knowledge of the domain, so that very few states are actually visited.

BACON can also be viewed as moving through a large search space, in this case a space of empirical laws and theoretical concepts. On close examination, we found that BACON actually carried out very little search, since its heuFistics were generally powerful enough to lead to the optimum concepts and laws. Most of these heuristics were data-driven, so that if different data were observed, the system would follow quite different paths and discover quite distinct laws. However, this is no different from an expert system like DENDRAL [12], which follows different paths when given different input. Thus, on this dimension, BACON may profitably be viewed as an expert system concerned with empirical discovery. However, this does not detract from either the generality or the simplicity of its methods, and we plan to continue using these criteria in directing our future work on discovery.

Fortunately, the generality and simplicity of BACON's heuristics have not detracted from the system's power, and it has shown itself capable of finding laws that were very significant when first discovered centuries ago. It remains to be seen whether the most recent versions of the system, with their ability to deal with noise, can aid modern-day scientists in discovering new empirical laws, and this is another obvious direction for future work. However, based only on the historical examples covered in the previous pages, we may conclude that BACON has led to significant improvement in our understanding of empirical discovery, and we fully expect that it will lead to deeper insights in the years to come.

References

1. Langley, P. BACON.1: A general discovery system. Proceedings of the Second National Conference of the Canadian Society for Computational Studies of Intelligence, 1978, pp. 173-180.

2. Langley, P. *Descriptive Discovery Processes: Experiments in Baconian Science.* Ph.D. Th., Department of Psychology, Carnegie-Mellon University, 1979.

3. Langley, P. "Data-driven discovery of physical laws." *Cognitive Science 5* (1981), 31-54.

4. Bradshaw, G., Langley, P., and Simon, H. A. BACON.4: The discovery of intrinsic properties. Proceedings of the Third National Conference of the Canadian Society for Computational Studies of Intelligence, 1980, pp. 19-25.

5. Langley, P., Bradshaw, G., and Simon, H. A. Rediscovering chemistry with the BACON system. In *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., Tioga Press, Palo Alto, CA, 1983.

6. Langley, P., Bradshaw, G., and Simon, H. A. BACON.5: The discovery of conservation laws. Proceedings of the Seventh International Joint Conference on Artificial Intelligence, 1981, pp. 121-126.

7. Langley, P., Bradshaw, G., and Simon, H. A. Data-driven and expectation-driven discovery of empirical laws. Proceedings of the Fourth National Conference of the Canadian Society for Computational Studies of Intelligence, 1982, pp. 137-143.

8. Langley, P., Bradshaw, G., Zytkow, J., and Simon, H. A. Three facets of scientific discovery. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 1983, pp. 465-468.

9. Langley, P., Zytkow, J., Simon, H. A., and Bradshaw, G. L. The search for regularity: Four aspects of scientific discovery. In *Machine Learning, Volume 2*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., Tioga Press, Palo Alto, CA, 1984.

10. Gerwin, D. G. "Information processing, data inferences, and scientific generalization." *Behavioral Science 19* (1974), 314-325.

11. Collins, J. S. A regression analysis program incorporating heuristic term selection. In *Machine Intelligence 2*, E. Dale and D. Michie, Ed.,American Elsevier, New York, 1968.

12. Feigenbaum, E. A., Buchanan, B. G., and Lederberg, J. On generality and problem solving: A case study using the DENDRAL program. In *Machine Intelligence 6*, Edinburgh University Press, Edinburgh, 1971.