Three Facets of Scientific Discovery

Pat Langley
Jan M. Zytkow
Gary L. Bradshaw
Herbert A. Simon
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213 USA

ABSTRACT

Scientific discovery is a complex process, and in this paper we consider three of its many facets - discovering laws of qualitative structure, finding quantitative relations between variables, and formulating sfructural models of reactions. We describe three discovery systems - GLAUBER, BACON, and DALTON - thr.t address these three aspects of the scientific process. GLAUBER forms classes of objects based on regularities in qualitative data, and states abstract laws in terms of these classes. BACON includes heuristics for finding numerical laws, for postulating intrinsic properties, and for noting common divisors. DALTON formulates molecular models that account for observed reactions, taking advantage of theoretical assumptions to direct its search if they are available. We show how each of the programs is capable of rediscovering laws or models that were found in the early days of chemistry. Finally, we consider some possble interactions between these systems, and the need for an integrated theory of discovery.

INTRODUCTION: THE DIVERSITY OF DISCOVERY

Scientific discovery is a process through which we acquire knowledge about the world This knowledge takes many forms, ranging from empirical regularities to structural models, and from qualitative relations to numerical laws. The diversity of scientific knowledge is accompanied by a diversity of processes for generating that knowledge. For instance, one would expect that quite different forms of reasoning led to the discovery of the ideal gas law, to the classification of organisms, and to the formulation of the atomic hypothesis.

Given the diversity of scientific discovery, two basic questions present themselves. First, what are the various types of scientific knowledge and the processes that lead to them? Second, how do these forms of reasoning interact to enable science as a whole to advance? In this paper we provide a response to the first of these questions in the form of three Al systems that address different aspects of the discovery process. One of these programs focuses on finding laws of qualitative structure, another is concerned with discovering quantitative relations between variables, and the third deals with the formulation of simple structural models. Below we discuss each of the systems and its application to some of facet of the history of chemistry. We will reserve our comments on the second question - how these systems might interact - until after we have described the systems themselves.

P. Langley is affiliated with The Robotics Institute, while J. M. Zytkow, G. L. Bradshaw, and H. A. Simon are associated with the Department of Psychology. This research was supported by Contract N00014-82-0168 from the Office of Naval Research.

DISCOVERING QUALITATIVE LAWS

By the 17th and 13th Centuries, chemists had made considerable progress in classifying substances on the basis of observable properties. For example, the class of acids had been defined in terms of its members' sour taste, their changing the color of organic dyes, dissolving metals, and so forth. Exceptions to these characteristics occurred, but sufficient regularity was present to make acid a useful concept. Along with acids, other classes such as metals, alkalis, and salts had been formulated in terms of similar properties. In addition to basing classes on properties of individual substances, the early chemists also noted relations between substances. Thus, they formulated the qualitative law "acids combine with alkalis to form salts", later generalizing this by replacing alkalis with the more abstract notion of a base.

GLAUBER: A Qualitative Discovery System

In order to better understand the processes through which such laws were found, we constructed a qualitative discovery system. We have named the program GLAUBER, after the 17th Century chemist who played an important role in developing the theory of acids and bases. GLAUBER inputs qualitative facts, such as "hydrochloric acid tastes sour" and "hydrochloric acid combines with sodium hydroxide to form sodium chloride", and produces two forms of output: a set of abstract classes, such as acids, alkalis, and salts, along with their members; and a set of laws, such as "acids taste sour" and "acids react with alkalis to form salts", stated in terms of these classes.

We should say a few words about GLAUBER'S representation of data, since it has implications for the system's discovery methods. Facts are represented using a simple frame-like structure, consisting of a predicate followed by a number of attribute-value pairs. The facts mentioned above are represented by the propositions (HAS-QUALITY OBJECT (HYDROCHLORIC-ACID) TASTE (SOUR)) and (REACTS INPUTS (HYDROCHLORIC ACID SODIUM-HYDROXIDE) OUTPUTS (SODIUM-CHLORIDE)). In the second expression the predicate is REACTS, the two attributes are INPUTS and OUTPUTS, and their respective values are (HYDROCHLORIC-ACID SODIUM-HYDROXIDE) and (SODIUM-CHLORIDE). In this example, the INPUTS attribute has two values, which represent the two substances that combine in the reaction. GLAUBER knows that the order of these values is not significant.

Noting Patterns and Defining Classes

GLAUBER inputs a set of facts such as the above, and iterates through all symbols that occur as values, searching for facts that have the same predicate and the same value for a given attribute. For example, upon considering the symbol SOUR, the system would note that a number of chemicals - hydrochloric acid, nitric acid, and sulfuric acid - all have a sour taste. When such a regularity is discovered, GLAUBER defines a new class and

stores the symbols that differ in these facts as members of the class. In addition, the program formulates a pattern that is identical with these facts, but in which the differing values have been replaced by the class name. If we call the class formed in this example SOUR TASTERS, then the three substances would be stored as members of this class, and the associated pattern would be represented as (HAS QUALITY OBJECT (SOUR TASTERS) TASTE (SOUR)).

Relational patterns can also be discovered. For instance, suppose that while focusing on sodium hydroxide, GLAUBER notes that this chemical combines v/ith hydrochloric acid to form sodium chloride, with nitric acid to form sodium nitrate, and with sulfuric acid to form Glauber's salt (Na_2SO_4)In this situation, GLAUBER would create two classes. The first (let us call it sodium-hydroxide-reactors) contains hydrochloric acid, nitric acid, and sulfuric acid, while the second (sodium hydroxide-results) contains sodium chloride, sodium nitrate, and Glauber's salt The associated pattern would be stated as (REACTS INPUTS (SODIUM HYDROXIDE SODIUM-HYDROXIDE REACTORS) OUTPUTS (SODIUMHYDROX IDE-RESULTS)).

Combining Classes and Recursing to Higher Levels

The early chemists noted that certain patterns tended to occur *together,* and GLAUBER achieves a similar insight. The system compares classes and combines those having a high percentage (determined by a system parameter) of elements in common. The new class is then compared to others so that further combinations can occur. For example, having generated the initial classes and patterns described above, GLAUBER notes that every member of the sour-tasting class also fits the pattern associated with the sodium hydroxide reacting class (and vice versa). As a result, the members of these two groups would be combined into a new class. This class would have two associated patterns, one involving taste and the other concerning reactions. The process is repeated, until eventually GLAUBER arrives at the three classes we know as acids, alkalis, and salts, each with a set of associated patterns.

Since patterns are stated in the same manner as the initial facts, GLAUBER can recursively apply its abstraction methods to the patterns themselves. Using this strategy, the system notes that HCl, $HN0_3$, and H_2S0_4 all react with with alkalis to produce salts, leading it to define a new class containing these elements. Upon realizing that this class is identical with the class of *acids* defined earlier, it combines the two concepts, and formulates the general law (REACTS INPUTS (ACIDS ALKALIS) OUTPUTS (SALTS)). Thus, GLAUBER arrives at one of the central results of the 17th Century chemists. When provided with additional data about metals and their reactions with acids, the system also defines the more abstract notion of *base,* along with the more general law that acids react with bases to form salts. Although the current version of GLAUBER treats all classes as equivalent, the introduction of more data will require future versions to focus attention on some classes (such as those occurring in the most patterns) in favor of others.

DISCOVERING QUANTITATIVE LAWS

Around the turn of the 10th century, three fundamental discoveries were made concerning quantities of substances forming chemical compounds. The first of these was Proust's (1799) law of constant proportions, which states that the weight ratio of constituent elements is constant for a given compound. The second advance was Dalton's (1804) formulation of the law of multiple proportions. This law asserts that when two elements combine to form several different compounds, the ratios of their combining weights are always small integer multiples of one another. The third was Gay Lussac's (1009) discovery of the law of combining volumes, which states that gases combine in small integer ratios by volume. These three discoveries provided the foundation for a quantitative theory of chemical reactions, and ultimately led to the determination of the atomic weights of the elements. Although the work of Dalton and Gay-Lussac was at least partially motivated by the atomic hypothesis, we shall see that data-driven methods are sufficiently powerful to find these laws.

Finding Numerical Relations in Noisy Data

We have explored the process of quantitative discovery through BACON.6, the sixth in a line of programs named after Sir Francis Bacon. Given a set of independent variables. BACON.6 varies one of them, looking for relations between that term and some dependent variable. Once a functional relation has been found, the parameters in that function are given the status of dependent terms at a higher level of description. When the system varies the next independent term, it looks for a relation between that variable and the new higher level terms. This process continues, with BACON.6 recursing to higher levels of description, until all the independent terms have been incorporated into a complex quantitative relationship We will not discuss this process in any great detail, since it has been described for earlier versions of BACON [1, 2].

Unlike previous versions, BACON.6 is capable of dealing with significant amounts of noise in its data. The program uses a differencing technique to find the best polynomial function for relating two terms. However, it also considers polynomial relations between *functions* of these terms, so that relations such as $y = ax^2 + bx +$ c, and $sin(y) -- a log(x) + b$ can be found. The differencing method accepts any relation that accounts for more than a user specified percentage of the variance in the data. When this requirement for explained variance is high, BACON.6 behaves much like its predecessors: if the data are noise-free, it generates only a single hypothesis; however, if the data are noisy, it fails to find any relation at all. In contrast, when the setting is low, the system tends to generate a number of alternate hypotheses, whether the data are noisy or not.

In cases where a number of competing hypotheses have been generated, BACON.6 must have some way to order these hypotheses. In addition to the explained variance, the system takes into account the *complexity* of each law, measuring this by the number of terms that make up the polynomial expression. The user can specify the exact role played by the two criteria, but in our experiments with the system, we have found that the *ratio* of explained-variance to complexity gives good results. Once the score for each hypothesis has been calculated, a threshold score is computed, and only those laws exceeding the threshold are retained. BACON.6 also takes the *generality* of each law into account. For example, if two laws are found to describe the relation between *y* and z when x = 7.0, but one of these laws does not fit well when x = *2.0,* then that law will be rejected. In this way, the system ensures that only hypotheses holding across a broad range of data are retained.

Intrinsic Properties and Common Divisors

While the above heuristics are useful for discovering relations between numerical terms, they cannot be used to relate *nominal* of symbolic independent terms to numeric dependent variables, and this is precisely the situation in which the early chemists found themselves. For instance, the independent terms in Proust's, Dalton's, and Gay-Lussac's chemical experiments were

the elements or compounds involved, while the dependent terms were numerical measures such as weight or volume. In such cases, BACON.6 defines *intrinsic properties* that take on numeric values, and associates these properties with the nominal terms.

Given control over the substances entering and resulting from a reaction, as well as the weight of the first substance that is used, the system gathers the data in Table 1. Upon varying the amount of oxygen used to form nitric oxide (NO), the program discovers that the two weights w_1 and w_2 are linearly related with a slope of 1.14 and a zero intercept. Upon varying the output of the reaction, BACON.6 examines the weight relations for the compound nitrous oxide (N_20). In this case, the law is also linear, but the slope has changed to 0.57. A similar result is obtained for nitrogen dioxide, and in this case the slope is 2.28.

$ELEMENT_1$	$ELEMENT_2$	COMP.	W_1	W_2	W_2/W_1
N	O	NO	1.0	1.14	1.14
N	O	NO	2.0	2.28	1.14
N	O	NO	3.0	3.42	1.14
N	O	N_2O	1.0	0.57	0.57
N	O	N_2O	2.0	1.14	0.57
N	O	N_2O	3.0	1.71	0.57
N	O	NO_2	1.0	2.28	2.28
N	O	NO_2	2.0	4.56	2.28
N	O	NO_2	3.0	6.84	2.28

Table 1. Determining the combining weights for reactions.

The slopes that BACON.6 finds in these experiments are closely related to the weight ratios found by Proust. Having found these ratios, the program defines an intrinsic property (say p) whose values are associated with the three nominal values under which they occur. Thus, the value of *p* for the triple nitrogen/oxygen/nitric oxide would be set to 1.14, the value for nitrogen/oxygen/nitrous oxide would be 0.57. and the value for nitrogen/oxygen/nitrogen dioxide would be 2.28. As stated, these intrinsic values simply store an already known fact. However, they can be retrieved in future experiments involving the same chemicals, and used to make predictions or to discover new empirical laws.

Proust's insight about combining weights laid the groundwork for Dalton's law of multiple proportions, and BACON.6 includes a heuristic which lets it discover just such a relation in the data from Table 1. This heuristic operates whenever the system is about to define a new intrinsic property, examining the values of the new property to see if they (or their inverses) have a common divisor. In this case, BACON would note that 1.14, 0.57, and 2.28 have the common divisor 0.57, and would replace these intrinsic values with their corresponding integers 2, 1, and 4. Later, if other common divisors were found for other pairs of elements, the program would define a higher level intrinsic property based on these divisors, and associate them with thoso pairs of elements. Thus, the common divisor 0.57 would be associated with the nitrogen/oxygen pair, the divisor 1.33 with carbon and oxygen, and so on. These relations are formally equivalent to Dalton's law of multiple proportions. BACON takes a similar path in discovering Gay-Lussac's common divisors for combining volumes, and has even arrived at the correct relative atomic weights of hydrogen, oxygen, and nitrogen from similar data. In summary, BACON'S mechanisms account for many of the quantitative laws found by chemists in the early 19th Century.

FORMULATING STRUCTURAL MODELS

Although Dalton's atomic hypothesis was readily accepted by many chemists, its application to specific reactions was far from clear. Dalton inferred the structure of various compounds using his rule of greatest simplicity, along with the assumption that all elements were monatomic. This led him to conclude that a molecule of water was composed of a single hydrogen atom and a single oxygen atom. In contrast, Avogadro (1811) employed Gay-Lussacs data on combining volumes and the assumption that equal volumes of gas contained equal numbers of particles. Using this information, he inferred diatomic models for hydrogen and oxygen and a different structure for water.

Searching the Space of Structural Models

In order to understand the process by which chemists constructed structural models of chemical reactions, we have implemented a third discovery system - DALTON - that focuses on this issue. The system knows that two quantities are important for any model of reaction - the number of *molecules* of each substance that takes part, and the number of *particles* in each molecule. Suppose the system is told that hydrogen reacts with oxygen to form water, and is asked to construct a molecular model of this process. In this case, the program must determine the number of hydrogen, oxygen, and water molecules, and the internal structure of each type of molecule. The system operates by starting with a model in which no commitments are made, and successively refines this model as it proceeds.

Starting with a model of the form (HO-> W), DALTON first considers the number of hydrogen molecules involved. Lacking any theoretical bias, the system assumes the simplest choice of a single hydrogen molecule. If this choice later causes difficulty, the model-builder can back up and try another path. Similar initial choices are made for oxygen and water. This is represented by the proposition ((H) (O) -> (W)), in which each molecule is enclosed in parentheses. Now DALTON must determine the internal structure of each type of molecule, and it assumes for both hydrogen and oxygen a single elementary particle (say h and o), giving the model ((h) (o) ->* (W)). At this point, the program invokes the theoretical assumption that the total number of particles in any reaction is conserved. This gives the final model ((h) (o) -* (h o)), which is equivalent to that originally formulated by the human Dalton. In this case, the program has arrived at an acceptable solution without needing to backtrack.

Altering the Search Process

In the above run, the system had no theoretical biases other than a belief in conservation of particles and a desire to construct as simple a model as possible. However, if we give DALTON some additional information about the water reaction, its behavior changes significantly.'Avogadro believed that the combining volumes which Gay-Lussac observed were related to the number of molecules involved in the reaction. Given this assumption (and knowledge of the combining volumes), our program instead postulates two molecules of hydrogen and water, while retaining the assumption of one oxygen molecule, giving the partially specified model ((H) (H) (O) -> (W) (W)).

At this point the system considers the internal structure of the hydrogen and oxygen molecules, and initially assumes both to be monatomic. However, for the resulting model, ((h) (h) (o) -> (W) (W)), there exists no decomposition of water in terms of h and o that satisfies the conservation assumption, so the program

backs up and considers another alternative. At this point DALTON hypothesizes the oxygen molecule as composed of two particles, and since this satisfies conservation, a final model is constructed: ((h) (h) (o o) -+ (h o) (h o)). While this model differs from the modern day one, it is consistent with Guy-Lussac's data and encounters difficulty only when other reactions are considered. For example, when the ammonia reaction is encountered, DALTON must revise its monatomic assumption for hydrogen, and arrives at the correct water model: ((h h) (h h) (o o) -> (h h o) (h h o)).

Since theoretical assumptions can influence DALTON's behavior to such a great extent, we should mention the form in which this information is presented. DALTON is stated as a production system, and in default mode it uses a few simple rules to formulate simpler models first, and more complicated ones as necessary. However, if new condition-action rules are added to the system, they take precedence over the default rules and can direct search down paths that might otherwise not be considered. Thus, one can insert a rule that would match if the combining volumes of substances are known, and use this information to determine the number of molecules used in the model. The conservation assumption is implemented in a similar fashion, so that it generates a molecular structure of a reaction's output that uses all particles occurring in the input. While the current version of DALTON is capable of formulating only very simple structural models, it does provide an initial account of this process, and the manner in which theoretical assumptions can alter the search strategy.

DISCUSSION

Although we have considered only chemical discoveries in our examples, each of the systems we have described is stated in a very general fashion and there is no reason they could not be applied to other domains as well. This is one direction in which we should apply our future research efforts. However, an even more interesting possibility presents itself. A complete theory of the scientific process must not only account for different types of discovery; it must also explain the *interactions* between these different facets. Although we have not yet linked our three systems computationally, we have considered some steps towards creating such an integrated model of discovery.

For example, qualitative laws generally appear earlier in the development of a field than do quantitative lawo. Thus, one can imagine a system like GLAUBER first discovering laws of qualitative structure, and then passing this information on to a BACON-like system, which would use it to determine the variables it should consider and the experiments it should run. Similarly, data-driven discovery often precedes theory-driven discovery. Thus, one can imagine BACON arriving at regularities such as Guy-Lussac's law of combining volumes, with DALTON employing this information to direct its search process. Of course, information could flow in the other direction as well. Once DALTON had determined the molecular structure of a pair of elements in one reaction, it might *predict* the combining volumes for new reactions; it could then pass these expectations on to BACON, where such expectations could play an important role in dealing with noisy data. In the field of genetics, GLAUBER might use data about inherited characteristics to classify offspring into genotypes, and DALTON might use this classification in replicating Mendel's two-trait model of heredity. In the other direction, GLAUBER might view DALTON's models as data, and note the distinction between *dominant* and *recessive* traits.

In addition to being interesting in their own right, such interactions would provide important constraints on our models of discovery. For instance, the current version of BACON must be supplied with a set of variables by the programmer, and the usefulness of BACON's discoveries is judged mainly by the user. Thus, there are very few constraints on either the system's inputs or its outputs. We do not feel that BACON fares any worse on these dimensions than other learning and discovery systems, but these are still issues that should be addressed. In attempting to construct an integrated discovery system, we expect that the interactions between different components will constrain the approaches we explore. Thus, by requiring BACON's expectations to come from GLAUBER or DALTON, and by insisting that BACON's discoveries be used by the other systems, we hope to account for facets of discovery that could not be explained by studying the various components in isolation.

Before closing, we should say a few words about the relations between our our systems and earlier AI research on discovery. For example, the patterns generated by GLAUBER bear some resemblance to those produced by Brown's [3] early system, while its approach to classification is related to Lenat's [4] heuristics for mathematical discovery, and to Michalski and Stepp's [5] conceptual clustering strategy. However, the details of GLAUBER'S operation differ considerably from each of these programs. BACON's techniques for finding numeric laws in the presence of noise are reminiscent of Gerwin's [6] early work in this area, though BACON can deal with more complex functions and employs a different curve-fitting method. Finally, DALTON's search for molecular models is similar in some ways to DENDRAL's [7] search for organic compounds to explain mass spectrographs, though the latter explored P. much larger space of hypotheses and required considerable knowledge of chemistry to direct its search through that space. Thus, while our discovery systems are related to earlier work in the area, they also differ in some important ways. Furthermore, unlike the earlier programs, our systems show a potential for being combined into a more complete, integrated theory of discovery.

REFERENCES

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

[2] 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, 137-143.

[3] Brown, J. S. Steps toward automatic theory formation. *Proceedings of the Third International Joint Conference on Artificial Intelligence,* 1973, 20-23.

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

[5] Michalski, R. S. and Stepp, R. E. An application of Al techniques to structuring objects into an optimal conceptual hierarchy. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* 1981, 460-645.

[6] Gerwin, D. G. Information processing, data inferences, and scientific generalization. *Behavioral Science,* 1974, 79, 314-325.

[7] 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: Edinburgh University Press, 1971.