CHAPTER

7

Planning to Learn About Protein Structure

Lawrence Hunter

1. Introduction

Discovery requires concerted effort. Human scientists actively seek out information that bears on questions they have decided to pursue. They design experiments, explore the implications of the knowledge they have, refine their questions and test alternative ideas. Although many discoveries are the result of unexpected observations, these surprises take place in the context of an explicit pursuit of knowledge.

Viewing scientific discovery as a kind of motivated action raises some basic issues common to goal-directed behavior generally: Where do desires (to know) come from? What are the actions that can be taken (to discover)? What are the resources those actions consume, and how are they allocated? How are decisions about selecting and combining actions made? The goal of this chapter is to describe a set of related systems for automated discovery in molecular biology, sketching a framework of a cognitive theory of discovery processes.

Automated process models of cognitive phenomena serve two functions. One is fundamentally scientific: such models provide a vocabulary for expressing theories of mental functioning and a framework for testing and comparing theories. The other role is a kind of engineering: these models are artifacts that, to the degree they are successful models, accomplish useful tasks, and which can extend human abilities. These functions are interrelated. The main scientific claim of most AI models of cognition is that a given model is *sufficient* to account for some complex cognitive phenomenon. Supporting a claim of sufficiency for a model of discovery involves writing a program that actually makes at least moderately significant discoveries.

Biologists hoping for useful tools from machine learning techniques can read this chapter as a description of some approaches to applying machine learning tools to biological problems, and as a promise for the eventual creation of an integrated framework for increased automaticity and coordination in the application of such tools. However, the main thrust of the chapter is to use the complexity and challenges inherent in the domain of molecular biology to argue for a new level of theorizing in machine learning, one that addresses issues such as the design of representations, the integration and coordination of multiple inference techniques, and the origin and transformation of specific desires for knowledge. I present arguments for the approach, and some examples, although this work does not offer significant empirical evaluation of the approach, nor a formal statement of its characteristics. This chapter is a preliminary exploration of a new set of problems for machine learning and discovery theories: expanding the scope of these theories to include the steps before and after data-driven inductive inference.

The chapter is divided into three sections: The first section outlines some theoretical concerns about existing approaches to automated discovery systems, and proposes a new kind of problem for machine learning research. The idea is to expand the purview of discovery systems to include the problems of data selection and representation, the automated selection and combination of inference methods, and the evaluation of alternative approaches. The second section describes in some detail an example of a partially automated scientific discovery process directed at the prediction of protein structure. This example includes the generation of appropriate representations and the integration of multiple inference methods. The process is analyzed to identify the kinds of decisions that scientists have to make before and after the inductive step itself, and to try to illuminate the context in which these decisions are made. In conclusion, the challenges inherent in developing testable theories that address these issues are considered.

2. Discovery in People and Machines

AI theories of scientific discovery have been around for more than a decade ([Lenat, 1979] was arguably the first such theory). Although the role of computation in science has grown enormously in the last ten years, AI theories of discovery have, as yet, played at best a minor part in this expansion. Scientific visualization, physical simulation systems and automated statistical analysis are now integral parts of scientific research, but as yet there isn't much AI.

There are a few examples of AI in molecular biology computing, particularly neural networks; for example, the cover of the 24 August 1990 issue of *Science* is a neural net prediction of the secondary structure of the HIV-1 Principal Neutralizing Determinant protein. However, to my knowledge, most of the applications of AI to scientific discovery have focused on *recapitulation,* or duplicating the historical record of a scientific insight [Shrager & Langley, 1990a]. Because the systems that have been so successful at recapitulation have yet to make any novel discoveries of their own, and because the computational methods embodied in these programs have not been adopted by the scientific community, there is reason to doubt at least the sufficiency of the theories underlying those systems to explain scientific thinking. Admittedly this is a high standard to apply, and there have been clear contributions of the previous work to understanding some of the subproblems of discovery. Nevertheless, demonstrations of the sufficiency of the proposed computational methods to accomplish useful scientific tasks are so far lacking.

What might be missing from the existing AI approaches to discovery? An indication can be found in the overview chapter from the leading collection on the topic, *Computational Models of Scientific Discovery and Theory Formation* [Shrager & Langley, 1990a]. In that overview, Shrager and Langley, two of the founders of the field, list the knowledge structures and processes addressed in their extensive survey of AI models of scientific discovery. The knowledge structures are: observations, taxonomies, laws, theories, background knowledge, models, explanations, predictions and anomalies (they also mention hypotheses, explorations, instruments and representations, but claim that the former set provides a sufficient basis for an account of scientific behavior). The processes found in AI models are: observation, taxonomy formation and revision, inductive law formation and revision, theory formation and revision, deductive law formation, explanation, prediction, experimental design, manipulation and evaluation (comparing a prediction with observations). They also note that assimilating a new theory into one's background knowledge, revising an entire theoretical framework, model formation and revision, and various activities related to the social and bodily embedded aspects of scientific activity are important, although not yet addressed by AI theories.

I found it striking that there is not a single reference to the interests or goals of the scientist in the entire survey. There is likewise no mention of a characterization of available inferential and data gathering abilities—no self-model of the scientist. As I will suggest in more detail below, discovery requires making decisions about how to pursue specific goals for knowledge, using a characterized collection of data-gathering and analytical tools under significant resource constraints. Explicit representations of desired knowledge, and models of the available methods for gathering and analyzing information are crucial components of this process. Some recent work in machine learning has raised this issue in other contexts, e.g., [Cox & Ram, 1992; desJardins, 1992; Hunter, 1989b; Hunter, 1990b; Ram & Hunter, 1992] and there is related psychological work, e.g., [Weinert, 1987] .

Existing AI theories of discovery are, almost universally, cast as methods for searching through a space of possible hypotheses for the point that somehow best fit the available data (e.g., [Langley, Simon, Bradshaw, & Zytkow, 1987; Shrager & Langley, 1990b] , although compare [Tweney, 1990]). The alternative presented here casts learning and discovery as planning processes, working from a knowledge goal to a selection of actions to take to achieve that goal, to the execution (and perhaps reaction or replanning) of those data gathering and inferential actions, ultimately resulting in the satisfaction of the goals for knowledge. Since planning is well known to be just another intractable search problem (in this case through the space of possible actions, rather than possible hypotheses), it is not immediately clear what the advantage of trading one intractable search space for another might be. The difference is in what these metaphors suggest about what the important research problems are in discovery. The hypothesis space view emphasizes the importance of evaluating theories in light of all the available data, and in revising theories given a new set of observations. The planning view emphasizes the importance of understanding how research questions are generated, how it is possible to characterize *a priori* or incrementally what information is likely to be relevant in addressing a question (and is therefore worth gathering or drawing inferences from) and how to select and combine inference methods to best address a particular question. Although the questions raised by the hypothesis space metaphor are clearly important, I believe that these other issues are also important, and currently underexplored.[1]

2.1 Selecting Data

It is perhaps obvious that scientists have particular interests in mind as they do their work. They do not simply examine all the information perceptually accessible to them and try to reach the best explanation of it. In fact, a large portion of scientific labor is devoted to acquiring information that is difficult to perceive, precisely because it is believed to be relevant to some question of interest. Making decisions about what data might be worth gathering is a fundamental component of the discovery process. Because there are limits on the kind and amount of information that can be gathered, and because there are limits on the amount of data that can be *considered* by any realistic inference process, the process of directing attention to potentially relevant data is a central one in scientific discovery.

Current machine learning (and automated discovery) approaches tend to use all the data that is available for inference. At first, this does not seem unreasonable; after all, why should a program ignore information that might be useful? For any particular learning or discovery task, it seems desirable to select as broad an array of potentially relevant phenomena as possible. It is within the ability of current methods to discover that an aspect of a training set is irrelevant, but hard to infer that something is missing, and even harder to infer what that missing element might be. However, this view makes an important tacit assumption, that it is possible to identify (and ignore) all the irrelevant aspects of the all information that may be available to a scientific discovery system.

As demonstrated in [Almuallim & Dietterich, 1991] , no existing machine learning system learns well in the presence of many irrelevant features. Almuallim and Dietterich present a system that exhibits somewhat better performance at this task, but the limitation is still significant. When compared to the amount of information potentially available to a discovery program, the problem of selecting relevant information is quite clear. In the field of molecular biology alone, the number and complexity of the datasets currently available over the Internet is staggering. The amount of information available to a human scientist in his or her local library is larger by many orders of magnitude, and the amount of information potentially gatherable with modern laboratory instrumentation is even larger. The vastness of “all available data” demands that decisions about what aspects of the universe are worth considering must be made in order to solve any significant scientific problem. The question of how a program can decide what data might be relevant to a particular question is a central concern to both developing a cognitive theory of how scientists think and to engineering a discovery assistant that is capable of navigating the information resources of the Internet effectively.

The requirement that potentially relevant aspects of a problem be selected before learning can occur is currently addressed by current automated discovery research in two ways. Most obviously, researchers select a dataset to which their algorithm will be applied. Existing datasets are generally winnowed down to select samples with various desirable characteristics. Some data is ignored as irrelevant, and others are transformed so that the distributions of values are better matched to the characteristics of the learning system. Possible transformations include discretizing, scaling, and combining multiple fields. Second, researchers are making decisions about relevancy when they craft the representations that their programs use. It is often the case that even radically different machine learning methods (e.g., decision tree induction and neural networks) offer similar levels of performance on a given induction problem. The key issue in the successful application of many of these methods turns out to be the selection of a suitable representation. However, the process of designing representations is generally taken to be outside the computational theory proposed (and evaluated) by AI discovery research.

One of the goals of this chapter is to bring the question of deciding upon the structure and content of input representations to a machine learning system into the realm of the theory itself. This work differs from related efforts in constructive induction (e.g., [Rendell & Seshu, 1990]) in that it addresses the entire process, from selecting and segmenting data sources to representational transformations both before and during learning. The task, given a specification of desired knowledge, is to make well-founded decisions that address the following questions:

- *What kinds of data might be relevant to acquiring the desired knowledge?*
- *What sources of potentially useful data exist?*
- *Given the available sources of potentially useful data, how can a dataset that best matches the relevancy specification be retrieved?*
- *How should retrieved data be sampled or segregated? (e.g., for cross-validation)*
- *How should retrieved data be transformed? (e.g., to match a particular inference method)*

How can a discovery system make these decisions about what data might be worth considering and how to find and transform it to address a given problem? A decision-theoretic approach would suggest defining utility and cost functions. The utility of considering a set of data might be estimated based on a characterization of the desired outcome. This is a difficult problem. The PAGODA system [desJardins, 1992] uses a decision theoretic approach to select which of several sensory modalities is worth learning about next, based on an estimate of expected utility of learning. However, the assumptions that make this estimate computationally tractable are extremely stringent, requiring, among other things, that the utilities of learning about the various modalities do not interact, and that the effects of learning in each modality can be modeled accurately. The example described in section 3, below, applies a computationally simpler method to a making a decision that does not fit PAGODA's assumptions.

A model of the costs of acquiring and using data is also necessary. It is possible to make estimates of the cost of obtaining and using data, e.g., as [Horvitz, Cooper, & Heckerman, 1989] does in evaluating the tradeoff between gathering more data and taking action in certain medical contexts, or as [Holder, 1991] does in estimating the amount of inference necessary for maximum predictive accuracy of certain machine learning systems. Other data-related costs can be estimated by an analysis of how the performance of a particular inferential method depends on the characteristics of its input, or by the network costs, time, disk space or other factors involved in acquiring and using the data.

No matter how large the machine, or how massively parallel, programs are fundamentally unable to make all inferences from all the potentially usable information in a realistic setting. Sampling methods, incremental experimentation and other methods for exploring very large spaces are applicable to this problem, but the space of possible "features" of the universe mandates some kind of selective attention. A novel set of problems for machine learning and discovery research to explore involves the interrelated issues of how to represent the contents of sources of information, and how to estimate the costs and benefits of using a potential source of information.

2.2 Knowledge Goals

An estimate of the expected utility of a source of information depends on what how that information relates to the goals of the learner. Not all information is equally relevant to all questions. The specific goal(s) for knowledge being pursued by a discovery program are the basis on which judgments about the relevancy (and hence utility) of a given collection of data must ultimately be made. *Knowledge goals* must describe the content of desired knowledge (e.g., Marvin Minsky's home phone number, or a computable method for calculating protein secondary structure from sequence) rather than just its structure (e.g., biases that prefer the induction of short hypotheses). Relevancy is an inherently semantic concept; a relationship between meanings. Programs without an explicit, content-based representation of the knowledge they desire will not, in general, be able to make effective relevancy decisions to focus attention on potentially useful knowledge. To the degree that programs that do not reason about their own goals for knowledge are successful in acquiring desired knowledge, they will either have had their input data prescreened by the researcher or they will include a built-in, inflexible bias that encodes relevancy judgments, or both. These methods will work in some circumstances, and built-in relevancy biases that are effective in particular situations are important contributions to attacking the general problem, but these methods are not alone sufficient for a building a flexible and powerful discovery system.

Programs that represent and draw inferences from their own goals for knowledge have other advantages as well. In addition to being able to make decisions about what external stimuli to focus on, they are also able to use those goals to focus their internal memory and inferential capacities in such a way as to improve their performance. Programs with limitations on processing ability and memory capacity need to allocate inferential resources so as to facilitate the accomplishment of their goals; explicit representation of those goals makes this process more flexible. Such decisions are important for agents with bounded rationality, and have been useful in addressing difficult inferential problems [desJardins, 1992; Hunter, 1989a; Ram, 1989]

A second important aspect of the explicit representation of desires for knowledge is that it makes possible the automatic and dynamic choice of the inference method or methods that are most appropriate for each particular knowledge goal. A recent proof demonstrated that learning algorithms, very broadly defined, can evaluate only a small proportion of the hypotheses compatible with the experiences they have. That is, there is no general learning method, and "different classes of learning problems may call for different algorithms." [Dietterich, 1989] A general (i.e. human-like) learning system will therefore have to make choices about what method(s) to learn or discover in a particular context.

A mechanism for making choices about what to learn and how to learn it is a crucial component of an automated learning system. On what basis can such decisions be made? How well a particular learning method performs on a particular task depends crucially on the characteristics of the concept to be learned [Rendell & Cho, 1990] . Rendell's work shows that the true character of a concept effects how well a particular learning method works. In order to use the relationships between concept character and learning method that Rendell identified to direct the selection of a learning mechanism, the learning system must have some internal characterization of its target concept(s). A knowledge goal is such an internal characterization of a target concept; knowledge goals are the appropriate basis for making decisions about learning methods, data selection, and representation.

Knowledge goals may have another role to play in an integrated learning and discovery system. They may facilitate experience-based improvement of the learning process itself: learning how better to learn. In order for a discovery program to be able to evaluate its own performance, it must compare the actual result of inference with its original knowledge goals. This comparison may identify areas where additional inference would be beneficial. A record of the decisions made and an internal model of the learning and discovery process could be used to identify alternative approaches that could be explored, or to support systematic modifications to decision making within the learning and discovery processes themselves. This potential use of knowledge goals remains unexplored, but is supported by analogy to the use of goals and internal models in other kinds of learning (e.g., [Hunter, 1989a]).

2.3 Problem Transformation

In order to address a desire for knowledge, a discovery system must have both data that bears on the question and inferential abilities that apply to it.

However, there is an interaction between the available data and the requirements of the inferential method. The structure and representation of the data is often a determining factor in the successful application of an inferential technique. Many of the considerations in selecting the data to attend to described above also apply to the selection of which features of that data should be made explicit, and how.

Often this question is complicated by the need to reduce the complexity of the data. In order to learn a complex mapping, an inference system needs many examples. Formal results relate the ability of any learning system to learn a concept to the number of examples it has seen [Valient, 1984]. It is not possible to accurately induce complex concepts from small amounts of data. Most interesting scientific problems face this challenge. This problem can be addressed by simplifying the space of possible concepts considered or increasing the amount of data, or both.

The process of addressing a complex desire for knowledge by transforming it into a more tractable problem that can be addressed with available data is an important aspect of scientific creativity. Individual scientists appear to have quite different approaches to this problem, which may depend on training, experience, the desire to try (or demonstrate) some particular approach, and many other difficult to capture factors. Computational models of creative processes in understanding [Kass, 1990] may be relevant to this addressing this question.

In the specific examples described in the next section, a variety of transformations were applied data to reduce the size of a problem. Generalizations of these approaches are potentially applicable to reducing the complexity of many other induction problems. Six interrelated classes of transformations were applied:

- *Identify invariances* so portions of the space can be collapsed into equivalence classes. A simple example in structural domains is to collapse all translations or rotations of a structure into a single class.
- *Creating approximate equivalence classes*, for example, by clustering the data and ignoring distinctions within clusters.
- *Prune the space* e.g., by focusing on areas with a high density of examples or with more available information.
- *Decrease the resolution* of the distinctions made, for example by discretizing real values or increasing the grain size of a discrete measure.
- *Find correlated attributes* and develop proxy measures that reduce the correlated attributes to a single one
- *Find independent subspaces*, and solve them one at a time

There are many possible ways to operationalize each strategies. The alternative operationalizations and strategies themselves are not mutually exclusive. Each class of reductions can be applied repeatedly and in combination with others, in an order sensitive way. For example, it may not be possible to identify equivalence classes until the problem has been divided into independent subspaces. Once those equivalence classes are identified, it may then be possible to prune the problem spaces by only considering problems that fall into the most common classes.

Different problem reductions lead to quite different results and it appears to be difficult to predict ahead of time which combination will work. There is also a complex interaction between problem reduction method and the selection of a specific inference method, since different inference algorithms place differing restrictions on problem structure and representation. The question of how to select among and apply these abstract problem reduction strategies is an open research problem. One detailed example is given in the next section.

In short, the view of learning as a kind of planning provides an set of novel problems not previously addressed by machine discovery work. Leaving aside the question of how desires for knowledge arise, this framework demands answers to questions about how a desire for knowledge is translated into an executable plan for acquiring that knowledge. How are potentially relevant sources of data identified? How is data screened, transformed, and represented so that desired inferences can be made? How are alternative inferential approaches selected among and combined to best use the available data? How are the conclusions drawn by an inferential method evaluated? The central claim outlined in this section is that explicit, content-based representations of the characteristics of desired knowledge play a role in each of these processes. The next section describes an set of examples illustrating how that might happen.

3. Planning to Learn About Protein Structure

In this section, I will describe a coordinated set of activities in service of the goal of being able to predict protein tertiary structure from sequence, paying special attention to the processes of selecting and representing relevant information. The collection of programs described here is implemented in the INVESTIGATOR framework, developed at the National Library of Medicine [Hunter, 1990a]. Although not all of the decisions described below were made in a meaningful way by a program[2], I do endeavor to provide a theoretical framework that illuminates the choices that must be made and the factors that influence those choices, as a prolegomena to an implemented theory. In some areas, however, even the details of the possible transformations remain unclear. Nevertheless, in order to create sufficient computational theories of scientific discovery (or human learning), these questions will have to be addressed. This section explores these issues in the context of a real problem in molecular biology: predicting protein structure from sequence.

The task of predicting three dimensional structure from amino acid sequence is described in detail in the introductory chapter of this volume. In brief, the genome of an organism specifies the makeup of all of the proteins that constitute that organism. The genes specify a linear sequence of amino acids, which are assembled at the ribosome. Although the proteins are constructed as a linear sequence, they only become chemically active when they have folded up into a particular three-dimensional conformation. The positions of each atom in the protein in three-space is called its structure (or, more specifically, its tertiary structure). Proteins are very large molecules, and the folded shape can hide some regions, expose others, and bring elements of the protein that were at opposite ends of the sequence close together in space. These factors are important in determining what the function of the molecule is in the living system, and how it performs that function. Determining the structure of biomolecules is important in designing drugs, understanding key functions such as development or neuronal signaling, and in practically every area of biology. Technologically, it is now relatively easy to determine the sequence of proteins, but it remains very difficult to determine their structures. It is easy to demonstrate that all the information needed to determine structure must be present in the amino acid sequence alone. It has proved to be quite difficult to find the mapping from sequence to structure.

Much related work in the field takes similar approaches to the ones presented here to learning aspects of this mapping (e.g., Holbrook, Muskal and Kim; Zhang and Waltz; Lathrop, et al, all in this volume). However, the goal of this chapter is to elucidate some of the cognitive processes that go unstated (although not undone) in that work, and bring those processes into the realm of AI discovery research. For example, nearly all researchers applying learning algorithms to the problem of protein structure prediction screen their dataset for homologies, and use a sliding window to segment the problem. These choices make a tremendous difference in the outcome of the work; how are they made? What are the alternatives?

As is often the case in machine discovery work, it is easier to define the space through which a program must search than it is to describe an effective method for traversing that space. The space of possible data-gathering and inferential actions is rather different than the space of possible hypotheses (or formulae) for describing a dataset. The hope of this approach to automated discovery research is that it will be possible to characterize knowledge generating actions on the basis of their expected difficulty or cost, and to develop a set of methods for estimating the distribution of expected outcomes of the application of these actions, given some information about the knowledge desired and the characteristics of the available data. Several related efforts have been made, such as [Holder, 1991] which empirically characterizes the expected performance of a learning algorithm given a partial execution, or [Rendell & Cho, 1990] which makes estimates of the performance of various learning methods based on the true character of the concepts they are trying to learn.

A human scientist attacking a large problem develops a research plan, consisting of many constituent approaches to relevant subproblems. In the example explored below, this research plan is built by the instantiation of an abstractly stated discovery strategy. The first step in this process is to identify the specific knowledge goal. Then, the statement of the problem is used to select one of three high level discovery strategies. Once a strategy has been selected, the information in the representation of the strategy is used to determine what data is necessary. Knowledge of various data sources is used to select a source, and then to identify and extract an appropriate dataset from that source. The next step is to transform the available dataset to meet the requirements of the strategy. This transformation is a complex process, and, in this case, is the area where scientific creativity is most apparent. Then a particular inference method is selected. This selection may place additional requirements on the dataset. A representation is selected on the basis of the data and the inference method, and the dataset is transformed into that representation. Parameters of the inference method must be set, or the space of possible parameterizations explored, and the inferences made. Finally, the outcome of the inference process must be evaluated. As the example unfolds, several more general points about the process become apparent as well.

3.1 Characterizing the Desired Knowledge

The protein structure prediction problem is to find a mapping from a linear sequence of amino acids to a set of three dimensional coordinates for all the atoms in each amino acid. Typical proteins contain hundreds of amino acids, and thousands of atoms. Large proteins (e.g., Apolipoprotein B-100) are composed of more than 4500 amino acids. Spatial resolution of 2Å is about the level of accuracy of the training data available from crystal structures, and a large globular protein (like Apolipoprotein B) may be 150Å along its longest dimension. The largest version of the problem therefore involves a mapping from any of 20^{4500} (~10^{5850}) strings to the positions of about 60,000 atoms in a lattice of 75^3 points (421,875 choose 60,000, or over $10^{46,000}$ possibilities). The number of possible mappings is proportional to the product of these two immense numbers! Fortunately, the problem is really much smaller than this. A vanishingly small portion of the large number of possible proteins is actually observed in nature. Most proteins are much smaller than 4500 amino acids and 150Å. A solution limited to proteins of 450 amino acids or less, using only 3 atoms per amino acid and 3Å resolution on a 90Å lattice would be a breakthrough. However, even this dramatically smaller problem has so many possible mappings to consider (merely 10^{585} strings and ~10^{1700} possible structures!) that it is extremely unlikely to be discovered by a search through the space of possible mappings described in this way. However intractable, this characterization of the problem space is useful for reasoning about the data and possible representations. A significant aspect of the discovery process involves transforming this space to a more tractable approximation of it that retains its essential character.

Given the large number of possible solutions, why is this problem thought to be solvable at all? Nature does it all the time. Denatured (i.e. unfolded) proteins will fold into their native conformation (i.e. the shape they take in living systems) in aqueous environments of suitable temperature, pressure and pH [Anfinsen, 1973]. Cells solve this problem millions of times a minute. The mechanism that determines how proteins fold in the cell can be explained in the same terms as any other physical phenomenon. The forces acting on atoms in the protein can be accurately described by quantum mechanics, and the molecule's folded state minimizes its free energy in its environment. In a system with an accurate causal model such as this, it may be possible to computationally simulate the process, and achieve the goal. Unfortunately, finding the minimum energy conformation of even a much simpler system from an arbitrary starting state using quantum mechanics (called an *ab initio*—from first principles—calculation) is a computationally intractable problem. The use of approximations and other methods to increase the tractability of simulation is discussed below.

Despite the obvious insolubility of the problem in these terms, it is still important to be specific about what the general problem entails. This mapping is the knowledge goal. The subproblem decompositions and approximations we will make along the way are methods of attacking this original, insoluble problem. In order to select among and evaluate these simplifications, there must be a reference to which they can be compared. The full statement of the problem, no matter how computationally intractable, provides a baseline from which simplifying assumptions can be made, and by which the results can be evaluated.

3.2 The Knowledge Acquisition Strategy

Knowledge goals are addressed by taking actions that change knowledge state, that is, by making inferences. Unfortunately, means-ends analysis applied to the space of knowledge states using inferences as operators is unlikely to work. However, inference steps can be assembled into plans to acquire knowledge, and skeletons of these plans can form general templates for assembling novel plans without the need for additional reasoning from first principles. These skeletal plans for acquiring knowledge are termed *knowledge acquisition strategies* [Hunter, 1989a; Hunter, 1989b].

Discovering a mapping from one complex, high dimensional space to another is a common problem confronting intelligent agents, and there are several distinct general approaches for addressing it. These approaches can be divided into three broad categories:

- *Simulation* using an effective causal model of the phenomena that underlie the transformation, reasoning about the transformation analytically.
- *Induction* of an empirical mapping between the input and output spaces based on a sample of I/O pairs.
- *Case-based* methods also work from a sample of I/O pairs, but instead of trying to induce a mapping between them, case-based methods make predictions about an input by finding a stored example with a similar input, and using the matching stored output as the basis for the prediction.

An autonomous discovery system would decide among these (and perhaps other) alternative strategies when trying to discover such a mapping. Ideally, each of these broad classes of methods would be characterized by a function that would estimate the expected cost and utility of each method given the characteristics of the transformation space, the available data or examples, and the amount and usefulness of any background knowledge or bias. Unfortunately, there is as yet no known method of making such a calculation in a reasonable period of time. All three of these methods might be successfully applied to the protein structure prediction problem, and human scientists are pursuing research that can be classified into each category. These scientists make their decisions about which strategy to pursue based on a variety of factors, including personal or social ones such as the kind of academic training they have had, how an available resource might be used (e.g., a private database or parallel computer) or where they perceive the competition is the least strong. It is, however, possible for a program to embody heuristic, qualitative characterizations of the problem situations best suited to each of these classes of methods, and make a selection based on a characterization of the desired knowledge.

For each possible strategy, there are costs, in terms of how much computational effort the strategy is likely to require, and expected benefits, usually cast in terms of how likely the strategy is to succeed. It may be possible to easily eliminate a strategy on the basis of its intractability, or to easily select one on the basis of its probability of success. The first step in the selection process is to eliminate strategies that are intractable.

It appears to be possible to directly assess the computational demands of a simulation strategy for protein structure prediction. In simulation, there is a always a computational model of the causal factors underlying the desired transformation. The expected running time and other resource consumption of the simulation of a model can be assessed either analytically or empirically, generating an estimate the resources required to execute a model given a particular problem characterization. The simulation of the movement of a molecule the size of a protein can take hundreds of hours of supercomputer time to simulate nanoseconds of folding, even using heuristic energy functions rather than *ab initio* quantum calculations [Karplus & Petsko, 1990] The entire process of protein folding in the cell can take several seconds, indicating that a simulation of folding a single protein would take more than 30 years.

However, it is worth noting here that such a conclusion, based on simple extrapolation, can easily be incorrect. Variations on the parameters of the simulation (e.g., lattice size or time step), the underlying model, the implementation (e.g., parallelism or clever optimization techniques) or other factors offer potential speed-ups or tradeoffs that might some form of simulation appropriate for the problem at hand. The difficulty in making this decision is reflected in the fact that human scientists working on this problem are currently pursuing all three strategies, and there is a great deal of research in variations on the simulation strategy (e.g., [Skolnick & Kolinski, 1990]).

The difficulty in making correct high level strategic decisions for scientific discovery is a quite general problem. Making discoveries about phenomena of significance often requires taking a method that appeared intractable and finding a way to apply it. The mere fact that a method appears intractable on one analysis does not mean that it is not worth inferential effort to refine or recast the method. People seem to be able to develop intuitions about what approaches are genuinely intractable, and which are merely difficult open problems; of course, these intuitions are not always correct.

The selection of a knowledge acquisition strategy for an unsolved problem reflects the learner's assessment of its own inferential abilities, as well as an assessment of the problem characteristics. It is hard to accurately assess the cost of instantiating and executing a complex strategy, or its likelihood of its success, especially since the learner needs to assess alternative strategies without wasting inferential resources on evaluating strategies that will not be used. This is an issue, since as [Collins, 1987] pointed out, there can be significant inferential work to be done in just figuring out how to apply a potential planning strategy to the problem at hand. The estimates of difficulty and likelihood of success that people use to select among strategies may well be based on their observations of how well other people have done using those strategies, or own their own history, rather than on a deep analysis of how a particular strategy will apply to a current problem of interest.

Returning to the specific problem at hand, the alternative to the analytical approach of simulation are the two empirical approaches, induction and case-based reasoning. Both methods are potentially achievable within reasonable resource limits, so the question becomes which is more likely to succeed in accomplishing the goal? Until success is achieved, there is no direct way to make this decision. Both methods have significant potential, but no clear solution. Because the strategy and set of strategy instantiation and transformation methods are better developed for the inductive methods (including neural networks) in the current implementation of INVESTIGATOR than CBR methods are, the choice to use them can be made on the basis of the internal abilities of the learner. This decision criterion must be secondary to an assessment of how likely a strategy is to succeed, since otherwise a less well developed strategy will never be used, even if it is assessed as more likely to succeed on a given problem. In the general case, it is also worth exploring a less well developed strategy periodically if there are potential opportunities to improve it (or learn more about its applicability conditions) through experience. The question of how often to try a less well developed strategy is related to the more general problem of deciding when to gather more knowledge [Berry & Fristedt, 1985] .

After selecting a strategy, a learner must instantiate it, mapping the abstract components of the plan to the specifics of the current goal. The strategies describe the steps of an abstract plan and constraints on the concepts that can be used to fill variablized slots in the plan.

3.3 Selecting Relevant Data

The first step in most knowledge acquisition strategies is to find relevant data from which inferences can be drawn. Few machine learning or discovery programs address this issue. Almost universally, these programs use all of the data that is available to them. One of the design goals in building INVESTIGATOR is that it have potential access to a great deal of information by accessing remote databases over the Internet. The computational (and sometimes financial) expense of accessing this data is non-trivial, so INVESTIGATOR must make decisions about what data it will use. These decisions are made on the basis of (1) the content-specific knowledge-acquisition goals that drive the entire process, (2) the selection of a knowledge strategy, which specifies the kind of information need in order to make the desired inferences, and (3) characterizations of the knowledge sources that are available to the system.

In the case at hand, the inductive learning strategy requires a large number of pairs of problem statements and solutions. When applied to the current knowledge goal, that requirement becomes a need for protein sequences and the structures associated with them. INVESTIGATOR's internal representations of its available data sources show only one source of protein structures, and that data source also contains the related protein sequences: the Brookhaven Protein Data Bank (PDB) [Arbola, Bernstein, Bryant, Koetzle, & Weng, 1987] . Although in this case, the desired information can be found in a single location, this is not generally the case. Some knowledge goals may require using data from several different sources. Earlier work with IN-

VESTIGATOR explored using multiple sources of data to address a particular knowledge goal [Hunter, 1990a]. The representation of PDB contains information about where to find the database, how large it is and procedures to parse its entries. The general information in INVESTIGATOR about the database specifies that each structure in PDB contains three dimensional location data for each atom in the molecule; most structures have well over than 1000 atoms; that database entries also generally include information about the bonds between the atoms, other atoms in the structure (such as cofactors, water molecules, or substrates), data about the certainty of the each atomic position, and that there are currently about 900 structures in PDB. Generating representations of available data is currently done by hand, although information about the size of the databases is updated automatically whenever a database is accessed. The movement towards the adoption of the ASN.1 data description standard for biological databases raises the possibility of the automatic generation of parsers as well [Karp, 1991].

A selecting a source of data is only the first step. The next step in instantiating the induction strategy is to select the particular data items that it will use, and then select an appropriate representation. There are several reasons why an inductive learning strategy may want to use only a subset of available data. In order to make estimates of the confidence in a prediction method, a learner must put aside a test set that is not used in the training procedure, e.g., for cross validation. This test set must not be used in any aspect of training. Another reason to use only a subset of the available data is the possibility of errors in the training collection. Many datasets are annotated in some way with characterizations of the certainty or believability of the data. Since many inductive methods are sensitive to noise, it may be appropriate to remove uncertain items from the training data, assuming that they can be identified. A more complex consideration is matching the distribution of the data items in the training set with the expected distribution of similar items the universe. Information about the true distribution in the world is rarely available, but some partial characterizations can be used to select a subset of the training data that is likely to be closer to the true distribution than is the entire dataset.

These 900 structures in PDB include several that are merely theoretical predictions of structure (not empirically derived) and several of very poor resolution. These structures can be easily identified and removed from consideration. PDB also contains many variant structures of a given protein; e.g., bound to inhibitors. These variants are given easily identifiable names and it is possible to select only one representative from each set of related structures. Removing all of these redundant structures reduces the total set to 324 distinct, empirical structures.

The proteins with known structures are not a random sample of proteins; the selection process is biased in many ways, some of which are likely to be biochemically significant. One source of bias is that the proteins in the database are those that are interesting to biologists, and that are (relatively) easy to crystallize, and therefore obtain structures from. It is not clear if any correction can be made for this source of bias.

Another bias results from the fact that once a protein's structure has been determined, scientists become interested exploring the structures of similar proteins for comparison. Technical problems that were solved in the creation of one structure may generalize best to proteins of similar structure, increasing the incentive to investigate similar proteins. These are reasons that entries in PDB may have sequences that are much more similar to each other than a randomly selected collection of proteins would be. If present and uncorrected, this bias will have a significant adverse effect on both the inference process and on estimates of its accuracy.

In general, correcting a bias requires a characterization of the true distribution, and a method for resampling a dataset to reflect the true distribution. There are many possible biases that might be present in a sample, and unless a mechanism for drawing an unbiased sample exists, there is no general way to detect them. However, given a specification of a possible source of bias it may be possible to test for it. The knowledge that the excess-similarity bias might exist in PDB is socially derived, but testing and correcting for it can be done automatically.

A source of an unbiased sample of proteins is needed in order to correct for selection biases in the PDB dataset. The bias introduced by the requirement of crystalizability is easy to address, since there are many sources of protein sequences that are not derived from (or related to) crystals, e.g., the protein information resource (PIR) database. However, finding a collection of sequences that is not influenced by the same socio-scientific interestingness considerations is difficult. The sequences that appear in PIR are determined by those that scientist deemed worth expending the effort to acquire. However, there are datasets that exhaustively sample some naturally defined collection of proteins, such as those that appear on a particular chromosome, or are expressed in a particular cell (e.g., [Adams, Dubnick, Kerlavage, Moreno, Kelley, Utterback, et al., 1992]). These datasets are intended to reflect the true distribution of proteins.

The true distribution of sequence similarity can be estimated by using one of these unbiased datasets, or a sample of it. There are very effective computational tools for determining if a pair of proteins have a greater than random similarity (e.g., BLAST, [Altschul, Gish, Miller, Myers, & Lipman, 1990]). Using a (putatitively) unbiased sample, the expected number of hits is roughly 0.0006 per pair of proteins. The same test on PDB yields nearly 0.002 hits per pair, three times the expected number. The collection of proteins in PDB is biased to excessive similarity.

Since the number of similar sequences in a sample the size of PDB under the true distribution would be close to zero, the induction strategy needs to generate a resampling of PDB to identify a set of proteins that are not similar to each other. Although it is necessary to ignore some data for this reason, the chance of successful induction goes up with the size of the training set, so it is desirable to ignore as little as possible. Since a measure of similarity (BLAST) exists, it is possible to generate a maximum size subset of PDB by using the similarity measure to define equivalence classes, and selecting a single representative from each class.

Even the choice of selecting which member of a class ought to be used to represent the class is a nontrivial decision. If it is possible to determine a selection criterion that facilitates successful inference, it should be used. In this case, members were selected for high resolution, since induction is sensitive to noise in the features of the data. The final reduced dataset has 183 structures in it, none of which have any significant sequence homology to any other.

Although the method described above was generated manually in response to the specific demands of this particular problem, it suggests that a more general strategy for addressing biased data. Resampling a dataset to find a maximum sized subset of it that reflects a specified distribution is a well defined problem that recurs often in inductive inference. Likewise, methods of generating estimates of the true distribution of data along some dimension is also a recurring problem. Detecting that unrepresentative biases exist is a much harder problem. In this context, making that inference appears to require knowledge of the way scientists make decisions about what work do to.

3.4 Reducing the Size of the Problem Space

The problem of inducing a mapping from the entire amino acid sequence of a protein to the positions of each of the constituent atoms is intractable. The space of possible inputs and possible outputs is enormous, and the number of examples is quite small. The problem space must be transformed so that the mapping to be learned is smaller, and the number of examples of this mapping is larger. As described in above in section 2.3, this kind of problem transformation is a significant component of scientific creativity. In [Kass, 1990] Kass describes a theory of creative explanation that involves finding an partial match with a prior explanation and making small changes to the structure of the previous example to meet the requirements of the current case. The following section takes an analogous approach to finding a suitable problem transformation.

A method widely used in the inductive protein structure prediction is the translation of atomic positions into secondary structures [Cohen, Presnell, & Cohen, 1990; Holley & Karplus, 1989; Qian & Sejnowski, 1988; Zhang, Mesirov, & Waltz, 1992]. Secondary structure (for these purposes) is an assignment of each amino acid in the protein sequence to one of three classes, based on the hydrogen bonding characteristics of that element in the final structure. This process involves several kinds of transformations. First, a set of approximate equivalence classes (secondary structures) were created; they were devised by Linus Pauling in the 1950's to coarsely describe local aspects of protein structure, long before any 3D atomic structures of proteins were known. The classes were defined based on invariances found in early crystallographic experiments. The application of this set of equivalence classes dramatically decreases the resolution of the description of the structure, and discards a great deal of information about the original structure, making the problem much smaller. This move is an example of the "creating approximate equivalence classes" problem transformation described in section 2.3.

The next step in this approach involves pruning the space of secondary structures. The secondary structure assignments used by modern biochemists involve eight classes, defined on the basis of hydrogen bonds formed in the molecule. The six least common of these classes are combined into a catch-all category called random coil, focusing on the two most common secondary structures, helices and strands. This also reduces the size of the problem, and is an example of the "decrease the resolution of the distinctions made" problem transformation.

The final step in this structure prediction strategy is to segment the problem into predicting the secondary structure assignment of each amino acid separately, based on a local window of sequence neighbors. This move, taking a large problem and segmenting it into many smaller problems, is an example of the "find independent subproblems" transformation.

The composite transformation of the general problem makes the problem computationally tractable for existing induction algorithms. The size of the problem was reduced by thousands of orders of magnitude, to a matter of learning a mapping from a short string of amino acids to one of three classes. However, each of the transformations introduces an assumption which may or may not be justified: namely, that secondary structure is an appropriate definition of equivalence classes of structural segments; that helices and sheets are the important classes of secondary structure; and that predicting the secondary structure class each amino acid based on is sequence neighbors decomposes the overall problem into independent subproblems. The association of underlying assumptions with transformations is useful in both diagnosing failure (should the inference fail) or in directing the exploration of alternative decompositions.

The existing work on protein secondary structure prediction provides one path through the space. The planning framework outlined here identifies the decisions that were made in generating that path, and suggests where variations could be tried. Exploring a space of alternative conceptions of a problem is part of the scientific discovery process.

The variation explored here is the replacement of secondary structure with another equivalence class defined over the structures. Secondary structure divides a three dimensional protein structure in subregions, and then classifies the subregions. In order to find alternatives to secondary structure, the protein structures must be divided into regions and those regions assigned to classes. Finding such a classification is a new knowledge goal, and can be planned for recursively.

The data is given for this problem, so data source and selection issues do not arise. The specification of a desired knowledge identifies the general strategy required as classification. There are, however, several issues that must be addressed in the instantiation of this strategy.

AI classification methods, e.g., [Cheeseman, Kelly, Self, Stutz, Taylor, & Freeman, 1988; Fisher, 1987] require that the examples to be classified be described by a fixed-length vector of feature values, and one with a relatively small number of features. Available inference methods require the transformation of the supplied protein structures to fixed length segments.

Long, variable length sequences can be transformed into a large collection of short, fixed length sequences by segmentation, as in the final transformation step in the secondary structure method described above. The segments can be mutually exclusive (end to end) or overlapping (sliding window). The division of structures in to fixed length feature vectors is complicated by several factors. First, the "size" of a feature vector for a structure segment can be measured in two different ways, which are not proportional to each other. First is the number of amino acids in the structure segment, and second is the number of dimensions required to describe the positions of the atoms in the segment (which is three times the number of atoms). Since different amino acids have different numbers of atoms, a segment of a fixed number of amino acids will have a variable number of atom description dimensions, and a segment with a fixed number of atom description dimensions will have a variable (and non-integral) number of amino acids. This mismatch can be resolved by a problem transformation. The positions of the atoms in an amino acid are highly correlated with each other. Knowing the position of three particular atoms (which biochemists call the "backbone") in amino acid is generally enough to identify the location of the remaining atoms with a high degree of accuracy. The value of the positions of these atoms can be used as a proxy for the positions of all the others. Protein structures can be segmented into fixed length feature vectors containing three numbers representing the positions of each of three atoms for each amino acid in the segment.

Other problem reduction transformations can also be applied. The molecular segments (and the molecules themselves) are rigid bodies. Similarity of rigid bodies is invariant under positional translation and rotation;

that is, if a structure is similar to another, then it will still be similar if one or both of the objects is moved or rotated. This invariance allows for another reduction in the complexity of the classification problem transforming all objects that are identical under rotation or translation to a single class. Adopting a uniform coordinate frame with which to describe the segments accomplishes this. The protein structure fragments can be translated to a uniform coordinate frame by defining the frame relative to the moment of inertia of each fragment. Similar fragments have similar moments of inertia, and will be oriented so that their constituent atoms will have similar absolute positions.

Decisions remain to be about the number of amino acids per segment and whether the segmentation should be mutually exclusive or overlapping. Overlapping segments are a superset of all possible mutually exclusive divisions, and are preferable unless they produce too many examples for the inference method to handle. The size of the segments should be as large as the inference method can handle.

Selection of the classification method itself must also be made. The alternatives available during this work were k-nearest-neighbor clustering, conceptual clustering and Bayesian classification. Bayesian classification is preferable for several reasons. First, unlike much of the conceptual clustering work, it is explicitly suited to clustering real-numbered location data. Unlike k-nearest-neighbor classification, it uses the data to estimate how many classes there are as well as their content. Finally, unlike other methods, it can also generate classifications that have significantly differing within-class variances. This is valuable both because the natural classes may differ in this way, and because variance information is useful in trying to fit new data to the model defined by the classification. Both k-nearest neighbor classification and conceptual clustering have a strong tendency to minimize differences in variance between classes. Bayesian classification therefore appears to be the most appropriate of the clustering methods for this problem.

Once the data have been transformed to match the requirements of the goal and a specific clustering method has been selected, the inference can be done. In this case Autoclass III [Cheeseman, Stutz, Hanson, & Taylor, 1990] was used to do the Bayesian classification. Other details of this clustering process are described in [Hunter & States, 1991].

The final step in most learning strategies is to evaluate the results. In this case, since the goal was to find an alternative to an existing classification of a particular dataset, the results can be evaluated by correlating the original and induced class assignments for each element. The clustering generated a much larger number of classes than traditional secondary structure recognizes, 27 vs. 8. Some of the induced classes appeared to be more fine-grained variations on traditional secondary structure, but others showed very little correlation with secondary structure. The relationship between the induced

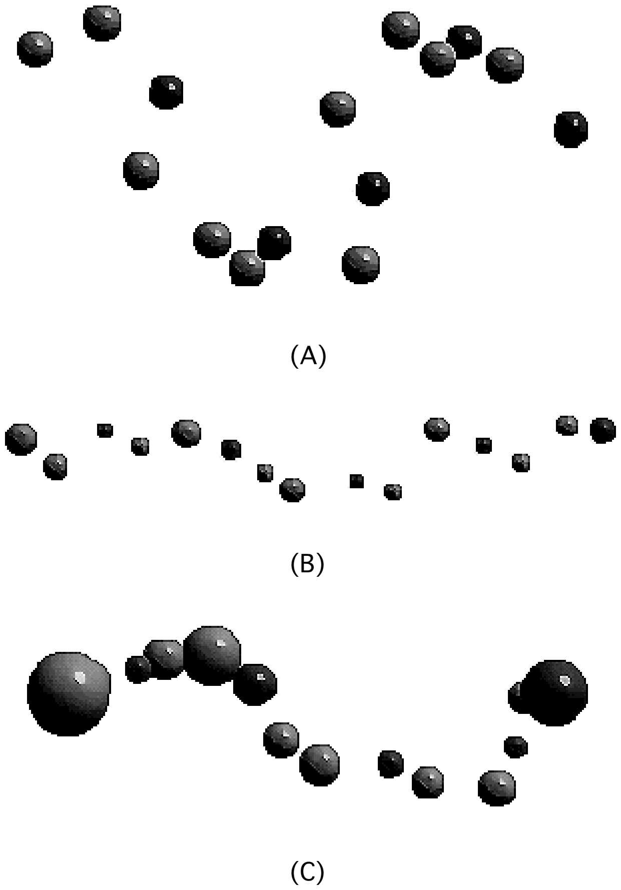

Figure 1. Three examples of the 27 protein structure classes used as an alternative to secondary structure. Two of the classes shown here (A & B) are similar to traditionally defined classes, and the other is not. The spheres depict the positions of the backbone atoms in five consecutive amino acids. The center of each sphere represents the mean position of the atom in the class; the size of the sphere represents the variation in position. Grey spheres are carbon atoms and black ones are nitrogen atoms. (A) is class 1 and is similar to alpha helix. (B) is class 8 and is similar to a beta strand. (C) is class 10, and structures placed in this class have secondary structure assignments in all eight secondary structure classes. Nevertheless, there are more than 800 examples of class 10 in the structure database. Table 1 completely describes the correlation between the induced classification and secondary structure..

class	β-bridge	β-strand	3_{10} helix	α-helix	bend	β-turn	none
1	23	1020	0	0	4	2	192
2	37	378	0	0	3	3	727
3	46	155	1	4	8	11	727
4	18	111	31	30	499	162	160
5	18	627	0	0	1	2	115
6	30	347	0	0	2	2	411
7	33	278	0	0	19	9	409
8	0	0	0	1077	0	1	1
9	0	0	0	1013	0	1	0
10	25	60	12	34	408	53	234
11	13	542	0	0	1	1	118
12	21	62	1	2	261	10	377
13	0	0	45	845	0	52	0
14	2	90	1	1	229	7	262
15	1	13	7	10	214	335	93
16	3	2	88	558	30	155	33
17	0	2	159	463	13	128	20
18	1	1	92	407	78	219	6
19	6	3	63	230	35	215	14
20	0	0	2	638	0	0	0
21	0	0	73	479	0	27	6
22	0	0	71	106	28	170	12
23	3	3	12	146	69	171	1
24	1	1	76	47	35	228	8
25	4	8	17	35	49	129	0
26	1	1	31	90	32	189	0

Table 1. The DSSP secondary structure assignments [Kabsch and Sander, 1983] for the amino acid in the center of the fragments placed into each class. So, for example, of the fragments assigned to class one, 1023 would be assigned to beta strand, and 221 would be assigned to random coil (not beta-strand or alpha-helix). Class ten, one of the most heterogeneous, would be assigned as 60 beta-strands, 34 alpha-helices and 732 random coils.

classification and the traditional one is shown in table 1. Some examples of the class definitions are shown in Figure 1.

This clustering provides an set of output classes which are an alternative to secondary structure classes. Many other alternatives to the original secondary structure plan are possible. The one described above provides one example, and illustrates the many decisions that a researcher must make in the course of addressing a knowledge goal. The actual classification itself is only a part of a much larger process.

The original method reduced the 8 secondary structure classes to three by identifying two classes as primary and combining the others into a single group. Since there is no equivalent identification in the new classification, and since induction methods can learn a mapping to 27 groups, this transformation was skipped. The final step in the original structure prediction strategy was to segment the problem into predicting the class assignment of each amino acid separately, based on a local window of sequence neighbors. This is straightforwardly applied to the new classifications, and no further transformations are necessary.

3.5 Choosing and Applying an Induction Method

The particular induction method used for a particular problem can be selected on the basis of either expected performance or on the cost of executing the method, or some combination. The identification of general methods for selecting appropriate inference methods for a particular problem is an open research problem. In the absence of an analytical method for distinguishing among alternative induction methods, running small scale comparison experiments on random samples of the data can provide justification for selecting one over the other.

The secondary structure prediction methods described above use feedforward neural networks trained with backpropagation to learn the mapping from the window of amino acids to the secondary structure class of the central amino acid in the window. The induction of decision trees based on expected information gain is the major alternative induction method used in machine learning [Quinlan, 1991] . In a set of sample runs on random subsets of the problem data, the accuracy of the two methods were statistically indistinguishable, and the decision tree learner runs several orders of magnitude faster than the neural network training. The neural network methods also require the setting of a free parameter, the number of hidden nodes. This parameter is usually set empirically, based on test performance, which requires a large amount of additional running time. On this basis, the decision tree learner was selected for the large scale induction run.

The fact that the radically different inference techniques of decision tree induction and backpropagation training of neural networks performed at nearly identical levels of accuracy may be somewhat surprising. It appears to be the case that in many circumstances the effectiveness of induction depends not so much on the particular inference method used, but on the data and representation of the data that it is applied to. Although algorithm development (and selection) is clearly an important component of machine learning, inference algorithms are not the sole factor involved in the satisfaction of a goal for knowledge. In particular, the selection and transformation of the data that the algorithms are applied to plays a central role in the outcome.

3.6 Evaluating the Outcome of Learning

The completed execution of a plan does not guarantee the success of the goal for which that plan was intended. The result of the learning strategy above is a decision tree that makes a mapping between amino acid sequence and the classification of protein structures induced before. How well does this tree address the original goal? At best, the resulting decision tree solves a transformed and dramatically simplified version of the original problem. Even if the tree were able to perfectly map from sequence to substructure class, it is not clear how to map from a set of substructure classes back to the positions of the constituent atoms. In particular, the segmentation of the structure lost information about the relationship of the segments which is difficult to reconstruct.

And the mapping from sequence to substructure class was far from perfect. The final application of the decision tree learner to the full dataset yields a decision tree that classifies an independent test set correctly slightly more than 36% of the time. In addition to this estimate of the absolute accuracy of this method, it is also possible to compare it to the accuracy of the secondary structure prediction strategy it was based on. Since the secondary structure method maps to one of three classes, and the variation maps to one of 27 classes, the accuracy statistics cannot be compared directly. The information content of the mappings can be measured, however, showing a slight edge for the variation: 0.9 bits per prediction for predicting one of three classes 63% of the time versus 1.68 bits per prediction for predicting one of 27 classes 36% of the time, under the observed class distributions. (The details of the this analysis, and a more specific description of the results can be found in [Hunter, 1992])

4. Conclusions

The purpose of this chapter was to illuminate the wide range of activities and decisions that go unstated in machine learning work, and to provide a framework for bringing these decisions into the realm of theories of learning and discovery. The process of getting from broadly stated problem to the final application of an inference method to a specific dataset contains many opportunities for machine learning research; this chapter is a preliminary attempt to explore those opportunities.

The exploration attempted to identify the decisions that had to be made in the pursuit of a particular strategy for making a discovery about protein structure prediction. Each decision point embodies alternative paths that might be taken towards the overall knowledge goal. The path actually taken above tried a variation on the previously described secondary structure prediction approach, generating an alternative classification of protein substructures. This alternative path produced only a minor gain in overall performance. However, the claim of this chapter is not that the particular alternative pursued solved the overall problem, but that it is one of many alternative plans for achieving the goal.

Machine discovery has generally been described as the search through a space of hypotheses for one that best fits a given collection of data. The task of this chapter was to make the collection of data seem less "given." Cast as a problem of selecting a potentially effective course of action in the service of an explicitly stated goal for knowledge, the question of what data to use (and in what form) becomes a central concern, not always part of the statement of the problem.

The many open questions and the plethora of on-line data, much of it symbolic, seems to make molecular biology an ideal domain for the testing of machine discovery tools. However, molecular biology offers too much of a good thing; the amount of data available is far too large for most existing machine learning methods, and is growing exponentially. The challenge posed by this domain to the machine learning and discovery community should now be clear. The task of making generalizable inferences about mappings between datasets, given a set of training examples, is not all there is to learning and discovery. A general theory of learning and discovery must also be able to figure out what mappings might be worth learning about, and what data might be relevant to learning them.

Notes

1. It is also worth noting that the metaphors used in science are not incidental to the research enterprise. They provide scaffolding for arguments, color the language used and guide inquiry (see, e.g., [Bloor, 1977; Hesse, 1966]).

2. It is always possible to write a computer program to make a particular choice. This decisionmaking is only meaningful if the program had alternative choices and a theoretically justifiable mechanism for making the choice. Arbitrary selection with backtracking in the case of errors, for example, is not an adequate mechanism for making complex choices.

References

Adams, M. D., Dubnick, M., Kerlavage, A. R., Moreno, R., Kelley, J. M., Utterback, T. R., Nagle, J. W., Fields, C., & Venter, J. C. (1992). Sequence Identification of 2,375 Brain Genes. *Nature*, 355(6361), 632-4.

Almuallim, H., & Dietterich, T. (1991). Learning with Many Irrelevant Features. In *Pro-*

ceedings of Ninth National Conference on Artificial Intelligence, vol. 2 (pp. 547-552). Anahiem, CA: AAAI Press.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). A Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215, 403-310.

Anfinsen, C. B. (1973). Principles that Govern the Folding of Protein Chains. *Science*, 181, 223-230.

Arbola, E., Bernstein, F., Bryant, S., Koetzle, T., & Weng, J. (1987). Protein Data Bank. In F. Allen, G. Bergerhoff, & R. Sievers (Eds.), *Crystallographic Databases - Information Content, Software Systems, Scientific Applications* (pp. 107-132). Bonn: Data Commission of the International Union of Crystallography.

Berry, D., & Fristedt, B. (1985). *Bandit Problems: Sequential Allocation of Experiments*. NY, NY: Chapman and Hall.

Bloor, D. (1977). *Knowledge and Social Imagery*. London: Routledge and Kegan Paul.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., & Freeman, D. (1988). AutoClass: A Bayesian Classification System. In *Proceedings of Fifth International Conference on Machine Learning,* (pp. 54-64). Ann Arbor, MI: Morgan Kaufman.

Cheeseman, P., Stutz, J., Hanson, R., & Taylor, W. (1990). Autoclass III. Program available from NASA Ames Research Center: Research Institute for Advanced Computer Science.

Cohen, B., Presnell, S., & Cohen, F. (1990). Pattern Based Approaches to Protein Structure Prediction. *Methods in Enzymology*, (May 23, 1990).

Collins, G. (1987) *Plan Creation: Using Strategies as Blueprints*. PhD diss., Yale University, Report YALEU/CSD/RR#599.

Cox, M. T., & Ram, A. (1992). Multistrategy Learning with Introspective Meta-Explanations. In *Machine Learning: Proceedings of the Ninth International Conference,* (pp. 123-128). Aberdeen, Scotland: Morgan Kaufman

desJardins, M. (1992) *PAGODA: A Model for Autonomous Learning in Probabilistic Domains*. Ph.D. thesis, University of California, Berkeley, Computer Science Division (EECS), available as technical report UCB/CSD 92/678.

Dietterich, T. (1989). Limitations on Inductive Learning. In *Proceedings of Sixth International Workshop on Machine Learning,* (pp. 125-128). Ithaca, NY: Morgan Kaufman.

Fisher, D. (1987). Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning*, 2, 139-172.

Hesse, M. (1966). *Models and Analogies in Science*. South Bend, IN: University of Notre Dame Press.

Holder, L. B. (1991) *Maintaining the Utility of Learned Knowledge Using Model-based Adaptive Control*. PhD thesis, University of Illinois at Urbana-Champaign, Computer Science Department.

Holley, L. H., & Karplus, M. (1989). Protein Secondary Structure Prediction with a Neural Network. *Proceedings of the National Academy of Science USA* 86 (January), 152-156.

Horvitz, E., Cooper, G., & Heckerman, D. (1989). *Reflection and action under scarce resources: Theoretical Principles and Empirical Study* (Technical report no. KSL-89-1). Knowledge Systems Laboratory, Stanford Univ.

Hunter, L. (1989a) *Knowledge Acquisition Planning: Gaining Expertise Through Experience*. PhD thesis, Yale University, Available as YALEU/DCS/TR-678.

Hunter, L. (1989b). Knowledge Acquisition Planning: Results and Prospects. In *Proceedings* *of The Sixth International Workshop on Machine Learning*, (pp. 61-66). Ithaca, NY: Morgan Kaufman.

Hunter, L. (1990a). Knowledge Acquisition Planning for Inference from Large Datasets. In *Proceedings of The Twenty Third Annual Hawaii International Conference on System Sciences*, vol. 2, Software track (pp. 35-44). Kona, HI: IEEE Press.

Hunter, L. (1990b). Planning to Learn. In *Proceedings of The Twelveth Annual Conference of the Cogntive Science Society*, (pp. 26-34). Boston, MA: Erlbaum Associates

Hunter, L. (1992). Classifying for Prediction: A Multistrategy Approach to Predicting Protein Structure. In R. Michalski (Ed.), *Machine Learning IV: Multistrategy Learning* San Mateo, CA: Morgan Kaufman. Forthcoming.

Hunter, L., & States, D. (1991). Applying Bayesian Classification to Protein Structure. In *Proceedings of Seventh Conference on Artificial Intelligence Applications*, vol. 1 (pp. 10-16). Miami, FL: IEEE Computer Society Press.

Karp, P. (1991). *ASN.1 parser and Printer Documentation* (Technical report 5). National Center for Biotechnology Information.

Karplus, M., & Petsko, G. A. (1990). Molecular Dynamics Simulations in Biology. *Nature*, 347(October), 631-639.

Kass, A. (1990) *Developing Creative Hypotheses By Adapting Explanations*. Ph.D. thesis, Yale University, Available as Institute for the Learning Sciences Technical Report #6.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific Discovery: An Account of the Creative Process*. Cambridge, MA: MIT Press.

Lenat, D. (1979). On Automated Scientific Theory Formation: A Case Study Using the AM Program. In J. Hayes, D. Mitchie, & L. I. Mikulich (Eds.), *Machine Intelligence* New York, NY: Halstead Press.

Qian, N., & Sejnowski, T. (1988). Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. *Journal of Molecular Biology*, 202, 865-884.

Quinlan, J. R. (1991). C4.5. Program available from the author: quinlan@cs.su.oz.au.

Ram, A. (1989) *Question-driven Understanding: An Integrated Theoryt of Story Understanding, Memory and Learning*. PhD thesis, Yale University, Report YALEU/CSD/RR#710.

Ram, A., & Hunter, L. (1992). A Goal-based Approach to Intelligent Information Retrieval. *Applied Intelligence*, to appear in vol. 2(1).

Rendell, L., & Cho, H. (1990). Empirical Learning as a Function of Concept Character. *Machine Learning*, 5(3), 267-298.

Rendell, L., & Seshu, R. (1990). Learning Hard Concepts through Constructive Induction: Framework and Rationale. *Computational Intelligence*, 6, 247-270.

Shrager, J., & Langley, P. (1990a). Computational Approaches to Scientific Discovery. In J. Shrager & P. Langley (Eds.), *Computational Models of Scientific Discovery and Theory Formation.* San Mateo, CA: Morgan Kaufmann.

Shrager, J., & Langley, P. (Ed.). (1990b). *Computational Models of Scientific Discovery and Theory Formation*. San Mateo, CA: Morgan Kaufmann.

Skolnick, J., & Kolinski, A. (1990). Simulations of the Folding of a Globular Protein. *Science*, 250(November 23), 1121-1125.

Tweney, R. D. (1990). Five Questions for Computationalists. In J. Shrager & P. Langley (Eds.), *Computational Models of Scientific Discovery and Theory Formation.* San Mateo, CA: Morgan Kaufmann.

Valient, L. (1984). A theory of the learnable. *Communications of the ACM* 27(11), 1134-1142.

Weinert, F. E. (1987). Introduction and Overview: Metacognition and Motivation as Determinants of Effective Learning and Understanding. In F. E. Weinert & R. H. Kluwe (Eds.), *Metacognition, Motivation and Understanding* Hillsdale, NJ: Lawrence Erlbaum Associates.

Zhang, X., Mesirov, J., & Waltz, D. (1992). Hybrid System for Protein Secondary Structure Prediction. *Journal of Molecular Biology*, 225, 1049-1063.