81-8

CONVART: A PROGRAM FOR CONSTRUCTIVE INDUCTION ON TIME DEPENDENT DATA

BY

JOHN HOWARD DAVIS

B.S., University of Illinois, 1979
A.B., University of Illinois, 1979

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1981

Urbana, Illinois

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor R. S. Michalski, my thesis advisor, for his guidance and ideas in the development of this work. I would also like to thank Bob Stepp for his suggestions, advice and ideas for the clustering and the change in slope algorithms, Paul O'Rorke for his editorial suggestions and ideas on relevancy measures, Albert Boulanger for the Black Cutworm information that he shared with me, and the entomologists Professor W. G. Ruesink and Chip Guse for providing their expertise. I would also like to thank the section of Economic Entomology of the Natural History Survey for providing the Black Cutworm Data analyzed in this paper.

This work was supported in part by the U. S. Department of Agriculture, grant No. 321512344 and by the National Science Foundation, grant No. MCS 79-06614.

Table of Contents

I. INTRODUCTION

A. Problem Statement

Induction is the process of creating generalized descriptions from specific descriptions (data). When the specific descriptions are grouped into classes then the general decriptions must characterize each class or distinguish each class from all the other classes. To create these generalized discriptions, inductive programs must be able to extract the relevant descriptors (variables) from the descriptors in the specific descriptions. Inductive programs like AQ/11 [Michalski and Larson 78] and ID3 [Quinlan 80] extract only those variables that are explicitly declared in the data. A variable representing, e.g., the arithmetic sum of two other variables could not be extracted by either AQ/11 or ID3.

If generalized descriptions can be expressed using the variables (declared in the data), their domains (the list of all possible values for a variable) and logical operators (such as conjunction, disjunction and negation), then an inductive program like AQ/11 can be used to create such descriptions. If generalized descriptions cannot be expressed in this way then a number of approaches can be taken to solve the problem.

If the data do not include sufficiently relevant variables, then new variables may be introduced, the original variables may be remeasured with greater precision and accuracy, or arithmetic and/or logical combinations of the original variables can be created, and used as new, potentially more relevant variables. The first two do not require any changes in the algorithm of induction.

The third method is performed by means of constructive induction, which can be of two kinds. In 'external' constructive induction (used by CONVART), variables are combined using arithmetic (and possibly logical) operators to create new derived variables, which then become part of the specific descriptions that are given to the inductive machinery. In 'internal' constructive induction, deriving new variables is done simultaneously with the process of generating descriptions. The operators used to generate derived variables can be very simple like arithemtic addition or very complicated like a series of special matematical functions. The number of ways in which numbers (values of variables) can be combined in this way is virtually infinite. Therein lies the problem, which combination is best or even which combination is sufficient to allow the inductive machinery to create satisfactory results? This problem is combinatorially very difficult, therefore it must be broken up

into subproblems. One such subproblem is constructing derived variables only from time dependent variables.

Time dependent variables are used to describe a history of changes that occurred in some process, instead of just describing a single snapshot of a process. With each such variable there is associated a time variable that specifies the moment at which the value of a given variable has been recorded. This pairing of values (between the dynamic variables and the time variable) allows CONVART to characterize the time course of an event. Derived-variables can be created that measure these changes. This is what CONVART was created to do. Such derived variables can then be given to an inductive machine like AQ/11. So by preprocessing the data, constructive induction on time dependent data can be accomplished using established inductive techniques.

B. Related Work

Constructive induction is just one of a growing number of topics of current interest in machine learning. General theory on learning systems and on methods of learning are described in [Buchanan, et. al. 77] and [Michalski 80]. There is also a growing list of implemented inductive programs. AQ/11 [Michalski

and Larson 78] is based on the A^q algorithm [Michalski 75b] and creates covers of events that are grouped into several classes. If events from only one class are available, then AQ7UNI [Stepp 79] can be used. As its name implies, it too is based on the A^q algorithm. ID3 [Quinlan 80] is also a general purpose program and uses a simle yet efficient algorithm.

Work on learning structural descriptions includes: INDUCE [Larson 77, Michalski 75a, Dietterich 78], SPROUTER [Hayes-Roth 78], THOTH [Vere 78] and Meta-DENDRAL [Buchanan and Fiegenbaum 78, Buchanan and Mitchell 78]. These methods can be compared using criteria such as method (data-driven versus model-driven), language (syntax and operators), generalization rules (like dropping a condition or turning a constant into a variable), efficiency (of algorithms not of the actual implementations) and extensibility (with regards to applications, noise immunity, domain knownledge and constructive induction) [Dietterich and Michalski 80]. INDUCE is the only one among these that has even a limited facility for constructive induction.

Among special purpose programs, Meta-DENDRAL [Buchanan and Fiegenbaum 78, Buchanan and Mitchell 78] is the most notable. Although it does not have a facility for constuctive induction, it does have a sophisticated facility to transform its raw data (mass

spectra) into bond-environment descriptions. Also, there is ELEUSIS [Dietterich 80], which plays the card game of the same name. It uses multiple models for induction to accomplish sequence extrapolation where both positive and negative training examples are available.

The program closest to CONVART is probably BACON.4 [Langely et. al. 80]. Although it does not analyze data with time dependent variables, it does combine variables mathematically (versus logically) to find constancies in the data.

Another method, besides constructive induction, which increases the power of inductive machines is the use of incremental learning. Systems with incremental learning must be able to test the rules they produce (perhaps using an expert system) and then they must analyze the results of these tests to see where and how improvements can be made. LEX [Mitchell 80] is a good example of this. The improvements to such systems may not be limited to changing the production rules created by induction, but may also include improvements in the control scheme, which governs rule selection and execution [Hayes-Roth, et. al. 80].

The use of time-dependent data in artificial intelligence is just in its infancy. For example, RX [Blum 80] is aimed at

finding methods to discover knowledge about the time course and treatment of long term diseases. The methods sought must be able to handle large amounts of data, which can be both highly variable and include missing (unknown) values. Also a formal logic specifically designed to describe time-dependent processed has been recently developed [McDermott 81].

Time-dependent information is also starting to be used in expert systems, such as in the ventilator management program VM [Fagan 80] for helping iron lung patients and ONCOCIN [Scott et. al. HPP working paper] for cancer therapy. There will be an increasing need in the future for powerful inductive programs to create and modify the knowledge bases for these and other expert systems [Michie 80].

For descriptions of active research in all areas of machine learning, there is a special issue of SIGART [Mitchell, Carbonell and Michalski 81], which also contains an extensive bibliography. Also, "Heuristic Programming Project 1980" created by H. Nii provides general information on the numerous projects they are working on.

C. Black Cutworm Damage as an Example

One problem area where constructive induction on time dependent data is needed is predicting black cutworm (BCW) damage to corn fields. The variables describing the damage are of two types: static and dynamic. Static variables describe permanent nonchanging attributes of a field, nonchanging in the sense that for a given corn field the variable will only have one value during one growing season. For example, "Last Years Crop", "Permanent Vegetation on Field Border", "Field Slope" and "Fertilizer" are all static variables, if considered over one growing season. In contrast, dynamic variables can have more than one value during a single season. Also, associated with each value is a time or date when the value was recorded. So "Weed Species", "Weed Density" and "Cumulative Degree Days" are all dynamic variables. (Cumulative degree days and other terminology used in this paper are explained in the glossary.)

It is with the dynamic variables that constructive induction is most useful and necessary for this type of problem. The static variables already contain sufficient information and programs like AQ/11 are capable of handling them. The opposite is the case for dynamic variables. The value of a dynamic variable at an arbitrary time is in general of very little use for distinguishing

between decision classes. For example, the weediness of a corn field on an aribtrary date can not provide very much information for predicting BCW damage, but the weediness on planting date or the average weediness when BCW moths are laying eggs can give quite a bit of information for predicting BCW damage [Busching and Turpin 77, Sherrod 76].

The other problem with dynamic data is that programs like AQ/11 can only handle variables that are single valued. So to use dynamic data directly, a separate variable would have to be created for each value of a dynamic variable. Also, the times when the values were measured varies greatly from event to event in BCW data, so not only would each event have a large number of unknown values, but the value of a dynamic variable at a specific time might be known for only a small percent of the events, thus making the values of little use.

It should be noted that CONVART distinguishes the time variable from all the other dynamic variables. This is because time is assumed to always be the independent variable and the other dynamic variables as dependent. CONVART considers time on two scales: absolute and relative. First there is absolute time, which uses the actual values of time found in the raw data. For example, MINTIME is an operator that can return an absolute time:

the time at which a dynamic variable reached its minimum. Relative time uses the same scale as absolute, but with the origin shifted to a data dependent time. For example, MINTIME with respect to Planting_date, would calculate the number of days before or after Planting_date at which a dynamic variable reached its minimum. CONVART could be extended to consider normalized time, where not only is the origin shifted, but a scaling factor is used. This could be used, for example, to make meaningful comparisons of time between years of fast and slow corn development. The scaling factor could be determined from data such as cumulative degree days data (see glossary for further information).

It should be noted that problems which do not have time dependent data, but do have data dependent one independent linear variable could be given to CONVART with that independent variable mapped onto the time variable assumed to be present by CONVART. One conceivable area is interpretting oil well log data (which the LOGIN project [Nii and Feigenbaum 78] hopes to accomplish). In this problem variables are measured with respect to depth into an oil well, so the depth variable would be given to CONVART as a replacement for the time variable.

One other nonstandard feature of the BCW data is the presence of a shared data set. This is weather station data where each event applies to a number of nearby corn fields. The simplest way of dealing with this kind of data would be to just copy the appropriate weather station data into every event (every corn field) in the regular data set, but this is much too inefficient when one is dealing with large amounts of data. Therefore, CONVART was provided with a pointer mechanism so that each regular event (a corn field) could be connected to a single copy of the appropriate shared data event (from a weather station). Figure 1a. shows the structure of a single event, note that any number of regular events could point to a single shared data event and figure 1b gives an example of an event.

D. A Solution

Clearly some mechanism is needed to convert the dynamic data into a few relevant static variables that can be directly useable by programs like AQ/11, which take the specific descriptions provided by each event and converts them into general descriptions for each class of events. (Where a class might be a soybean disease or a specific amount of BCW damage.) CONVART aims to be just that, a mechanism for CONstructing VARiables that depend on

Specific Data | Shared Data

s1 s2 s3 ...si	s1 s2 s3 ...sj
t1: d1 d2 d3 ...dk	t1: d1 d2 d3 ...d1
t2: d1 d2 d3 ...dk	t2: d1 d2 d3 ...d1
t3: d1 d2 d3 ...dk	t3: d1 d2 d3 ...d1
.	.
.	.
.	.
tm: d1 d2 d3 ...dk	tn: d1 d2 d3 ...d1

s - the value of a static variable
t - the time at which a set of dynamic values were measured
d - the value of a dynamic variable

Figure 1a. The Structure of a Single Event in CONVART

Specific Data

damage	crop	tillage	risk index	date	weeds	weed count
low	corn	plow	11	Mar 15	grass	few
				Mar 23	jimson	some
				June 3	?	few

Shared Data

stationname	date	temperature
Urbana	Feb 1	32
	Mar 1	43
	Apr 1	66
	May 1	72
	Jun 1	88

Figure 1b. An example of an Event

Time (hence the name CONVART), i.e. it creates time-dependent derived variables that may be useful, tests their relevancy and then produces output in a format directly useable by AQ/11. Also, for CONVART to be able to successfully attack the BCW problem, it must be able to handle real world data. This implies it must be efficient enough to handle large amounts of data and that it must be reasonably immune to noisy data. The pair CONVART and AQ/11 are part of a loop (figure 2.) for incremental learning. The elements of this loop are similar to the elements of the general learning system of Buchanan et. al. [77].

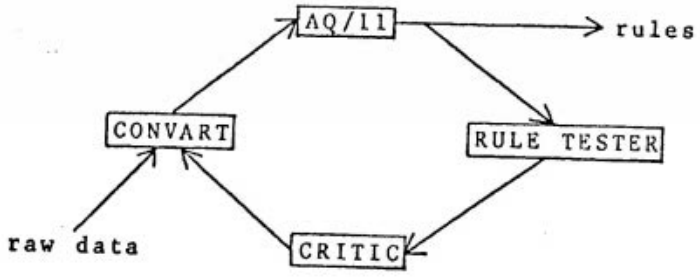

Figure 2. The Incremental Learning Loop for CONVART

After AQ/11 has produced rules using the data prepared by CONVART, the Rule Tester is invoked to compile statistics on rule performance. The Rule Tester can be thought of as an expert system run in batch mode using many known examples. Next, the critic is used to analyze the results of the rule tester to see if the rules need improvement. If not the loop ends, otherwise CONVART and AQ/11 can be rerun with new directives. This loop is described in more detail in section II. D. In this paper, operator refers to specific ways of creating new variables (like average or minimum). While derived variable refers to specific instanciations of these combinations (like "Average Rainfall" see figure 3.).

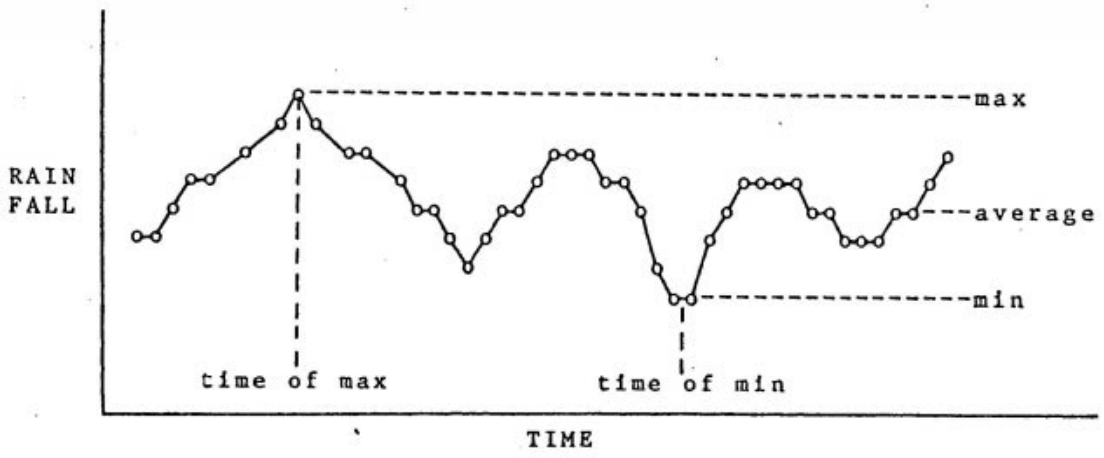

Figure 3. Rainfall vs. Time

II. METHOD

The problem that CONVART is attacking is certainly a very difficult one. The combinatorics of constructive induction are such that in general, it is impossible to search for an optimal set of derived variables for a given problem. Therefore, only a satisficing set can be searched for. The method used is quite different than that used in Bacon.4 [Langley, Bradshaw and Simmon 80] to combine variables because CONVART is designed to handle data which are not tightly controlled, i.e. in which the independent variables are not controlled by the inductive system. For example, when Bacon.4 "discovered" Ohms law (V=IR) it used data where one, then another of the variables was held constant while another was varied in a controlled way. CONVART on the other hand is aimed at problems where the data is not or cannot be controlled tightly. In the BCW example it is either too costly or impossible to control all the variables that way and so such data is not available.

Since CONVART is producing variables and not rules, it could be adapted to any number of inductive systems using any particular logic system. This generality is one reason for making CONVART a separate program from the inductive machinery of AQ/11. Since CONVART is designed to be directly linked to AQ/11, the logic

system being used for this research is a variable valued logic, VL_1 [Michalski 74]. The facilities in CONVART for attacking problems like BCW damage are:

A. Real World Constraints (suggestions): These are used to suggest to CONVART which operators should be used on which variables. Thus limiting the seach space of possible operator-variable pairs.

B. Operators and Operator Types: this is a carefully selected general purpose set. They create the values of derived variables and thus are the backbone of CONVART.

C. Relevancy Measures: These are used to decide whether or not a newly created derived variable could potentially help discriminate one decision class from the others.

D. Incremental Learning: CONVART was designed to be a part of a closed loop. The loop allows gradual improvement and correction of the inadequacies and inaccuracies in the system.

E. Clustering Raw Derived Values: AQ/11 needs variables with domains of limited size, so the values of numeric derived variables are grouped into intervals for AQ/11.

(CONVART was implemented in Pascal on a Cyber 175.) Let us look at each of these features in detail:

A. Real World Constraints

Since it is impossible to calculate all of the combinations of all of the variables, a reasonable method of selecting which combinations to calculate is needed. A reasonable method must allow the use of real world knowledge, ranging from specific to general information and simultaneously it must control the combinatorial explosion as much as possible.

CONVART's solution is to have two sets of suggestions: positive and negative. The positive suggestions allow the user to specify which derived variables or which types of derived variables to calculate. The negative suggestions specify which not to calculate. Suggestions (either positive or negative) can be of several types, corresponding to levels of control.

Level 1 is the most specific. It provides a very tight control on computation and allows specific real world knowledge to be used. Level 1 suggestions are of the form:

specific_operator< list specific variables >

Level 1 suggestions specify to derive or not derive a particular derived variable for a completely enumerated list of variables. For example,

AVG< weed_density >

specifies that the average value of "weed_density" should be (or should not) be derived. So if the user knows exactly which derived variables he or she needs, then level 1 suggestions are completely adequate.

Note that operators like AVG above are vector valued and so if more than one variable is given to them then they may produce more than one derived variable. Since some operators combine the values of several variables into one derived variable, there is not a one to one correspondence between number of variables given to an operator and the number of derived variables produced. For example, the operator DYNATSTAT (the value of a dynamic variable at a time given by a static variable) must be given a dynamic, a static and the time variables to calculate the value of one derived variable.

If users always knew exactly which derived variables were needed there would not be a great need for the convenience of a program like CONVART. Therefore, more general types of suggestions are allowed. In level 2, the user still supplies a specific operator to use, and a list of variables, but a control number is also given. For example:

SLOPE< 2, rainfall, temperature, degree_days >

specifies that two relevant derived variables should be searched for. The operator used to calculate these derived variables is the slope operator, which calculates the slope of a dynamic variable with respect to time. Two relevant variables means that CONVART will calculate the slope of enough variables in the list until it finds two that pass the relevancy test (see section II. C.) or until it exhausts the list. So if the slope of both rainfall and temperature appear to be useful (relevant) then CONVART stops there, otherwise it calculates the slope of "degree_days" as well. In either case only those derived variables that pass the relevancy tests are passed on to the output and AQ/11. (For Negative suggestions the control number, if present, is ignored.)

Notice that CONVART is only looking for two derived variables that are adequate, not the two that are the "best". There are two reasons for this: first CONVART was designed to find satisficing

not optimal solutions and second (and more importantly) the current relevancy measures are still rather crude and so the extra calculations needed to find the two derived variables that have the most optimal relevancy measures are not nesesarrily the "best" derived variables. (The best derived variables are those that are needed to create the most optimal rules, optimal with regards to comprehensiblity, correctness, simplicity, completness or even some problem specific criteria.)

In the next level, level 3, the list of specific variables is replaced by a variable type such as nominal, linear or structured. For example:

SLOPE< 2, linear >

specifies that slope should be calculated for linear variables until 2 good slopes are found. If rainfall temperature and degree_days (from the last example) were all of the linear dynamic variables then

SLOPE< 2, linear >

SLOPE< 2, rainfall, temperature, degree_days >

would be equivalent if those variables were defined in that order (with possibly some nonlinear variable definitions between them).

Level 4 suggestions allow a user to specify that a type of operator should be tried on a specific list of variables. For example:

D< 3, weed_density >

where D is the set of one arguement operators that only use the values of a single dynamic variable and nothing else like the time variable or a static variable. Average, maximum and minimum are D operators. The other operator types are given in the next section. The above suggestion specifies that 3 relevant derived variables should be found that only need the values of weeddensity for their calculation.

Level 5 is the logical combination of levels 3 and 4: an operator type, an optional control number and a variable type. For example:

DT< real >

specifies that all derived variables using two variables: a single dynamic variable and the time variable (like the operators, slope and intercept) should be (or not be calculated) for all the real valued dynamic variables.

If no suggestions are given then a default list is used (See section III. c. on program features for details). Level four and five suggestions have not been implemented and will not really be needed until data sets with a large number of dynamic variables are analyzed. Also, if a conflict arises between the positive and the negative suggestions (i.e. they both specify either directly

or indirectly the same derived variable) the conflict is resolved in favor of the negative suggestion, as their reliablity tends to be much greater. This greater reliablity stems from the origin of the suggestions. Positive suggestions are initially educated guesses, which become refined, while negative suggestions are generally given only when a user or CONVART is reasonably certain that the derived variable is irrelevant. CONVART's ablity to provide negative suggestions is explained in section II. C. on relevancy tests.

B. Operators and Operator Types

For CONVART to be a powerful tool it must have a carefully selected set of operators that it can use. This set must be able to create derived variables that are comprehensible and that can express a wide range of concepts. A derived variable that is some obscure mathematical formulation with no real world interpretation, is of no practical use. This is because any system that would use such a derived variable (like an expert system) would appear to be a black box to user. Experience shows such systems do not get used very much because they can not explain their results in a meaningful way to a user.

Another way of putting it is that the derived variables must adhere to the comprehensiblity postulate [Michalski 80] which applied to this problem states that the set of operators should describe the dynamic data in a conceptual and understandable way and not merely in a numeric way. For example, if the data was resonably linear then its slope and y intercept would characterize it numerically. However, if it was only the x intercept that was relevant to the problem at hand, then rules using slope and y intercept would be needlessly complex and not very understandable. Also, the slope and either x or y intercept, which characterize the data numerically, may not be capable of characterizing the data conceptually for some problems.

So although statistical methods must not be overlooked, they can not be relied on solely. There are two ways statistical methods can be made more comprehensible and conceptually more powerful. The easiest way is to give each subrange of values of a statistical measure a mnemonic name. For example, the select :

[relative constancy of weediness = 0.98]

tells you nothing unless you know and are familiar with the method of calculating relative constancy. If the range of its values is [0..1] then this is a high value, but if the range is [0..100] then this is a low value. On the other hand the selector

[relative constancy of weediness = very constant]

instantly tells anyone how constant weediness was. Now of course 0.98 is much more accurate than "very constant", but for many problems such descriptions are sufficiently precise. When the precise definition is needed it can easily be looked up in the user's guide for CONVART. Also, if greater precision is needed then the range of values of a derived variable could be split up into more, smaller intervals (with perhaps some loss of comprehensiblity if good mnemonics can not be found).

The second and much harder way to add to the conceptual power and comrehensiblity of statistical measures is to change and/or augment them so that they derive numbers of more conceptual content. For example, instead of trying to fit a single line to a set of data, one could look for two lines, each of which fits the data for one interval. Then not only can the slope and intercept of each segment be returned, but also the time at which the data switched from one line to the other, the length of time the data fit each line, the average value of each of those time intervals and so on. This in turn could be generalized to the search for an arbitrary piecewise linear function that fits the data.

One other constraint on operators is that they must be able to handle real world data. This means that they must be efficient, allow data with unknown values and be as immune to noise as

possible. Noise immunity is not just a function of the definition of an operator, but is also dependent on the algorithm used to calculate it. For example, to calculate the change in slope for ideal data only the first two and the last two points (of the list of values for an event) are needed. Instead a much more sophisticated method (described below) is used in CONVART.

Since the list of operators is constantly growing, a means of organizing or classifying them is needed. If CONVART is to remain a general purpose utility program, the classification schmeme cannot rely on problem specific groupings, but instead should be based on the structure of the list of derived variables themselves. One such structure is based on the number and type of arguements that an operator needs for its calculation. "Average Weediness" and "Minimum Weediness" only need the values of a single dynamic variable to be calculated, while a derived variable like "Weediness on Planting Date" needs a static variable, a dynamic variable and the time variable for its calculation. Since average and minimum are much more similar to each other than an operator which uses the value of a dynamic variable at a time specified by a static variable, they are grouped together.

A shorthand notation has been developed to name each of the possible types of operators. The type name is a string of

letters, one for each variable needed for the calculations. S stands for static variables, D for dynamic variables and T for the time variable. So SDD would be a operator type that needs a single static variable and two dynamic variables (but not the time variable) for its calculation. The types currently implemented are:

. D - those using only a single dynamic variable for their calculation

. DT - those using only a single dynamic variable and the time variable

. DD - those using two dynamic variables

. SDT - those using three variables: 1 dynamic, 1 static and the time variables

The following is the current list of derived variables for each group.

D Operators

. MAX	The maximum value of a variable in one event.
. MIN	The minimum value of a variable in one event.
. AVG	The average value of a variable in one event.
. RELCONST	The relative constancy of a variable. Calculated as (2 * SD) / RANGE, where SD is the standard deviation of the the values of a variable for one event and range is the size of the domain of the variable (i.e. for numeric variables this is the difference between the largest and the smallest values in the domain of the variable). This derived variable has the predefined mnemonic values: "extremely nonconstant", "not very constant", "somewhat constant", "very constant", and "extremely constant".
. MOSTCOM	The most common value of a nominal variable.
. LEASTCOM	The least common value of a nominal variable.

. NBRSAME — The length of the longest string of equal consecutive values from one event for a nominal variable.

. COUNTCHANG — The length of the longest string of values (from one event) where no two consecutive values are equal. (for nominal variables)

DT Operators

. TIMMIN
. TIMMAX — The time when a variable attained its minimum (TIMMIN) or maximum (TIMMAX) value. In the case where there is not a unique time then either the first or last time can be returned. (Currently only absolute time is supported, but relative time could be easily added.)

. SLOPE
. YINTERCEPT
. XINTERCEPT — The variables's values are fitted to a line using linear regression with time as the x axis. The intercept can be either the y or the x intercept.

. CORCOEFF — The correlation coefficient, a standard measure of how well the data fits a line. This derived variable also has a predefined domain similar to relative constancy's.

. CHNGSLOP2 — Change in slope, an attempt is made to fit two lines to the data. (see figure 4.)

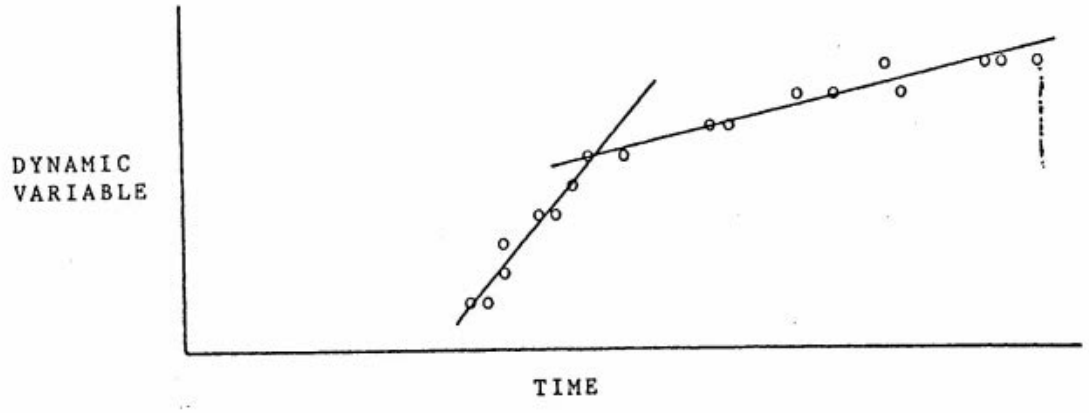

Figure 4. Change in Slope

To calculate it, the first few and the last few data values are used to calculate approximate lines, then their intersection point is found. Since this is only an approximatation the points immediately around the intersection point are ignored and the rest of the points are used to calculate the

final two slopes and intercepts. (Although an algorithm for this has been developed it has not been implemented yet.) The values returned are : the slopes and intercepts of each line, the difference (change) in slope or intercept and the values of the variable and time at the intersection point. If insufficient data is present then all of those statistics will have the value of unknown. (Eventually the 2 will be replaced with a parameter k so that data which fits k line segments could be analyzed.)

. CHNGLEV

A change in level is defined as a set of values that can be split into two intervals during each of which the values are reasonably constant and between the intervals the variable changes significantly relative to the constancy within each interval. (This operator has not been implemented at this time.) The values returned are the time and magnitude of the change and the average value before and after (see figure 5.).

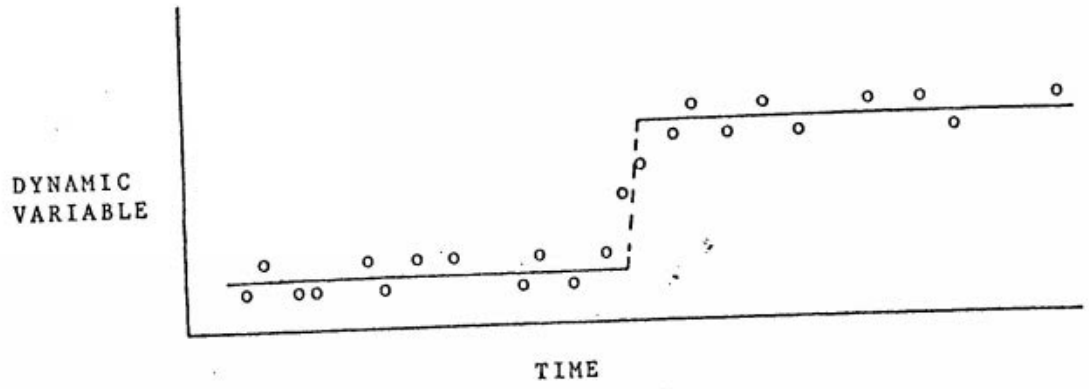

Figure 5. Change in Level

. TIMMAX-MIN
. TIMMIN-MAX

The length of time between the minimum and the maximum. For the operator TIMMAX-MIN this time is negative if the maximum is first (instead of the minimum). If the reverse is desired then the operator TIMMIN-MAX is used. (Currently only absolute time is supported, but relative time could be easily added.)

DD Operators

. SLOPE
. YINTERCEPT
. XINTERCEPT

Same as the D operator except that a second dynamic variable is used in place of the time variable.

SDT Operators

. DYNATSTAT — A static variable with the same units as the dynamic time variable is needed. It is used to find (or interpolate if need be) the value of a dynamic variable. For example, weediness on planting date is such a derived variable.

. TIMEDYN=STAT — The time when a dynamic variable equals a static. This is similar to the above derived variable except that a static variable with the same domain as a dynamic variable is needed. (This operator is not implemented and could not be used with the BCW data, but would add to the generality of CONVART.)

Suggestions for derived variables using shared data are given separately from the suggestion using only the regular data and since all of the above operators can be used with shared data, there is no need for a separate section of operator types for shared data operators.

C. Relevancy Measures

After CONVART has selected a derived variable and done the calculations for it, it still is not done. It must help AQ/11 control its computation and memory requirements by eliminating those derived variables that appear to be irrelevant. Some derived variables are obviously irrelevant. The extreme worst case is when the value of a derived variable is the same for all the events (assuming their is more than one decision class in the data, a derived variable that is constant cannot possibly be of any use in distinguishing between those classes.) The other extreme is the ideal derived variable that can distinguish between the decision classes all by itself. The latter rarely happpens, but the former can happen often with some types of data. For example, "cumulative degree days" is a cumulative sum so its minimum value is always 0 and the time of the first minimum is the same day each year, namely January 1 (that is if one is looking for the first instead of the last day with the minimum "cumulative degree days"). So any relevancy test must be able to handle that extreme case (of course the user could avoid the needless calculation of minimum "cumulative degree days" with a negative suggestion).

The first relevancy test in CONVART was designed to eliminate the worst derived variables. It is basically a test to see if the values for a derived variable are predominantly one value. This can be found by a simple calculation, namely go through the list of values (of the derived variable) and incrementing a counter everytime the next value is significantly different from the current one. Then if this count is too low the variable is declared irrelevant and discarded. Two values are significantly different if after all of the values are grouped into intervals, (see section II. E. on clustering) the two values are in different intervals.

Although this calculation is rather simple, it does make sure that the derived variable has at least two different values and that both occur with some minimum frequency and. In practice this simple test is actually useful. This is because when an irrelvant derived variable gets past CONVART, it goes to AQ/11 which will probably not use it or only use it to cover a few events and then if the rules from AQ/11 are not good enough CONVART can be rerun with a new set of negative suggestions for the irrelvant variables (more on this in the next section). The biggest problem with this measure is that it cannot distinguish between a truely relevant derived variable and one that has purely random values.

Since relevancy is so important for producing high quality rules, one fixed relevancy test is not sufficient. Ideally a whole list of tests should be available, but that is another project in itself. To start this list CONVART has one other, more sophisticated relevancy test. Instead of being based on eliminating the worst case, it is based on retaining the ideal case. Recall the ideal case is when each value of a derived variable is associated with only one decision class. If one calculates the average number of decision classes per value (of the derived variable) one gets a 1.0 in the ideal case and a number equal to the number of decision classes in the worst case. So, if the average number of decision classes per value is too high the derived variable is potentially irrelevant and can be discarded.

The minimum number of decision classes per value may also be a valuable measure, although this has not been investigated. If the minimum is a one then at least one value of the variable is associated with only one decision class. This would mean that the variable could distinguish one decision class from all of the others.

Although conceptually the average number of decision classes is a more sophisticated test, it is also more dangerous as

genuinely relevant derived variables can be discarded. This can happen when a derived variable is not independent of the other variables. For example, consider a case with two decision classes A and B, which are described by the rules:

[x=1][m=high] v [x=2][m=low] ::> [class=A]

[x=2][m=high] v [x=1][m=low] ::> [class=B]

Given a sufficient sampling of events for both of these classes, the variable m would have the worst possible value for the average number of decision classes per value: 2, but M is clearly a relevant variable.

The effect of noisey data is exactly opposite for the two methods. Since noise will tend to scatter values, the count of different values will tend to be higher making a derived variable appear more relevant. However the same scattering of values would tend to make the average number of decision classes also higher, thus making a derived variable appear less relevant. Perhaps combining these measures could result in a more noise immune measure of relevancy, although this has not been investigated.

D. Incremental Learning

In general, the only completely reliable way to accurately test for relevant variables like m, whose relevancy depends on their interrelations with other variables, is to try to produce rules using them, but producing rules is the purpose of inductive machines like AQ/11, not CONVART. To get around this problem incremental learning is needed. This loop is similar to the incremental learning loop in LEX [Mitchell 80]. The basic loop for incremental learning with CONVART is shown in figure 6.

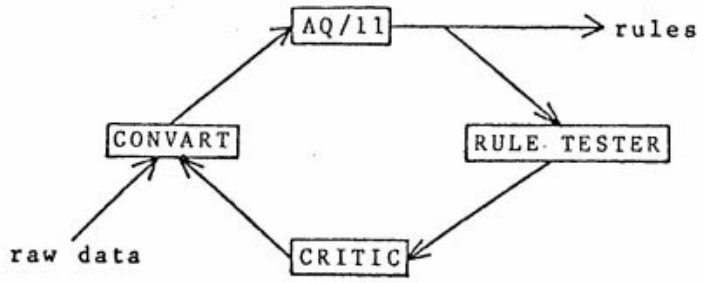

Figure 6. The Incremental Learning Loop for CONVART

The user supplies raw data and suggestions to CONVART which calculates some potentially useful derived variables and prepares an input file for AQ/11. AQ/11 then uses that data to produce the first version of a set of rules. The rules are then tested by an expert system capable of checking production rules against known examles, like the batch version of PLANT/ds [Michalski, Chilausky 80], which would be using data transformed by CONVART. The statistics for rule performance are then analyzed by the critic (which is for now human), that decides whether the rules are satisfactory or not. If so then the loop ends, otherwise a new set of suggestions is created for CONVART and the loop is repeated with the possible addition of new data. Errors due to inaccurate data and sampling errors could also be found by the critic if it was given enough real world knowledge.

Part of this new set of suggestions has already been created by CONVART itself. When a derived variable or list of derived variables does not pass the relevancy test, CONVART automatically derives negative suggestions. The critic can also derive negative suggestions by checking to see which derived variables were not used or which were used to cover only a few events. The critic could also adjust some of the parameters to CONVART or AQ/11. Adjusting parameters is not a trivial problem. Simple hill climbing algoritms do not always work. Personel experience with

AQ/11 seems to indicate that a successful algoritm might have to use trial and error as one of its methods.

E. Clustering for AQ/11

For CONVART to connect directly to AQ/11 it must create variables whose domains are relatively small. (The domain size limit of the pascal implementation of AQ/11 is 58, but quite often significantly smaller domains are completely adequate and require less computation time.) Since many of the derived variables that CONVART creates are arbitrary numbers, a means of splitting up the number line into a few intervals, (which will be the domain elements) is needed.

To accomplish this a nearest neighbor method is employed. To descritize an arbitrary list of numbers into a finite number of levels (domain elements) the following algorithm was used:

1. Compute each numbers nearest neighbor
2. Create an ordered list of nearest neighbors (with no pair of neighbors repeated in the list)
3. REPEAT

    3.1 Remove the smallest pair from the list (these two points are now in the same level)

    3.2 Check to see if one level was extended or two levels were merged

3.3 Add to the list a nearest neighbor pair for the point which is the average of the pair removed from the list (rememeber to keep the list ordered)

UNTIL number of levels < threshold(data size)

Basically this algoritm iteratively puts data points that are near each other in the same interval (level) and repeats until the number of intervals is below a threshold which is a function of the data set size (number of events in the data). This is shown graphically in figure 7.

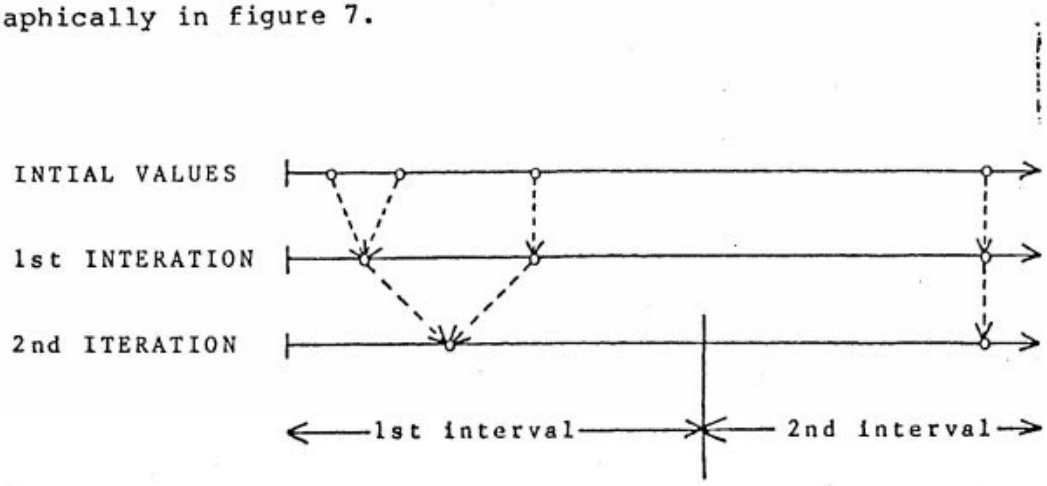

Figure 7. The Clustering of 4 Values into 2 Intervals

There are also control parameters which can give a more direct control of the domain size if desired. It should also be noted that bad values (noise) will tend to be put in separate intervals, thus making them more visible.

F. Summary of Method

CONVART is a program that derives variables from data that is dependent on time. It has a suggestion taking facility that allows it to accept real world knowledge and use that knowledge to control its calculations. The calculations it does do are aimed at describing the data in a conceptual versus a purely numerical way. These conceptual calculations are then tested to see if anything potentially meaningful has been discovered and if so the results after clustering are passed on to the induction machinery of AQ/11. The main loop in CONVART that accomplishes this is shown in figure 8.

```
while (more suggestions) and (nbr good suggestions < threshold) do

        get next suggestion (one operator and a variable)

        if derived-variable is numeric
                then cluster its values

        if derived-variable is relevant
                then
                   increment nbr good suggestions count
                  increment specific suggestion counter

endwhile
```

Figure 8. Main Calculation Loop in Deriver

If AQ/11's results prove to be inadequate then the suggestion facility provides a natural mechanism for incremental improvement of the results. These tools allow CONVART and AQ/11 to tackle problems together that would be intractable for AQ/11 alone.

III. RESULTS

A. A Simple Example

As an example to the kind of problems that CONVART was designed for, a hypothetical problem from molecular biology was created. A Professor had been working on isolating a recently discovered enzyme, but his technique was still experimental and so his preparations were not always biologically active. To avoid wasting a great deal of time and money on experiments with inactive enzyme preparations he needed a reliable and cheap method for testing his preparations for biological activity. The only cheap method he knew of for testing this activity was an unreliable colorimetric reaction (a chemical reaction whose progress is marked by a gradual change of color in the reacting solution). The rate of such reactions is directly related to the rate of change in color of the solution and the degree to which the reaction went to completion is directly related to the final

color of the solution. Unfortunately, in this case neither of these two measures were completely adequate to accurately predict biological activity, except when the preparation was either very active or very inactive. So, CONVART was called in to see if it and AQ/11 could find a pattern in the data from the colorimetric reaction that could accurately predict biological activity.

Each event in the data consists of a list of absorbances (a quantitative measure of color, see the glossary) and the times when the reading were recorded, also there was the name of the technician that made the enzyme preparation and whether or not the preparation was active (see figure 9.). The output from CONVART for this problem is listed in appendix A.

First CONVART was run with no suggestions; so only the default operators were applied (average, slope, y intercept, max, min, and the time when the maximum and minum occrred). AQ/11 was then run using CONVARTS output as data. (Derived-variables whose values are absorbances have been given the mnemonic value set: low, medium-low, medium-high, and high. The precise definition of these values depends on the particular derived variable.) The resulting rules were: (The values for y intercept of absorbance, which are pure numbers with no units, where intentionally left as numbers.)

Static Data		Dynamic Data	
preparation	technician	minutes	absorbance
active	Stepp	0.5	0.2
		1.0	1.1
		1.5	2.3
		2	3.5
		2.5	4.4
		3	5.6

(absorbance has no units.)

Figure 9. One Event for the Enzyme Activity Example

```
[maximum absorbance ≥ medium-high]
[time of maximum absorbance < 6 min.]
                    V
[average absorbance = high]
[slope of absorbance = low]

        ::>  [preparation = active]


[average absorbance = high]
[y intercept of absorbance > 0.522]
                    V
[average absorbance < medium]
[y intercept of absorbance < -0.81]
                    V
[average absorbance = high]
[y intercept of absorbance < -0.35]

        ::>  [preparation = inactive]
```

To obtain these results CONVART used 0.37 seconds of central

processor time (cp secs.) and 18k sixty bit words of central memory (cm) and AQ/11 used 0.26 cp secs. and 32k cm. These and all other results in this paper were run on a Cyber 175 computer.

Although these rules are not bad, the biologists were not completely satisfied, because the negative absorbances calculated for the "y intercept of absorbance" had no meaning to them. Therefore CONVART was run again with a suggestion to calculate the x intercept of the line for absorbance versus time. The x intercept corresponds to the time when there was zero absorbance which is also the time when the reaction actually started (as opposed to time 0 which is when the measurement of time began in the experiment). The time of the start of the reaction did not correspond to the time when the clock was started because the reacting solutions had to be incubated for a fixed period of time before inserting them into the spectrometer (a device that measures absorbances) and so the reaction might start before or after the incubation period was over. (Biology, like artificial intelligence, is not an exact science.) The results this time were:

[maximum absorbance $\geq$ medium high]
[time of maximum absorbance < 6 min.]
V
[x intercept of absorbance > -0.58 min.]
[slope of absorbance < 1.14]

::> [preparation = active]

[x intercept of absorbance < -0.58 min.]
[slope of absorbance < medium-high]
V
[x intercept of absorbance > 0.32 min.]
[slope of absorbance < medium-high]

::> [preparation = inactive]

(the processor time and central memory used were similar to those for the first set of rules.)

These rules were much more pleasing to the biologists as there was a natural interpretation for all of the derived variables: maximum absorbance relates to how close to completion the reaction went, the time of this maximum is the time it took for the reaction to occur, the x intercept of absorbance is the time when the reaction actually started (and not when the clock was started) and the slope of absorbance is the rate (how fast) the reaction occurred. However, there was still one problem: if a set of measurements for a preparation yielded an x intercept over 0.32 min. and a low slope then the second complex of both rules would be satisfied and so the preparations activity would not be known.

This problem was easily solved by rerunning just AQ/11 (without rerunning CONVART) with the mode set for disjoint covers, instead of intersecting covers. (When AQ/11's mode is set for intersecting covers the complexes for more than one rule are allowed to cover don't care events (i.e. events not in the data

set), while the opposite is true if the mode is set for disjoint (non overlapping covers). With the change in mode the final set of rules were created:

[slope of absorbance $\geq$ medium-high]
V
[x intercept of absorbance = -0.58 to 0.32 min.]

::> [preparation = active]

[x intercept of absorbance < -0.58 min.]
[slope of absorbance < medium-high]
V
[x intercept of absorbance > 0.32 min.]
[slope of absorbance < medium-high]

::> [preparation = inactive]

This time AQ/11 only used 0.18 cp secs. (as opposed to 0.26 for the first set of rules). The central memory needed was about the same.

These rules are clearly superior to the previous ones with respect to simplicity: only two derived variables, slope and x intercept, are needed to determine the biological activity of a preparation. The technicians were also happy as none of them were associated with the inactive preparations.

These rules have a simple interpretation: if the rate of the reaction is fast enough (slope > medium-high) or if the reaction started during a prescribed time period (x intercept = -0.58 to

0.32 min.) then the preparation was active, otherwise the preparation was not active. The creatation of these rules shows how CONVART and AQ/11 can incrementally generate and refine a set of production rules, although admittedly this was a simple problem of this process.

B. Black Cutworm Damage Examples

Predicting black cutworm (BCW) damage is a very difficult problem. Expert entomologists have yet to find a satisfactory solution. [Busching and Turpin 77]. The expert's models are still very sketchy. They are not sure that they know what all of the relevant descriptors are and quantitative analysis of their ideas has just begun.

In the available data there are both static and dynamic variables for each corn field and a single dynamic variable from weather data (shared data). The static variables describing corn fields are:

Field History Variables

. Previous Crop - the crop grown in the field the year before. Its values have been grouped into three categories: corn, soybeans and other.

. Cutworm History - describes past damage to the field (Ideally this would be a dynamic variable with values for the past several years, but even as a static value its value was seldom recorded.

Field Characteristic Variables

. Permanent Vegetation - indicates the presence or absence of permanent plant growth adjacent to the field.

. Border Water - indicates the presence or absence of a body of water adjacent to the field.

. Direction of Slope - The direction (north, south, etc.) of the slant of the field.

. Surface - describes the fields surface contours. Eg. flat or rolling.

Field Treatment Variables

. Fall Tillage - the type of plowing that was done to the field at the end of the previous growing season.

. Spring Tillage - the type of plowing that was done to the field just before the beginning of the current growing season.

. Spring Tillage Date - the time of spring tillage. This variable was not available for any of the events.

. Manure - indicates whether or not manure (animal feces) was applied to the field.

. Fertilizer - specifies which chemical fertilizer (if any) was used. Its values were grouped into three types: none, anhydrous (ammonia), and other.

. Insecticide - specifies what insecticides (if any) were used. The groups for its values are: none, seed treatment, old soil and new soil.

Corn Planting Variables

. Planting Date - the julian date when the corn was planted.

. Planting Rate - the measure of how many seeds were planted in a given area.

Miscellaneous Variables

. Risk Index - this is a derived variable that the entomologists supplied. Its calculation is based on a model for determining how well synchronized the developments are of the black cutworm and corn. .

. Moth Flight - the date on which the maximum number of moths where laying eggs in the fields. This can be estimated using counts from scented moth traps placed in the fields. This variable was not available for almost all of the data.

The dynamic variables for describing corn fields are:

Weed Variables

. Weed Species - gives a list of the specific types of weeds that were present over the course of the growing season.

. Weed Density - gives a list of the number of plants for each of the weedspecies given in weed species.

Developmental Variables

. Corn Height - records how tall the average corn plant was on a particular date. This was not available for most of the data.

. BCW Population - a count of the number of BCW larvae present. This was not present in almost all of the data

The single shared variable was:

A Weather Variable

. Cumulative Degree Days - this is weather station data, which provides a measure for plant and animal development rates. (see the glossary for further details).

The first observation that can be made is that there are not many dynamic variables available. This limits (to some extent) the number of derived variables that CONVART can construct. However, the real limitation was the amount of dynamic data in most of the data. Remember that dynamic variables can have any number of values for a single event and in general, as the number of values goes up so does CONVART's ablity to accurately characterize an event. Unfortunately, the average number of values per event for weed species and weed density was less than three values each and the maximum number was six for all data recorded before 1980. The 1980 data was significantly better in this respect, but there were only 24 events available from 1980 as opposed to well over 300 events from previous years (1976-1978).

Inorder to get reasonable results, data that has a sufficient amount of information in it is needed. The small 1980 data set meets this requirement and the results from analyzing it will be presented later in the section: An Example With Real World Data. However, since there was so much data from the previous years, an attemp was made to analyze it. After initial attempts failed, it was realized that the data did not contain sufficient information. Therefore, some information was artifically added to the data from one year to see how CONVART would do on a larger set of data.

An Idealized Example

The data used in this example was created by starting with the actual data set for 1977. This data had over 20% of its values missing, and the holes in the data were filled in with values that were consistent with the models of the entomologists. The 53 events from 1977 were chosen because the data from all of the other years (except 1980) had an even higher percentage of missing values. None of the values that were already in the data set were changed, so the data is still about 80% real data. Also, no new variables were added. See appendix B. to see output from CONVART for this data.

Besides the default derived variables several other derived variables were generated. The list includes:

- most common group of weed species
- least common group of weed species
- number equal consecutive values of weed species
- weed density at planting date
- cumulative degree days at planting date

These derived variables and the defaults (such as average and maximum) were calculated in 6.0 cp seconds and required 204k words of memory. (the size of these numbers is due to the large number of dynamic values for cdd (181) for each event.) The final set of rules were calculated by AQ/11 in 1.6 cp seconds and required 100k words of memory. Two decision classes were used: damage predicted

or not predicted, where damage is when 25% or more of a corn field was cut (damaged) by the black cutworm larvae.

The idealized rules were created fairly quickly as known regularities had been added to the data. The final rules for the idealized data set are:

```
[risk index = 10 to 19]
[maximum weed density = low]
                V
[most commmon weed species group
                ≠ other broadleaf or weedkill]
[average weed density = low]
                V
[planting date < 119]
[most common weed species group ≠ weedkill]
[cumulative degree days x intercept = 43.5 to 44.3]
                V
[fall tillage = none, plow]
[most common weed species group ≠ other broadleaf]
[weed density on planting date = low]
[average weeddensity = low][minimum weed density = low]

        ::>  [damage = less than 25% cut predicted]


[maximum weed densityity = medium to high]
([average weed density = medium to high]
([cumulative degree days x intercept > 44.35]
                V
[planting date > April 30])
                V
[most common weed species group = other broadleaf]
([cumulative degree days x intercep < 42.9]
                V
[planting date > April 30])
                V
[most common weed species group = weedkill]
[weed density on planting date = medium to high])
                V
[risk index > 20]([fall tillage = reduced]
[most common weed species = weedkill]
                V
[most common weed species = other boadleaf]
```

[cumulative degree days x intercept < 42.9])

::> [damage = over 25% cut predicted]

These rules created from partially real and partially hypothetical data, express ideas similar to those in the models of the experts (the entomologists). For example, "Risk Index" is low for no predicted damage and high when damage is predicted. Also, the "Weed Density" measures of maximum and average are always low for no damage and medium to high when damage is predicted, and later "Planting Dates" were associated with damage. The latter could be due to synchronization: by planting later one gives the cutworms more time to develop and they are larger and so can cause more damage when the corn is susceptible. Both weedkill and other broadleafs were associated with damage and could be due to the attractivines of low dense growth forms to egg laying moths. (weedkill are weeds that are in the process of dying from cultivation or herbicides so their value as a long term habitat for the developing larvae is questionable.)

An Example With Real Data

The data used in this examle was collected in Illinois by the section of Economic Entomology of the Natual History Survey. The 24 events from 1980 were analyzed without any changes. No missing values were filled in, although one approximation was made: the

cumulative degree days data for the ten events (farms) near the Rochelle weather station were missing; so the weather station data from Mount Carrol, which is reasonably nearby, was used.

Due to the prevalence of low damage events in this data, the dividing line between damage and no damage was lowered from 25% down to 5%. Otherwise, the variable definitions for this and the previous example were almost identical.

The "best" set of rules was obtained after five iterations of generating derived variables with CONVART and creating rules with AQ/11. The rule testing could not be done due to lack of data; so the critic (the author) based its analysis of the rules on two criteria: 1) brevity, the fewest number of complexes and selectors was desired and 2) semantics, rules that had a meaningful real world interpretation were sought. These rules are:

[count changes of weed species < 10]
[most common group of weed species = weedkill]
[time of first minimum weed density = before June 10th]
[cumulative degree days on planting date < 864.65]

V

[most common group of weed species ≠ weedkill]
[average weed density ≥ 42.1 weeds / square meter]

V

[premanent vegetation on field border ≥ 50%]
[cumulative degree days on planting date < 864.65]

:::> [damage = less than 5% cut predicted]

[time of first maximum weed denstiy = before April 5th or after June 7th]
[cumulative degree days on planting date ≥ 864.5]

V

[permanent vegetation on field border < 50%]
[count changes of weed species = 0 or 1 or 3 or
at least 10]

::> [damage = over 5% cut predicted]

These rules required 3.0 seconds and 40.5k of 60 bit words of memory for CONVART and 1.6 seconds and 33k words for AQ/11 on the Cyber 175.

There are a number of interesting concepts in these rules. The first complex covers 13 of the 16 no damage events. The first selector of this complex states that if one examines the list of weed species observered over time for one field, and a count is made of the number of times two successive values are different, then this count will be less than 10. This could mean either that the groups of weed species observed were mostly of one type or that fewer than 10 observations of weeds were made in the no damage fields. The next selector states that the most common (most often observed) group of weed species was not "weedkill". This means that the types of weeds present in a field are not important if those weeds are not dying ("weedkill" weeds are those that are in the process of dying). The third selector states that the first minimum weed density occurred before June 10th, and the last selector of the first complex states that the cummulative degree days on planting date is below the 864 mark. These last two selectors may be related to the synchronization (or lack thereof) of BCW larva and corn development.

The second complex describing no damage fields covered 2 events. It states that the most common group of weeds was any group except "weedkill" and that the average weed density was high. This is counter intuitive, since a large density of healthy (not dying) weeds could foster BCW larva development.

The last complex describing no damage fields states that there was a high percentage of permanent vegetation surrounding the field and the cumulative degree days on planting date was low. The high percentage of field border containing vegetation might lure the BCW moths away from the field and the low cdd measure could be interpreted as saying that not enough heat had flowed into the field to allow the BCW larva to develop fast enough to damage corn.

Both complexes describing damaged corn fields covered half of the eight events in that class. The first complex states that the first maximum weed density occurred either early or late in the season and that there was a high amount of cumulative degree days by the time when the corn was planted. The other complex states that there was a low percentage of the field border with permanent vegetation and that the number of times the weed species changed in the field was either low or high.

Since only a few events were analyzed, deep insights into this complex problem can not be expected. Still, there is one very striking (and to the author very satisfying) feature to these rules: permanent vegetation was the only static variable used in the rules and it was only used twice. All of the other variables were derived variables created by CONVART! This would seem to indicate that most of the static variables are not relevant to the problem. This also justifies the claim that the shortage of dynamic values in the larger data sets was the reason why their analysis was a failure.

Since this was just a preliminary analysis of the BCW problem, it is important to identify any shortcomings so that they may be corrected in the future. Most importantly, it appears that detailed histories of weed populations for a large set of data must be acquired, as all but one of the complexes in the rules used a variable derived from this data. Also, data for variables like moth flight dates and spring tillage dates, which were not available, could be quite useful; so it is recommended that such data be collected. Once this is done, more elaborate operators could be used to measure such values as relative constancy of weed density from moth flight date to planting date or the net increase in cumulative degree days for that same period. (At this time such operators have not been implemented, but they could be easily

added to CONVART.) Such enrichments to the data may make it possible for CONVART and AQ/11 to find some deep insights into the BCW problem.

C. CONVART Program Features

No matter how powerful CONVART is made, it still would not be a useful tool if it was hard to use. Therefore, a number of features have been added to make it pleasant and easy to use and to help explain what it has done. First for the naive user, there are defaults for all of the parameters which can be overridden as needed. The default suggestions are:

```
  Mostcom< nominal >
 Leastcom< nominal >
  Average< linear >
    Slope< linear >
Intercept< linear >
      Min< linear >
      Max< linear >
lstmintime< linear >
lstmaxtime< linear >
```

Since most of the work in using CONVART or AQ/11 is in file preparation, special attention was given to that. AQ/11, which is an older program, requires that one counts such quantities as the number of: variables, decision classes, levels per variable and several others. Clearly, a better way to interface with users is needed. CONVART was written with the philosophy that computers

are much better at counting than people and so they should do the counting whenever possible. Therefore, CONVART does not ask the user to count anything, and it does all the counting that is needed for AQ/11, including encoding all the values in the data as integral levels. Also, AQ/11 requires events that are sorted by decision class, while CONVART does not require this. Therefore CONVART sorts the events before writing them out. Not only does CONVART create the data file for AQ/11, but it also creates a trans file that AQ/11 uses to give names to the variables and their values. For the static variables this is simply a matter of copying the definitions given to CONVART, while for the derived variables the names must be created for the variables and sometimes for their values. To make numeric levels as readable as possible, scientific notation is only used for very large numbers and for reals, zero fractional parts are never printed.

The input to CONVART is all free format so the user never has to worry about columns or line boundaries. All events, suggestions and variable declarations start with the syntactic marker '!'. This allows the user to insert a title of any length at the start of any input file as long as it does not have any '!'s in it. As far as limits are concerned, CONVART has no inherent limit on the number of variables, the number of values for any variable, or the number of events except the limits

imposed by the physical machine. Also, all input is in human readable form versus the integer encoded data of AQ/11. CONVART also allows events with an arbitrary number of values for dynamic variables (and of course one never has to count how many there are).

If the data does have counts of how many dynamic values there are for each event, CONVART can handle that as well. All that is needed is to declare the count variable as a non-applicable variable (in the variable declaration file) and it will not be passed on to AQ/11. For that matter, any static variable can be declared non-applicable to prevent it from being passed on to AQ/11. Another non-standard variable type is the key variable type. This is used to tell CONVART which variables are to be used to link the regular data to the shared data via a correspondence table supplied by the user. (Key variable's values are not passed to AQ/11 either.) CONVART also supports the value of unknown for all variables including all derived variables.

When CONVART has completed its calculations it produces a data file for AQ/11, but since all the values in that file are encoded as integers, its hard to tell directly what CONVART has done. Therefore, a text version of the data file can be created by CONVART which looks like the input data file except that the

dynamic variable's values are replaced by derived variable's values, all the numeric data is rounded off to the level assigned to it (because that is the value that AQ/11 will be given) and the non-applicable and key variables are surrounded by square brackets to signify that they were not put in the data file for AQ/11.

There are still two other output files for CONVART, one is a debug file which can be filled with a variable amount of details and intermediate results; so all of CONVART's results can be checked easily. The other is an error message file to which CONVART prints messages when it finds anything is wrong, like it was given a derived variable name it does not know about.

Lastly, if a user wants to define his own derived variable, he can write an external pascal procedure that CONVART will call to do the calculations. This means that new derived variables can be added and debugged without changing or recompiling CONVART.

IV. CONCLUSION

Constructive induction is clearly needed to do meaningful inductive inference on low grade time dependent data. CONVART is program designed to deal with this problem in a natural,

incremental way that does not ignore valuable real world knowledge and produces comprehensible results. Although to get truely meaningful results, it must be given data that has a sufficient amount of information in it. Since CONVART was designed to handle real world problems with real world amounts of data, it was made efficient, easy to use and able to handle large amounts of data.

Possible Extensions:

. To add constructive induction on static variables alone. This would allow the output text data file to be used as input to CONVART, thus enabling it to create derived variables of derived variables.

. To add more operators. Although a complete list would be infinite, more could always be added as external procedures.

. To use normalized time and improve the relative time operators.

. To develop a more elaborate control structure for suggestions with a cost for each operator and each input variable.

. To add new relevancy tests.

. To implement the Critic as a computer program to close the incremental learning loop. Although difficult this might be a very interesting and fruitful project.

V. REFERENCES

Blum, R. L., Automating the Study of Clinical Hypotheses on a Time-Oriented Data Base: The RX Project, MEDINFO80 Proceedings, 1980.

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

Buchanan, B. G. and T. M. Mitchell, Model Directed Learning of Production Rules, in D. A. Waterman and F. Hayes-Roth (Eds.) Pattern-Directed Inference Systems, Academic Press, New York, 1978.

Buchanan, B. G., T. M. Mitchell, R. G. Smith and C. R. Johnson, Models of Learning Systems, in Encyclopedia of Computer Science and Technology. J. Belzer, et. al. (Eds.), Marcel Dekker, Inc., New York, 1977.

Busching, M. K. and F. T. Turpin, Oviposition Preferences of Black Cutworm Moths Among Various Crop Plants, Weeds and Plant Debris, J. of Economic Entomology, Vol. 69, no. 5, Oct. 76.

Busching, M. K. and F. T. Turpin, Survival and Development of Black Cutworm (Agrotis ipsilon) Larvae on Various Species of Crop Plants and Weeds, Environmental Entomology, Vol. 6, no. 1, Febuary 1977.

Dietterich, T. G., The Methodology of Knowledge Layers for Inducing Descriptions of Sequentially Ordered Events, MS thesis and internal report no. UIUCDCS-R-80-1024, Univ. of Illinios, Urbana IL, May 1980.

Dietterich, T. G., User's Guide for INDUCE 1.1. Internal Report, Dept of Comp. Sci., Univ. of Illinois, Urbana IL, 1978.

Dietterich, T. G. and R. S. Michalski, Learning and Generalization of Structural Descriptions: Evaluation Criteria and Comparative Review, internal report no. UIUCDCS-R-80-1007, Dept. of Comp. Sci., Univ. of Illinios, Urbana IL, Feb 1980.

Fagan, L. M., VM: Representing Time-Dependent Relations in a Medical Setting, Doctoral dissertation, Computer Science Department, Stanford University, Stanford California, June 1980.

Hayes-Roth, F., P. Klahr and D. J. Mostow, Advice-Taking and Knowledge Refinement: An Iterative View of Skill Acquisition, Rand Paper no. P-6517, Santa Monica CA, July 1980.

Hayes-Roth, F. and J. McDermott, An Interference Matching Technique for Inducing Abstractions, CACM 21:5, pp. 401-410, 1978.

Langley, P., G. Bradshaw and H. Simmon, Rediscovering Chemistry with Bacon.4, Proceedings of the Workshop on Machine Learning, Carnegie-Mellon University, Pittsburg PA, June 1980.

Larson, J. B., Inductive Inference in the Variable Valued Predicate Logic System VL_{21}: Methodology and Computer Implementation, Ph. D thesis and internal report no. UIUCDCS-R-77-869, Dept. of Comp. Sci., Univ. of Illinois, Urbana IL, 1977.

McDermott, D., A Temporal Logic for Reasoning About Processes and Plans, research report no. 196, Dept. of Comp. Sci., Yale University, March 1981.

Michalski, R. S., Variable-valued Logic: System VL_1, Proceedings of the Fourth International Symposium on Mulitple Valued Logic, Morgantown West Virginia, May 1974.

Michalski, R. S., Pattern Recognition as Knowledge-Guided Computer Induction, internal report no. UIUCDCS-R-75-927, Department of Computer Science, University of Illinois, Urbana IL, 1975a.

Michalski, R. S., Synthesis of Optimal and Quasi-Optimal Variable-Valued Logic Formulas, Proc. of the 1975 Inter. Symposium on Multiple-Valued Logic, Indiana University, Bloomington Indiana, May 1975b.

Michalski, R. S., Inductive Learning as Rule-Guided Generalization and Conceptual Simplification of Symbolic Descriptions: Unifying Princples and a methodology, Proceedings of the Workshop on Machine Learning, Carnegie-Mellon University, Pittsburg PA, June 1980.

Michalski, R. S. and R. L. Chilausky, Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquistion in the Context of Developing an Expert System for Soybean Disease Diagnosis, Inter. J. of Policy Analysis and Information Systems, Vol.4, No2. 1980.

Michalski, R. S. and J. B. Larson, Selection of Most Representative Training Examples and Incremental Generation of VL_1 Hypotheses: the Underlying Methodology and the Description of Programs ESEL and AQ11, Department of Computer Science, University of Illinois, Internal Report No. 867, 1978.

Michie, D., Knowledge-Based Systems, internal report no. UIUCDCS-R-80-1001, Dept. of Comp. Sci., Univ. of Illinois, Urbana, Jan. 1980.

Mitchell, T. M., J. Carbonell and R. S. Michalski, SIGART Special Issue on Machine Learning, SIGART Newsletter, No. 76., April 1981.

Mitchell, T. A., P. E. Utgoff and R. B. Banerji, Learning Problem-Solving Heuristics by Experimentation, Proceedings of the Workshop on Machine Learning, Carnegie Mellon University, Pittsburg PA, June 1980.

Nii, H. P. (ed.), Heuristic Programming Project 1980, pub. by Heuristic Programming Project, Comp. Sci. Dept., Stanford Univ., Stanford CA. 1980.

Nii, H. P. and E. A. Feigenbaum, Rule Based Understanding of Signals, in D. A. Waterman and F. Hayes-Roth (eds.), Pattern-Directed Inference Systems, Academic Press, New York, 1978.

Quinlan, J. R., Inductive Inference as a Tool for the Construction of High Performance Programs, Proceedings of the Workshop on Machine Learning, Carnegie-Mellon University, Pittsburg PA, June 1980.

Scott, A. C., M. B. Bishoff and E. H. Shortliffe, Oncology Protocol Management Using the ONCOCIN System: A Preliminary Report, HPP (Hueristic Programming Project) working paper,

Sherrod, D. W., Concepts on Black Cutworm Biology and Oviposition, MS thesis, Department of Entomology, University of Illinios, 1976.

Stepp, R., Learning Without Negative Examples via Variable-Valued Logic Characterization: The Uniclass Inductive Program AQ7UNI, internal report no. UIUCDCS-R-79-982, Dept. of Comp. Sci., Univ. of Illinois, Urbana IL, July 1979.

Vere, S. A., Inductive Learning of Relational Productions, in Patter-Directed Inference Systems, D. A. Waterman and F. Hayes-Roth (Eds.) Academic Press, 1978.

VI. GLOSSARY

absorbance - a measure of the percent of light that is absorbed by an object (like a test tube filled with a chemical solution). A reading of 0 means the object is perfectly transparent and a reading of infinity means the object is opaque (assuming reflection is ignored). Usually absorbance is measured a specific wavelength (color) of light. Negative absorbances have no meaning.

AQ/11 - a computer program that performs inductive inference and produces rules in a variable-valued logic: VL_1 [Michalski and Larson 78].

BCW - an abbreviation for black cutworm, an insect that damages crop plants like corn.

cdd - an abbreviation for cumulative degree days.

colorimetric reaction - a chemical reaction whose rate (speed at which reactants become products) can be measured be a change in color of the reacting solution. The degree to completion (percent of reactants that became products) is directly related to the finaly color of the solution.

cumulative degree days: is a measure of the total amount of energy useful for plant of animal development for the current season. It is usually measured by averaging the minimum and maximum daily temperature, subtracting a threshold temperature and adding the result to a running total. The threshold temperature is used because temperatures below a certain level do not contribute to plant or animal development. So by using a proper threshold, cumulative degree days become a reasonable measure of how far a crop or insect pest should have developed. Of course there are other factors too like planting date, amount of rainfall, etc. but the main point is that it is a useful statistic.

derived variable - a variable whose value is calculated by combining the values of one or more other variables,

generally this means using raw data values to create a new variable.

clustering - (for this paper, only one type of clustering is considered) the process were by a arbitrary list of numbers is split into groups (intervals) so that the numbers relatively close to each other are in the same group. This is similar to what teachers do when they split up numeric test scores into letter grades except here the number of intervals (grades) is not initially known.

dynamic variable - a variable whose value changes over time and so can have more than one value for one event. Also, each value of it must be associated with a time or date when the value was measured.

event - a specific example or description for a given problem giving the values of all the variables. i.e. the single values for all the static variables, an arbitraryily long list of values and times for all the dynamic variables and possibly additional static and dynamic values from a shared data set if thes is one. The value of unknown if by definition a part of the domain of all variables. (see also shared data).

induction - for this paper it is sufficient to define induction as the process of creating a few general descriptions (production rules) from many specific descriptions (events).

operator - a method for combining raw values into a new value such as averaging, finding the minimum value or any other mathematical or logical combination of values.

relevancy - that which pertains and is useful to the problem at hand. a derived variable is relevant if it can help distinguish between decision classes.

shared data - a collection of events associated with another set of events where every shared event is actually a part of one or more events in the other set. For example, weather

station data is shared (and is apart of) the events describing fields around each station.

static variable - a variable whose value does not change over the period of time during which the values of an event are measured. Such variables can only have one value per event.

suggestion - a means of telling CONVART which variables should be created (a positive suggestion) or which variables should not be created (a negative suggestion).

tillage - the practice of loosening the soil of a field before planting crops. Plowing is the most disruptive of the soil as it turns it over. Other types such as disc or chisel plow merely cut the soil leaving much more vegetation on top of the soil, which helps prevent erosion, but attracts egg laying BCW moths.

time - a special variable to CONVART which is always an independent variable that gives the time (in any units) when a dynamic variable was measured. Therefore derived variables like average<time> are never created.

VII. APPENDICES

A. Output from the Enzyme Activity Example

The Transformed Data in Text Form

Prep	Tech	X-Inter	Slope	Y-Inter	Max	MaxTim	Min
ACTIVE	STEPP	0.32	1.8	<-0.81	5.05	<6	<0.5
ACTIVE	?	-0.58	<1.1	-0.0014	3.45	<6	0.5
ACTIVE	BOULANGER	0.22	2.5	-0.81	5.05	<6	0.9
ACTIVE	O'RORKE	0.22	1.1	-0.35	5.05	<6	<0.5
ACTIVE	DAVIS	-0.086	2.5	-0.35	5.05	<6	1.1
INACTIVE	BOULANGER	<-0.58	<1.1	0.52	8.15	8.5	1.1
INACTIVE	O-RORKE	0.32	1.1	-0.81	3.45	<6	<0.5
INACTIVE	DAVIS	0.32	1.1	-0.81	<3.45	<6	<0.5
INACTIVE	STEPP	0.32	1.1	-0.81	8.15	6	<0.5

Abbreviations:

Prep - Preparation
Tech - Technician
X-Inter - X Intercept
Slope - of Absorbance vs. Time
Y-Inter - Y Intercept
Max - Maximum Absorbance
MaxTim - Time of Maximum
Min - Minimum Absorbance

The Transformed Data as Integers for AQ/11

```
NV=7   NCL=2   NEVE=9   NPASS=2                          * AQ/11
                                                        *
TRANS='1'B   OPT='0'B   CPXEV='0'B   GEN='0'B ;         * Parameters
                                                        *
          1    'F'     1                                * Nominal Variable
                                                        *
    4     5     4     5     4     3     4               * Levels / Variable
                                                        *
    5     4                                             * Events / Class
                                                        *
    0     0                                             *   of Input Formula
                                                        *
 0.50    1.00                                           *   of Events / Pass
                                                        *
   0    4    2    0    2    0    0                      * The First Event
  -1    1    0    3    1    0    1                      * (each line
   1    3    3    1    2    0    2                      * is one
   2    3    1    2    2    0    0                      * event)
   3    2    3    2    2    0    3                      * (-1 represents
   1    0    0    4    3    2    3                      * unknown)
   2    4    1    1    1    0    0                      *
   3    4    1    1    0    0    0                      *
   0    4    1    1    3    1    0                      *
```

The Variable Declaration File for AQ/11

```
ACTIVE
INACTIVE

TECHNICIAN
    STEPP
    BOULANGER
    O-RORKE
    DAVIS


X-INTERCEPT OF ABSORBANCE
 < -0.58
   -0.58
   -0.086
    0.22
    0.32

SLOPE OF ABSORBANCE
  < 1.1
    1.1
    1.8
    2.4

Y-INTERCEPT OF ABSORBANCE
 < -0.80
   -0.80
   -0.35
   -0.0014
    0.52

MAXIMUM ABSORBANCE
  < 3.45
    3.45
    5.05
    8.15

MAXTIME OF ABSORBANCE
  < 6
    6
    8.5

MINIMUM OF ABSORBANCE
  < 0.5
    0.5
    0.9
    1.1
```

B. Output from the Idlealized BCW Data

The Transformed Data in Text Form

Each Column is One Event. The events are not numbered as there is no need for any numbering with CONVART. The Names for Variables 0 to 27 are listed below under the heading: Variable Decalaration File For AQ/11. Variable 0 is the decisions class, variables 1 - 12 are static variables copied directly from input to CONVART and variables 13 - 27 are derived variables created by CONVART.

s# - are static variables
d# - are derived variables

Variable Number	Events		
s0. no damage	nodamage	nodamage	no damage
s1. [HELMS]	[WENZEL]	[SOLT]	[FAMS]
s2. [77]	[77]	[77]	[77]
s3. BEANS	OTHER	OTHER	BEANS
s4. YES	YES	YES	NO
s5. [NO]	[NO]	[NO]	[YES]
s6. [LEV]	[ROL]	[ROL]	[BLD]
s7. REDUCED	PLOW	PLOW	REDUCED
s8. REDUCED	REDUCED	REDUCED	REDUCED
s9. [COUNTER]	[COUNTER]	[NONE]	[YES]
s10. 110	110	110	110
s11. 15	20	20	15
s12. [6]	[3]	[2]	[3]
d13. OTHERBROAD	WEEDS	WEEDS	GRASS
d14. OTHERBROAD	WEEDKILL	WEEDS	WEEDKILL
d15. 1	1	2	1
d16. LOW	LOW	LOW	LOW
d17. LOW	LOW	LOW	LOW
d18. -0.005244	-0.005244	-0.005244	-0.005244
d19. -0.2671	-0.2671	-0.2671	-0.2671
d20. LOW	LOW	LOW	LOW
d21. 79	79	79	79
d22. LOW	LOW	LOW	LOW
d23. 83	83	83	83
d24. 462.14	?	?	?
d25. 9.7228	?	?	?
d26. -432.48	?	?	?
d27. 1866.5	?	?	?

s0.	no damage	nodamage	nodamage	no damage
s1.	[HERTER]	[SCHLEU]	[CARROL]	[WIEBUR]
s2.	[77]	[77]	[77]	[77]
s3.	OTHER	OTHER	CORN	BEANS
s4.	YES	YES	NO	NO
s5.	[NO]	[NO]	[NO]	[YES]
s6.	[ROL]	[ROL]	[BLD]	[BLD]
s7.	NONE	NONE	NONE	PLOW
s8.	REDUCED	NONE	REDUCED	REDUCED
s9.	[COUNTER]	[NONE]	[COUNTER]	[DIAZINON]
s10.	110	135	120	0
s11.	20	0	20	15
s12.	[3]	[3]	[3]	[2]
d13.	OTHERBROAD	GRASS	GRASS	GRASS
d14.	OTHERBROAD	LEGUME	OTHERBROAD	GRASS
d15.	2	1	2	2
d16.	LOW	LOW	LOW	LOW
d17.	HIGH	LOW	LOW	LOW
d18.	< -0.01915	-0.005244	-0.005244	?
d19.	2.4486	-0.2671	-0.2671	?
d20.	HIGH	LOW	LOW	LOW
d21.	79	79	79	79
d22.	LOW	LOW	LOW	LOW
d23.	111	83	83	83
d24.	400.81	?	387.40	?
d25.	8.6695	?	8.4074	?
d26.	-389.36	?	-389.36	?
d27.	1668.7	?	1668.7	?

s0.	no damage	nodamage	nodamage	no damage
s1.	[HISSON]	[DALTON]	[LIGHTL]	[POE]
s2.	[77]	[77]	[77]	[77]
s3.	BEANS	CORN	CORN	CORN
s4.	YES	YES	YES	NO
s5.	[YES]	[YES]	[NO]	[YES]
s6.	[LEV]	[ROL]	[ROL]	[BLD]
s7.	PLOW	NONE	NONE	PLOW
s8.	REDUCED	PLOW	PLOW	REDUCED
s9.	[COUNTER]	[COUNTER]	[FURADAN]	[COUNTER]
s10.	100	110	135	0
s11.	20	20	0	0
s12.	[2]	[2]	[3]	[2]
d13.	GRASS	GRASS	WEEDKILL	GRASS
d14.	GRASS	GRASS	WEEDKILL	GRASS
d15.	2	2	2	2
d16.	LOW	LOW	LOW	LOW

d17. LOW	LOW	LOW	LOW
d18. -0.005244	-0.01419	-0.005244	?
d19. -0.2671	1.7245	-0.2671	?
d20. LOW	HIGH	LOW	LOW
d21. 79	79	79	79
d22. LOW	LOW	LOW	LOW
d23. 83	117	83	83
d24. 419.44	387.40	?	361.48
d25. 9.0333	8.4074	?	7.9621
d26. -422.66	-402.51	?	-378.39
d27. 1729.7	1635.2	?	1561.7

---

s0. no damage	nodamage	nodamage	no damage
s1. [WIRTH]	[JEFFCO]	[DASCHE]	[THOREN]
s2. [77]	[77]	[77]	[77]
s3. BEANS	OTHER	OTHER	OTHER
s4. NO	YES	NO	YES
s5. [NO]	[NO]	[YES]	[NO]
s6. [LEV]	[LEV]	[BLD]	[ROL]
s7. REDUCED	NONE	NONE	NONE
s8. REDUCED	PLOW	PLOW	PLOW
s9. [DIAZINON]	[DIAZINON]	[COUNTER]	[COUNTER]
s10. 135	110	120	120
s11. 0	20	20	20
s12. [4]	[2]	[2]	[5]
d13. WEEDKILL	GRASS	GRASS	WEEDKILL
d14. OTHERBROAD	GRASS	GRASS	WINTER_ANN
d15. 1	2	1	2
d16. LOW	LOW	LOW	LOW
d17. LOW	LOW	LOW	LOW
d18. -0.005244	?	?	-0.005244
d19. -0.2671	?	?	-0.2671
d20. LOW	LOW	LOW	LOW
d21. 79	79	79	79
d22. LOW	LOW	LOW	LOW
d23. 83	83	83	83
d24. 419.44	?	?	?
d25. 9.0333	?	?	?
d26. -422.66	?	?	?
d27. 1729.7	?	?	?

---

s0. no damage	nodamage	nodamage	no damage
s1. [PATTER]	[BUNSEL]	[LEIGH]	[BEELER]
s2. [77]	[77]	[77]	[77]
s3. OTHER	CORN	CORN	CORN

s4. YES	NO	NO	YES
s5. [NO]	[NO]	[NO]	[YES]
s6. [ROL]	[LEV]	[LEV]	[BLD]
s7. NONE	PLOW	PLOW	PLOW
s8. PLOW	REDUCED	REDUCED	REDUCED
s9. [COUNTER]	[FURADAN]	[COUNTER]	[COUNTER]
s10. 120	110	120	100
s11. 20	20	15	15
s12. [2]	[7]	[4]	[4]
d13. GRASS	OTHERBROAD	OTHERBROAD	OTHERBROAD
d14. GRASS	WINTER_ANN	OTHERBROAD	OTHERBROAD
d15. 1	2	2	2
d16. LOW	LOW	LOW	LOW
d17. LOW	LOW	LOW	LOW
d18. ?	-0.01206	-0.005244	-0.005244
d19. ?	1.2443	-0.2671	-0.2671
d20. LOW	HIGH	LOW	LOW
d21. 79	79	79	79
d22. LOW	LOW	LOW	LOW
d23. 83	83	83	83
d24. ?	400.81	< 255.09	?
d25. ?	8.6695	< 5.7981	?
d26. ?	-389.36	-264.98	?
d27. ?	1668.7	1257.4	?

---

s0. no damage	nodamage	nodamage	no damage
s1. [GUSTAF]	[WINKEL]	[FRICKE]	[JOHNSO]
s2. [77]	[77]	[77]	[77]
s3. BEANS	BEANS	CORN	CORN
s4. NO	NO	NO	YES
s5. [NO]	[NO]	[YES]	[YES]
s6. [ROL]	[LEV]	[BLD]	[ROL]
s7. REDUCED	REDUCED	NONE	REDUCED
s8. REDUCED	REDUCED	REDUCED	REDUCED
s9. [THIMIT]	[THIMIT]	[COUNTER]	[COUNTER]
s10. 110	110	135	135
s11. 20	20	0	0
s12. [2]	[2]	[5]	[2]
d13. GRASS	GRASS	WEEDKILL	GRASS
d14. GRASS	GRASS	OTHERBROAD	GRASS
d15. 1	2	2	2
d16. LOW	LOW	LOW	LOW
d17. LOW	LOW	LOW	LOW
d18. ?	?	-0.01915	?
d19. ?	?	1.7245	?
d20. LOW	LOW	HIGH	LOW
d21. 79	79	79	79

d22. LOW	LOW	LOW	LOW
d23. 83	83	94	83
d24. ?	361.48	361.48	387.40
d25. ?	7.9621	7.9621	8.4074
d26. ?	-378.39	-378.39	-389.36
d27. ?	1561.7	1561.7	1668.7

---

s0. damage	damage	damage	damage
s1. [MUSSEL]	[SKAER]	[TALBOT]	[DANNEL]
s2. [77]	[77]	[77]	[77]
s3. CORN	CORN	BEANS	BEANS
s4. YES	YES	YES	YES
s5. [NO]	[YES]	[YES]	[YES]
s6. [LEV]	[ROL]	[BLD]	[LEV]
s7. PLOW	NONE	REDUCED	NONE
s8. REDUCED	NONE	REDUCED	REDUCED
s9. [COUNTER]	[FURADAN]	[FURADAN]	[YES]
s10. 110	120	110	120
s11. 20	0	15	15
s12. [4]	[3]	[2]	[3]
d13. OTHERBROAD	GRASS	OTHERBROAD	OTHERBROAD
d14. OTHERBROAD	GRASS	OTHERBROAD	GRASS
d15. 2	1	2	1
d16. HIGH	HIGH	LOW	LOW
d17. HIGH	HIGH	HIGH	LOW
d18. 0.01439	-0.005244	?	0.01893
d19. -1.0779	0.5488	?	-2.5705
d20. HIGH	HIGH	HIGH	HIGH
d21. 89	79	116	126
d22. LOW	HIGH	LOW	LOW
d23. 83	83	83	83
d24. 308.42	462.14	462.14	255.09
d25. 6.8854	9.7228	9.7228	5.7981
d26. -363.03	-432.48	-432.48	-318.16
d27. 1402.3	1866.5	1866.5	< 1257.4

---

s0. damage	damage	damage	damage
s1. [BRIGGS]	[FINFRO]	[ALBIN]	[LOCK]
s2. [77]	[77]	[77]	[77]
s3. CORN	CORN	CORN	BEANS
s4. YES	YES	YES	NO
s5. [YES]	[YES]	[YES]	[NO]
s6. [ROL]	[ROL]	[BLD]	[LEV]
s7. NONE	NONE	REDUCED	REDUCED
s8. REDUCED	PLOW	REDUCED	REDUCED

s9.	[COUNTER]	[COUNTER]	[COUNTER]	[THIMIT]
s10.	120	110	135	135
s11.	10	20	0	20
s12.	[4]	[3]	[4]	[3]
d13.	OTHERBROAD	WEEDKILL	OTHERBROAD	WEEDKILL
d14.	GRASS	OTHERBROAD	WINTER_ANN	OTHERBROAD
d15.	1	1	1	1
d16.	LOW	HIGH	HIGH	LOW
d17.	LOW	HIGH	HIGH	LOW
d18.	0.009507	0.002719	0.002719	-0.005244
d19.	-1.0779	-0.2671	-0.2671	-0.2671
d20.	HIGH	HIGH	HIGH	LOW
d21.	151	< 79	79	79
d22.	LOW	LOW	LOW	LOW
d23.	83	83	83	83
d24.	400.81	?	387.40	387.40
d25.	8.6695	?	8.4074	8.4074
d26.	-402.51	?	-402.51	-402.51
d27.	1668.7	?	1635.2	1668.7

---

s0.	damage	damage	damage	damage
s1.	[MUELLE]	[MUELLE]	[WETZEL]	[WETZEL]
s2.	[77]	[77]	[77]	[77]
s3.	OTHER	CORN	CORN	BEANS
s4.	YES	YES	YES	YES
s5.	[YES]	[YES]	[YES]	[YES]
s6.	[BLD]	[BLD]	[BLD]	[BLD]
s7.	REDUCED	PLOW	PLOW	PLOW
s8.	REDUCED	REDUCED	REDUCED	REDUCED
s9.	[DIAZINON]	[DIAZINON]	[DIAZINON]	[DIAZINON]
s10.	100	100	100	100
s11.	20	20	20	20
s12.	[2]	[2]	[2]	[2]
d13.	OTHERBROAD	OTHERBROAD	OTHERBROAD	OTHERBROAD
d14.	OTHERBROAD	OTHERBROAD	OTHERBROAD	OTHERBROAD
d15.	1	1	2	1
d16.	LOW	LOW	LOW	LOW
d17.	LOW	LOW	LOW	LOW
d18.	?	?	?	?
d19.	?	?	?	?
d20.	LOW	LOW	LOW	LOW
d21.	79	79	79	79
d22.	LOW	LOW	LOW	LOW
d23.	83	83	83	83
d24.	483.28	483.28	483.28	483.28
d25.	10.063	10.063	10.063	10.063
d26.	-453.71	-453.71	-453.71	-453.71

d27.	1926.7	1926.7	1926.7	1926.7

s0.	damage	damage	damage	damage
s1.	[ZINDAR]	[BLACKE]	[GREGOR]	[GOODIN]
s2.	[77]	[77]	[77]	[77]
s3.	BEANS	CORN	BEANS	CORN
s4.	YES	YES	YES	YES
s5.	[YES]	[YES]	[YES]	[YES]
s6.	[BLD]	[ROL]	[BLD]	[LEV]
s7.	REDUCED	NONE	NONE	NONE
s8.	REDUCED	PLOW	REDUCED	REDUCED
s9.	[NONE]	[COUNTER]	[MOCAP]	[FURADAN]
s10.	135	110	120	135
s11.	0	20	10	0
s12.	[6]	[3]	[2]	[2]
d13.	OTHERBROAD	WEEDKILL	OTHERBROAD	OTHERBROAD
d14.	OTHERBROAD	OTHERBROAD	OTHERBROAD	OTHERBROAD
d15.	2	2	2	2
d16.	HIGH	HIGH	HIGH	HIGH
d17.	HIGH	HIGH	HIGH	HIGH
d18.	-0.005244	-0.005244	-0.005244	0.01604
d19.	0.5488	0.5488	0.5488	-1.5607
d20.	HIGH	HIGH	HIGH	HIGH
d21.	79	79	79	126
d22.	HIGH	HIGH	HIGH	LOW
d23.	83	83	83	83
d24.	387.40	387.40	387.40	?
d25.	8.4074	8.4074	8.4074	?
d26.	-402.51	-402.51	-402.51	?
d27.	1635.2	1635.2	1635.2	?

s0.	damage	damage	damage	damage
s1.	[DUFREN]	[GEBHAR]	[WIRTH]	[RUFF]
s2.	[77]	[77]	[77]	[77]
s3.	CORN	OTHER	OTHER	BEANS
s4.	YES	YES	YES	NO
s5.	[NO]	[YES]	[YES]	[NO]
s6.	[BLD]	[BLD]	[LEV]	[LEV]
s7.	PLOW	NONE	PLOW	REDUCED
s8.	REDUCED	PLOW	REDUCED	REDUCED
s9.	[COUNTER]	[NONE]	[DIAZINON]	[COUNTER]
s10.	100	110	135	135
s11.	20	20	0	0
s12.	[4]	[2]	[4]	[4]
d13.	OTHERBROAD	GRASS	OTHERBROAD	WEEDKILL

d14. WINTER_ANN	GRASS	GRASS	OTHERBROAD
d15. 2	1	1	2
d16. HIGH	HIGH	HIGH	HIGH
d17. LOW	HIGH	HIGH	LOW
d18. 0.03098	?	0.01439	0.002719
d19. < -2.5705	?	-1.2198	-0.7548
d20. HIGH	HIGH	HIGH	HIGH
d21. 99	79	99	126
d22. LOW	HIGH	LOW	LOW
d23. 83	83	83	83
d24. 506.22	361.48	419.44	419.44
d25. 10.549	7.9621	9.0333	9.0333
d26. < -453.71	-378.39	-422.66	-422.66
d27. 2011.5	1561.7	1729.7	1729.7

---

s0. damage	damage	damage	damage
s1. [MONTGO]	[RINCKE]	[HENTON]	[GROEZI]
s2. [77]	[77]	[77]	[77]
s3. CORN	CORN	CORN	OTHER
s4. YES	YES	YES	YES
s5. [YES]	[YES]	[YES]	[YES]
s6. [LEV]	[LEV]	[LEV]	[ROL]
s7. REDUCED	NONE	NONE	NONE
s8. PLOW	PLOW	PLOW	PLOW
s9. [NONE]	[NONE]	[NONE]	[NONE]
s10. 135	135	135	120
s11. 0	0	0	20
s12. [3]	[4]	[3]	[4]
d13. OTHERBROAD	WEEDS	OTHERBROAD	WEEDKILL
d14. GRASS	WEEDS	WEEDKILL	WEEDKILL
d15. 2	1	1	2
d16. HIGH	HIGH	HIGH	HIGH
d17. HIGH	HIGH	HIGH	HIGH
d18. 0.01604	0.009507	0.009507	-0.005244
d19. -1.5607	-0.7548	-1.0779	0.5488
d20. HIGH	HIGH	HIGH	HIGH
d21. 126	99	116	79
d22. LOW	LOW	LOW	HIGH
d23. 83	83	83	83
d24. 419.44	419.44	387.40	?
d25. 9.0333	9.0333	8.4074	?
d26. -422.66	-422.66	-389.36	?
d27. 1729.7	1729.7	1635.2	?

---

s0. damage	damage	damage	damage

s1.	[OSTRUM]	[NAGEL]	[BUCHAN]	[IVERS]
s2.	[77]	[77]	[77]	[77]
s3.	OTHER	OTHER	BEANS	CORN
s4.	YES	YES	YES	YES
s5.	[YES]	[NO]	[YES]	[NO]
s6.	[ROL]	[ROL]	[ROL]	[ROL]
s7.	NONE	NONE	REDUCED	REDUCED
s8.	PLOW	PLOW	REDUCED	REDUCED
s9.	[NONE]	[MOCAP]	[NONE]	[FURADAN]
s10.	120	120	135	135
s11.	20	20	0	0
s12.	[2]	[2]	[4]	[4]
d13.	OTHERBROAD	GRASS	OTHERBROAD	WEEDKILL
d14.	OTHERBROAD	GRASS	GRASS	OTHERBROAD
d15.	2	2	1	2
d16.	LOW	LOW	HIGH	HIGH
d17.	HIGH	HIGH	HIGH	HIGH
d18.	?	?	-0.005244	0.009507
d19.	?	?	0.5488	-1.0779
d20.	HIGH	HIGH	HIGH	HIGH
d21.	138	138	79	108
d22.	LOW	LOW	HIGH	LOW
d23.	83	83	83	83
d24.	?	?	438.37	438.37
d25.	?	?	9.3597	9.3597
d26.	?	?	-422.66	-422.66
d27.	?	?	1790.8	1790.8

---

s0.	damage	s1.	[HAMPTO]
s2.	[77]	s3.	BEANS
s4.	YES	s5.	[YES]
s6.	[BLD]	s7.	REDUCED
s8.	REDUCED	s9.	[NONE]
s10.	110	s11.	20
s12.	[3]	d13.	OTHERBROAD
d14.	OTHERBROAD	d15.	2
d16.	HIGH	d17.	HIGH
d18.	-0.005244	d19.	0.5488
d20.	HIGH	d21.	79
d22.	HIGH	d23.	83
d24.	387.40	d25.	8.4074
d26.	-389.36	d27.	1668.7

The Transformed Data as Integers for AQ/11

The first two lines are parameter settings for AQ/11. The next six lines specify which variables are nominal, the number of levels per variable (two lines), the number of events per class, the number of input formuala per class and the fraction of the events to use during each pass over the data, respectively. The single spaced lines are events, one event per line. For further information see Michalski and Larson [78].

```
NV=21  NCL=2  NEVE=53  NPASS=2

TRANS='1'B  OPT='0'B  CPXEV='0'B  GEN='0'B ;

        7   'F'    1    2    3    4    7    8    9

3   2   3   3   6   5   9   9   3   3  3

11  11   3   9   3   5  11  11  10  11

  24   29

   0    0

0.50   1.00

 1  0  2  2  2  2  8  8  1  0  0  4  6  0  1  0  1  8  8  2  8
 2  0  1  2  2  3  1  4  1  0  0  4  6  0  1  0  1 -1 -1 -1 -1
 2  0  1  2  2  3  1  1  2  0  0  4  6  0  1  0  1 -1 -1 -1 -1
 1  1  2  2  2  2  5  4  1  0  0  4  6  0  1  0  1 -1 -1 -1 -1
 2  0  0  2  2  3  8  8  2  0  1  0 10  1  1  0  3  5  5  5  5
 2  0  0  0  4  0  5  6  1  0  0  4  6  0  1  0  1 -1 -1 -1 -1
 0  1  0  2  3  3  5  8  2  0  0  4  6  0  1  0  1  4  4  5  5
 1  1  1  2  0  2  5  5  2  0  0 -1 -1  0  1  0  1 -1 -1 -1 -1
 1  0  1  2  1  3  5  5  2  0  0  4  6  0  1  0  1  6  6  3  6
 0  0  0  1  2  3  5  5  2  0  0  2  9  1  1  0  4  4  4  4  4
 0  0  0  1  4  0  4  4  2  0  0  4  6  0  1  0  1 -1 -1 -1 -1
 0  1  1  2  0  0  5  5  2  0  0 -1 -1  0  1  0  1  3  3  6  3
 1  1  2  2  4  0  4  8  1  0  0  4  6  0  1  0  1  6  6  3  6
 2  0  0  1  2  3  5  5  2  0  0 -1 -1  0  1  0  1 -1 -1 -1 -1
 2  1  0  1  3  3  5  5  1  0  0 -1 -1  0  1  0  1 -1 -1 -1 -1
 2  0  0  1  3  3  4  7  2  0  0  4  6  0  1  0  1 -1 -1 -1 -1
 2  0  0  1  3  3  5  5  1  0  0 -1 -1  0  1  0  1 -1 -1 -1 -1
 0  1  1  2  2  3  8  7  2  0  0  3  8  1  1  0  1  5  5  5  5
 0  1  1  2  3  2  8  8  2  0  0  4  6  0  1  0  1  0  0  9  1
 0  0  1  2  1  2  8  8  2  0  0  4  6  0  1  0  1 -1 -1 -1 -1
 1  1  2  2  2  3  5  5  1  0  0 -1 -1  0  1  0  1 -1 -1 -1 -1
 1  1  2  2  2  3  5  5  2  0  0 -1 -1  0  1  0  1  3  3  6  3
 0  1  0  2  4  0  4  8  2  0  0  1  9  1  1  0  2  3  3  6  3
```

0	0	2	2	4	0	5	5	2	0	0	-1	-1	0	1	0	1	4	4	5	5
0	0	1	2	2	3	8	8	2	1	1	7	4	1	2	0	1	2	2	7	2
0	0	0	0	3	0	5	5	1	1	1	4	7	1	1	1	1	8	8	2	8
1	0	2	2	2	2	8	8	2	0	1	-1	-1	1	5	0	1	8	8	2	8
1	0	0	2	3	2	8	5	1	0	0	9	1	1	6	0	1	1	1	8	0
0	0	0	2	3	1	8	5	1	0	0	6	4	1	8	0	1	5	5	4	5
0	0	0	1	2	3	4	8	1	1	1	5	6	1	0	0	1	-1	-1	-1	-1
0	0	2	2	4	0	8	7	1	1	1	5	6	1	1	0	1	4	4	4	4
1	1	2	2	4	3	4	8	1	0	0	4	6	0	1	0	1	4	4	4	5
2	0	2	2	1	3	8	8	1	0	0	-1	-1	0	1	0	1	9	9	1	9
0	0	1	2	1	3	8	8	1	0	0	-1	-1	0	1	0	1	9	9	1	9
0	0	1	2	1	3	8	8	2	0	0	-1	-1	0	1	0	1	9	9	1	9
1	0	1	2	1	3	8	8	1	0	0	-1	-1	0	1	0	1	9	9	1	9
1	0	2	2	4	0	8	8	2	1	1	4	7	1	1	1	1	4	4	4	4
0	0	0	1	2	3	4	8	2	1	1	4	7	1	1	1	1	4	4	4	4
1	0	0	2	3	1	8	8	2	1	1	4	7	1	1	1	1	4	4	4	4
0	0	0	2	4	0	8	8	2	1	1	8	2	1	6	0	1	-1	-1	-1	-1
0	0	1	2	1	3	8	7	2	1	0	10	0	1	3	0	1	10	10	0	10
2	0	0	1	2	3	5	5	1	1	1	-1	-1	1	1	1	1	3	3	6	3
2	0	1	2	4	0	8	5	1	1	1	7	3	1	3	0	1	6	6	3	6
1	1	2	2	4	0	4	8	2	1	0	5	5	1	6	0	1	6	6	3	6
0	0	2	1	4	0	8	5	2	1	1	8	2	1	6	0	1	6	6	3	6
0	0	0	1	4	0	1	1	1	1	1	6	5	1	3	0	1	6	6	3	6
0	0	0	1	4	0	8	4	1	1	1	6	4	1	5	0	1	4	4	5	4
2	0	0	1	3	3	4	4	2	1	1	4	7	1	1	1	1	-1	-1	-1	-1
2	0	0	1	3	3	8	8	2	0	1	-1	-1	1	7	0	1	-1	-1	-1	-1
2	0	0	1	3	3	5	5	2	0	1	-1	-1	1	7	0	1	-1	-1	-1	-1
1	0	2	2	4	0	8	5	1	1	1	4	7	1	1	1	1	7	7	3	7
0	0	2	2	4	0	4	8	2	1	1	6	4	1	4	0	1	7	7	3	7
1	0	2	2	2	3	8	8	2	1	1	4	7	1	1	1	1	4	4	5	5

The Variable Declaration File for AQ/11

The first two names given below are the names of the two decision classes, following them are the names and domains of all of the variables that were sent to AQ/11. These variables are explained in section III. B. In the actual trans file for AQ/11 there are no variable numbers as here. They have been added here to help document the text version of the transformed data listed above. Notice that some of variable numbers appear to be missing. This is because some of the variables given to CONVART were declared to be non-applicable variables and so were not passed on to AQ/11. Therefore their values are surrounded by square brackets in the text version of the transformed data, and their values and definitions were omitted from the two input files to AQ/11: the transformed data as integers and the variable declaration file listed below. For the sake of completeness; the names and domains of the non-applicable variables are given following the variable definitions for AQ/11.

No Damage
Damage

3. Previous Crop
   corn
   beans
   other

4. Permanent Vegetation
   yes
   no

7. Fall Tillage
   none
   plow
   reduced

8. Spring Tillage
   none
   plow
   reduced

10. Planting Date
    0
    100
    110
    120
    135

11. Risk Index
    0
    10
    15
    20
    25

13. Most Common Weed Species
    none
    very few weeds
    weeds
    onion
    weedkill
    grass
    legume
    winter annual

14. Least Common Weed Species
    none
    very few weeds
    weeds
    onion
    weedkill
    grass
    legume
    winter annual

other broadleaf

15. Number Same of Weed Species
< 1
1
2

16. Weed Density on Planting Date
low
medium
high

17. Average Weed Density
low
medium
high

18. Slope of Weed Density
<-0.01915
-0.01915
-0.01419
-0.01206
-0.005244
0.002719
0.009507
0.01439
0.01604
0.01893
0.03098

19. Y Intercept of Weed Density
<-2.5705
-2.5705
-1.5607
-1.2198
-1.0779
-0.7548
-0.2671
0.5488
1.2443
1.7245
2.4486

20. Max Value of Weed Density
low
medium
high

21. Time of 1st Maximum of Weed Density
<79
79
89
99
108
116
126
138
151

22. Minium Weed Density
low
medium
high

(cdd stands for Cumulative Degree Days)

23. Time of 1st Minimum Weed Density
    <83
     83
     94
    111
    117

24. Average cdd
    <255.09
     255.09
     308.42
     361.48
     387.40
     400.81
     419.44
     438.37
     462.14
     483.28
     506.22

25. Slope of cdd
    <5.7981
     5.7981
     6.8854
     7.9621
     8.4074
     8.6695
     9.0333
     9.3597
     9.7228
    10.063
    10.549

26. Y Intercept of cdd
    <-453.71
     -453.71
     -432.38
     -422.66
     -402.51
     -389.36
     -378.39
     -363.03
     -318.16
     -264.98

27. Maxium cdd
    <1257.4
     1257.4
     1402.3
     1561.7
     1635.2
     1668.7
     1729.7
     1790.8
     1866.5
     1926.7
     2011.5

:Definitions for the Non-applicable variables

1. Farmer

Mussel	Skaer	Talbot	Helms	Wenzel	Solt	Fams	Herter	Schleu
Dannel	Briggs	Finfro	Albin	Lock	Carrol	Muelle	Muelle	Wetzel
Wetzel	Wiebur	Hisson	Zindar	Blacke	Dalton	Gregor	Lightl	Goodin
Dufren	Gebhar	Poe	Wirth	Wirth	Ruff	Montgo	Rincke	Henton
Jeffco	Dasche	Thoren	Patter	Groezi	Ostrum	Nagel	Buchan	Ivers
Bunsel	Leigh	Beeler	Gustaf	Winkel	Fricke	Johnso	Hampto	

2. Calender Year
   77
   78.t
   79
   80

5. Field Border Water
   yes
   no

6. Field Surface
   rolling
   level
   buildings

9. Insecticide

yes	counter	furadan	thimit
diazinon	mocap	none	
ramrod	lorsban	dyfonate	rlorsban
thimet	aldrin	isotoxdia	heptachlor
rescuetrt	rpenncap	rsevin	belt

12. Dynamic Count
{a count of the number of dynamic values per event}
   0
   1
   3
   6