Toward Automated Discovery in the Biological Sciences

Bruce G. Buchanan and Gary R. Livingston

■ Knowledge discovery programs in the biological sciences require flexibility in the use of symbolic data and semantic information. Because of the volume of nonnumeric, as well as numeric, data, the programs must be able to explore a large space of possibly interesting relationships to discover those that are novel and interesting. Thus, the framework for the discovery program must facilitate proposing and selecting the next task to perform and performing the selected tasks. The framework we describe, called the agenda- and justification-based framework, has several properties that are desirable in semiautonomous discovery systems: It provides a mechanism for estimating the plausibility of tasks, it uses heuristics to propose and perform tasks, and it facilitates the encoding of general discovery strategies and the use of background knowledge. We have implemented the framework and our heuristics in a prototype program, HAMB, and have evaluated them in the domain of protein crystallization. Our results demonstrate that both reasons given for performing tasks and estimates of the interestingness of the concepts and hypotheses examined by HAMB contribute to its performance and that the program can discover novel, interesting relationships in biological data.

The biological sciences are rich with observational and experimental data characterized by symbolic descriptions of organisms and processes and their parts as well as numeric data from high-throughput experiments. The complexity of the data and the underlying mechanisms argue for providing computer assistance to biologists. Initially, computational methods for investigations of relationships in biological data were statistical (Sokal and Rohlf 1969). However, when the DENDRAL project demonstrated that AI methods could be used successfully for hypothesis formation in chemistry (Buchanan and Feigenbaum 1978; Buchanan, Sutherland, and Feigenbaum 1969), it was natural to ask whether AI methods would also be successful in the biological sciences.[1]

Data in the biological sciences have been growing dramatically, and much of the computational effort has been on organizing flexible, open-ended databases that can make the data available to scientists. After the initial demonstrations of the power of applying machine learning to biological databases (Harris, Hunter, and States 1992; Qian and Sejnowski 1988), the application of machine learning to biological databases has increased. It is now possible to carry out large-scale machine learning and data mining from biological databases. Catalysts for this research were the Intelligent Systems in Molecular Biology conferences, the first of which was held in 1993. This conference brought together people from diverse groups, all with the realization that biological problems were large and important and that there was a need for heuristic methods able to reason with symbolic information.

Toward Automated Discovery

The end point of scientific discovery is a concept or hypothesis that is interesting and new (Buchanan 1966). Insofar as there is a distinction at all between discovery and hypothesis formation, discovery is often described as more

Copyright © 2004, American Association for Artificial Intelligence. All rights reserved. 0738-4602-2004 / $2.00

opportunistic search in a less well-defined space, leading to a psychological element of surprise. The earliest demonstration of self-directed, opportunistic discovery was Doug Lenat's program, AM (Lenat 1982). It was a successful demonstration of AI methods for discovery in a formal domain characterized by axioms (set theory) or rules (games). AM used an agenda-based framework and heuristics to evaluate existing concepts and then create new concepts from the existing concepts. It continued creating and examining concepts until the "interestingness" of operating on new or existing concepts (determined using some of AM's heuristics) dropped below a threshold. Although some generalization and follow-up research with AM was performed (Lenat 1983), this research was limited to discovery in axiomatic domains (Haase 1990; Shen 1990; Sims 1987).

Our long-range goal is to develop an autonomous discovery system for discovery in empirical domains, namely, a program that peruses large collections of data to find hypotheses that are interesting enough to warrant the expenditure of laboratory resources and subsequent publication. Even longer range, we envision a scientific discovery system to be the generator of plausible hypotheses for a completely automated science laboratory in which the hypotheses can be verified experimentally by a robot that plans and executes new experiments, interprets their results, and maintains careful laboratory records with the new data.

… we envision a scientific discovery system to be the generator of plausible hypotheses for a completely automated science laboratory ….

Currently, machine learning and knowledge discovery systems require manual intervention to adjust one or more parameters, inspect hypotheses to identify interesting ones, and plan and execute new experiments. The more autonomous a discovery system becomes, the more it can save time, eliminate human error, follow multiple discovery strategies, and examine orders-of-magnitude more hypotheses in the search for interesting discoveries (Zytkow 1993).

AI research on experimental planning systems has produced numerous successful techniques that can be used in an automated laboratory. For example, Dan Hennessy has developed an experiment planner for the protein crystallization problem discussed later that uses a combination of Bayesian and case-based reasoning (Hennessy et al. 2000). Because the number of possibly interesting discoveries to be made in any large collection of data is open ended, a program needs strong heuristics to guide the selection of lines of investigation.

No published system completely combines all phases of the empirical discovery process, although planning systems for knowledge discovery in databases (KDD), such as the framework presented in Engels (1996), perform sequences of tasks for a discovery goal provided by a user. Similarly, multistrategy systems such as that developed by Klosgen (1996) perform multiple discovery operations, but again, the discovery goals are provided by a user, as is evaluation of the discovered patterns. The research presented here describes and evaluates an agenda- and justification-based framework for autonomous discovery, coupled with heuristics for deciding which of many tasks are most likely to lead to interesting discoveries.

A Framework for Discovery

It is essential that a discovery program be able to reason about its priorities because there are many lines of investigation that it could pursue at any time and many considerations in its selection of one. Keeping an explicit agenda allows examination of the open tasks, and keeping explicit reasons why each task is interesting allows comparing relative levels of interest. We use an agenda- and justification-based framework, which is similar to the framework of the AM and EURISKO programs (Lenat 1983, 1982): It consists of an agenda of tasks prioritized by their plausibility. As in AM, a task on the agenda can be a call to a hypothesis generator to produce more hypotheses or explore some of the properties of hypotheses (or objects mentioned in them) already formed. *Items* are the objects or hypothesis (and sets of these) examined by the discovery program, and a *task* is an operation on zero or more items. For example, one task might be to find patterns (using an induction engine) in a subset of the data that have an interesting common property, such as being counterexamples to a well-supported rule. Although Lenat's programs discovered interesting conjectures in axiomatic domains such as set theory and games, those programs also contained general, domain-independent heuristics of the same sort used in empirical domains.

To evaluate our framework, we developed the prototype discovery program HAMB (Livingston 2001) that finds interesting, new relationships in collections of empirical data.[2] A key feature of HAMB is its domain-independent heuristics that guide the program's choice of relationships in data that are potentially interesting. HAMB's primary generator of plausible hypotheses is an inductive generalization program that finds patterns in the data; in our case, it is the rule-induction program RL (Provost and Buchanan 1995). RL is an inductive generalization program that looks for general rules in a collection of data, where each rule is a conditional sentence of the form

IF f_1 and f_2 and ... and fn
THEN class = K (with $CF = c$)

Each feature (f) relates an attribute (a variable) of a case to a named value, and a degree of certainty (CF) is attached to each rule as a measure of evidential support in the data; for example:

IF SEX = male and AGE < 30 and
ADDRESS = pittsburgh
THEN PERS = geek (CF = 0.6)

The conditional rule, which is easily understood by anyone who knows the meanings of the variable names, thus says that if a case matches all the antecedent conditions, then it is likely to be a member of the named class (K). Thus, the items in hamb's ontology are attributes, cases, rule conjuncts, and rules, plus sets of these. The cases and the attributes used to describe them are taken directly from the database.

On each cycle, heuristics can create tasks that result in new items or hypotheses, or tasks that examine some of the properties of those items or hypotheses. Each task must have accompanying text justifications for performing it, which are called *reasons*, qualitative descriptions of why a task might be worth performing (for example, sets of exceptions to general rules are likely to be interesting), and each reason must have an assigned strength, which is a relative measure of the reason's merit.

A task's plausibility is an estimate of the likelihood that performing the task will lead to interesting discoveries, and it is calculated as the product of the sum of the interestingness of the items involved in the task and the sum of the strengths corresponding to the reasons assigned to the tasks, as illustrated in the following equation:

$$\text{Plausibility}(T) = (\Sigma\, R_T) \bullet \{\Sigma\, \text{Interestingness(IT)}\}$$

where T is a task, (R_T) is the set of the strengths of T's reasons, and {Interestingness(IT)} represents the sum of the estimated interestingness of T's items.

Tasks are performed using heuristics and, when executed, create new items for further exploration and place new tasks on the agenda. When proposing a new task, a heuristic must also provide reasons and corresponding strengths for performing the task. If new reasons are given for performing a task already on the agenda, then they are attached to the existing task, increasing its plausibility. Therefore, the framework provides three additional properties that Lenat (1982) identified as desirable when selecting the next task to perform:

First, the plausibility of a task monotonically increases with the strength of its reasons. Therefore, with all else being equal, a task with two reasons will have a greater plausibility than a task with only one of those reasons. If a new supporting reason is found, the task's value is increased.[3] The better that new reason, the bigger the increase.

Second, if a task is reproposed for the same reason(s), its plausibility is not increased.

Third, the plausibility of a task involving an item C should increase monotonically with the estimated interestingness of C. Two similar tasks dealing with two different concepts, each supported by the same list of reasons and strengths of reasons, should be ordered by the interestingness of those two concepts.

Thus, the top-level control of the framework is a simple loop: (1) calculate the plausibilities of the tasks; (2) select the task with the greatest plausibility; and (3) perform the task, possibly resulting in the creation or examination of items, the evaluation of relationships between items, and the proposal of new tasks. At the end of each iteration of this loop (called a *discovery cycle*), a stopping condition is checked to determine if further exploration is warranted. In our prototype program, HAMB, the stopping condition is that either the plausibility of all tasks on the agenda falls below a user-specified threshold (that is, no task is interesting enough), or the number of completed discovery cycles exceeds a user-defined threshold. In cases of repeated consideration of the same task, the system detects the possible deadlock and moves to the next most interesting task.

The use of the agenda- and justification-based framework to propose and select data-mining tasks is unique to HAMB. The explicit representation of the determination of plausibility in the framework allows implementing programs to easily reorder the tasks when evidence is found in the data to do so. Other data-mining systems use an explicit notion of the interestingness of its findings to guide their behavior. This explicit representation of plausibility, reasons and strengths for performing tasks, and interestingness allows notions of interestingness to be changed without reprogramming and facilitates other changes to behavior, such as modifying how strongly HAMB should consider a reason when assessing the plausibility of a task.

Application to Protein Crystallography

X-ray crystallography is the primary means of determining three-dimensional structures of proteins and other macromolecules. After isolating and purifying a macromolecule, crystallographers must grow crystals that are sufficiently large and regular that the data pro-

Most crystallographers acknowledge that growing good crystals is a major (perhaps the major) rate-limiting step in structural studies.

duced when they diffract X-rays can be interpreted as a high-resolution structure. Most crystallographers acknowledge that growing good crystals is a major (perhaps the major) rate-limiting step in structural studies. Growing crystals can take many weeks or months when it is successful, with little theoretical guidance about the laboratory conditions that promote success.

The problem we address is discovering conditions under which macromolecules of different classes are likely to crystallize and grow large, regular crystals. We started with published data from numerous crystal-growing experiments, described later, and asked HAMB to find interesting relationships that could be useful to crystallographers and technicians in the laboratory. The input to the system is a set of records of successful and unsuccessful crystal-growing experiments, including the size of the protein, the chemicals used as precipitating agents and buffers, and the experimental conditions such as temperature. The output of the discovery system is a set of associations and other relationships that might constitute interesting heuristics for promoting crystal growth in new cases.

Methods

The macromolecule crystallization data set consists of reports of experiments for growing crystals of proteins, nucleic acids, or larger complexes, such as proteins bound to DNA, for X-ray diffraction and subsequent determination of three-dimensional structure (Hennessy et al. 2000). These data have been problematic for machine learning and clustering techniques for a variety of reasons: no clear target attribute, a high level of redundancy in the data, heterogeneous data (for example, nucleic acids crystallize under a very different set of conditions than proteins), and a high degree of noisy and incomplete data.

The database was derived from a subset of the Biological Macromolecule Crystallization Database (Gililand, Tung, and Ladner 1996), from which we selected 2,225 cases. These data were supplemented with additional chemical information: macromolecule type, the "perceived role" of the additives in the experiment (for example, precipitant, buffer), the ions the additives break down into, the net charge of the ions, the buffering capacity of the growth medium, and so on (Hennessy et al. 2000). The total number of attributes in this new database was 170, with several attributes having missing values for many of the cases. The intent of adding new information was to give the discovery system more possibilities for finding plausible discoveries. However, much of the new information overlaps information already present in the data and contains many known dependencies. Thus, although the additional information augments the database and can be useful to a discovery program, it also increases the redundancy and the number of nonnovel patterns in the database, which can make it difficult to identify the interesting discoveries and can lead to overfitting (Mitchell 1997).

The attributes in our augmented data set include (1) macromolecular properties—macromolecule name, macromolecule-class name, and molecular weight; (2) experimental conditions—pH, temperature, crystallization method, macromolecular concentration, and concentrations of chemical additives in the growth medium; and (3) characteristics of the grown crystal (if any)—descriptors of the crystal's shape, for example, crystal form, and space-groups-description and its diffraction limit (which measures how well the crystal diffracts X-rays).

We ran HAMB on the full database, seeding the agenda with examine tasks for examining each of the attributes, which were put onto the agenda when HAMB loaded the database. While performing these seed tasks, HAMB examined simple relationships among the attributes, one of which examines, using heuristics, how predictive one attribute might be of another. When a good predictor of an attribute was found, heuristics caused HAMB to propose the task of selecting a training set of cases that will be used to induce a rule set to predict the values of the attribute. After selecting a training set of cases, heuristics cause HAMB to propose the task of selecting a set of attributes, the *feature set*, that will be used to form the rules during the inductive process. After HAMB has selected a feature set, a task for selecting the parameters will be given to an induction program to induce the rules. Once an initial set of rules was created, HAMB used its heuristics that apply to rule sets, individual rules, attributes, and relationships to create new tasks, which were added to the agenda. We let HAMB run without intervention until there were no tasks on the agenda with plausibility above threshold (1.0). After 33,204 discovery cycles, HAMB found 575 items it considered interesting. Descriptions of three cycles are shown in figure 1 and illustrate the program's abilities to propose new tasks, reason about the appropriateness of the tasks, and direct its behavior toward tasks more likely to produce interesting discoveries.

Cycle 5 in figure 1a illustrates how an attribute is selected as an interesting target concept for rule induction. During cycle 5, a new task is added to the agenda to select a training set for inducing rules predicting CRYSTAL-FORM, which is the first step in inducing rules predicting CRYSTAL-FORM.

A. Cycle 5

HAMB examines predictivity relationships involving the attribute crystal-form before the other attributes because it has a higher estimated interestingness at this point in the discovery cycle.

- HAMB discovers that there are two strong predictors of crystal-form and that crystal-form is highly predictive of several others.
- One of HAMB's heuristics suggests that it is easier to induce good rules for a target attribute with many good predictors, causing hamb to propose a task to select a training set for inducing rules predicting crystal-form each time HAMB discovers a good predictor of crystal-form.
- Similarly, each time HAMB discovers that crystal-form is a good predictor of another attribute, HAMB proposes a task to select a training set for that attribute.

This excerpt illustrates a heuristic opportunistically identifying a potential rule-induction target.

B. Cycle 1,029

HAMB selects a training set for inducing rules predicting crystal-form's values before doing so for other attributes primarily because crystal-form has the greatest a priori interest to the user but also because HAMB has discovered many good predictors of crystal-form.

- HAMB creates the training set from the discovery database's examples that do not have uninformative values for crystal-form (for example, missing values, or "miscellaneous"), avoiding the induction of many uninteresting rules.
- The selected training set is small, containing only 164 of 1,482 possible examples.
- HAMB proposes the next step of inducing rules that predict crystal-form's values—selecting a feature set.
- Because HAMB believes that it is harder to induce good rules using a small training set, HAMB assigns very low strengths to the reasons it provides for selecting a feature set for crystal-form, causing HAMB to postpone performing the newly proposed task until cycle 11,172.
- When HAMB finally does induce a rule set for crystal-form, the accuracy of the induced rule set on the testing database is 0.02.

This excerpt shows HAMB assigning low strengths to a task its heuristics suggest would not lead to a promising line of investigation.

C. Cycle 1,202

HAMB selects a training set for inducing rules predicting the attribute add-iron-[ii]-citrate.

- HAMB selects a training set for add-iron-[ii]-citrate before most of the attributes, not because it has a high estimated interestingness but because HAMB found many reasons for performing this task.
- The selected training set contains 1,482 of 1,482 possible examples.

This excerpt depicts HAMB performing a task because it has determined that add-iron-[ii]-citrate might be an easy rule-induction target (not shown are the many reasons hamb found for doing so), not because add-iron-[ii]-citrate is especially interesting to the user.

Figure 1. Excerpts Taken from a Run of HAMB.

Parts A, B, and C show three separate discovery cycles. These excerpts are taken with the macromolecule crystal-growing data illustrating HAMB's ability to be opportunistic as well as postpone less promising tasks. The bullets identify what HAMB's heuristics note when examining the results of performing a task and what tasks they propose and the reasons and strengths given for performing the tasks.

Category	Description	Number	Percent
4	Individually, category 4 discoveries could be the basis of a publication in the crystallography literature, being both novel and extremely significant to crystallography.	0/575	0
3	In groups of about a dozen, category 3 discoveries could form the core of research papers in the crystallography literature.	92/575	16
2	Category 2 discoveries are about as significant as category 3 but are not novel.	192/575	33
1	Category 1 discoveries are not as interesting as category 2 or 3 but still are of some interest.	51/575	9
0	Category 0 discoveries are any discoveries that are not category 1, 2, 3, or 4.	240/575	42

Table 1. Significance and Novelty of 575 Discoveries.

Categorization of the significance and novelty (interestingness) of 575 discoveries made by HAMB from the macromolecule crystallization database. The 144 redundant rules removed during the semimanual filtering are counted as category 0 discoveries. Removing these 144 rules from the calculations results in only 22 percent (96/431) of category 0 discoveries.

Cycle 1029 in figure 1b shows HAMB assigning low strengths to a task its heuristics suggest would not lead to a promising line of investigation. One of HAMB's heuristics encodes the "common knowledge" of statistics that more reliable rules can be induced from larger rule sets. Thus, in this instance, HAMB assigns low strengths to the reasons given for selecting a feature set for the attribute CRYSTAL-FORM.

Cycle 1202 in figure 1c depicts HAMB beginning the process of inducing a rule set for a target attribute because it has found many reasons for doing so. To avoid unnecessary detail, we did not show the individual reasons.

Evaluation of HAMB's Discoveries

We manually removed 144 discoveries that we considered to be equivalent to others and asked our collaborator,[4] John Rosenberg, to assess the novelty and interest of the remaining 431 discoveries. He categorized them by their significance and novelty according to the categories shown in table 1, along with the numbers (and percents) of discoveries in each category. The redundant rules are counted as category 0 (uninteresting) discoveries in the table. Some of the specific items found to be interesting are shown in figure 2.

Some of HAMB's discoveries are interesting, not because they are extremely significant and novel but because they increase our confidence in HAMB's ability to detect patterns. Some of these patterns are rediscoveries of some of the crystallography "lore," which most practitioners would already know and use. Two of these known associations are given in figure 2a. Nucleic acids are highly negatively charged, requiring stabilization, and the most common reagent used to stabilize them is magnesium chloride. However, with proteins, magnesium chloride is avoided because of its tendency to promote the growth of crystals of magnesium salts rather than protein crystals. HAMB's ability to rediscover some of the patterns already known suggests that HAMB might be able to find equally useful patterns that are novel.

Other discoveries made by HAMB that increase our confidence in the results are its discovery of chemically reasonable patterns. These patterns often overlap those already known but have an obvious chemical interpretation. Two of these discoveries are given in figure 2b. Divalent cations (which include magnesium) effectively stabilize nucleic acids because of their chemical properties. They tend to be avoided with proteins because another (undesirable) property of divalent cations is that many salts of divalent cations are insoluble.

A few of the more interesting category 3 discoveries (apparently novel and significant) are presented in figure 2c. Category 3 discoveries can be helpful to crystallographers, but because the data are noisy and are biased by human preferences, further investigation is needed to confirm their validity. The first three rules given in figure 2c suggest that different procedures should be used for specific types of macromolecules. The second three rules in figure 2c suggest that different ionic strengths might be required when there are crystallizing

A. Rediscoveries known in the crystallography "lore" (category 2)

Macromolecule class is nucleic-acid ⇒ magnesium chloride is present
(true positives: 60; false positives: 54; sensitivity: 0.45; positive predictive value: 0.53; *p*-value: < 0.001)

Macromolecule class is protein ⇒ magnesium chloride is not present
(true positives: 1862; false positives: 60; sensitivity: 0.90; positive predictive value: 0.97; *p*-value: < 0.001)

B. Rediscoveries having clear chemical explanations (category 2)

Macromolecule class is nucleic-acid ⇒ inorganic divalent ions are present
(true positives: 81; false positives: 36; sensitivity: 0.35; positive predictive value: 0.69; *p*-value: < 0.001)

Macromolecule-class is protein ⇒ inorganic divalent ions are not present
(true positives: 1660; false positives: 262; sensitivity: 0.92; positive predictive value: 0.86; *p*-value: < 0.001)

C. Discoveries interesting and novel enough to warrant further investigation (category 3)

Macromolecule class is P.RNA.E ⇒ crystallization method is batch
(true positives: 141; false positives: 88; sensitivity: 0.39; positive predictive value: 0.62; *p*-value: < 0.001)

Macromolecule class is P.S.H ⇒ crystallization method is temperature crystallization
(true positives: 22; false positives: 30; sensitivity: 0.73; positive predictive value: 0.42; *p*-value: < 0.001)

Macromolecule class is P.S.L.O ⇒ concentration by evaporation
(true positives: 67; false positives: 14; sensitivity: 0.65; positive predictive value: 0.83; *p*-value: < 0.001)

Macromolecule class is enzyme ⇒ ionic strength is greater than 2.21 and less than or equal to 5.98
(true positives: 151; false positives: 638; sensitivity: 0.53; positive predictive value: 0.21; *p*-value: < 0.001)

Macromolecule class is P.S.H ⇒ ionic strength is greater than 5.98
(true positives: 90; false positives: 139; sensitivity: 0.28; positive predictive value: 0.39; *p*-value: < 0.001)

Macromolecule class is P.S.L ⇒ ionic strength is less than or equal to 2.21
(true positives: 114; false positives: 137; sensitivity: 0.32; positive predictive value: 0.45; *p*-value: < 0.001)

D. Other types of discoveries (categories 3 and 1)

Equivalence classes of attributes:
(CH < = –4, CH > = 4, ADD-SODIUM_AZIDE, SPEC-AZIDE)

Frequency of missing values of attributes:

Buffering capacity	0.69
Crystal form	0.67
Temperature	0.35
Ionic strength	0.26
Macromolecule concentration	0.25
Diflim	0.23
PH	0.13

Figure 2. Some of HAMB's Discoveries.

Parts A, B, C, and D show different kinds of discoveries from categories 1, 2, and 3. The statistics reported for each rule are calculated from a validation set not used to learn the rule. The p-value of a rule's positive predictive value is computed using the Fisher's exact test (Sokal and Rohlf 1969). p.rna.e macromolecules are proteins that bind to RNA and catalyze a chemical reaction that modifies it; p.s.h. ("heme"-containing) macromolecules are soluble proteins containing an iron-porphyrin prosthetic group (for example, hemoglobin and cytochrome); P.S.L. macromolecules are small proteins and peptides; and P.S.L.O. macromolecules are heterogeneous subgroups of P.S.L.

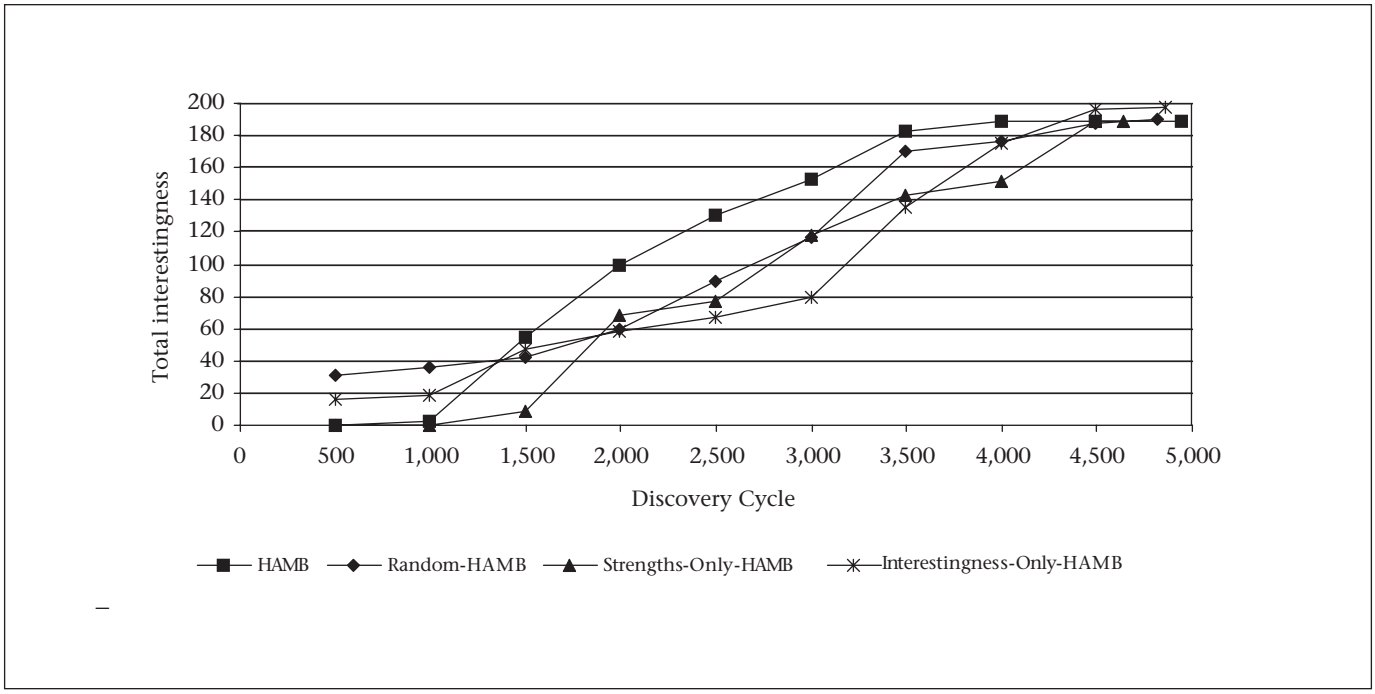


Figure 3. Plot of the Total Interestingness of Reported Discoveries Versus the Time Taken to Make Them for HAMB and Each of the Three Variations.

Total interestingness is the sum of the category numbers assigned to the reported discoveries. Discovery cycle is the discovery cycle after which the discoveries are reported. The curves converge because eventually nearly all the interesting discoveries from the initial database are found by all the methods (as a result of our only using a subset of the data during this study).

enzymes, "heme"-containing proteins, and small proteins. Because many crystallographers have their preferred crystallization techniques, experimental work is needed to discern whether these patterns represent human preferences or real chemical associations. However, in either case, the results would be useful.

Figure 2d presents some of the types of discoveries HAMB makes in addition to rules—attributes with equivalent extensions and attributes with frequently missing values. Both of these types of discoveries are useful in evaluating the quality of the data. The discovery of extensionally equivalent attributes that are not intensionally equivalent demonstrates that the data are insufficient for discriminating between the two attributes. The identification of attributes with a large proportion of missing values suggests that additional data might be needed to induce good rules for those attributes.

Lesion Studies of the Plausibility Function

Given an assessment of the degree of interest of more than 400 discoveries, we performed lesion studies to further evaluate the plausibility function. To perform this experiment, we used HAMB and three variations: (1) RANDOM-HAMB, which selects the next task to perform randomly from the tasks on the agenda; (2) REASONS-ONLY-HAMB, which computes the plausibility of a task as the sum of the strengths of the task's reasons; and (3) INTERESTINGNESS-ONLY-HAMB, which computes a task's plausibility as the sum of the estimates of the interestingness of the items involved in the task.

Figure 3 presents a graph of the sum of the interestingness of the discoveries given in periodic reports by the four versions of HAMB versus the discovery cycles that followed the generation of the reports. When comparing the graphs for HAMB and RANDOM-HAMB in figure 3, note that for the first approximately 1,500 discovery cycles, RANDOM-HAMB's total interestingness is greater than HAMB's, after which HAMB's scores equal or exceed random-HAMB's. We attribute this difference to preliminary investigations that HAMB performs on the attributes and their relationships before inducing rule sets. The additional time HAMB spends examining the attributes is useful in reducing the redundancy of its discoveries. The plot shown in figure 3 only reports the total interestingness of the reported discoveries, including redundancies. Another study, reported later, shows that HAMB is able to use this discovered knowledge to reduce the redundancy in its induced rule sets by 19 percent.

The difference between the two plots for HAMB and RANDOM-HAMB after 1,500 discovery cycles is statistically significant (p-value is 0.007).[5] Differences are significant between HAMB and REASONS-ONLY-HAMB (p-value is 0.003) and between HAMB and INTERESTINGNESS-ONLY-HAMB (p-value is 0.049).

We also compared the average interestingness of the discoveries made by the four versions of HAMB and found that the plot for RANDOM-HAMB was slightly better and that the p-value of the difference between these two plots is 0.015.

RANDOM-HAMB's tasks are randomly selected from the agenda, not from the space of all possible tasks; therefore, RANDOM-HAMB's behavior is aided by the heuristics that propose the tasks, biasing its task selection toward more appropriate tasks. If RANDOM-HAMB's tasks had been selected from the space of all possible tasks rather than the space of tasks chosen by the heuristics, RANDOM-HAMB's performance would probably have been much worse.[6]

Note that HAMB often uses the results of tasks performed earlier when performing the current task; therefore, the order in which the tasks are performed is important. For example, HAMB uses its discoveries of equivalence classes to reduce the number of redundant discoveries.

Evaluation of HAMB's Use of Domain-Specific Knowledge

We performed a lesion study to evaluate the effectiveness of some of HAMB's heuristics that use domain-specific knowledge. To perform this study, we used 500 cases randomly selected from the macromolecule crystallization data.

The unmodified version of HAMB with the complete domain theory was also run on this set of cases, as was a version of HAMB that used no domain knowledge. In the study described later, this second version is called *no-domain-knowledge*.

The heuristics tested during this study are as follows:

Heuristics reducing redundancy by eliminating synonyms: If an attribute is found in the feature set that is synonymous (as either discovered by HAMB or stated in the domain theory) with another attribute in the feature set, the attribute with the lesser estimated interestingness is removed from the feature set. A baseline version of HAMB omitted these heuristics and did not eliminate synonyms. It allowed the creation of 40 (19 percent) more redundant rules than did HAMB. This result was surprisingly low because the data contain many similar attributes. However, HAMB's definition of redundancy is very strict, requiring either intensional or extensional equivalence; therefore, only a few pairs of attributes met its strict criterion of similarity.

Heuristics reducing the number of uninteresting discoveries: The domain theory can contain knowledge about attributes and values that are uninteresting or meaningless to the user. HAMB's heuristics use this knowledge to avoid inducing rules containing uninteresting features (either in the left-hand side or right-hand side of a rule). A baseline version of HAMB omitted this heuristic and allowed the generation of 300 (141 percent) additional uninteresting rules.

Heuristics reducing the number of non-novel discoveries: HAMB uses domain knowledge to remove attributes that have a known association (by causation, definition, association, and so on) with the current target attribute. It also removes attributes that are discovered to be extensionally equivalent to the target attribute. The baseline version of HAMB used to test these heuristics omitted this use of domain knowledge and allowed the generation of 2,897 (1,367 percent) additional non-novel rules.

The regular version of HAMB induced 212 rules in this experiment, whereas the baseline no-domain-knowledge version induced 3,936 rules. Thus, HAMB was able to use domain knowledge to avoid the creation of 3,724 rules that are uninteresting by definition. Although the number of interesting rules is about the same in each case, the percentage of interesting rules shown to the user is much higher in the first case. In addition, HAMB only required 5,030 discovery cycles to finish, but the baseline no-domain-knowledge version required approximately 45,000 cycles. From these results, we conclude that the domain-specific knowledge used by these heuristics focuses HAMB on a smaller set of interesting rules in less time.

Evaluation of Generality

To evaluate the generality of the system, we used HAMB to perform discovery from a second database that was quite different in content: There were 930 cases of patients in rehabilitation after a medical disability, such as stroke or amputation. There are 11 attributes in the database, ranging from demographic data to admission and discharge scores of the patients' functional independence measures (FIM). Thus, this database represents a domain that is dissimilar to the macromolecule crystallization domain.

After running HAMB with these data, we presented HAMB's discoveries to the physician who provided us with the data, Dr. Louis Penrod of the University of Pittsburgh Medical Center. Some domain-specific knowledge was provided, which was referenced by some of the heuristics.

This knowledge consisted of specialization (for example, disability class is specialized by disability) and derivational (for example, admit FIM and discharge FIM are used to derive the amount of improvement) relationships among the attributes and their values. Penrod decided that there were 26 (9 percent) category 3 discoveries (novel and significant), of which 2 were bordering on category 4 (revolutionary), 5 (2 percent) were category 2 discoveries (non-novel but significant), 53 (18 percent) were category 1 discoveries (novel and marginally interesting), and 215 (71 percent) were uninteresting.

Because of the smaller number of attributes, we were able to represent almost all the known relationships among the attributes, which HAMB's heuristics were able to use to reduce the proportion of category 2 (previously known but significant) discoveries. Because this domain is a clinical domain in which discoveries must have strong empirical support before they can be used in a clinic, the criteria for the usefulness of a discovery are more stringent than those for the macromolecule crystal-growing domain, in which the crystallographer is looking for any clue or hint that would help him/her increase his/her chances of growing a useful crystal. Therefore, it is understandable that the proportion of uninteresting discoveries is greater for this domain (rehabilitation) than for the crystal-growing domain.

Our conclusion from this study is that the heuristics are general enough to make significant discoveries from two dissimilar domains.

Details of the HAMB Program

HAMB's input consists of the file containing the set of cases that it will use to make its discoveries (the *discovery database*), an optional testing set of examples (the *testing database*), and a domain theory file containing problem- and domain-specific information. HAMB reports as discoveries those items with interesting relationships or properties. A property or relationship is *interesting* if its value exceeds a threshold provided for each relationship or property. HAMB creates a report for each relationship or property, where the items having values for that property or relationship greater than or equal to the threshold are listed in decreasing order.

HAMB uses a variety of knowledge, both general domain-independent heuristics about performing the tasks and problem- and domain-specific information. The problem- and domain-specific information is kept in a separate file called a *domain theory file*. Thus, we have a clean separation of HAMB's knowledge; allowing the application of HAMB to a new problem only requires the changing of the problem-specific information in the domain theory file, not the framework or HAMB's heuristics. A study of HAMB's generality shows that the framework and heuristics are domain independent. The general heuristics we have implemented to date fall into three classes: (1) heuristics that select rule-induction targets and other goals worth pursuing; (2) heuristics that keep an item's properties and relationships sufficiently up to date, allowing a discovery system to select appropriate tasks without needlessly reexamining these properties and relationships after every task; and (3) heuristics that reference domain-specific properties to improve the quality of reported discoveries.

Although this set of heuristics is incomplete, we provide evidence that they are useful and partially accomplish our goal of guiding a nearly autonomous discovery system by separating interesting hypotheses from other associations found in a database. Because heuristics create and execute tasks, we sometimes refer to them interchangeably, although tasks are instantiated with specific arguments.

HAMB's Domain-Independent Knowledge about Performing Discovery

HAMB's domain-independent knowledge about using rule induction to perform discovery comes in a variety of types, three of which are discussed in the following paragraphs: (1) relationships among attributes that are useful to examine; (2) properties of items that help identify interesting items; and (3) heuristics that are used to perform the tasks, during which new items might be created, and new tasks might be proposed.

Relationships The space of relationships among items that could be examined is immense. One of the types of general knowledge about discovery that HAMB uses is knowledge about which of these possible relationships to examine. We defined a set of relationships that seemed useful when we manually performed the knowledge discovery process; these relationships overlap many of those used in data mining and association mining. Although some machine learning and data-mining systems look for one or two of these relationships, few systems look for as many as HAMB does.

Unexpectedly equivalent items: Relationships identifying unexpectedly equivalent items consist of pairs of items of the same item type (for example, two attributes) that have nearly identical extensions among the examples but that are not equivalent (as defined by the domain theory). Currently, only attributes and rule clauses are tested for unexpected

equivalence. During the evaluations of HAMB, several pairs of attributes and rule clauses were identified that were essentially equivalent in the cases given to HAMB but were not specified as equivalent in the domain theory.

Coupled items: Pairs of attributes or rule clauses are said to be *coupled* when one of them is always (allowing for a small degree of noise) used in a rule whenever the second is used. For now, the default in the domain theory is that no two attributes or rule clauses are coupled. Any couplings that HAMB discovers would be inconsistent with this assumption and assumed to be novel and interesting. Although potentially interesting, HAMB discovered none of these relationships in the study described here.

Exceptions: These relationships identify pairs of items where one specializes the other and has different implications. In particular, pairs of rules are exceptions when the first rule's left-hand side is specialized by the left-hand side of the second, but the class predicted by the general rule differs from the class predicted by the specialization. For example, "(age > 50) → has-pneumonia" and "(age > 50) and (fitness = good) → no-pneumonia."

Related items with significantly differing performance: Pairs of related items of the same type with significantly different performance are likely to be surprising and, thus, interesting. HAMB checks for either of two cases:

In case one, pairs of rules predict the same class, the right-hand side of the first rule is specialized by that of the second, and the more general rule has a significantly lower positive predictive value. An example of this type of relationship is the following pair of rules:

Rule 1 (general rule): "An organic amine is present → the macromolecule being crystallized is a nucleic acid," which predicts the type of macromolecule being crystallized correctly for 63 examples and incorrectly for 97 examples (PPV = TP/(TP + FP) = 0.39).

Rule 2 (specialization): "An organic amine is present AND ammonium is not present → the macromolecule being crystallized is a nucleic acid," which predicts 62 examples correctly and 25 examples incorrectly (PPV = 0.71).

In case two, pairs of rule sets are such that one is a subset of the other, and the subset has a significantly higher accuracy. This result is surprising because adding more rules would be expected to increase predictive accuracy. An example would be the pair of rule sets rule-set-1, which has an accuracy of 0.90 and consists of rules A and B, and rule-set-2, which has an accuracy of 0.80 and consists of rules A, B, and C.

Other tests described by Livingston, Rosenberg, and Buchanan (2001a) that signal interesting relationships among items include other kinds of surprises, consistent variation of measured values, and high information gain (Quinlan 1993) of an outcome variable with the addition of a predictive variable.

Properties Similar to relationships, the space of possible properties of items that could be examined is huge. Therefore, HAMB uses general knowledge to guide which of the possible properties to examine. There are 10 properties that make attributes interesting, 3 for example sets, 5 for the predicted values, 6 for rule clauses, 10 for rules, and 8 for rule sets. These properties are summarized here and described in more detail in Livingston (2001).

Properties measuring syntactic simplicity are designed to measure various aspects of syntactic simplicity, such as the number of cases in a set of cases, or the number of clauses in a rule's left-hand side. There are five of these properties: (1) the number of unique clauses used in the rules of a rule set, (2) the number of values of an attribute, (3) the number of clauses in the left-hand side of a rule, (4) the number of members of a rule set or a set of cases, and (5) the ratio of the number of cases in a set of cases to the total number of cases.

Properties measuring the performance of rules or rule sets measure the empirical support of rules and rule sets. There are seven of these properties: two for rule sets and five for rules. The two properties for measuring the performance of rule sets are (1) positive predictive value (TP / (TP + FP)) and (2) accuracy, including false negatives to be incorrect predictions. The five properties measuring performance for rules are (1) sensitivity (TP/(TP + FN); (2) specificity (TN / (TN + FP)); (3) positive predictive value (PPV); (4) negative predictive value (NPV) (TN / (TN + FN)); and (5) *p*-value, which is the statistical significance of a rule's PPV over the prior frequency of a rule's right-hand side (note that TP = true positives, FP = false positives, TN = true negatives, and FN = false negatives). The *p*-value is computed using the Fisher's exact test (Sokal and Rohlf 1969).[7]

Other properties of items that make them interesting include considerations of utilities and significance in the domain, the frequency of use of attributes in rule clauses, and the number of examples with missing values for an attribute. For example, this last property might indicate that 75 percent of the examples have missing values for the attribute weight, which might indicate that the data are of poor quality, particularly with respect to weight.

HAMB's Heuristics and Associated Tasks HAMB's agenda- and justification-based framework depends heavily on the heuristics that

Tasks Created by Heuristics	Summary
Examine-item (item)	Check item's membership in item groups, evaluate item's properties, and estimate its interestingness.
Examine-relationships-with (item)	For each type of relationship *R* defined to hold with item, propose an examine-*r*-relationships-with task.
Examine-r-relationships (item, *R*)	For each possible relationship of type *R* that could hold with item, if the relationship can be evaluated quickly, then do so; otherwise, propose a task for examining that relationship.
Examine-relationship ($item_1$, $item_2$, ... $item_n$, *R*)	Evaluate the *R* relationship among $item_1$, $item_2$, ... $item_n$.

Table 2. An Overview of HAMB's Tasks for Examining Items.

Tasks indicate the type of task, and the summary provides a brief description of the purpose of a task. Examine-item tasks that are proposed during initialization are shown.

create and perform tasks, evaluate the results of performing the tasks, and assign reasons to the newly proposed tasks. In addition, heuristics inspect the items and their relationships and estimate the interestingness of the items. We have tried to make our heuristics problem independent so that applying HAMB to new problems would not require revising the heuristics.

When it was practical, we decomposed the tasks HAMB would perform so that each task would only involve one fundamental operation. For example, we decomposed the induction of a rule set into several tasks: selecting a training set, selecting a feature set, selecting the parameters for the induction program, and calling the induction program to induce the rule set. This decomposition provided two benefits: (1) making the tasks and heuristics more modular, which facilitates their comprehension, modification, and the addition of new tasks and heuristics, and (2) putting the subtasks onto the agenda, allowing HAMB to use the results of performing earlier subtasks of a goal to reason about the fruitfulness of achieving the goal and allowing HAMB to delay less promising goals in favor of more promising ones.

HAMB's tasks can be divided into three groups based on function: (1) heuristics for examining items; (2) heuristics for creating new items using rule induction; and (3) heuristics for creating *exception sets,* sets of cases that are misprediced by induced rule sets.

Heuristics for examining items: These heuristics solve a novel problem for autonomous discovery systems—keeping the values of an item's properties, relationships, and interestingness sufficiently up to date without recalculating their values after every discovery cycle. Typically, this task is performed by the user when he/she is examining the results of performing a task. With potentially millions of items in a discovery program's ontology, reexamining the properties and interestingness of the items every time a task is performed is computationally expensive. Instead, our heuristics periodically re-evaluate the items to identify any new findings about them and to keep the values of their properties and estimated interestingness reasonably up to date. We hypothesize that merely keeping the property and estimated interestingness values close to the correct values will be sufficient. Whenever a heuristic indicates that some aspect (property, relationship, or interestingness) of an item might have changed, a task is proposed for reexamining the item.

Five types of tasks achieve the general goal of examining items: (1) *examine-item tasks*, which evaluate an item's properties and interestingness, and four additional tasks, which examine the item's relationships—(2) *examine-relationships-with*, (3) *examine-r-relationships*, (4) *examine-rule-families*, and (5) *examine-relationship tasks*. These tasks are presented in table 2.

Heuristics for creating new items by induction: When HAMB determines that an attribute might make an interesting target variable (such as an attribute is defined in the domain theory as one of several target attributes, or it discovers a potentially good predictor of the attribute's values), heuristics propose a task for selecting a training set of examples for inducing rules predicting the value of that variable. Performing the task of selecting a training set for a target variable begins the sequence of subtasks that HAMB performs to induce a rule set for the target variable: selecting a training set; selecting a feature set; selecting

Tasks	Summary
Select-training-set (target attribute)	Select a training set from which rules predicting target attribute will be induced.
Select-feature-set (target attribute, training set)	Select a feature set from which rules predicting target attribute will be formed.
Select-bias (target attribute, training set, feature set)	Select a bias for inducing rules predicting target attribute using hill climbing. Use cross-validation on training set to evaluate the biases.
Induce-rule-set (target attribute, training set, feature set, bias)	Induce a rule set predicting target attribute using the selected training set, feature set, and bias.
Create-exception-set (rule set, training set)	Create an exception set for rule set.

Table 3. An Overview of HAMB's Tasks for Creating New Items.

Tasks indicates the type of task; *summary* provides a brief description of the purpose of a task; and *calls* indicates which other tasks are proposed while tasks are being performed. Not shown is that examine-item tasks are proposed during initialization.

a parametric bias; and calling an induction program (RL) with the selected training set, feature set, and bias. Each of these subtasks corresponds to one of HAMB's tasks shown in table 3: select training set, select feature set, select bias, and induce rule set, respectively. To ensure that HAMB performs the tasks in the order given, the heuristics are designed so that after the completion of one of these tasks, the next task in the sequence is proposed.

HAMB often uses the results of the current task to determine the strengths of the reason given for performing the next task in the sequence, allowing HAMB to factor the results of the current task into its decision about when to perform the next task in the sequence, if at all. For example, the heuristic for selecting a training set proposes a task for selecting a feature set, with the strengths of the reasons given for the proposed task being proportional to the size of the training set (encoding the rule of thumb that inducing a rule set from a small training set might result in a rule set with poor performance on future cases).

Heuristics for exception sets: We added heuristics allowing HAMB to create exception sets for rule sets and then induce a new rule set from the exception set. We found this process to be useful when manually performing discovery because inducing rule sets from these exception sets can fill in the gaps of the "theory" formed by the initial rule set. An exception set is created for a rule set by selecting the rule set's counterexamples from the training set. After creating the exception set, HAMB proposes an *examine-item task* to examine the set and a *select-feature-set task* that will begin the process of inducing rules from the exception set.

HAMB's Background Knowledge: Domain- and Problem-Specific Knowledge

Most KDD and machine learning programs are only capable of using one or two types of background knowledge. In contrast, HAMB's heuristics use a wide variety of knowledge, which allows HAMB to use its discoveries to tailor its choice of tasks to the domain and the database provided to it for discovery.

HAMB's domain- and problem-specific knowledge consists of file and directory pointers, parameters that control its behavior, simple user models for estimating the interestingness of items, and semantic knowledge about the domain. This information is provided to HAMB using a domain theory file.

Parameters HAMB was written with flexibility as one of its primary design principles. Therefore, it has many parameters that affect its behavior: (1) file pointers, such as pointers to the discovery and testing databases; (2) parameters controlling the generation of output, such as the frequency with which reports of discover-

ies are printed; (3) parameters controlling the discovery process, such as the maximum size of the agenda; and (4) parameters used by the heuristics, such as similarity threshold, which HAMB uses to determine that two sets are essentially the same.

Simple User Models for Estimating Interestingness HAMB provides two methods for modeling a user's preferences: (1) a simple user model formed by adjusting weights in a hierarchical weighted sum that HAMB utilizes to estimate an item's interestingness, with the adjusted weights indicating the user's preferences, and (2) user-defined item groups, a facility for defining groups of items and assigning utilities that are attributed to members of the groups.

Simple user model: The hierarchical weighted sum used to estimate an item's interestingness is evaluated using three levels of abstractions and normalization. For the sake of simplicity, the abstraction hierarchies are mostly uniform. However, the properties and weights used in the hierarchies for the other item types vary considerably. This hierarchical weighted sum is presented in detail in Livingston (2001).

First, the values of the item's properties are calculated.

Second, the values of the properties are abstracted using weighted sums. Then, the values of the abstractions are normalized to fit between 0 and 100. For example, when estimating the interestingness of a rule, the properties PPV, NPV, *p*-value, rule-set usage ratio, specificity, and sensitivity are abstracted into empirical support, with a raw score of 0.18 (2.52 • 0.07) for semantic simplicity. Compared to the raw empirical support scores for other rules, this value is relatively low; after normalization, the rule's score for empirical support is low, 12.

Estimated interestingness is computed from the top-level abstractions using a final normalized weighted sum.

Our results demonstrate that HAMB's estimation of the interestingness of items is useful in guiding its discovery process. Moreover, in our experience, HAMB's behavior is not sensitive to minor changes to the weights.

Item groups: To define an item group, the user provides in the domain theory file a name for the group, a utility, and a predicate for determining membership in the group. These groups allow HAMB to factor a user's a priori interests in groups of items into its estimates of the items' interestingness through a special property, *item group utility,* which sums the utilities of the item groups for which an item is a member. This property is used in HAMB's estimation of interestingness. For example, suppose a user with a set of cases of macromolecule crystallization records from which discovery will be performed is more interested in attributes describing the outcomes of the experiments than attributes describing characteristics of the macromolecules and also is more interested in attributes describing characteristics of the macromolecules than attributes describing the presence of additives. To model these interests, three item groups could be defined with utilities of 100, 50, and 25, respectively.

Domain-Specific Knowledge HAMB's domain-specific knowledge can be divided into the following groups:

First is simple semantic information, such as potential target attributes and the value used to denote missing values.

Second is information about uninteresting attributes and values. This information is used to avoid the generation and reporting of many discoveries that are uninteresting because they use uninteresting attributes and values. Uninteresting values can be removed from the feature set of attributes from which rules are induced, eliminating the induction of many rules that are uninteresting because they use uninteresting values. Cases with uninteresting values for the target attribute are removed from the training set of cases used to induce the rules, avoiding the induction of rules predicting uninteresting values. Rules containing clauses using uninteresting values are eliminated by HAMB after the rules are induced.

Third is a simple time model of the attributes. This model is used by HAMB to remove from a feature set of attributes being formed for a target attribute those attributes whose values are set after the target attribute's values have been set. Thus, HAMB can avoid rules that represent the "future predicting the past" and generate a set of rules that is more likely to be causal.

Fourth is known relationships among the attributes and their values. These relationships are currently used by HAMB to avoid inducing rules representing relationships that are already known. We are currently improving HAMB's ability to perform more sophisticated reasoning with these attributes, which will allow HAMB to do things such as merging similar rules or identifying semantically redundant rules and eliminating them. Both of these capabilities can allow us to dramatically simplify induced rule sets. The types of known relationships that can be given to HAMB are detailed in the following subsection.

Known Relationships among Attributes The user can provide a variety of known relationships among attributes and their values:

Definitionally related: This one-to-many relationship indicates the construction of one

attribute from other attributes. For example, area would definitionally be related to length and width.

Translational equivalence: This one-to-one relationship indicates when two attributes measure the same feature but possibly (but not necessarily) using a different scale, such as the Celsius and Fahrenheit temperature scales.

Semantic equivalence: This one-to-one relationship indicates when two attributes should have identical values, such as two gauges that are used redundantly to ensure that a valid reading is obtained.

Abstraction: This one-to-one relationship indicates when the first attribute is an abstraction of the second, such as the second attribute representing the presence of a chemical compound and the first attribute representing the presence of a chemical grouping to which it belongs, such as organic or inorganic.

Discretization: This one-to-one relationship indicates when the first attribute is a discretization of a second numeric attribute, such as the discretization of temperature in Fahrenheit into below zero (< 0), subfreezing (> = 0 and < 32), subboiling (> = 32 and < 212), and above boiling (> = 212).

Known related: This one-to-one relationship indicates when one attribute is known to be related to a second attribute in a manner not expressible using the other relationships.

Conclusions

Data from the biological sciences are so voluminous and growing so rapidly that many potentially useful discoveries are not being made. Computer assistance is already available in the form of statistical and string-matching programs, but biological processes and organisms are often described in terms of symbolic features that carry considerable semantic content with them.

The primary conclusion from this work is that HAMB's general framework for discovery can be used with experimental and observational data in science to make interesting, novel discoveries of utility to laboratory scientists. An essential component in automating discovery is providing heuristics for selecting promising items to explore, that is, the next task to work on, because the space of possible items to explore is so large. The agenda- and justification-based framework gives us an explicit means for selecting tasks. Moreover, it is general enough, as are the heuristics guiding discovery, to work with empirical data from a wide variety of biological and nonbiological domains. We also believe that the heuristics that identify interesting discoveries capture many of the expert's criteria of interestingness, which were only made explicit in the context of the working program.

HAMB's strengths are (1) the large number of hypotheses that it examines as a result of its increased autonomy; (2) its ability to adapt its behavior to what it discovers and to select its own objectives (factoring in the user's interests), allowing it to pursue potentially interesting areas of investigation that were not expected; and (3) the ability to use domain- specific knowledge to aid its discovery process and to avoid reporting many uninteresting discoveries. In addition, HAMB is robust to noisy and incomplete data. Although HAMB would be suitable for most data-mining tasks, HAMB is particularly well equipped for performing exploratory analysis of new data where numerous hypotheses should be explored or where the learning goals are not well defined. Because of its ability to use domain-specific knowledge to avoid reporting many redundant or uninteresting discoveries, HAMB would be a useful tool for analyzing data with redundant attributes and many known relationships. For example, HAMB is well suited for gene-expression analysis—researchers would be interested in relationships involving many of the genes, gene-expression data are often noisy, and domain knowledge exists that HAMB could make use of.

In the problem area of protein crystallography, HAMB has been demonstrated to find interesting and novel relationships in published data about crystal-growing experiments. Some of these discoveries are rediscoveries in the sense that they are well known to crystallographers, just not to the program. Some discoveries, however, constitute interesting enough suggestions for what to do in the laboratory to promote crystal growth that laboratory resources have been spent on them.

Acknowledgments

We are particularly grateful to John Rosenberg, from the Department of Biological Sciences at the University of Pittsburgh, for the time and expertise he provided in critiquing the program's output and refining its heuristics. Lou Penrod was also very generous with his time and the data from his clinic that we used to test the generality of the program. We also thank Dan Hennessy, Joe Phillips, John Aronis, and Foster Provost for their numerous discussions and suggestions. This work was funded in part by grants from the National Library of Medicine (LM006625), the National Center for Research Resources (RR14477), and the National Science Foundation (9412549).

Notes

1. Because one of the principals involved in DENDRAL was Joshua Lederberg, a pioneer in molecular genetics, we frequently asked him whether there were questions in biology he considered suitable for the AI methods we had in hand. The MOLGEN project at Stanford University, begun in the late 1970s, grew from these discussions and was the first AI project in molecular biology, specifically for planning gene- cloning experiments (Martin et al. 1977; Stefik 1981a, 1981b). Considerable other work in biology followed from this context.

2. HAMB is presented in detail in Livingston (2001) and Livingston, Rosenberg, and Buchanan (2004, 2001a, 2001b). We review some of this information here to provide sufficient detail for readers.

3. All supporting reasons have strengths greater than zero.

4. We plan to automate this task but have not yet.

5. A paired t-test is used to test for the statistical differences in the plots presented in this article; significance level for rejecting H0 is $p \leq 0.05$. The p-value of the difference between the entire plots for RANDOM-HAMB and HAMB is 0.158.

6. Moreover, to conserve time, the complexity of many of the tasks was reduced: for example, HAMB uses an iterative cross-validation process to select a bias to be given to the rule-induction program. For this study, only a few bias points were examined using twofold cross-validation. In addition, the size of the database used during this study

was reduced to 500 examples. When more complex tasks are added, or the size of the database is increased, we expect that the improvement in HAMB's performance over that of the other versions will increase.

7. Rich Ambrosino first implemented the Fisher's exact test to measure *p*-values associated with rules in RL. To make the computation tractable, we approximated the calculations of factorials using the algorithms in Press et al. (1988).

References

Buchanan, B. G. 1966. Logics of Scientific Discovery. Ph.D. dissertation, Department of Philosophy, Michigan State University.

Buchanan, B. G., and Feigenbaum, E. A. 1978. DENDRAL and META-DENDRAL: Their Applications Dimension. *Artificial Intelligence* 11(1–2): 5–24.

Buchanan, B. G.; Sutherland, G. L.; and Feigenbaum, E. A. 1969. Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry. *Machine Intelligence, Volume 4,* eds. B. Meltzer and D. Michie, 209–254. Edinburgh, U.K.: Edinburgh University Press.

Engels, R. 1996. Planning Tasks for Knowledge Discovery in Databases: Performing Task-Oriented User Guidance. Paper presented at the Second International Conference on Knowledge Discovery and Data Mining, August 2–4, Portland, Oregon.

Gililand, G. L.; Tung, M.; and Ladner, J. 1996. The Biological Macromolecule Crystallization Database and Protein Crystal Growth Archive. *Journal of Research of the National Institute of Standards and Technology* 101(3): 309–320.

Haase, K. W. 1990. Exploration and Invention in Discovery. Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

Harris, N. L.; Hunter, L.; and States, D. J. 1992. Mega-Classification: Discovering Motifs in Massive Datastreams. In Proceedings of the Tenth National Conference on Artificial Intelligence, 837–842. Menlo Park, Calif.: American Association for Artificial Intelligence.

Hennessy, D.; Buchanan, B.; Subramanian, D.; Wilkosz, P.; and Rosenberg, J. 2000. Statistical Methods for the Objective Design of Screening Procedures for Macromolecule Crystallization. *Acta Crystallographica* Section D: 817–827.

Klosgen, W. 1996. EXPLORA: A Multipattern and Multistrategy Discovery Assistant. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 249–271. Menlo Park, Calif.: AAAI Press.

Lenat, D. 1983. EURISKO: A Program That Learns New Heuristics and Domain Concepts. The Nature of Heuristics III: Program Design and Results. *Artificial Intelligence* 21(1–2): 61–218.

Lenat, D. 1982. AM: Discovery in Mathematics as Heuristic Search. In *Knowledge-Based Systems in Artificial Intelligence,* eds. R. Davis and D. Lenat, 3–225. New York: McGraw-Hill.

Livingston, G. R. 2001. A Framework for Autonomous Knowledge Discovery from Databases. Ph.D. dissertation, Department of Computer Science, University of Pittsburgh.

Livingston, G. R.; Rosenberg, J. M.; and Buchanan, B. G. 2004. An Agenda- and Justification-Based Framework for Discovery Systems. *Journal of Knowledge and Information Systems*. Forthcoming.

Livingston, G. R.; Rosenberg, J. M.; and Buchanan, B. G. 2001a. Closing the Loop: An Agenda- and Justification-Based Framework for Selecting the Next Discovery Task to Perform. In Proceedings of the 2001 IEEE International Conference on Data Mining, 385–392. Washington, D.C.: IEEE Computer Society Press.

Livingston, G. R.; Rosenberg, J. M.; and Buchanan, B. G. 2000b. Closing the Loop: Heuristics for Autonomous Discovery. In Proceedings of the 2001 IEEE International Conference on Data Mining, 393–400. Washington, D.C.: IEEE Computer Society Press.

Martin, N.; Friedland, P. E.; King, J.; Stefik, M. 1977. Knowledge Base Management for Experiment Planning in Molecular Genetics. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 1977), 882–887. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Mitchell, T. 1997. *Machine Learning*. New York: McGraw-Hill.

Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; and Vetterling, W. T. 1988. Numerical Recipes in C. In *The Art of Scientific Computing*. Cambridge, U.K.: Cambridge University Press.

Provost, F. J., and Buchanan, B. G. 1995. Inductive Policy: The Pragmatics of Bias Selection. *Machine Learning* 20(1): 35–61.

Qian, N., and Sejnowski, T. J. 1988. Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. *Journal of Molecular Biology* 202(4): 865–884.

Quinlan, J. R. 1993. C4.5: *Programs for Machine Learning*. San Francisco, Calif.: Morgan Kaufmann.

Shen, W.-M. 1990. Functional Transformations in AI Discovery Systems. *Artificial Intelligence* 41(3): 257–272.

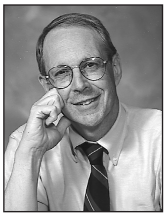
Sims, M. H. 1987. Empirical and Analytic Discovery in IL. In *Proceedings of the Fourth International Workshop on Machine Learning,* 274–280. San Francisco, Calif.: Morgan Kaufmann.

Sokal, R. R., and Rohlf, F. J. 1969. *Biometry: The Principles and Practice of Statistics in Biological Research*. San Francisco, Calif.: W. H. Freeman.

Stefik, M. 1981a. Planning and Meta-Planning (MOLGEN Part 2). *Artificial Intelligence* 16(2): 141–169.

Stefik, M. 1981b. Planning with Constraints (MOLGEN Part 1). *Artificial Intelligence* 16(2): 111–140.

Zytkow, J. M. 1993. Cognitive Autonomy in Machine Discovery. *Machine Learning* 12(1–3): 7–16.



Bruce Buchanan received a B.A. in mathematics from Ohio Wesleyan University (1961) and M.S. and Ph.D. degrees in philosophy from Michigan State University (1966). He is university professor emeritus at the University of Pittsburgh, where he has joint appointments with the Departments of Computer Science, Philosophy, and Medicine and the Intelligent Systems Program. He is a fellow of the American Association for Artificial Intelligence (AAAI), a fellow of the American College of Medical Informatics, and a member of the National Academy of Science Institute of Medicine. He has served on the editorial boards of several journals and has served as secretary-treasurer of the AAAI (1986–1992) and as president (1999–2001). His e-mail address is buchanan@cs.pitt.edu.

Gary Livingston received B.S. degrees in computer science and math from Arkansas Tech University (1988) and M.S. (1994) and Ph.D. (2001) degrees in computer science from the University of Pittsburgh. He recently joined the faculty of the Department of Computer Science at the University of Massachusetts at Lowell as an assistant professor. Livingston's current research is focused on the application of machine learning techniques to infer gene influence networks from gene-expression microarray data; the learning and representation of cellular processes; the capture, representation, and use of background knowledge in machine learning; and the development of nearly autonomous discovery systems that might be guided at a high level. His e-mail address is gary@cs.uml.edu.