Report 82-28
Stanford -- KSL

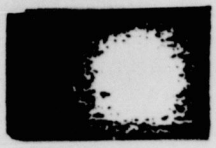
Scientific DataLink

Mechanizing the Search for Explanatory Hypotheses.

Bruce G. Buchanan,
1982

card 1 of 1

HPP-82-28



Mechanizing the Search for Explanatory Hypotheses[1]

Bruce G. Buchanan

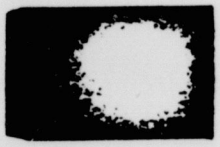
Stanford University

1. Introduction

Mechanized methods in science have attracted much attention among philosophers since Bacon. Of the many facets of scientific activity discovery has seemed most inscrutable. Until recently, Peirce and Hanson were the only ones to claim publicly that there can be rational methods for discovering hypotheses as well as for testing them. Their arguments, and more recent ones, are based on historical examples and analysis, and thus lack the convincingness of an existence proof. In this paper I take an empirical look at the question of whether there are rational methods of discovery and claim that computer programs provide a laboratory for experimentation on this question. Recent work in artificial intelligence, or AI, has produced programs capable of serious intellectual work in science. Results from AI will be used to show that there exist mechanized procedures for discovering hypotheses and that these methods often lead to plausible hypotheses (but do not guarantee always finding the correct hypothesis).

Statistical methods have been used successfully in science for decades. Methods such as regression and clustering provide hypotheses for explaining data that can be expressed quantitatively. Computers have certainly increased their applicability and ease of application. But mathematical methods are not the only means of finding relationships in descriptions of data, although we understand the scope and limitations of numerical methods much better than methods of symbolic reasoning. Therefore I wish to explore this lesser known area of computation to examine the extent to which today's methods answer Bacon's questions about mechanizing some aspects of science. The computer is an important part of the investigation because it removes such psychological elements as "creativity", "intuition",

---



As reprinted from PSA 1982, Vol.2, pp.129-146. Edited by P.D. Asquith and P. Kitcher, 1983.

"gestalt" and "mind set" from the discussion whereas past discussions of discovery trip over these subjective elements. In principle, the methods could be exercised and investigated by paper and pencil, just as mathematical methods. But only a machine can keep track of immense numbers of details and combinations of possibilities in practice.

Searching for explanatory hypotheses is one scientific activity in which discovery is central, another is searching for laboratory plans that accomplish desired goals. Both require finding descriptions that link starting assumptions and theory with observables. On Hempel's model of explanation, the explanatory links are deductive: from H and relevant background statements one can deduce the phenomenon to be explained. The links in the description of the experiment are also deductive: from the initial conditions, one can achieve the result by following the prescribed steps which are linked deductively through the theory. Unfortunately in both explanation and experiment planning, uncertainties in the initial conditions and in the theory often prevent us from knowing we have the correct hypothesis or plan until we test it, although we can have some assurance that we have a plausible one. This paper will be limited to hypothesis formation methods because there are more examples in AI and because the methods are better understood.[2]

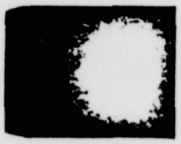
It is commonplace to say that a computer cannot discover anything it has not already been told. There is a grain of truth in that, but it depends on the ambiguity in how much you have to tell a machine, or a person, before you have told it "everything interesting". Do the axioms of a theory say everything interesting? By telling a chess machine the rules of chess, have you told it everything? When you add some heuristics that suggest good chess moves, have you now told it everything? And so on.

The programs we will discuss briefly have been told a great amount about their areas of specialization and they have been given methods for reasoning with that factual knowledge in novel situations. The programs can find explanations of data from cases they have never encountered. Even though part of what they have been told is how to cope with new cases, it is false that they have been told in advance what are the correct explanations for all combinations of data. Nor have they been given a single procedure to follow in every situation. As is true also with numerical methods, a computer's symbolic reasoning can produce explanations of new data that are not anticipated by the persons writing the programs. The reasoning becomes too complicated to follow by hand.

The question to be answered in the next section is: By what methods can a computer program find plausible explanations of empirical data? To answer this, we first look at a common metaphor for describing several AI programs and then look at examples of programs actually constructed. From there, we generalize to some conclusions about mechanizing hypothesis formation. I am deliberately blurring the distinction between discovering hypotheses that explain single events and more general ones that explain a collection of events. Computer programs that formulate both kinds are cited as examples.

2. Hypothesis Formation as Search

The metaphor of search is a powerful one in AI for thinking about problem solving.[3] Under the search model, a program systematically explores a space of alternative solutions and stops when a solution meets its criteria of correctness. The model has been elaborated in



different ways, as we will see, but there are two essential elements. First, it is necessary that elements of the space can be *generated* so they can be examined. A trivial generating scheme is to take elements from a pre-defined list, but there are more complex and interesting generators. Second, it is necessary that potential solutions can be *tested* against the criteria of success. The so-called "generate and test" method incorporates these two elements in a search method that is powerful, but often inefficient. Some of the elaborations of the search model beyond generate and test take account of the following facts:

- With a very large search space, systematic generation needs guidance to be effective;
- Guidance can come from a set of rules of thumb, or heuristics, as well as from theoretical laws;
- Generating and testing complete solutions (as contrasted with partial solutions) can be very expensive;
- Finding the "correct" solution is much more difficult than finding one that is "good enough", and may not be cost effective.

The search model has been used in several AI programs that formulate hypotheses to explain or interpret data. Some of these programs will be discussed briefly later, but let us look at the tasks together to see the range of problems and their difficulty. For example:

Chemical Analysis

DENDRAL — *Given* spectroscopic data from an unknown organic chemical sample, *Determine* the molecular structure of the unknown [Lindsay et al. 1980].

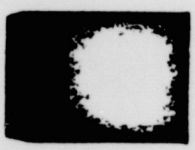
CRYSALIS — *Given* x-ray diffraction data from crystallized proteins, *Interpret* the 3-dimensional structure of the protein [Engelmore and Terry 1979].

META-DENDRAL — *Given* a collection of analytic data from a mass spectrometer, *Discover* rules explaining the fragmentation behavior of chemical samples in that instrument [Buchanan and Mitchell 1978].

Signal Understanding

HASP — *Given* underwater acoustic signals emanating from ships at sea, *Identify* the types and locations of the ships [Nii et al. 1982].

HEARSAY — *Given* acoustic waveforms from human speech, *Understand* the meaning the speaker is trying to convey [Erman et al. 1980].



Medicine

MYCIN
CADUCEUS
PUFF
CASNET

Given signs and symptoms from a patient with a medical problem,
Diagnose the cause of the problem [Shortliffe 1976], [Miller et al. 1982], [Kunz et al 1978], [Weiss et al. 1979].

RX

Given a collection of patient records with observations over time,
Discover new causal relationships [Blum 1981].

Physics

BACON

Given a set of numerical measurements from natural bodies,
Discover the general laws governing their observed behavior [Langley 1979].

Psychology

BUGGY

Given a record of a child's correct and incorrect solutions to mathematical problems,
Find the misconceptions causing the errors [Brown and Burton 1978, Sleeman 1982].

Electronics

DART

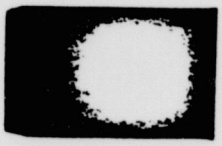
Given data on the malfunction of a computer system,
Diagnose the cause of the trouble [Genesereth 1982, Davis et al. 1982].

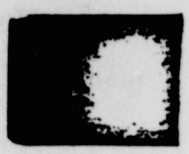
Geology

PROSPECTOR

Given data about the geological features of a region,
Determine whether significant mineral deposits can be found [Campbell et al. 1982].

In these areas, and more, AI programs with demonstrated competence have been written. Time does not allow detailed descriptions of how these programs work, but I will go over some of their general principles, in terms of the search metaphor and variations on it. Some of these applications are from engineering, in which the data are signals from man-made devices such as computers. In these applications, a program has the advantage of knowing many of the principles by which the device was constructed and is intended to function. Medical diagnosis has something of that flavor because data are interpreted with respect to partial models of how the body functions. In other applications such as chemistry, psychology and physics, however, the programs have only weak models of the systems -- in fact, their purpose is to discover partial models. In all of the applications there is much in common, though, and I claim, but do not show, that there is much carryover among all these





applications by virtue of their explaining empirical data. To a large extent, each of the programs mentioned earlier represents one data point in the inductive argument presented in this paper. Because each program represents considerable effort -- as much as 50 man-years or more -- details beyond the brief sketches will have to be found in the literature.

The Data

Descriptions of the data are given to a computer in a vocabulary whose terms are known in advance. Some programs receive as input the "raw" data from electronic devices such as voltmeters or TV cameras. But they still interpret these signals within a framework of descriptions that is fixed by their designers. This does not imply a finite set of descriptions in the domain since some, for example, variables may have continuous values. But in practice, computer programs for hypothesis formation deal with a finite number of values (or ranges of values) for any variable. A patient's temperature, for example, can be measured with arbitrary precision, but there are only a few ranges of values that make a difference in either diagnosis or therapy. However, the domain of data descriptions can be so large as to be infinite for all practical purposes. With 100 binary variables the domain covers 2^{100} or about 10^{10} individuals.

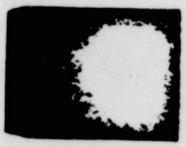
In science there is no reason to assume that the data are complete or totally correct. Even the most careful observer, or best engineered instrument, fails to record some significant events and fails to measure and record all events with total accuracy. Measurement errors are common. But so are spurious data. In electronic instruments, spurious data points are introduced by electronic noise in the line. Visual observation suffers from similar introductions of spurious sightings. Even the most honest observer sees evidence confirming his favorite hypothesis, which other observers do not see.

Not only are the data partially incomplete and incorrect, they also may be redundant and ambiguous. Redundancy is often a virtue in that multiple observations of the same event should help overcome the uncertainty due to knowing that some measurements are incorrect. But too many recordings of the same thing may swamp our processing capabilities so that we skip over the one-in-a-million event that is an interesting counterexample. Ambiguity in the data is another problem in trying to mechanize hypothesis formation in science. Ambiguities occur because multiple interpretations of the data arise from either the incompleteness of the data or incompleteness of the interpretive theory. In either case, a central problem for scientists is to remove the ambiguity through experimentation and theory reformulation.

The reason for pointing out these obvious characteristics of empirical data is to highlight the problems of automating hypothesis formation. Hypothesis formation methods must be robust enough to explain the data as given. Not all hypothesis formation work in AI has had this goal because it is useful to experiment first with methods to explain "cleaned up" data. If AI programs are going to aid scientists, however, they ultimately have to work with messy data.

The Hypotheses

Just as the domain of data descriptions is determined by the vocabulary of features, the range of hypotheses is also circumscribed by the vocabulary and syntax of allowable H's. This is frequently called the hypothesis space. The size of the space is a function of the number of primitive terms we are willing to consider in allowable H's and the number of allowable ways of combining those terms. For example, in organic chemistry it makes sense,



for some purposes, to limit the vocabulary of chemical compounds to just a half-dozen or so types of chemical atoms and to limit the size of the hypotheses to 50 or 100 individual atoms.

By stating the ways to restrict the complexity of allowable H's in the hypotheses space, we are stating the characteristics of the H's that define their complexity or simplicity. There are many arbitrary means of defining for a program what we want it to consider as a "simpler" hypothesis. Counting the number of terms is not philosophically satisfying, but may be pragmatically justified within a fixed vocabulary in which all terms are judged to have more or less equal complexity. Some choices must be made, however, in spite of the difficulty of satisfying philosophers that they are good choices.

Just as the complexity of a hypothesis is difficult to define satisfactorily, there is no satisfactory measure of the degree of difficulty of a data interpretation problem. One measure is the size of the hypothesis space, although there is a threshold above which all sizes are classified as "immense". With numerical terms, the size of the space is obviously related to the desired precision of the explanations. Integers as coefficients and exponents certainly simplify the space of algebraic relations. There are analogous levels of precision with symbolic terms. For example, chemical atoms can be referred to by class name (e.g., halogen) by atom name (e.g.,chlorine) or by isotope (e.g., ^{37}CL).

We know that a trivial "explanation" of *n* data points is a set of *n* descriptions. However, we wish for more compact explanations. We also know in many cases that we have collected data describing different processes or relations, that the data need to be classified as instances of different phenomena, with explanatory rules for each class. Whether or not a program is told how many phenomena it is trying to explain certainly influences the amount of search it must do. When analytic data are known to come from a pure sample of a single chemical compound, for example, analysis is easier than when they may have come from a mixture of several compounds.

In the next sections, we discuss methods of searching the hypothesis space in order to find explanations of the given data. As stated before, the methods considered here are all variations on search. Although random trial and error can be considered a kind of search, it is not even discussed here because there are more rational alternatives. Fig.1 summarizes the types of search discussed in the sections below.

- RANDOM SEARCH -- unsystematic trial and error
- EXHAUSTIVE SEARCH -- systematic generate and test
- SELECTIVE SEARCH -- evidence gathering for fixed categories
- HEURISTIC SEARCH -- generate and test with guidance and pruning
- OPPORTUNISTIC SEARCH -- search focused by best opportunities

Figure 1: Variations on the Search Model

2.1. Exhaustive Search [or Blind Search]

A baseline, unintelligent program could in principle enumerate allowable H's in the space and test to see if each one predicts the observed data or other data in the domain. This exhaustive search procedure is not recommended for man or machine. We are not likely to recommend exhaustive search as a method of discovery even though it is reproducible and can, in principle, find the best hypothesis. It is brute-force search not guided by any knowledge. That is, it is purely syntactic and exhaustive.

In the DENDRAL program, [Lindsay et al. 1980] the baseline, exhaustive generator actually exists at the heart of the system and can be exercised for small problems. It generates topological descriptions of chemical graph structures from any number of chemical atoms of various types. It knows about the syntax of molecular structure descriptions in that carbon atoms are known to have valence four, nitrogen valence three or five, oxygen valence two, and so on. Within that simple syntax, it will combine, for example, six carbon atoms, thirteen hydrogen atoms, one nitrogen and two oxygens ($C_5H_{13}NO_2$) into over 10,000 different structural descriptions (in two dimensions -- many more exist when three dimensional information is considered). Some are shown in Fig. 2 below.

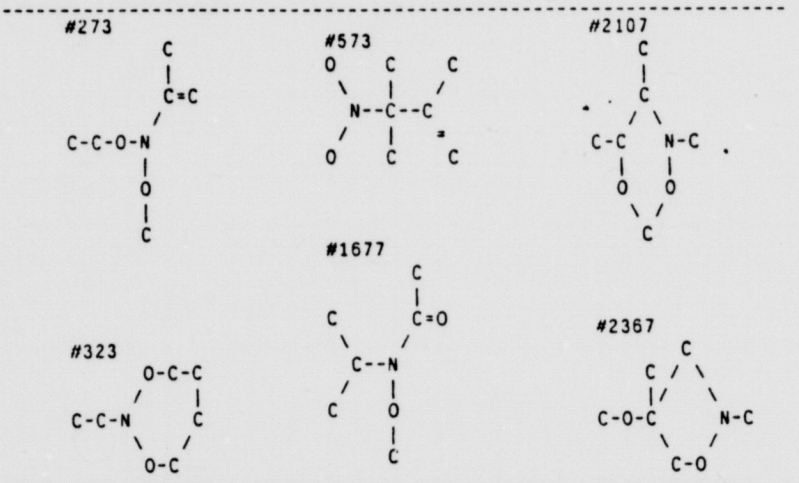


Figure 2: Some of the possible isomeric chemical graph structures with composition $C_6H_{13}NO_2$. Numbers are sequence numbers printed by CONGEN.

Conceptually, the program is combining all the atoms into all possible labelled graphs, using a method that is known to be exhaustive and not redundant. It should be obvious that the simple generate and test method would be hopelessly inefficient for explaining chemical data on unknown samples. There are just too many combinations of atoms into possible hypotheses. And the number increases exponentially with problem size and complexity, a fact that has been referred to as the "combinatorial explosion".

2.2. Selective Search [or "Evidence Gathering" or Classification]

A valuable limiting case of exhaustive search is selective search, or evidence gathering for and against a fixed set of hypotheses. With a small set of possible hypotheses, a program can ask about evidence for each in turn -- using the list as a generator -- and select the best one. In medical diagnosis, for example, a patient's abnormal signs and symptoms are explained by assigning the patient to one or more of a fixed set of disease categories. The MYCIN program [Shortliffe 1976], for example, gathers evidence for and against any of about 120 bacteria as possible causes of an infection. The evidence-gathering dialog and MYCIN's conclusions are shown in Fig. 3.

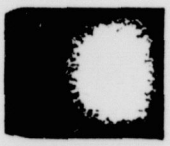
```
--------PATIENT-538--------
1) Patient's name:
** PT538
2) Age:
** 34 YEARS

.                                        Several questions are skipped.
.
.

14) Has Pt538 recently had symptoms of persistent headache or other
    abnormal neurologic symptoms (dizziness, lethargy, etc.)?
** YES
15) Has Pt538 recently had objective evidence of abnormal
    neurologic signs (nuchal rigidity, coma, seizures, etc.)
    documented by physician observation or examination?
** YES

The CSF cultures will be considered to be associated with MENINGITIS.
```

Figure 3: A portion of a dialog with MYCIN in which the program requests relevant data and then suggests the most likely hypotheses. *User's responses are in upper case and follow double asterisks. Editorial comments are in italics.*



.
.
.

23) For how many days has Pt538 had abnormal neurologic signs?
** 7.2 HOURS

.
.
.

Please enter CBC results in the following table

	WBC in 1000's	Percent PMN	Percent| BANDS
29)**	1.9	UNKNOWN	UNKNOWN

.
.
.

35) Is meningitis a hospital-acquired infection?
** WHY

The user asks "WHY" to see an explanation of the line of reasoning MYCIN is following.

.

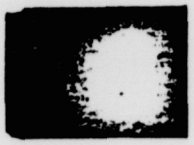
[i.e., WHY is it important to determine whether the infection was acquired while the patient was hospitalized?]

[2.0] This will aid in determining the organisms (other than those seen on cultures or smears) which might be causing the infection.
It has already been established that
[2.1] the infection which requires therapy is meningitis, and
[2.2] organisms were not seen on the stain of this pending csf culture, and
[2.3] the type of the infection is bacterial, and
[2.4] neurosurgery is not one of the relevant items from the history of pt538, and
[2.5] the age of pt538 is greater than 10 days
Therefore, if
[2.6] the infection was acquired while the patient was hospitalized, then
there is evidence that the organisms (other than those seen on cultures or smears) which might be causing the infection is
e.coli (.75f) staphylococcus-coag-pos (.3) pseudomonas-aeruginosa (.3) klebsiella-pneumoniae (.5)

Author: Yu
[RULE545]

The numbers in parentheses after each conclusion indicate the measure of importance of the conclusion. This single number combines frequency & utility considerations, as discussed in [Shortliffe 1976].

Figure 3. continued



```
.
.
.
   My therapy recommendations will be designed to treat for organisms
that  either are very likely  or, although less likely,  would have a
significant effect  on therapy selection if they  were present. It is
important  to  cover  for the  following  probable  infection(s)  and
associated organism(s):

INFECTION-1 is MENINGITIS

  + <ITEM-1>  E.COLI [from clinical evidence only]
  + <ITEM-2>  PSEUDOMONAS-AERUGINOSA [from clinical evidence only]
  + <ITEM-3>  KLEBSIELLA-PNEUMONIAE [from clinical evidence only]
  + <ITEM-4>  DIPLOCOCCUS-PNEUMONIAE [from clinical evidence only]

   41) Does Pt538 have a clinically significant allergic reaction to
       any antimicrobial agent?
   ** NO
[Considering AMPICILLIN-AND-GENTAMICIN for use against INFECTION-1...]
   42) Do you have reason to suspect that Pt538 may have impaired
       renal function?
   ** NO

   43) Pt538's weight in kilograms (or <number> POUNDS):
   ** 70

[REC-1] My preferred therapy recommendation is as follows:
   In order to cover for Items <1 2 3 4>:
        Give the following in combination:
        1) AMPICILLIN
        Dose: 3.5g (28.0 ml) q4h IV [calculated on basis of 50 mg/kg]

        2) GENTAMICIN
        Dose: 119 mg (3.0 ml, 80mg/2ml ampule) q8h IV [calculated on
              basis of 1.7 mg/kg] plus consider giving 5 mg q24h
              Intrathecal
        Comments: Monitor serum concentrations

        Since high concentrations of penicillins can inactivate
        aminoglycosides, do not mix these two antibiotics in the same
        IV bottle.
```

Figure 3, continued

Similarly, the CADUCEUS program [Miller et al. 1982] gathers evidence for any of about 800 diagnoses of internal medicine. Most evidence gathering programs do not search exhaustively, however. Both the MYCIN and CADUCEUS programs are intelligent about using known data to avoid whole classes of unlikely explanations, and they gather some initial data to focus on classes of likely explanations. The search space includes all combinations of likely causes so a one-by-one testing, as in the generate and test method, is too inefficient. Instead, the MYCIN program structures its search by chaining backward through sets of inference rules -- from conclusions to the premises that need to obtain for the conclusions to be made. Thus not all possible questions about the patient are asked -- only the relevant data

are gathered. As evidence is gathered about a patient, some premises are found to be unsatisfied while others are followed up with more evidence gathering. In the end, there is positive and negative evidence for several diagnoses which needs to be weighed and put together in a coherent hypothesis mentioning one or more of the likely causes.

2.3. Heuristic Search [or Guided Search]

A more open-ended search procedure is heuristic search. Starting with a generator of possible hypotheses, for example, chemical structure descriptions, we use heuristics to focus and constrain the search. The heuristics are part of an evaluation function that guides the search by pruning or reordering paths through the search space. Pruning has the effect of ruling out absurd or implausible parts of the space. Reordering has the effect of selecting the most plausible parts of the space. The DENDRAL program, which explains spectroscopic data from unknown chemical samples, depends on rules from chemistry for pruning. These constraints are pieces of general information about chemistry, such as chemical stability, and about specific structural features present or absent in the structure of the unknown.

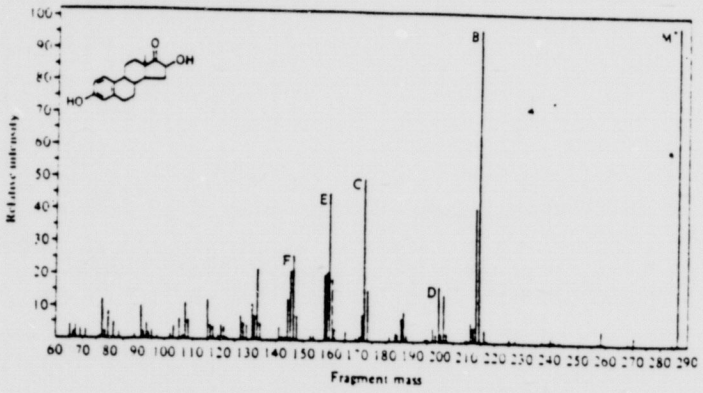


Figure 4: Analytic data (a mass spectrum) and the chemical structure that DENDRAL correctly hypothesized as the best explanation. In this case, constraints from other data were used as well. A mass spectrum is a plot of the masses of fragments of the compound (x-axis) against the relative abundance of fragments at each mass (y-axis).

The essential component of heuristic search is a generator of hypotheses. In many cases, such as DENDRAL, the base-level generator is exhaustive: it can enumerate the complete space of hypotheses of a class.[4] As mentioned, the program has the capability of producing an exhaustive list of structures from any collection of chemical atoms. But with only one or two dozen atoms the number of combinations of those atoms in structural descriptions explodes to the tens of millions. Thus the generator must be constrained. As soon as general

rules and heuristics about chemical stability are introduced, the number of hypotheses consistent with the data can be reduced by one or two orders of magnitude. Introducing specific structural information about the sample further reduces the number of structures consistent with the data to a very few, sometimes to a unique structure.

It is not necessary that the generator be exhaustive if we are willing to relinquish the assurance that every possible hypothesis in the space has been considered implicitly or explicitly. Instead of a generator of all syntactically possible hypotheses, then, substitute a generator of plausible hypotheses. In the Meta-DENDRAL program [Buchanan and Mitchell 1978], the task is to find explanatory rules that explain a collection of data. The program searches the space of rules by generating plausible rules, under constraints. A rule constructed by Meta-DENDRAL relates important structural features of chemical compounds with mass spectrometry fragmentation processes. As shown in Fig. FIG5, the syntax of a mass spectrometry rule is a conditional sentence relating (premise) a generalized description of part of a chemical structure and (conclusion) a description of where the molecular structure will fragment in the environment of the described part. The observed data from these processes in the mass spectrometer are spikes in a plot of fragment masses against relative abundance of fragments of each mass. The explanation of a large collection of data is a set of fragmentation rules describing the general behavior of a class of compounds.

IF *<description of two fused six-membered rings>*
THEN *<description of fragmentation at the ring juncture>*

Schematically (collapsing left and right sides of rule):

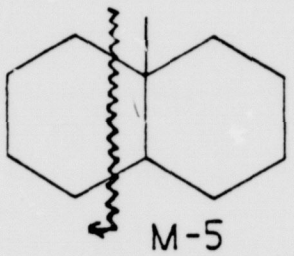


Figure 5: A general rule of mass spectrometry that Meta-DENDRAL rediscovered. This rule predicts that a 6-membered ring will fragment next to a ring juncture.

Meta-DENDRAL's search space is enormous, partly because the program is looking for a set of rules that explain most of the data, not just a single hypothesis. It reasons about underlying processes occurring in the instrument, which further compounds the search because any one data point may have several plausible explanations in terms of processes. It constrains the search in three general ways. (1) It generates only rules for which there is some positive evidence in the first place -- thus avoiding exhaustive generation. (2) It considers only rules that are consistent with a crude, but helpful, model of mass spectrometry

processes, which we call the "half-order theory". (3) It considers rules in two stages: in the initial search it estimates the amount of negative evidence a rule is likely to encounter (based on its level of generality), and in a refined search it modifies rules based on a careful look at the negative evidence actually found in the data.

2.4. Opportunistic Search [or Problem Solving with a Blackboard Model]

Another computational model for hypotheses formation programs is known as opportunistic search. While it is still correct to use the search metaphor to describe the exploration of plausible hypotheses, there is no single generator in this model. The generator is replaced by a collection of heuristic procedures that suggest new hypotheses, or modifications to old ones. This often referred to as the Blackboard Model because communication among all the heuristic procedures is through a common data structure called the blackboard. It is called opportunistic because the procedures respond to changes on the blackboard. New pieces of data trigger some new inferences; expectations about confirming or disconfirming data trigger other procedures.

The HASP/SIAP program [Nii et al. 1982], for example, explains underwater acoustic data using the Blackboard Model. Noises from ships' engines, generators, pumps and other equipment are monitored and recorded in a sonogram. The program interprets those data as a collection of ships of specific types in the region. Interference from islands or from other ships in the ocean can cause ships to be momentarily lost, but expectations are then created that they will reappear shortly along extrapolations of their previous paths.

Another explanatory program that uses this model is CRYSALIS [Engelmore and Terry 1979]. CRYSALIS interprets X-ray crystallographic data from proteins by finding three-dimensional descriptions of amino acid sequences. The space of possible hypotheses is very large so the interpretation must be driven as much by the data as by a model of what constitutes a plausible hypothesis.

In both these programs, and others using the blackboard model, there is no single generator of possible, or even plausible, hypotheses. Inference rules postulate changes to the current best interpretation of the data based on new data or other changes in the partial interpretation so far.

3. Conclusions

We have mentioned over a dozen AI programs that formulate hypotheses to explain empirical data. Using these programs themselves as data points, what can we conclude about methods for hypothesis formation? Because these programs fit within the paradigm of search, we conclude first that hypothesis formation can be mechanized by systematic search of a space of possible hypotheses. The essential components of a search program are (a) one or more generators of hypotheses; (b) a well-defined vocabulary and syntax of hypotheses; (c) knowledge about the domain that can be used to constrain the search; (d) evaluation functions that can guide the search and define a satisfactory solution. Of these elements, the hypothesis generator is the most critical. There are several ways of constructing a generator, which we discussed as exhaustive, selective, heuristic and

opportunistic search. None of the programs mentioned here uses exhaustive search because of its appetite for computer cycles.

Selective search assumes that the data can be explained by selecting the most appropriate classification, as in medical diagnosis. For example, a patient's signs and symptoms are explained by diagnosing the illness as meningitis caused by an e coli infection. Both the type of infection (meningitis) and the cause (e coli) are selected from relatively short lists of possibilities. In principle, the selection is made by gathering evidence for and against each hypothesis on a list. But in practise, whole classes of hypotheses may be eliminated by small clusters of evidence early in the search. When the lists are organized in a hierarchy, selective search can prune classes of hypotheses without considering the individual numbers of the class. Thus selective search is more efficient than exhaustive search.

More than the other search methods, evidence gathering involves determining degrees of confirmation. With imprecise and noisy data and inexact inference rules, data can be found to lend some support to almost any hypothesis. A fever, for example, is evidence for almost every infection. Similarly, data can almost always be found to reduce one's confidence in any hypothesis. How much one should weigh individual pieces of evidence and how one should combine weights are subjects treated theoretically in the literature on confirmation. They have found practical expression in several AI programs such as MYCIN, PROSPECTOR and CADUCEUS.

Heuristic search combines the virtue of systematic exploration of a hypothesis space with the efficiency of a focused, selective search. The main strengths of the method are:

1. We can guarantee that all implausible hypotheses have been excluded and the remaining hypotheses are all and only the plausible ones. (The guarantee holds only within the conceptual frame defined for the program but is good for the life of that framework).
2. We can encode a scientist's incomplete and uncertain knowledge of the domain to help define the criteria of plausibility. The programs can be given these items in many levels of certitude--from those most central to the paradigm to those that are little more than ephemeral intuitions [Buchanan 1979].

The major limitation of heuristic search in any domain is the necessity of finding (or inventing) a generator of possible solutions. In the case of molecular structures, finding the generating algorithm took many years. Lederberg's notational algorithm for unringed graph structures (described in [Lindsay et al. 1980]) was mapped into a generating algorithm with little difficulty, but the symmetries of cyclic graphs complicated the generation problem immensely.

A second major limitation on heuristic search is the necessity of finding heuristics, or rules of thumb, that guide the generator and constrain it from producing all syntactically allowable hypotheses. For rule generation it was necessary to find heuristics that steer the generator toward the small number of interesting rules and away from the very large number of uninteresting rules. The problem is that it is difficult to find these guiding principles.

Opportunistic search buys us the benefits of the search model without the necessity of defining a single, uniform generator of hypotheses within the space. The method takes

advantage of a scientist's knowledge about what to do next: how to construct a partial explanation of data and how to modify and refine it into a satisfactory, more complete explanation. In problem areas for which a generator of the entire space cannot be designed, or implemented efficiently, a collection of opportunistic rules can provide a method for finding plausible explanations. They start with small "islands of certainty", that is, interpretations of parts of the data that are nearly unambiguous. Then the rules augment the best hypothesis, postulate alternatives to try to disconfirm and note the expected data that would confirm or disconfirm an hypothesis. Since hypothesis elements are considered because there is some positive reason to do so, the exploration is highly focused on the relevant parts of the space.

The major drawback of opportunistic search, however, is the absence of a guarantee that all relevant classes of hypotheses have been considered. The total space is not defined by the method and only parts of the total space are considered explicitly or implicitly.

In summary, we initially asked by what methods a computer program can find plausible explanations of empirical data. One answer is by searching a space of hypotheses. The next question is how a computer can search a space. To this we offered four related (and non-exclusive) answers based on generalizations from several programs:

- Avoid exhaustive search, since it is inefficient for large spaces.
- Use selective search if the problem of explaining data is essentially one of classification.
- Use heuristic search if the space is large and if there exists both a generator of hypotheses and sufficient knowledge to guide it;
- Use opportunistic search if there is no systematic generator of hypotheses and scientists' heuristics can be used for plausible inference from data to partial hypotheses and from partial hypotheses to confirming and disconfirming experiments.

The efficiency of those methods, and of the programs using them, depends very much on the knowledge they have been given for guiding the search. Because much of the world's knowledge is qualitative, however, the methods that use this knowledge for guidance must be symbolic and not strictly numerical. That is the power of AI.

NOTES

[1]I am grateful to Dr. Derek Sleeman, and Mr. Tom Dietterich for comments on early drafts of this paper, and to Dr. Lindley Darden for discussions. This work was supported in part by DARPA [Contract # MDA903-80-C-0107], ONR [Contract # N00014-79-C-0302], NLM [Contract # NLM 1 P01 LM03395], and NIH [Contract # NIH RR 00785-10.]

[2]Two experiment planning programs worth noting are MOLGEN [Friedland 1979, Stefik 1980] and SECS [Wipke et al. 1977]. MOLGEN designs experiments in molecular genetics to accomplish goals such as splicing genes into bacteria. SECS designs chemical synthesis plans to accomplish goals of synthesizing organic chemical samples in the laboratory. Both use large amounts of factual and heuristic knowledge and both have demonstrated their ability to aid scientists.

[3]Newell and Simon, among others, have advocated the fundamental role of search in both human and machine problem solving since the earliest days of AI. See, for example, [Newell and Simon 1976]. For a full discussion of search see [Barr and Feigenbaum 1982].

[4]It is worth noting that the DENDRAL structure generator also avoids duplication. Thus there is no possibility of an infinite loop of generating duplicate structures repeatedly, an important feature of a mechanized procedure.

References

Barr, A. and Feigenbaum, E.A. (eds.). (1982). The Handbook of Artificial Intelligence. (Vols. I and II). Los Altos, CA: Kaufmann.

Blum, R.L. (1981). Discovery and Representation of Causal Relationships from a Large Time-Oriented Clinical Database: The RX Project. Unpublished Ph.D. Dissertation, Stanford University. Xerox University Microfilms Publication Number DA8208819.

Brown, J.S. and Burton, R. (1978). "Diagnostic Models of Procedural Bugs in Basic Mathematical Skills." Cognitive Science 2: 155-192.

Buchanan, B.G. and Feigenbaum, E.A. (1978). "DENDRAL and Meta-DENDRAL: Their Applications Dimension." Artificial Intelligence 11: 5-24.

---------- and Mitchell, T.M. (1978). "Model-Directed Learning of Production Rules." In Pattern-Directed Inference Systems. Edited by D.A. Waterman and F. Hayes-Roth. New York: Academic Press. Pages 297-312.

----------. (1979). "Steps Toward Mechanizing Discovery." Tech. Report HPP-79-28 Computer Science Dept., Stanford University. Presented at Pittsburgh University Conference on the Logic of Diagnosis.

Campbell, A.N.; Hollister, V.F.; Duda, R.O.; and Hart, P.E. (1982). "Recognition of a Hidden Mineral Deposit by an Artificial Intelligence Program." Science 217: 927-929.

Davis, R.; Shrobe, H.; Hamscher, W.; Wieckert, K.; Shirley, M.; and Polit, S. (1982). "Diagnosis Based on Description of Structure and Function." In Proceedings of the Second National Conference on Artificial Intelligence. Menlo Park, CA: American Association for Artificial Intelligence. Pages 137-142.

Engelmore, R.S. and Terry, A. (1979). "Structure and Function of the CRYSALIS System." In Proceedings of the International Joint Conference on Artificial Intelligence. Pages 250-256.

Erman, L.D.; Hayes-Roth, F.; Reddy, D.R.; and Lesser, V.R. (1980). "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty." Computing Surveys 12: 213-253.

Friedland, P. (1979). Knowledge-Based Experiment Design in Molecular Genetics. Unpublished Ph.D. Dissertation, Stanford University. Xerox University Microfilms Publication Number DEM80-11638.

Genesereth, M. (1982). "Diagnosis Using Hierarchical Design Models." In Proceedings of the Second National Conference on Artificial Intelligence. Menlo Park, CA: American Association for Artificial Intelligence. Pages 278-283.