ADAPTION THROUGH GENERALIZATION

by

Ryszard S. Michalski

Invited paper at the Fifth Annual Ann Arbor Adaptive Systems Workshop,
University of Michigan, Ann Arbor, Michigan, July 21-23, 1980.

Invited paper at the Fifth Annual
Ann Arbor Adaptive Systems Workshop,
University of Michigan, Ann Arbor,
Michigan, July 21-23, 1980.

4. TUESDAY AFTERNOON (De Jong chairing)

4.1. Adaptation through Generalization - Ryszard Michalski

The adaptation to be discussed occurs in a highly structured environment, which provides a lot of information to the object or organism or system that learns. The system itself is equipped with knowledge of various high-level concepts. A very important part of learning is the ability to generalize over tasks. The adaptive processes to be considered are very general, applicable to solving all kinds of problems which produce generalizations of initial information. We have tried to identify, in the various methods which have been developed in machine learning and related fields, certain principles which are common, regardless of the application domain or the terminology used.

We think of learning as a process of building or modifying or improving descriptions, which here will be of the symbolic variety. Inductive learning involves building a new description from an initial one by a process of generalization. In addition to greater generality, the new description should meet the criterion of being the most plausible among the possible generalizations.

A taxonomy of learning types includes: (1) learning by creating a data base, through simple memorization or by being programmed; (2) learning by being taught, in which the system

accommodates new information within a given high-level conceptual organization; (3) learning by analogy, as in the application of Ohm's law to fluid dynamics; (4) learning from example, in which classification of new observations serves to generalize a given set of concepts (e.g., diseases); and (5) learning from observation, where the observations must be categorized without benefit of a pre-existing conceptual structure.

Buchanan suggested that another dimension, cutting across learning from example and observation, was the presence or absence of a teacher to provide examples that were already classified. Michalski felt that the presence of a teacher was what actually characterized learning from example. Buchanan said there was an intermediate case in which non-classified examples and non-examples of a single known concept were encountered.

The types of inductive learning Michalski has found dominant in the literature include: (1) concept acquisition, in which there are several examples representing certain concepts; (2) classification learning, or learning discriminating descriptions which contain sufficient information to distinguish between already known concepts; (3) sequence prediction, or learning from a (possibly non-deterministic) generating rule; and (4) "conceptual clustering," a type of learning from observation that reveals a taxonomic structure underlying an arbitrary collection of entities.

A paradigm which seems to encompass these kinds of inductive learning has been developed. (See, for example, [Michalski, "Pattern recognition as rule-guided inductive inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-2*, 349-361, 1980].) We start with data rules, descriptions in the form of production rules, frames, conditional statements, or the like. In learning discriminant descriptions, there may be a fixed number of generalization classes; these are not usually provided in learning from observation. Another important component of the paradigm is the set of (problem) knowledge rules, which include both specific knowledge of available descriptors (reflecting the semantics of the problem domain) and various generalization rules (reflecting problem-solving knowledge). The problem is to determine a new, more general set of decision rules, called hypotheses, which are complete, consistent, and in some sense preferable.

Consistency means that the hypotheses do not engender violations of the classification scheme of the original rules. Completeness means that the hypotheses do not fail to classify anything classified by the original rules. The preference criterion is used to select among competing sets of complete and consistent hypotheses. The preference criterion varies with the problem domain and can involve a priority ordered group of factors, such as the computational simplicity of the rules, the cost of measuring information needed for rule evaluation, and the degree-of-fit to the data.

Data rules, hypotheses, problem knowledge rules, and generalization rules are all expressed in an extended version of first order predicate calculus. The main extensions are (1) typing of variables (and functions and predicates) to handle domains which are unordered, linearly ordered, or hierarchically structured; (2) numerical quantifiers which can specify a range within a domain; and (3) the operators of internal conjunction and internal disjunction.

The transformation of data rules into hypotheses involves application of generalization rules, which transform one or more decision rules associated with a given generalization class into a new rule which is equivalent to, or more general than, the initial ones. Generalization rules, which are similar to processes by which people generalize information, include (1) dropping one or more conditions from descriptions which are logical products of conditions to be satisfied; (2) turning constants into variables, so that two descriptions referring to different objects can be combined; (3) extending quantification to an interval between two extreme values, in a linearly ordered domain; (4) climbing up one or more levels in the generalization tree, in a hierarchically organized domain; (5) changing the representation space by deriving new descriptors from combinations of old ones; and (6) associating descriptors, as is done to infer physical laws from correlations of variables.

A complete but simple example of these kinds of induction processes is a program which incrementally learns a

rule for diagnosing soybean diseases. Initially, the system is given only a few cases of each of the several diseases, where a case may involve on the order of 30 descriptors of the symptoms. By dropping apparently irrelevant conditions, the program generates a first set of rules, which discriminate new cases with 96% accuracy. In a second iteration, attention to the various kinds of wrong classifications produces a new, more complicated set of rules, which now discriminate new data with 98% accuracy. One more iteration leads to a final, and again simpler, set of hypotheses which are 100% accurate.

Buchanan asked if complete knowledge of the instances seen so far was retained, in order that a new iteration would not destroy prior correct relations. Michalski said yes, that to assume the system had such a memory was more interesting than the alternative. MacLaren requested clarification of how the representation space could be changed. Michalski answered that this occurred when new descriptors were derived. After some discussion, it became clear that a bottom level of basic attributes, like color and size, was not subject to modification but only to successive levels of transformation. Martinez asked how the set of possible transformations was determined. Michalski indicated that it was dependent on the problem and the insight of the investigators. De Jong followed this up with a comment that some of the transformations seemed quite domain dependent, such as one having to do with arithmetic. Michalski responded that there were many domains in which counting made

sense.

Another example involves a hypothetical organism confronted with an environment containing two kinds of structurally complex "cells," edible and inedible. The task is to survive and prosper by discovering those properties of the cells which discriminate with respect to edibility. The process of learning here again involves repetition of a three-phase cycle. The first phase is "focus of attention," in which the organism uses its extensive built-in knowledge of concepts, like the number of segments in a cell's circumference, to generate a symbolic description of each cell. In the second phase, the program alters the current descriptors in an attempt to formulate partial hypotheses which are preferable according to the criteria of being logically irredundant and sufficient to distinguish between the two types of cells. In the third phase, the hypotheses are tested on new data. Then the cycle begins again.

4.2. Problems Illustrating the Potential Use of Reproductive Plans - Hugo Martinez

Three kinds of problems will be described. The first two are typical of information processing problems arising in molecular biology and illustrate the need for efficient search methods in high dimension spaces. Genetic algorithms might be useful.

The first problem is the commonality problem, which arises in the comparative analyses of DNA segments responsible for a given function, such as the promoter sequences from a variety of species. You want to find out

what 50 or so such sequences have in common, in order to determine the functionally significant structural component.

For pairs of sequences, commonality has been analyzed in terms of evolutionary distance, or kinship. The question becomes one of the minimum number of mutations (including additions and deletions) to be made in either one of the two sequences in order to render them alike. Useful algorithms have been devised to solve this problem. But since the time complexity is proportional to the product of the sequence lengths, it is a much harder problem for many sequences with lengths on the order of 100 bases each.

The common structure aspect is also complicated by the fact that it may be necessary to view a linear sequence in the context of geometrical constraints, such as those imposed by a helical three-dimensional form. There seems to be a need for a uniform approach, starting perhaps with a careful assessment of how the structure of a linear sequence can be characterized. An idea currently being considered is based on the old string completion problem, which was alluded to by Michalski. Given a finite sequence over a specified alphabet, we would like to know what the next letter would be if the sequence were extended in a manner conforming to its structure. The notion of structure is thus made equivalent to "rules of formation"; and a common structure corresponds to a common set of rules. It may be possible to generalize to multiple strings the "Adaptive Production Systems" of Waterman [*Proceedings of the Fourth IJCAI*, pp. 296-303, 1975], which use a heuristic template for generating

inference rules about string formation.

This contemplated generalization would use a search space in which each point is a set of inference rules. Two difficulties confronting application of genetic algorithms are (1) the string representation of sets of inference rules is not evident, and (2) it is not clear what would constitute a critic for guiding the search in such a space.

The second problem to be discussed relates to the secondary structure of RNA molecules, in particular transfer RNA's. The bonding of complementary base sequences in such molecules gives rise to double-helical stems, with attached single-stranded loops. The number of three-base or longer sequences which can participate in stem formation is about the same as the number of bases in the molecule. This means that a molecule N bases long has 2**N possible secondary structures. In addition, loops can interact to form tertiary structures called knots; but approaches to date have neglected tertiary structure simply because of the complexity of the secondary structure problem itself. Out of all the possible secondary structures, the biochemist is looking for those that are most probable in that they have the least free energy.

De Jong asked if it was necessary to find a single global optimum, or if it was a satisficing situation, in which you could be happy with something reasonably close. Martinez said that, while you would always like to have one to play with, the experimenter normally wants to see the several, competing near-optimal structures, in order to

choose among them according to other considerations. Michalski inquired whether experimental data could be used to guide the search. Martinez noted that diffraction studies could help, when recrystallization could be achieved. Morgan said that mutation experiments could also be helpful, notably when changes in structure could be associated with changes in function. MacLaren asked if there were both upper and lower bounds on the size of the loop. Martinez said no, that it was still a wide open game. Even very large molecules could not usually be divided into subproblems.

Martinez has developed an algorithm for the problem which contrasts with previous approaches (as described by Studnicka, et al. ["Computer methods for predicting the secondary structure of single-stranded RNA," *Nucleic Acids Research*, 5, 3365-3387, 1978]). The new algorithm can be implemented on a minicomputer because of space efficiency. The computation time is reasonable (about an hour) for problems with 200 or 300 potentially double-helical regions. But significantly more powerful methods are needed to deal with newly arising problems involving thousands of regions.

The algorithm limits the search for minimum energy secondary configurations to "orthodox" structures, in which no two double-helical regions overlap (have a base in common) or form knots with one another. In a significant further reduction of the search space, attention can be limited to "maximum exclusion paths" of regions nested within larger regions, without loss of generality. Loosely, a region's maximal exclusion path is that linear sequence of least

deeply nested non-intersecting subregions which most completely spans the region; a rigorous definition has been developed in terms of the relative values and interrelations of parameters specifying region endpoints. Also rigorously developed is a series of recursive relations which determine the minimum orthodox-structure free energy of an RNA strand, ultimately in terms of the base pairing and stacking energies of its component regions.

In a post-workshop discussion, De Jong and Martinez developed a representation of the RNA secondary structure search space which would allow fairly natural application of genetic algorithms. The (totally ordered) N possible double-helical regions can easily be found, and their free energies precalculated. An N-bit string represents a secondary structure in which the regions corresponding to ones participate. The critic simply sums the free energies of the participating regions. Some complications arise in insuring that evolving structures do not contain overlapping or knotting pairs of regions.

The third problem is the "navigation problem," which has been devised to examine the potential of reproductive plans in a setting which clearly requires maintaining information in both the population and individual modes. Search is carried out over a space of programs which are formulated in a production system language like that used in the cognitive system of Holland and Reitman. As in biological systems, a program is assumed to have a "genome" and a "phenome"; the latter is partially derived from the genome through an

internally directed developmental process. The underived part of the phenome is a learning algorithm for carrying out the task.

In the simplest form of the navigation problem, a population of "explorers" traverse an environment of points specified as fixed length binary numbers. In one time step, an explorer can move from a current point to one of its K neighbors (by pushing one of K levers), remembering only the binary label of the previous point and which lever was pressed. Although most point labels can partially vary with time (to introduce the complication of focusing on the relevant information), a subset designated as "ports" have unique, invariant labels which are known to the explorer. In an overall attempt to find the shortest paths between all pairs of ports, each explorer reports the set of such paths it was able to discover.

The randomly generated productions comprising the initial explorers' genomes have binary string conditions which specify 0, 1, or "don't care" with respect to point label strings; the associated actions specify which lever to push. The productions are randomly divided into as many classes as there are ports. After being generated, an explorer undergoes development. In the current implementation, the genome is simply copied; but more sophisticated phenome generation methods are under consideration. The resulting production system is the data base part of the phenome.

The other part of the phenome is the explorer's route finding algorithm, which currently works as follows. Placed initially at random in the environment, the explorer must begin by finding a designated first port, in a semi-random search that is biased by the initial structure of the phenome. Once the first port is found, the label of the immediately preceding position is rendered "familiar" by addition to the phenome of a production whose condition perfectly matches that label and whose action is the lever actually used to reach the port. Subsequently, positions already familiar are used in the same way to cause "waves of familiarity" to spread out from discovered ports, which are sought in order. Eventually, each pair of ports will become linked by a familiarity path. The length of these paths is combined with the number of genome productions, the final number of phenome productions, and the total exploration time, producing an evaluation of the explorer that reflects both the efficiency of its exploration and the goodness of its solution.

The reproductive plan is then applied to the current population of explorers, producing offspring in proportion to parental goodness and modifying them by mutation and (optionally) crossover. New explorers whose performance is superior to that of the worst performing parents replace them in the population.

In this work, the genome is regarded as a generative description of the phenome, a description on which genetic operators should act. The critic selectively acts at the

phenotype level. The collective set of genomes constitutes the population memory, while that of the individual is the individual genome expressed in the corresponding phenome. Several generalizations of the above paradigm are envisaged. A most important one would permit adaptive development of the explorer's route finding procedure, through evolution of alternatives to gradual familiarization.