Knowledge-Based Generation of Machine Learning Experiments: Learning With DNA Crystallography Data*

Dawn Cohen† and **Casimir Kulikowski**† and **Helen Berman**‡
Departments of †Computer Science and ‡Chemistry
Rutgers University
New Brunswick, NJ 08855
dcohen,kulikows@cs.rutgers.edu, berman@dnarna.rutgers.edu

Abstract

Though it has been possible in the past to learn to predict DNA hydration patterns from crystallographic data, there is ambiguity in the choice of training data (both in terms of the relevant set of cases and the features needed to represent them), which limits the usefulness of standard learning techniques. Thus, we have developed a knowledge-based system to generate machine learning experiments for inducing DNA hydration pattern classifiers. The system takes as input (1) a set of classified training examples described by a large set of attributes and (2) information about a set of learning experiments that have already been run. It outputs a new learning experiment, namely a (not necessarily proper) subset of the input examples represented by a new set of features. Domain specific and domain independent knowledge is used to suggest subsets of training examples from suspected subpopulations, transform attributes in the training data or generate new ones, and choose interesting ways to substitute one experiment's set of attributes with another. Automatic hydration pattern predictors are of both theoretical and practical interest to DNA crystallographers, because they can speed up a labor intensive process, and because the extracted rules add to the knowledge of what determines DNA hydration.

Introduction

In trying to learn to predict *DNA hydration patterns* (or particular kinds of 3-D arrangements of water molecules around a DNA molecule) from crystallographic data, several problems preclude direct application of traditional classifier learning techniques. This is because the domain violates certain assumptions underlying the application of inductive learning systems like ID3 [Quinlan, 1983], CART [Breiman *et al.*, 1984] or Swap-1 [Weiss and Indurkhya, 1991]. First, most learning techniques require a reasonably strong knowledge of which features are necessary for building classifiers. In our domain, experts have only uncertain knowledge. Second, the learning algorithms assume that cases are drawn from a homogeneous population, and that the derived rules will only be used to classify cases from the same population. However, the DNA hydration data that are available are from a mixture of several subpopulations, which may or may not have distinct factors governing their hydration. Finally, most learning algorithms produce only a single classifier from a particular training set, but since our data are noisy, and the knowledge of what determines DNA hydration is uncertain, it would be useful to identify a set of classifiers. It is this combination of considerations which led us to develop a knowledge-based system that will apply the learning algorithms effectively despite the violations of these assumptions.

We showed in [Cohen *et al.*, 1992] that it is possible to learn classifiers for a class of hydration pattern prediction subproblems. In that study, classifiers for some of the subproblems could not be derived, due in part, to limitations of predictive power in the restricted set of features used. Thus, it was necessary to extend the set of attributes available to the learner.

In principle, it is possible to represent training cases with every possible feature that could be considered useful for learning classifiers. However, with many noisy or irrelevant features, a learning algorithm is likely to find spurious correlations at an early stage of classifier induction, which may lead greedy induction methods to produce very poor classifiers. In addition, some computationally intensive learning algorithms may be quite slow when used with many attributes. On the other hand, it is not feasible to try learning with every possible subset of attributes. The set of training cases to be used poses a similar dilemma. It is possible to learn on the entire set of training examples, irrespective of whether or not they are from distinct subpopulations. But it is likely that better classifiers can be obtained from homogeneous train-

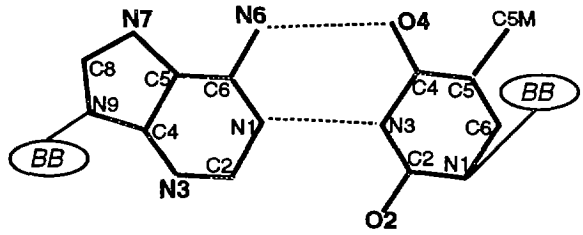
*This material is based on work supported by the National Science Foundation under grant NSF BIR 90 12772 and the National Institutes of Health, under grant GM21589 and a fellowship for DC under grant GM08319.

ing sets. Exhaustive testing of every possible subset of training examples is not likely to be either feasible or productive. We resolve these conflicts by making use of prior knowledge of the problem which provides constraints on the choice of data to supply to a learner.

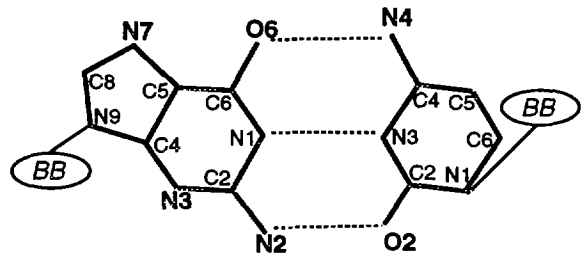
In our framework, a *learning experiment* consists of running a machine learning algorithm on a particular data set. Our system generates learning experiments by continually varying experimental conditions, namely the set of training cases and the features used to represent them. An incremental approach to experiment generation is used. Input to the system consists of a set of training examples and a set of experiments that have already been generated and run. New experiments are created by examining the results of previously run experiments. Domain specific knowledge and domain-independent heuristics are used both for designing the experiments, and ruling out unpromising ones.

Domain-specific knowledge helps select (or stratify) subsets of training cases which are likely to have been drawn from more homogeneous populations than the entire set of examples. Furthermore, it suggests interesting ways of choosing subsets of attributes based on the interrelationships and relative importance of particular attributes for different kinds of hydration pattern prediction subproblems. Finally, knowledge of how the subproblems affect each other allows us to exploit previous learning in designing experiments for related subproblems. Domain independent heuristics constrain the experiments so that the training cases are likely to be adequate in number and reasonably well distributed over the classes to allow learning.

The system has successfully been used to generate learning experiments. The experiments have yielded classifiers which provided some interesting insights about factors affecting DNA hydration.



Adenine base-paired with Thymine



Guanine base-paired with Cytosine

Figure 1: The DNA base pairs: cross sections through a double helix. Dashed lines represent hydrogen bonds between bases that hold the chains of the double helix together. "BB" denotes the sugar-phosphate backbone. All atoms of the bases (except hydrogens) are labeled. Hydration predictors were learned for atoms shown in bold.

The DNA Hydration Problem

DNA is the molecule which carries the genetic code and much research has been done to determine how its structure affects its ability to transmit the genetic information. Some of its interactions with proteins and drugs are mediated by water molecules (e.g. [Aggarwal *et al.*, 1988], [Neidle *et al.*, 1980]) which is one motivation for working on understanding what determines a DNA molecule's hydration.

Much of what we know about the structure and function of DNA comes from x-ray crystallographic analysis. In such an analysis, a DNA crystal structure is *solved* by using x-ray diffraction to identify 3-D coordinates for every atom in the crystal. DNA crystals generally contain water molecules as well as DNA, and coordinates must be found for both. Constraints on the arrangements of DNA atoms are well known, whereas those of water arrangements are not, making the process of finding coordinates for waters more difficult and error-prone than for DNA. By developing predictors of DNA hydration, we provide a tool which can make structure solution simpler.

A number of researchers have discussed DNA hydration in qualitative terms. Type B- and Z-DNA structures have frequently been found to have a "spine of hydration" in their minor grooves, and other regular arrangements have been described [Berman, 1991]. However, relatively little *quantitative* work has been done to predict coordinates of water molecules around DNA, using information about known structures (i.e. solved DNA crystal structures).

Our previous work [Cohen *et al.*, 1992], [Schneider *et al.*, 1992] showed how all known DNA-water structures can be studied in a common framework, in order to define hydration patterns. We also showed in [Cohen *et al.*, 1992] that it is possible to learn hydration pattern predictors, based on the defined patterns. These studies identified hydration pattern building blocks found around the bases which are the building blocks of DNA.

In both this study and [Cohen *et al.*, 1992], the problem of predicting the hydration pattern of a DNA molecule is seen as a set of subproblems of predict-

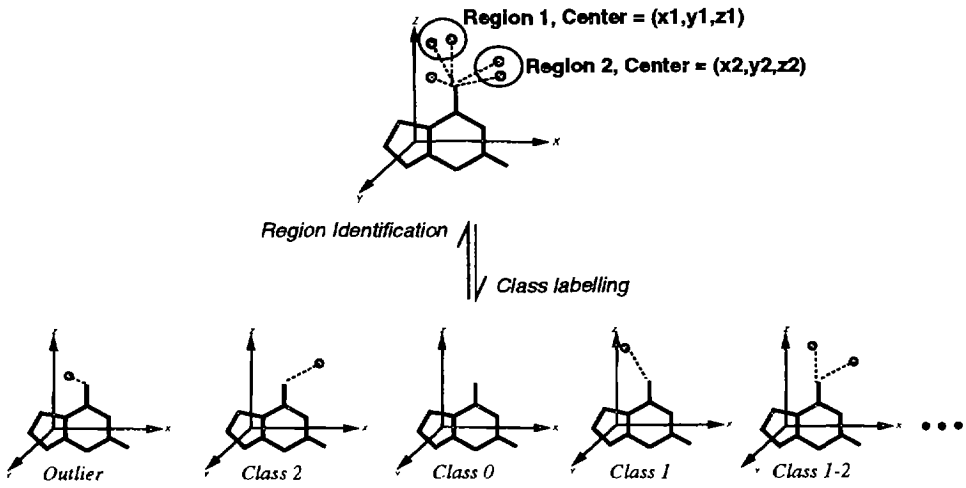


Figure 2: Determining hydration patterns. Bases (without class labels) are superimposed to find regions with many waters. Regions are used to define the classes, and then to label the bases with classes.

ing hydration near particular atoms (those capable of hydrogen bonding with water) in each base of the molecule. Figure 1 shows the DNA bases and the atoms for which hydration predictors were learned.

The method for defining hydration patterns used here is almost identical to that used in [Cohen *et al.*, 1992]. Forty-seven DNA crystal structures were chosen from the Nucleic Acid Database [Berman *et al.*, 1992] as data for identifying possible patterns. All bases of a specific type (taken from these structures) were superimposed on each other, along with their associated water molecules. (Our training sets consisted of 247 guanines, 240 cytosines, 107 adenines and 100 thymines.) Regions where waters tended to be found around the particular type of base were then located. (In [Cohen *et al.*, 1992] this was carried out by statistical cluster analysis, whereas in the present study, the regions were found from electron density maps of the superimposed bases with their waters, which is a more plausible method for localizing hydration. The details of this procedure are given in [Schneider *et al.*, 1993].)

The hydration pattern around an atom in a specific base is defined as follows (see Figure 2). Given the waters around that atom in the base, the classification of the base is the set of regions that the waters fall in. These newly classified bases, described by a set of attributes, are used as training examples for the decision tree learner CART. The classifiers can then be used to predict hydration around bases whose waters are unknown. If such a base is classified into some class, C, it is predicted to have waters at the centers of the regions that C represents.

The exact structure in the vicinity of a base places two kinds of constraints on possible positions of waters near it. First, water cannot physically occupy the same space as the DNA atoms around the base. Second, the waters are frequently observed placed in such a way as to be able to hydrogen-bond with more than one DNA atom. Though these two constraints are correct, they are too weak to be helpful for prediction. There is no known simple measure of the structure which can be used to compute hydration positions. Several easily determined attributes which measure aspects of DNA structure have been proposed as influencing hydration, but the evidence supporting them has been quite limited [Berman, 1991].

Many attributes can act as measures of the local structure around a base. For example, each *type of DNA* (e.g. left-handed helical Z-DNA or right-handed A- or B-DNA) puts different spatial constraints on the backbone and the bases above and below a base, so there will be different limitations on the free space where waters may be found. Bases with *chemical modifications* (extra atoms added onto the standard DNA bases) will have different shapes and hence, different allowed water positions. Likewise, because different types of bases have different shapes, the *types of the neighboring bases* around a base also limit its possible water positions. The twelve *torsion angles* that determine the exact structure of a nucleotide (which are strongly related to the DNA type) could also affect the space available for waters. In addition, some attributes related specifically to crystallographic properties (rather than the structure of the DNA molecule itself) may affect where waters are observed around DNA molecule. These include measures of *packing* (which tells how DNA molecules fit together in a crystal, possibly forcing water away from a base) and *resolution* (which gives a measure of the observability of waters in a crystal structure). All of these attributes

have been used in the classifier learning experiments of this study.

Domain knowledge specifies some subpopulations of DNA molecules which might be important to study separately. In particular, molecules of several different DNA types are used in this study, and it is thought (but not certain) that the hydration of each type may be governed by distinct factors. In addition, the known molecules are from a mixture of high and low resolution structures, and since the coordinates of waters in low resolution structures are quite uncertain, it is thought that it may be appropriate to study only high resolution structures. (It is not known, however, if this will make any difference in learning classifiers). Thus, it is necessary to experiment with learning classifiers with data from each of these subpopulations, as well as for the entire set of examples.

The specific subpopulation of DNA being studied affects the likely relevance of different attributes for particular learning problems. It is known [Saenger, 1984], for example, that the value of the χ torsion angle of guanine and adenine nucleotides in Z-DNA is always in a narrow range, called the "syn" conformation (an unusual range for this angle). Thus, it is not likely to be productive to include χ as an attribute if we are studying hydration around a guanine atom, using only the cases from Z-DNA. As another example, B-DNA molecules tend to pack differently in crystals, depending on their chain length, and since packing is thought to affect hydration patterns, chain length was expected to be a useful attribute in predicting hydration, when considering cases from B-DNA. Various properties of neighboring atoms and neighboring bases (above and below a base) could affect the hydration pattern around an atom.

In [Cohen *et al.*, 1992] we found that classifiers of hydration patterns for atoms that are close together in 3-D space often have similar forms. We can use this knowledge to design experiments for learning about hydration around one atom, based on information about experiments that have been run for *related* atoms. This includes selecting from experiments of base-pairing atoms (e.g. using an experiment from cytosine O2 or guanine N3, for generating one for guanine N2), or for neighboring atoms on the same base (e.g. guanine O6 and guanine N7).

Generating Learning Experiments

To perform a *learning experiment* in our framework is to induce a (DNA hydration) classifier using some particular set of training data. Each experiment is distinguished by the subset of the possible training examples used, and the subset of all possible features used to represent them. (We are *not* concerned with biological experiments, per se. Instead, each of our learning experiments selectively focuses on different results of crystallographic experimentation, in order to draw general conclusions from large numbers of DNA structure determination experiments. In this sense, we are investigating a type of meta-experimentation.)

We have developed a knowledge-based system which automatically generates learning experiments. It takes as input, a set of classified training examples over a large set of features and information about a set of experiments that have already been run. It outputs a set of data in a format which can be input directly to a learner, as well as information about how the data were chosen.

An iterative approach to experiment generation is used. A new experiment is produced by modifying an earlier one, based on some interesting aspect of its outcome. The new experiment is then run, and is included in the set of experiments, where it can serve as a basis for further experimentation. Figure 3 shows the overall structure of the design process as implemented in this study.

As mentioned earlier, in the domain of hydration pattern prediction, we are not faced with a single classification problem, but rather a set of problems of predicting hydration around each hydrogen bonding atom in each of the four types of bases. Thus, much of the encoded knowledge can be shared across multiple subproblems.

Knowledge used in this process describes various aspects of the problem of finding classifiers in our domain. It includes:

DNA knowledge

- properties of particular atoms (e.g. proximity to other atoms, similarity to other atoms);
- which problems are most likely to have similar classifiers;
- the DNA parts hierarchy;

Knowledge of particular attributes

- how useful an attribute is likely to be in learning classifiers for a particular atom or class of atoms (e.g. guanine N7 or all atoms in Z-DNA guanine);
- how different attributes tend to depend on each other (for particular atoms or classes of atoms);
- how numeric attributes can be (meaningfully) discretized;

Learning experiment knowledge

- constraints on acceptable learning experiments;
- heuristics for modifying previous experiments;
- methods for generating new attributes by applying various operations on the initial attributes;

The first step in designing an experiment is to choose the *data set*, a subset of the possible training examples. This strongly constrains the following steps. The second step is to choose an experiment that has already been run, to serve as a *template* for the experiment to be designed. The third step is to choose a *goal attribute* to focus on as an anchor for modifying the

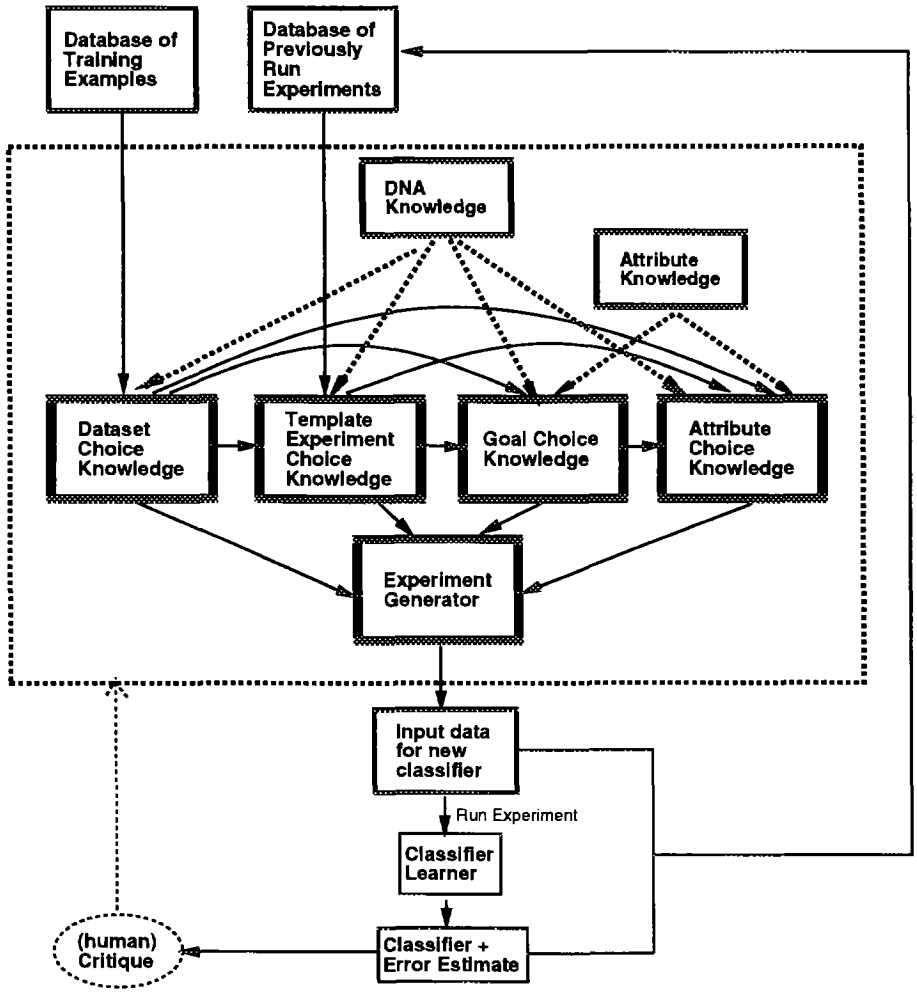


Figure 3: Flow of information in the experiment generation process. Dotted line encloses portions of the system implemented in this study.

template and generating the new experiment. The last step is to choose the set of attributes to represent the training examples. Each step in the process constrains those that follow it, and information about rules used for each decision may be used to make later decisions (indicated by the solid arrows in Figure 3).

The knowledge for making decisions at each step consists of a set of rules. The rules fall into two classes: those for suggesting possible ways to make decisions and those for ruling out certain decisions. These rules are for the most part domain independent (in that they do not generally refer to particular attributes or data sets). However, they are instantiated by attribute and DNA objects which encode domain specific knowledge, and as such, the rules can be used to generate experiments intelligently. (Indicated by dashed arrows in Figure 3.)

We now describe in more detail each step in the experimental design process.

Domain knowledge for choosing subsets of training examples captures experts' opinions of interesting subsets of data, or those which may give rise to distinct classifiers. In this study we have confined ourselves to considering sets of bases from a single DNA type at a time or from high resolution structures. The knowledge can be thought of as specifying different database selection operations on the training examples based on values of their attributes. Alternatively, for decision tree learning, it can be thought of as forcing a particular split (say, on DNA type) to be at the root of a tree. Domain independent heuristics rule out the possibility of choosing data sets with poor distributions of cases. (For example, there is not much to learn about a training set with (almost) all cases in one class.)

	Ade-N3				Ade-N6			Ade-N7		
	All-I	All-II	B-I	B-II	All-I	All-II	B	All	B-I	B-II
Error %	23	21	24	20	22	28	25	37	35	33
Error % prior	45	45	48	48	31	31	25	39	35	35
Change error	22	24	24	28	9	3	0	2	0	2
Rel. % change	49	53	50	58	29	10	0	5	0	6

Table 1: Summary of results for hydration predictors for adenine atoms N3, N6 and N7. Error %: n-fold cross validation error estimate for classifier. Error % prior: % of cases in largest class (guessing error). Change error: difference between error of classifier and error of guessing. Rel. % change: % of guessing error reduced by using classifier. Set I contains twenty attributes; Set II contains all but five of these. N6 B-DNA results were identical for Sets I and II, as were N7 All-DNA. There were 107 cases in the entire set and 80 in B-DNA's.

The second step in experimental design is to choose an experiment that has already been run. The new experiment will be very similar to the template experiment with some modification. There are a number of heuristics which are used to identify useful candidates. For example, experiments which yielded good classifiers for one atom may be used as templates for neighboring (i.e. physically close) atoms, on the assumption that the two atoms will be subject to similar physical constraints. Template experiments must have a reasonable overlap of training cases with the data set chosen in the first step. They are generally chosen from among those which have produced the lowest error-rate classifiers.

The third step in experiment design is to choose a goal attribute to focus on. Changes to the template experiment are made by varying the goal attribute's use in the new experiment, compared to the old. Goal attributes may be from the initial set of attributes or may be generated in this step. A number of rules for choosing goals have been implemented, including:

- choose a "surprising" attribute used in the classifier produced by the template experiment (i.e. one which was not expected to have much effect on hydration for the current atom, but did);
- choose an attribute which is expected to have a strong effect on hydration for the current atom;
- choose an attribute that has not previously been used for the current atom;
- generate an attribute which may be useful for predicting hydration around the current atom.

An attribute choice is excluded from becoming the goal if there are not at least some fixed number of cases for each of at least two values of the attribute. (In other words, not all of the training cases should have the same value on the goal attribute.)

The last step in designing an experiment is to choose the entire set of attributes that will be used to represent the training cases. In most cases, this is the set of attributes used in the template experiment, with some modifications determined by the goal attribute. Specifically, the new experiment may add or delete this attribute from the template experiment, discretize it, or substitute it with another (set of) attribute(s). In addition, some more drastic changes may be made, such as using only the attributes that appeared in the classifier generated by the template or discretizing all of the numeric variables. The exact choice depends on the data set and goal attribute and how they were chosen. For example, one kind of goal is an attribute that experts had not expected to be useful in predicting hydration for some class of cases, but did show up in the template's classifier. The system tests the importance of such an attribute by deleting it in the new experiment. If the learned classifier is less accurate than the template experiment's classifier, we infer that there was some true relevance of the attribute.

Once all choices have been made, the data can be output in a format appropriate to a classifier learner (generally CART, in this study). A record of the experiment and rules used to generate it is also output. A classifier is learned for the chosen data. The new experiment and its results can then be incorporated into the set of template experiments.

The system has been implemented on a Sun 4 workstation, using CLOS running under CMU Lisp. Objects represented include DNA objects (such as molecules, bases and atoms), attributes, experiments, rules, choices, sets of training cases and relationships between attributes.

Results

A number of experiments in learning hydration pattern predictors have been run. All of them used CART with the gini splitting index and 0.0 SE tree pruning rule.

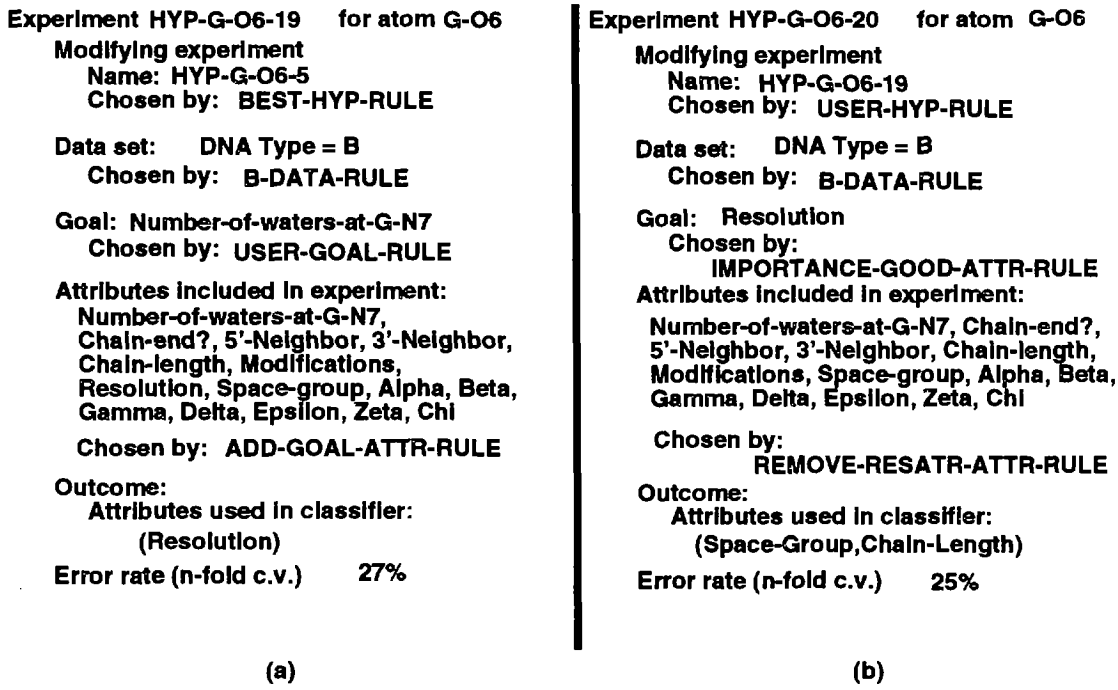


Figure 4: Two experiments in learning hydration predictors for guanine O6. (a) Template experiment (b) Experiment derived from (a).

Error rates were estimated using n-fold cross validation (leave-one-out).

The first set of experiments were run with manually chosen data and attribute sets. These served as initial templates for the system. These experiments were run for each atom capable of hydrogen bonding with water (see Figure 1). Two different sets of attributes were used for each set of training examples. Hydration predictors were learned for guanine and cytosine atoms, using only cases from A-, B- or Z-DNA, besides the entire set of cases. Predictors were learned for adenine and thymine atoms using only cases from B-DNA and all cases (there were very few cases of these from A- or Z-DNA). The attribute sets included features both of individual bases and of the crystal structures containing them. Attributes of crystal structures included crystallographic resolution, space group, length of DNA chain in the crystal, and the type of DNA. Attributes of individual bases included the $\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \chi$ torsion angles of the backbone, type of bases that are 5' and 3' (above and below) neighbors of the base, chemical modifications of the base and a predicate denoting whether or not the base is at the end of a DNA chain. One set (II) of template experiments used exactly these attributes. Another set (I) used these plus the $\nu_0, \nu_1, \nu_2, \nu_3, \nu_4$ torsion angles of the sugar attached to the base. (See [Saenger, 1984] for details of all of the attributes.) Table 1 summarizes the results of these initial template experiments for adenine.

From Table 1, it is clear that making all possible attributes available to a learning algorithm may not always be a desirable strategy for inducing classifiers. In several cases where the ν's were included, worse results were obtained than for the corresponding experiments that omitted them. (The ν's are known to be particularly noisy, so this result is not entirely surprising.) In these cases, either no tree was obtained at all or less accurate trees were obtained. In a few cases, worse classifiers were obtained when these attributes were omitted. Thus, it is clear that it is possible to obtain better results when we allow experimentation with training data.

Figure 4 shows two experiments generated by the system for learning hydration pattern predictors for guanine O6. The first of these is a template experiment. The second is derived from the first by removing the attribute *resolution*. This change was suggested because the template relied on resolution, which is thought to be an important attribute for determining whether or not hydration is observable (for any atom). The system tries to test whether resolution is really essential, or whether it can be replaced by any other features in the set. The results indicated that chain-length and space group together can be substituted for resolution to obtain a slightly improved classifier. The

error rate of a classifier which always guesses the most likely class is 37%. The tree learned in the template experiment has an error rate of 27% and contains 1 decision node. The tree learned in the second experiment has a slightly lower error rate of 25% but uses 2 decision nodes.

Some interesting negative results were obtained. Experts believe that for B-DNA structures, the attributes *chain-length* and *packing* are related to each other. In particular, chains of length 10 pack in such a way as to leave space for waters around guanine N3, whereas chains of length 12 do not. (Almost all of our B-DNA structures are one of these two lengths.) Chain length turned out to be a useful attribute for classifying hydration, for this subproblem. Surprisingly, however, experiments designed by the system which substituted packing for chain length indicated that packing is not useful for learning hydration predictors for B-DNA. On the other hand, it was useful in learning predictors for high-resolution structures.

Conclusion

Domain knowledge is usually needed to apply classifier induction systems effectively. This knowledge is generally implicit in an expert's choice of training examples and the attributes used to represent them. If these are not chosen appropriately, the learned classifier is not likely to be useful for cases outside the training set. In domains where the choice of training data is not clear-cut, it is necessary to experiment with the data presented to a learning algorithm. In this study, we have shown how it is possible to use the limited expert knowledge available to guide the process of designing experiments for learning DNA hydration predictors.

Other techniques have been developed to address some of the issues examined here. The problem of choosing attributes (from a fixed set) to represent examples is frequently addressed with feature selection and feature extraction techniques [Fukunaga, 1990]. However, these methods tend to be most useful when all attributes are numeric. In our domain, we use a combination of numeric and categorical features. Also, if the scales of the features vary greatly (as in our domain), some important features may be ruled out by these techniques. The FRINGE system [Pagallo and Haussler, 1990] addresses the possibility of generating new features and learning new trees with these. However, the attributes that are generated are only boolean combinations of the original ones. SEEK2 [Ginsberg *et al.*, 1988] iteratively changes a rule set by adding and deleting variables in rules so that they classify a set of cases with greater accuracy. It assumes, however, that the initial rule set produces very good performance, and merely needs some tuning. Swap-1 [Weiss and Indurkhya, 1991] induces a rule set and then iteratively improves it by swapping variables in and out of rules and changing their thresholds. It thereby decreases the likelihood that extra attributes will adversely affect the induced classifier. However, it requires a lot of computation. None of these methods can make use of domain knowledge to bias the process of generating classifiers. Also, it is necessary to choose the training examples manually with all of these techniques.

There are several properties of our approach which are useful in our domain. First of all, we can make use of whatever limited knowledge is available to automate the design of learning experiments, and to rule out the generation of certain poor experiments. Second, because of our iterative approach, we generate alternative classifiers for the same training cases which may provide different insights to the expert. Third, though we are dealing with multiple classification problems, rules do not have to be written for each specific problem: much of the knowledge applies to many problems, and by writing one rule, we may be helping to design many different experiments. Finally, domain knowledge can be used to generate attributes which are potentially relevant to the problem at hand.

Work in progress extends the current framework of designing learning experiments. Specifically, the method used to label the training cases with classes is noisy, and it is useful to be able to consider alternative ways of viewing the problem. For example, in many cases, it may be reasonable to learn predictors of whether or not there is any hydration around an atom (a two-class problem), rather than trying to learn a predictor of all classes. Given our rather small sets of training cases, this is likely to be very helpful, and initial results of using the system to suggest such problem reformulations seem quite promising. Another possible extension to the system would take greater advantage of information about weaknesses of a learned classifier. It may be possible to use knowledge to characterize the cases that are misclassified, and design an experiment to reduce the error rate.

Acknowledgements

Many people have provided great assistance in carrying out this study. These include Bohdan Schneider, Leah Schleifer, A.R. Srinivasan, Jordi Bella, and the staff of the Nucleic Acid Data Base.

References

Aggarwal, Aneel K.; Rodgers, David W.; Drottar, Marie; Ptashne, Mark; and Harrison, Stephen C. 1988. Recognition of a DNA operator by the repressor of Phage 434: A view at high resolution. *Science* 242:899–907.

Berman, H.M.; Olson, W.K.; Beveridge, D.L.; Westbrook, J.; Gelbin, A.; Demeny, T.; Hsieh, S.H.; Srinivasan, A.R.; and Schneider, B. 1992. The nucleic acid database: A comprehensive relational database of three-dimensional structures of nucleic acids. *Biophysical Journal* 69:751–759.

Berman, Helen 1991. Hydration of DNA. *Current Opinions in Structural Biology* 1(3).

Breiman, L.; Friedman, J.H.; Olshen, R.A.; and Stone, C.J. 1984. *Classification and Regression Trees.* Wadsworth and Brooks, Monterey, CA.

Cohen, Dawn M.; Schneider, Bohdan; Berman, Helen M.; and Kulikowski, Casimir A. 1992. Learning to predict DNA hydration patterns. In *Proceedings of the Ninth IEEE Conference on AI for Applications.*

Fukunaga, Keinosuke 1990. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, second edition.

Ginsberg, Allen; Weiss, Sholom M.; and Politakis, Peter 1988. Automatic knowledge base refinement for classification systems. *Artificial Intelligence* 35:197–226.

Neidle, Stephen; Berman, Helen M.; and Shieh, H.S. 1980. Highly structured water networks in crystals of a deoxydinucleoside-drug complex. *Nature* 288(5787):129–133.

Pagallo, Giulia and Haussler, David 1990. Boolean feature discovery in empirical learning. *Machine Learning* 5:71–99.

Quinlan, J. Ross 1983. Learning efficient classification procedures and their application to chess end games. In Michalski, R.S.; Carbonell, J.G.; and Mitchell, T.M., editors 1983, *Machine Learning: An Artificial Intelligence Approach.* Morgan Kaufmann, Los Altos, CA.

Saenger, Wolfram 1984. *Principles of Nucleic Acid Structure.* Springer-Verlag, New York.

Schneider, Bohdan; Cohen, Dawn; and Berman, Helen 1992. Hydration of DNA bases: Analysis of crystallographic data. *Biopolymers* 32:725–250.

Schneider, Bohdan; Cohen, Dawn; Schleifer, Leah; Srinivasan, A.R.; Olson, Wilma; and Berman, Helen M. 1993. A new method to systematically study the spatial distribution of the first hydration shell around nucleic acid bases. *Forthcoming.*

Weiss, S. and Indurkhya, N. 1991. Reduced complexity rule induction. In *Proceedings of IJCAI-91*, Sydney. 678–684.