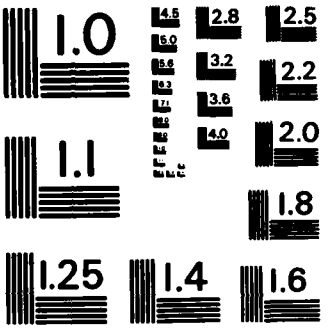
AD-A120 950 DATA-DRIVEN AND EXPECTATION-DRIVEN DISCOVERY OF EMPIRICAL LAWS(U) CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST P W LANGLEY ET AL. 10 CT 82 CMU-RI-TR-1 1/1

UNCLASSIFIED N00014-82-K-0168 F/G 9/2 NL

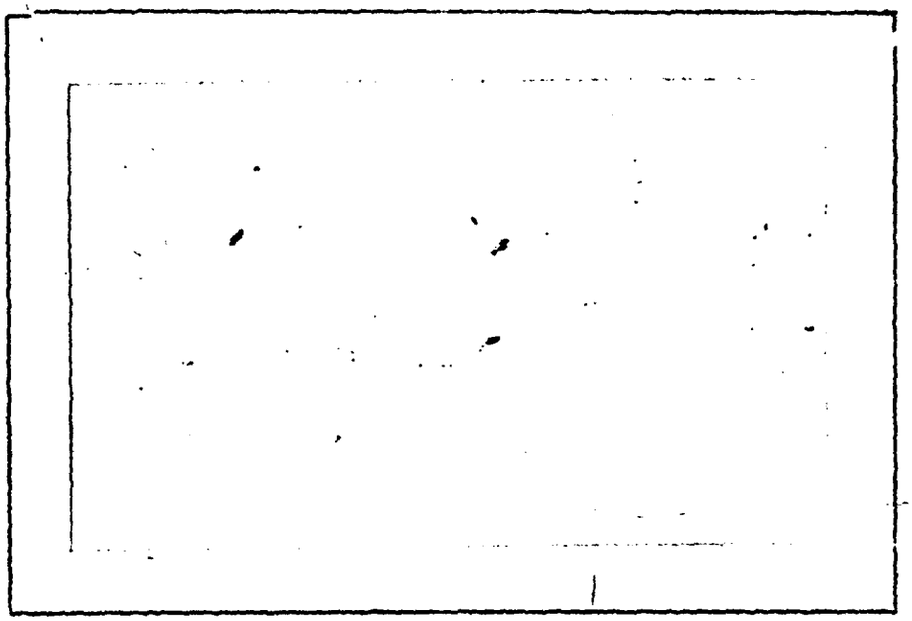
END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A



DEPARTMENT of PSYCHOLOGY

Carnegie-Mellon University

12

CIP Working Paper No. 433

Data-Driven and Expectation-Driven Discovery of Empirical Laws

Patrick W. Langley
Gary L. Bradshaw
Herbert A. Simon
The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report No. 1 (CMU-RI-TR-1)	2. GOVT ACCESSION NO.: AD-A120950
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE *(and Subtitle)*: Data-Driven and Expectation-Driven Discovery of Empirical Laws	5. TYPE OF REPORT & PERIOD COVERED: Interim Report 2/82-10/82
	6. PERFORMING ORG. REPORT NUMBER: CIP No. 433
7. AUTHOR(s): Patrick W. Langley, Gary L. Bradshaw, Herbert A. Simon	8. CONTRACT OR GRANT NUMBER(s): N00014-82-K-0168
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The Robotics Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 049-514
11. CONTROLLING OFFICE NAME AND ADDRESS: Information Sciences Division, Office of Naval Research, Arlington, Virginia 22217	12. REPORT DATE: 10 October, 1982
	13. NUMBER OF PAGES: 14
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

Presented at the Fourth National Conference of the Canadian Society for Computational Studies of Intelligence, Saskatoon, Saskatschewan, 1982

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

scientific discovery
physical laws
data-driven heuristics
expectation-driven heuristics
levels of description
intrinsic properties
common divisors
symmetrical laws

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

BACON.5 is a program that discovers empirical laws for summarizing data. The system incorporates four data-driven heuristics for relating numeric terms, recursing to higher levels of description, postulating intrinsic properties such as mass and specific heat, and finding common divisors. BACON.5 also includes expectation-driven strategies for directing search based on discoveries that the program has already made. These include heuristics for expecting similar forms of laws, reducing the amount of data that must be

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

1. Introduction

Scientific discovery is a complex, ill-defined activity, and one of the most profitable ways to study such phenomena is to construct intelligent programs that model them. In this paper we describe BACON.5, a program that discovers empirical laws for summarizing data. The core of this system is a set of general, data-driven heuristics for detecting numerical relations and proposing new terms to express these relations. However, the program also incorporates expectation-driven rules that let it take advantage of its earlier discoveries. Before moving on to describe the system in detail, we should first review some of the earlier Artificial Intelligence research on discovery, and outline the scope and limitations of the current project.

One of the earliest attempts to model scientific discovery was the simulation work of Gerwin [1]. Gerwin was interested in how humans could infer numerical laws or functions, given knowledge of specific data points. Of course, such ***descriptive discovery*** is only one part of the total scientific process. In order to understand this process, he gave subjects several sets of data and asked them to find the relationships which best summarized each data set. Using the verbal protocols collected from this task, Gerwin built a working simulation of the subjects' behaviors. The model first attempted to identify a general pattern in the data, such as a periodic trend with increasing amplitudes, or a monotonic decreasing trend. A class of functions was stored with each pattern the program could recognize; once a class was hypothesized, the system attempted to determine the specific function responsible for the data. If unexplained variance remained, the program treated the differences between the observed and predicted values as a new set of data. This procedure was used to elaborate the hypothesis until no pattern could be found in the residual data. The program also had the ability to backtrack if the latest addition to the rule failed to improve predictions. One limitation of Gerwin's simulation was that the program incorporated specific knowledge about the shapes of functions within a specified range. Therefore, these functions could not have variable parameters associated with them. Even though Gerwin's model could only solve a very restricted range of problems, it was an important step in understanding the discovery process.

Another early discovery system was DENDRAL [2], a program that identified organic molecules from mass spectrograms and nuclear magnetic resonances. The system identified chemical structures in three main stages – planning, generating plausible structures, and testing those structures. The first stage used patterns in the data to infer that certain familiar molecules were present. Considering these molecules as units drastically reduced the number of structures produced during the generation stage. This second phase used knowledge of valences, chemical stability, and user-specified constraints to generate all plausible chemical structures. In the final testing stage, the system predicted mass spectrograms for each of these structures, which were then ranked according to their agreement with the data. DENDRAL relied on considerable domain-specific knowledge, which was laboriously acquired through interaction with human experts in organic chemistry.

In order to reduce their dependence on human experts, the same researchers designed META-DENDRAL [3], a system that acquired knowledge of mass spectroscopy which could then be used by the DENDRAL program. META-DENDRAL was provided with known organic compounds and their associated mass spectrograms, from which it formulated rules to explain these data. Two types of events were used to explain spectrograms – ***cleavages*** in the bonds of a molecule and ***migrations*** of atoms from one site to another. Although plausible actions were determined using domain-specific chemical knowledge, the conditions on rules were found through a much more general technique [4]. META-DENDRAL has successfully discovered new rules of mass spectroscopy for three related families of organic molecules.

Lenat [5] has described AM, a system that has rediscovered important concepts from number theory. The program began with some 100 basic concepts such as ***sets***, ***lists***, ***equality***, and ***operations***, along with some 250 heuristics to direct the discovery process. These heuristics were responsible for filling the facets of concepts, suggesting new tasks, and creating new concepts based on existing ones. New tasks were ordered according to their ***interestingness***, with tasks proposed by a number of different heuristics tending to be more intersting than those proposed by a single rule. Using this measure to direct its search through the space of mathematical concepts, AM defined concepts for the integers, multiplication, divisors-of, prime numbers, and the unique factorization theorem. Like META-DENDRAL, Lenat's system incorporated some very general strategies, as well as some domain-specific knowledge about the field of mathematics.

In our work on BACON, we have attempted to develop a general purpose descriptive discovery system. Rather than relying on domain-dependent heuristics, as many of the earlier discovery systems have done, BACON incorporates weak yet general heuristics that can be applied to many different domains. The current version addresses only the descriptive component of scientific discovery. It does not attempt to construct explanations of phenomena, such as the atomic theory or the kinetic theory of gasses, but we will have more to say on this in a later section. Neither is the system meant to replicate the historical details of various scientific discoveries, though of course we find those details interesting. Instead, it is intended as a model of how discoveries ***might*** occur in these domains.

Descriptive discovery may take either of two basic forms: one may start from the data and use very general strategies to uncover regularities in those data; or one may bring certain expectations to the task and examine the data to see if they match those expectations. Earlier versions of BACON [6, 7, 8] relied entirely on ***data-driven*** discovery methods. The current version takes advantage of these heuristics, but also incorporates a number of ***expectation-driven*** discovery techniques. The latter take advantage of discoveries that have already been made to direct and simplify the search process in new situations. We have chosen to organize the paper around the system's discovery methods. Since the expectation-driven heuristics work with the results of the data-driven approaches, we will begin by focusing on the data-driven components and then move on to their expectation-driven

counterparts. Both types of heuristics are implemented as condition-action rules in Forgy's [9] OPS4 production system formalism.

2. Discovering Numeric Relations

BACON.5's most basic heuristic attempts to discover polynomial relations between two variables that take on numeric values. This rule computes the successive derivatives of one term with respect to the other, until it arrives at a set of constant values. The level of the constant derivative tells BACON the highest power necessary in the polynomial it seeks, while the constant determines the coefficient of this term. As in Gerwin's system, this component is subtracted out, and the technique is repeated on residual values. This process continues until all of the variance has been accounted for, and the program has determined the complete functional relation between the two variables.

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

Table 1. Determining the coefficient of a quadratic term.

As an example, let us consider BACON.5's use of this heuristic to discover the law $y = 3x^2 + 2x + 1$. The program begins by examining values of the dependent term y for different values of the independent term x, as shown in Table 1. Since y is not constant, the system computes the values of y', the first derivative with respect to x. In the table, the first value of y' is (34 - 6)/(3 - 1) = 14, while the second value is (121 - 34)/(6 - 3) = 29. Since these values are not constant either, BACON examines the second derivative y'', basing its computation on the values of y' and x. Thus, the first value of y'' is (29 - 14)/(6 - 1) = 3, while the second is (50 - 29)/(10 - 3) = 3. In this case, the program finds the constant value it seeks; this tells BACON that an x^2 term is present in the final law, and that its coefficient is 3.

However, more remains to be done before the discovery is complete. After subtracting out the $3x^2$ term, BACON attempts to relate the values of $y - 3x^2$ to the independent term x, as shown in Table 2. This time the first derivative is the constant 2, implying that an x term with a coefficient of 2 is also present in the final law. Subtracting this new component out as well, the constant value 1 immediately results, as we see in Table 3. BACON.5 includes this value as the final term in the law it has discovered, $y = 3x^2 + 2x + 1$, which completely summarizes the original set of observed data.

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

Table 2. Determining the coefficient of a linear term.

This method lets BACON.5 discover any of a large class of functions that can be expressed as polynomials with integer powers and real coefficients. In cases where no polynomial can be found, the system considers various powers of the dependent term, so that an even larger set of relations can be discovered. Thus, BACON can uncover relations such as $y^2 = 6.71x^3 + 4.23x$ and $y^{-1} = 3.5x^2$. The system entertains only one hypothesis at a time, and since simpler relations are considered before more complex ones, they are preferred if they are found to hold.

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

Table 3. Determining the constant term in an equation.

3. Recursing to Higher Levels of Description

By itself, the above differencing heuristic can discover numeric relations between *two* variables, but more complex relations lie beyond its scope. In order to find laws relating many terms, BACON.5 invokes a second data-driven heuristic that lets it summarize regularities at different *levels of description*. Upon discovering a law at one level, this method stores the coefficients from that law at the next higher level. Once enough of these higher level values have been gathered, BACON attempts to relate them to the independent term that was varied in each of the experiments. The system employs the same differencing technique to find the second level law as it did at lower levels. After a law at the second level has been found, the program recurses to still higher levels, until all of the data have been summarized.

BACON.5'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 uncovering this law, BACON first finds the relation $V^{-1} = aP$, where a is a parameter that varies with different values of T and N. Upon comparing the values of a and T, the system finds the law $a^{-1} = bT + c$, where b and c represent second level parameters that potentially vary with N. Finally, the program finds that $b = dN$, and that $c = eN$. Substituting these relations into the first law, we arrive at the equation $V^{-1} = P(dNT + eN)^{-1}$. BACON.5 calculates the value of d to be 8.32, and e to be -2271.36. When the factor 8.32 is divided out, e becomes -273, or the absolute zero point expressed in the Celsius scale. Thus, the equation is equivalent to the standard form of the ideal gas law. Table 4 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.

BACON'S VERSION	STANDARD VERSION	CONSTANT TERMS
1/V = aP	PV = k	N, T
1/V = P/(bT + c)	PV = k(T-273)	N
1/V = P/(dNT + eN)	PV = 8.32N(T-273)	

Table 4. Summary of ideal das law discovery.

Taken together, the heuristics for relating numeric terms and recursing to higher levels give BACON.5 considerable power. Using these two strategies, the system has successfully rediscovered versions of Coulomb's law of electrical attraction, Kepler's third law of planetary motion, and Ohm's law for electrical circuits. Table 5 presents the forms of these laws, along with that for the ideal gas law. Variables are shown in upper case, while coefficients are given in lower case. Superficially, the equations in the table have quite different forms, yet all can be expressed as combinations of the polynomial relations for which BACON searches.

Ideal gas law	$PV = rNT$
Coulomb's law	$F = aQ_1Q_2/D^2$
Kepler's third law	$D^3/P^2 = k$
Ohm's law	$v = rI + IL$

Table 5. Numeric laws discovered by BACON.5.

4. Postulating Intrinsic Properties

The heuristics we have discussed so far are fine for relating numeric terms, but they are of little use when an independent term takes on *nominal* or *symbolic* values. In such cases, BACON.5 draws on a third data-driven heuristic that postulates *intrinsic properties*. This rule associates the values

of the numeric dependent term with the nominal independent values, and retrieves them in later situations. In this context, BACON moves beyond the relatively simple process of curve fitting, and takes on some features of explanatory discovery.

For example, consider a version of Ohm's experiment in which the batteries and wires take on nominal values, so that one can distinguish between them but measure none of their characteristics. Ohm's law may be stated as $I = V/R$, where I is the current flowing through a circuit, V is the voltage associated with a wire, and R is the resistance of the wire. (We assume here that the internal resistance is negligible.) Table 6 presents data that might be gathered in an experiment with three batteries (A, B, and C) and three wires (X, Y, and Z). The values of the current were calculated on the assumption that $V_A = 4.613$, $V_B = 5.279$, $V_C = 7.382$, $R_X = 1.327$, $R_Y = 0.946$, and $V_Z = 1.508$.

BATTERY	WIRE	CURRENT	CONDUCTANCE	SLOPE
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

Table 6. Postulating the property of conductance.

Focusing on the first three rows of this table, BACON.5 finds that with the battery set to A and varying the wire, the current of the circuit varies as well. Since it cannot apply its numeric heuristic in this situation, the program proposes *conductance* as an intrinsic property of the wire, and bases the values of this new term on those of the current. Having done this, BACON can apply its differencing heuristic, and finds a linear relation between the current and the new property, with a slope of one. Of course, this is hardly surprising, since the conductance was defined so that this relation would hold.

However, upon varying the values of the battery, BACON retrieves the same values of the conductance in the new situations, as shown in the fourth through ninth rows. When these are compared to the currents, the system discovers other linear relations with different slopes. After recursing to a higher level of description, BACON uses these new parameters to postulate an intrinsic property associated with the battery, which we would call the *voltage*. The retrieval technique is actually stated as a separate heuristic, and shows more similarity to the expectation-driven heuristics we shall discuss later than to the data-driven ones. We have mentioned it here because the data-driven process of postulating an intrinsic property has little purpose without the ability to retrieve the associated values at later times.

Ohm's law	$v = rI$
Archimedes' law of displacement	$d = W/v$
The law of definite proportions	$k = W_e/W_c$

Table 7. Laws discovered with intrinsic properties.

Unfortunately, the discovery of intrinsic properties is more complex than we have made it appear. Some properties exist which are associated not with one, but with many, nominal terms. An obvious example is the coefficient of friction, which is a function of *pairs* of surfaces. To avoid difficulties in such cases, BACON.5 takes a conservative path by comparing different sets of intrinsic values. If a linear relation is found, the system generalizes and retrieves values as in the Ohm's law example. However, if no relation is found, it retains the additional conditions. Table 7 lists some of the laws rediscovered by BACON.5 that incorporate intrinsic properties. These include a version of Archimedes' law of displacement, in which the system computes the volumes of irregular solids as well as their density, and Proust's law of definite proportions, in which a constant weight ratio is associated with an element-compound pair.

5. Finding Common Divisors

The history of chemistry from 1800 to 1860 provides some additional examples of the discovery of intrinsic properties, with an interesting complication. In 1808, John Dalton set forth the ***law of simple proportions***, which stated that when two elements could combine to form different compounds, the weights contributed by one element for a constant weight of the other always occurred in ***small integer proportions*** to each other. In 1809, Joseph Gay-Lussac found evidence for his ***law of combining volumes***, which stated that a similar relation held for the relative volumes contributed by gaseous elements in chemical reactions. Again, in 1815, William Prout noted that the atomic weights of the known elements were all very nearly divisible by the weight of hydrogen. And finally, in 1860, Stanislao Cannizzaro pointed out that when a given element took part in different reactions, the ratios of the element's weight and the volume of the resulting compound always occurred in small integer proportions.

BACON.5 incorporates a fourth data-driven heuristic that enables it to discover these regularities in the chemical data. When the system is about to postulate a new intrinsic property, this rule examines the dependent values to see if they have a common divisor. If none can be found, then the process continues as described in the last section. However, if the numbers can be evenly divided, then the resulting integers are used as the intrinsic values instead of the original numbers. Also, the common divisor is associated with the terms that were held constant, instead of the *1.0* that would normally be used. This means that even in cases where BACON.5 cannot generalize and so retrieve a set of intrinsic values in a new situation, the common divisors let the system break out of the tautological circle and make further interesting discoveries.

ELEMENT	COMPOUND	W_E/V_C	INTEGER	DIVISOR
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	N_2O	0.715	1.0	0.715
OXYGEN	SO_2	1.430	2.0	0.715
OXYGEN	CO_2	1.430	2.0	0.715
NITROGEN	N_2O	1.250	2.0	0.625
NITROGEN	AMMONIA	0.625	1.0	0.625
NITROGEN	NO_2	0.625	1.0	0.625

Table 8. BACON.5's rediscovery of Cannizzaro's law.

Table 8 summarizes BACON.5's reformulation of Cannizzaro's discovery. The system is given control over two independent nominal terms - one of the elements entering into a reaction, and the resulting compound. The dependent variable is w_e/v_c, or the weight of the element used in the reaction, divided by the volume of the compound that results. For the element ***hydrogen***, different compounds lead to different values of w_e/v_c, so the system postulates an intrinsic property. However, the dependent values are all divisible by ***0.0446***, so the integers ***2***, ***3***, and ***2*** are used as the intrinsic values instead of the originals. This process is repeated with the elements ***oxygen*** and ***nitrogen***, but in these cases the divisors ***0.715*** and ***0.625*** are found instead. The integers in the table correspond to the coefficients on the given elements in the balanced equations for each reaction, while the divisors correspond to the relative atomic weights of the elements. When these divisors are carried along to the next level of description, BACON.5 also notes that they can all be divided by the value associated with ***hydrogen***; this statement is a variant on Prout's hypothesis. By searching for common divisors, BACON has replicated some of the major empirical discoveries of nineteenth century chemistry.

6. Expecting Similar Relations

We have now completed our survey of BACON.5's data-driven heuristics. The remainder of the system's strategies draw upon information gathered in this bottom-up manner to reduce search at later stages. Thus, when we speak of expectation-driven heuristics, we do not mean to imply that BACON starts with knowledge of a particular domain. Rather, we mean that the program is capable of taking advantage of discoveries it has made at early stages to simplify this process at later points.

The simplest of these heuristics proposes that if BACON.5 has found a law in one context (i.e., when certain variables 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, this ***similar relations*** heuristic could be used after the system has discovered Kepler's third law for the planets orbiting the

sun, to predict an analogous law to hold for the moons of Jupiter. Specifically, if the law $D^3 = 1.0P^2$ were found in the first situation, BACON.5 expects that a law of the form $D^3 = kP^2$ would hold in the new case, though it would not yet know the value of the parameter k. Such a prediction allows BACON to replace its search through the space of possible relationships between two variables with a simple calculation designed to test the expected relationship. If this relationship holds, BACON calculates the values of the unknown parameters and moves on to further discoveries.

Previous versions of BACON always utilized the same number of observations to find relationships between variables in its experiments. 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. Using this *data reduction* heuristic, BACON only collects the minimum number of observations necessary to complete its description of the current law. If D were being expressed as a function of P in the above example, BACON.5 would need only three data points to determine the value of k for the Jovian moons.[1]

Taken together, these two heuristics significantly reduce the program's search through both the space of data and the space of rules. The actual amount of savings depends on the number of superfluous data points. In order to evaluate the impact of the new heuristics, BACON was given six values of each independent variable in four separate discovery tasks. Performance of the purely data-driven system was compared to systems incorporating the expectation-driven heuristics, and is shown in Table 9. From this table, it can be seen that the similar relation heuristic only resulted in a small amount of savings. This result is somewhat misleading, because the amount of search required by the differencing technique was significantly reduced; however, the OPS4 interpreter was slowed by the inclusion of an additional condition-action rule, so the effect was masked. For more complex forms of laws, the computational savings would be greater.

	DD	DD + SR	DD + SR + DR
IDEAL GAS LAW	35	34	21
COULOMB'S LAW	35	35	23
OHM'S LAW	3	3	3
KEPLER'S THIRD LAW	3	3	3

Table 9. Time to discover numeric laws in CPU seconds.[2]

The present system employs a few simple heuristics for dealing with noise. In executing the differencing technique, BACON.5 checks the current derivative term to see if its values are constant.

[1] Of course, more would be required if significant noise were present, but the principle of reduced data would remain.

[2] DD = data-driven heuristics, DD + SR = data-driven and similar relation heuristics, DD + SR + DR = data-driven, similar relation, and data reduction heuristics.

All values which fall within a small interval of one another are accepted as equivalent. The program also calculates the number of outliers, or *exceptions* to the current relationship. If the number of exceptions is a small proportion of the total number of data points, BACON.5 decides the current term is constant, and updates its functional description. Although these methods allow BACON.5 to cope with modest amounts of noise, more sophisticated techniques might be required to deal with very noisy data.

One such technique might be to check the dependent term for systematic trends. The values of *y* in Table 1 are monotonically increasing, for example, which suggests a higher order derivative should be calculated. If no such trends were found, BACON.5 could accept the current relationship, even though the number of outliers was large. A second technique would be to allow the program to store several possible relationships between the current independent and dependent terms. Beam searching techniques could be used to limit the number of competing hypotheses BACON entertained at any given time, and the program could design critical experiments to determine the best description of the data. Finally, if the system discovered promising relationships in parts of the data, the expectation-driven heuristics discussed above could help BACON to develop a consistent interpretation of the data, even in the presence of substantial noise. Combining these techniques should allow BACON to deal with realistic amounts of noise in data in a robust manner.

7. Discovering Symmetrical Laws

The assumption of symmetry has been a powerful aid in the discovery of physical laws. Table 10 presents three well-known laws that exhibit symmetry. Although BACON.5 could discover these laws without any heuristics other than those we have already described, the inclusion of a new component that *postulates* symmetry significantly reduces the search required to find these laws. This new heuristic waits until all the terms associated with an object have been related, and then *assumes* that the same relation will hold for a second set of terms that are associated with an analogous object. The resulting complex terms are then combined into a symmetrical law.

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

Table 10. Symmetrical laws discovered by BACON.5.

As an example, consider BACON.5's discovery of Snell's law of refraction, as summarized in Table 11. 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 $sine\theta_1$ constant, the system postulates an intrinsic property, n_2, whose values are associated with different media. Of course, the ratio $sine\ \theta_2/n_2$ has the constant value

1.0. At this point, BACON.5 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.

$MEDIUM_1$	SIN θ_1	$MEDIUM_2$	SIN θ_1	N_2	SIN θ_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

Table 11. Discovering Snell's law of refraction.

The BACON.5 system has discovered two other symmetrical laws – conservation of momentum and Black's specific heat law – following very similar paths. Table 10 presents the full form of the laws; directly observable terms are shown in upper case, while intrinsic properties are shown in lower case. The program has also discovered two different versions of Joule's law of energy conservation, using a simple form of reasoning by analogy. This strategy states that if the same set of terms occurs in more than one experiment, one should consider combining them in the same fashion as proved useful before. For a more complete description of this heuristic and its application to Joule's law, the reader is directed to an earlier article on BACON [10].

In summary, we have seen that BACON's expectation-driven heuristics – expecting similar relations, reducing the data that is gathered, and postulating symmetrical laws – allow it to discover empirical laws with considerable reduction in search. Actual computational savings for three symmetric laws are shown in Table 12. From this table, it can be seen that, when combined, BACON.5's expectation-driven heuristics result in major savings. Moreover, these heuristics accomplish this with little loss in generality, since relations such as symmetry can be found in a wide variety of scientific domains.

	DD	DD + SR + DR	DD + SR + DR + SY
MOMENTUM	515	212	8
SNELL'S LAW	40	40	5
BLACK'S LAW	8433	2200	23

Table 12. Time to discover symmetric laws in CPU seconds.[3]

[3] DD = data-driven heuristics, DD + SR + DR = data-driven, similar relation, and data reduction heuristics, DD + SR + DR + SY = data-driven, similar relation, data reduction, and symmetry heuristics.

8. The Importance of Structure

In the previous sections, we have described the empirical discovery system BACON.5. Given a set of numeric or nominal variables, this system employs a number of heuristics to determine the relation between those terms. Yet it is worth noting that the most interesting of BACON's heuristics address aspects of discovery that lie beyond the simple relation of variables. For example, when an intrinsic property is postulated, it is always associated with some *object* or class of objects. Similarly, the symmetry and analogy heuristics apply only in situations where the same terms are associated with different objects. (In the symmetry case, identical terms are attached to different objects ***within*** an experiment, while in the analogy case the identity falls ***across*** experiments.)

In summary, these heuristics appear to incorporate some notion of ***structure*** which extends beyond the simple variable-value representation used in BACON.5. Given this view, one drawback of BACON is that it represents this structure ***implicitly*** rather than ***explicitly***. Thus, in replicating Ohm's experiment, the program is told about the battery, the length of the wire, and the current, but it does not understand that the wire must be connected to the battery to generate the current. Similarly, in the conservation of momentum experiment, BACON is given variables for the objects along with their initial and final velocities; however, it is unaware that the initial velocities are transformed into the final velocities by a collision, and that if no collision occurs, the velocities will remain unchanged. In other words, BACON.5 attempts to discover quantitative laws before it has mastered the ***qualitative laws of structure*** [11]. This feat can be accomplished, but only if the system is presented with a set of variables that have been carefully selected to contain those qualitative relationships.

Future versions of BACON should represent structural relations explicitly, and should attempt to discover the qualitative laws of a situation (e.g., that objects collide and change direction, or that some chemicals combine to form new chemicals) before moving on to considering quantitative laws. Such an approach would be much more consistent with historical developments in science than the current implementation. Moreover, once the system has arrived at a structural model for a situation, this model may find another use as an ***explanation*** for a quantitative law found in some other situation. This is an important point, since many explanatory theories - including the atomic theory, the kinetic theory of gasses, and the germ theory of disease - are primarily structural models. Thus, by exploring the role of structure in a descriptive discovery system like BACON, we may come to a fuller understanding of explanatory science as well.

For instance, consider the kinetic theory of gasses, which can be used to explain the ideal gas law. Central to the kinetic theory is the notion of colliding molecules that obey conservation of momentum. The hypothesis that a gas is composed of microscopic objects (similar to their macroscopic counterparts in the momentum experiment) provides an explanation of the macroscopic relation between temperature, volume, and pressure. We do not claim to fully understand the process by which such explanations are constructed, though some form of reasoning by analogy seems a

likely candidate. In any case, the relation between qualitative laws of structure and explanation is a promising direction for future research.

It is interesting to note that one of BACON's current heuristics – searching for common divisors – could play an important role in such an explanatory discovery system. This results from the fact that the existence of a common divisor for a set of data suggests an important structural aspect of those data, namely that the objects involved in the experiment consist of quanta. Thus, one can imagine an extended version of BACON that, upon finding common divisors in chemical reactions, would invoke a prototype atomic theory to explain this fact.

Finally, we should note that an emphasis on qualitative laws of structure may provide a new approach to the dual problems of noise and irrelevant variables. Given an understanding of the structure of some situation, it may be possible to eliminate some relationships and some variables even before any quantitative data are gathered. For example, given the principle "no action at a distance" and an experimental context in which two objects never touch or even approach each other, one can immediately predict that the variables associated with these objects will be unrelated. Again, this is an area in which our ideas remain vague, but it is also an area that deserves further attention.

9. Conclusions

In this paper we have described BACON.5, an empirical discovery system that draws on data-driven heuristics for finding numeric relations between two variables, recursing to higher levels of description, postulating intrinsic properties, and finding common divisors. In addition to its data-driven techniques, BACON also incorporates expectation-driven heuristics for expecting similar relations, reducing the amount of data that must be gathered, assuming symmetrical laws, and reasoning by a simple type of analogy. These latter rules take advantage of discoveries BACON has made itself instead of drawing on knowledge about some particular domain. Thus, the program retains considerable generality, as evidenced by the broad range of laws it has been able to discover. In addition, the expectation-driven methods reduce the overall search that BACON must perform in discovering a law.

We have also seen that some of BACON's heuristics incorporate a notion of structure, but that this knowledge is represented implicitly. Future versions of the system should represent structural information explicitly, and attempt to discover qualitative laws before moving on to quantitative ones. This approach should provide new methods for handling noise and determining relevant variables, but it may do more than simply improve BACON's techniques for discovering descriptive laws. We hope that a concern with qualitative laws of structure will shed light on the process of explanatory discovery as well.

References

[1] Gerwin, D. G. Information processing, data inferences, and scientific generalization. ***Behavioral Science***, 1974, *19*, 314-325.

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

[3] Buchanan, B. G., Feigenbaum, E. B., and Sridharan, N. S. Heuristic theory formation: Data interpretation and rule formation. In D. Michie (ed.), ***Machine Intelligence 7.*** New York: American Elsevier, 1972, 269-290.

[4] Mitchell, T. M. Version spaces: A candidate elimination approach to rule learning. ***Proceedings of the Fifth International Joint Conference on Artificial Intelligence***, 1977, 305-310.

[5] Lenat, D. B. Automated theory formation in mathematics. ***Proceedings of the Fifth International Joint Conference on Artificial Intelligence***, 1977, 833-842.

[6] Langley, P. Data-driven discovery of physical laws. ***Cognitive Science***, 1981, *5*, 31-54.

[7] 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, 19-25.

[8] Langley, P., Bradshaw, G., and Simon, H. A. Rediscovering chemistry with BACON.4. To appear in J. Carbonell, R. Michalski, T. Mitchell (eds.), ***Machine Learning.***

[9] Forgy, C. L. The OPS4 reference manual. Technical report, Department of Computer Science, Carnegie-Mellon University, 1979.

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

[11] Newell, A. and Simon, H. A. Computer science as empirical enquiry: Symbols and search. ***Communications of the ACM***, 1975, *3*, 113-126.