USING ACCURACY IN SCIENTIFIC DISCOVERY

M. MOULET

Equipe Inférence et Apprentissage
LRI, Bât 490, Université Paris-Sud
91405 Orsay Cedex. France
Email : marjorie@lri.lri.fr

Abstract

Learning by discovery aims at bringing to light laws from a set of numerical or symbolic data. Our work deals with the improvement of the discovery system ABACUS created by Michalski and Falkenhainer, and in particular, with the way the system makes use of informative accuracy of the data. ABACUS, like most others current discovery systems does not use this information in the real physical sense, that means accuracy given by the measure device. However, in experimental domains accuracy cannot obviously be separated from the data. In this paper, we show how, when used in a more realistic manner, this information can significantly improve not only the accuracy of the results but also the efficiency of the search algorithm. Several additional modifications to ABACUS to improve the robustness of the system without losing generality will also be described.

Key-words

Scientific discovery, learning by observation, numeric - symbolic integration.

1 Introduction

Our work is concerned with the field of learning by discovery and consists in the improvement of the discovery system **ABACUS** created by Falkenhainer (Falkenhainer & Michalski, 1986) and improved by Greene (1988). This system is well-known as one descendant of the **BACON** system created by Langley, Simon, Bradshaw and Zytkow (1985; 1986 and 1987). Without changing the algorithm itself and its basic heuristics, we add a new kind of information which improves the system's efficiency, especially its search algorithm. Moreover, this information is general enough to be applicable in most scientific domains. With this goal, we have integrated into the system the notion of uncertainty of data as an inherent part of the data.

Indeed, as soon as we have to deal with numerical data, whether it be the result of physical experiments or statistical results of a sociological study, the question of the accuracy associated with these data is raised. By accuracy, we mean the uncertainty of the data, expressed either with a **relative** value as in "the town contains 25 000 inhabitants, plus or minus two percent", or with an **absolute** value as in "it costs 400 pounds, plus or minus 20 pounds".

When the data is experimental, the accuracy associated with a measure is characterized most of the time by the quality of the **device used.** Consequently, the accuracy of these measures cannot be computed automatically from the data alone, but must be part of the data itself and given by the expert or the system's user.

Despite the fact that the necessity for specifying the accuracy of data during the acquisition step is self evident, we must emphasize that this requirement is not always observed by researchers in the learning community, even by those interested in data analysis or statistics. However, we can note one approach concerned with the problem of uncertainty. FAHRENHEIT, created by Zytkow (1987 ; Zytkow & Zhu & Hussam, 1990), is able to compute "errors" on the data by asking to the user for repeated experiments. However, these errors obviously correspond to a lack of fidelity of the device used, but they do not represent its sensibility (if a small change can be detected) or its rightness (with regard to a standard). Unfortunately, these two last properties can only be evaluated by a user experimentally.

In the majority of cases, only one value is chosen to represent the accuracy of all the numerical data. But this solution cannot be used in learning by discovery where data can frequently be provided by different measuring devices and moreover where the research is done by iterating computations, decreasing accuracy proportionately with the number of steps. In this context, there is no longer any justification for associating the same accuracy with all values.

In this paper, we want to introduce our new version of the system ABACUS and the improvements we have made concerning the use of uncertainty. We will first present ABACUS's aim and the general principles we have chosen to improve the system. Then, we describe in detail some definitions needed to manipulate uncertainty within ABACUS before we present the improvements. The first one and the most important of our work is based on our new approach to dealing with accuracy. The second one concerns mathematical cancellations and tautologies. Finally, we present a simple but useful way to improve the iterative search and to correct the drawbacks of the evaluation criteria used by ABACUS to control the search.

2 ABACUS's aim and algorithm

ABACUS deals with finding relations which are of interest to the experimenter, within a set of experimentations. This data set can be viewed as a matrix where each row represents an example and each column represents one measured parameter. In this context, the aim of the discovery system is to bring to light the mathematical relation(s) between all or some of the parameters such that all or a majority of the examples verify this (or one of these) relation(s). Thus, the search space to look through is constituted of all the mathematical formulas one can build from the initial parameters.

Let us precise that in the following, we will note either parameter or variable a column of the matrix, but *parameter* will rather represent the initial measured parameters and *variable* will rather represent the next columns created by the system.

ABACUS's discovery process consists in an iterative building of variables starting with the initial parameters as first variables. It chooses two among the four arithmetic operators (+, -, / and *) to apply to each pair of variables according to the existing monotonic dependency (the second variable increases or decreases when the first increases). As in BACON, the dependencies between two parameters can be computed only when all the other parameters are holding constant. The process stops when a variable is found which takes the same value for all examples. This variable will then represent the equation or law verified by the data set.

The main advantage ABACUS provides with regard to BACON is to propose to the user either one law as in BACON, or a collection of laws with each one summarizing a subset of the examples. Two solutions can be used to find the partition : the first one is to compute it with a clustering algorithm and then to search for an equation on each subset independently. The second is to search for only one variable such that all the examples can be gathered into different subgroups according to their value on this variable.

3 New principles

1- The accuracy rate of each parameter is provided by the user : only one **relative** uncertainty rate is provided if all the values of the parameter have been recorded with the same relative uncertainty, and if not, **absolute** uncertainty rates associated with each value. In the following, we will denote the relative and absolute uncertainty of a value x_0 by the respective expressions ρ (x_0) et δ (x_0).

In spite of the fact that learning by discovery deals usually with experimental data, this work constitutes the first work where the user is allowed to provide along with the data their associated measures of accuracy. However, in an experimental field, like physics or chemistry, the uncertainty rates of the values are not only supplementary information but necessary and fundamental to the interpretation of the measures. These new values take the place in ABACUS of the previous parameter "uncertainty" initially used to represent the uncertainty rate of all the numerical values the system had to manipulate. We will show in the next parts how, as a contrast to this simpler way of dealing with uncertainty, our addition of the real uncertainties can improve the system.

2- One difficult problem in this kind of "blind" search, is how to avoid the generation of tautologies, that is to say equations which are self-evident like y = y, which are unfortunately often considered by the system as being good solutions. Although ABACUS have succeeded in controlling some of these tautologies, we have noticed that numerous tautologies were still not taken into account.

With this goal, we have integrated together with the notion of accuracy the one of negligibility, which definition is based on accuracy since a variable is negligible with regard to another according to the accuracy of the later. Moreover, we have implemented a simplification module, which is particularly useful to avoid the creation of tautologies. This has been done thanks to some additional computations based on arithmetical rules, quite simple when done "by hand" but quite problematic when done automatically, because of their complexity.

3- Finally, we have improved the search algorithm, by analyzing the way ABACUS evaluates variables in order to choose the best path in the search graph. At each step, the new variables are evaluated before the graph is splitted into two subgraphs which will be explored one after the other. Although this heuristic is useful in order to reduce combinatorial explosion, it is difficult to avoid a bad splitting of the search space. We propose a compromise, which is quite simple but has still proved to be useful, based on delaying the splitting.

4 Improvements based on new accuracies

We present now the principal definitions used to compare and manipulate different uncertain values, before we show the advantages of the use of accuracy for a discovery system.

4.1 Definition of a new equality

In the description of ABACUS above, we have described how the whole process relies on the finding of a variable which is constant on all or the majority of examples. We are therefore principally interested in detecting when two or more values of a same variable are equal. But once we are given the accuracy of these values, we can no longer use the standard definition of equality. In place of this standard definition, we introduce the notion of the **possibility** of two values being equal.

Note that in the previous version of ABACUS, the uncertainty parameter was also used to compare two values of a same variable, but it did not allow the values to have their own absolute uncertainty. Now, we have to generate a new comparison function that is valid for values taking different absolute or relative uncertainties. The simplest way is to define this comparison function from the absolute uncertainties.

Let us call *interval of definition of x* the set $[x - \delta(x), x + \delta(x)]$, then two values x_i and x_j *have the possibility of being equal* if their two intervals of definition are not disjoint. Let us call this new property the **δ-equality** and let us define it thus :

> **x_i et x_j are δ−equal**
> *if and only if*
> $$|x_j - x_i| \leq \delta(x_i) + \delta(x_j)$$

Example : $x = 10 \pm 3$ is δ-equal to $y = 12 \pm 1$ since $|10 - 12| = 2 \leq 3 + 1 = 4$

We can then generalize this property to p values. Let $x_0, x_1, \ldots, x_p$ be the values such that $x_0 \leq x_1 \leq \ldots \leq x_p$ (this condition helps simplifying the definition). If we have already verified that $x_0 \ldots x_{p-1}$ are **δ−equal**, then

> **x_p is δ−equal to $x_0, \ldots, x_{p-1}$**
> *if and only if*
> $$x_p - \delta(x_p) < \min_{0 \leq i \leq p-1} \{x_i + \delta(x_i)\}$$

Finally, when is found a variable which takes constant values for all the examples in fact we have to set which value will represent all of them, given that all the values of the variable are only δ-equal. Seeing that his value will only be used in the presentation of the result to the user, and not in the incremental process, and in order to avoid possible large mistakes, we choose to represent them by the most precise one (12 ± 1 in our example).

4.2 Accuracy computation formulas

One of the most important improvements to the initial version of ABACUS is its additional capacity to compute automatically the uncertainty rates of the variables the system generates. These new accuracies must obviously be computed according to the uncertainty rates of the previous variables and according to the function used to generate these new variables. It is then necessary to introduce the computation formulas able to provide the relative or absolute uncertainties of a function from either the absolute or the relative uncertainties of the previous variables.

We recall in the table 1 the computation formulas used to derive uncertainties when applying the different arithmetic operators to a pair of variables. Some of them can be found in the physics course of Joyal (1956) or Eurin and Guimiot (1953).

op	ρ (x op y) (relative form)	δ (x op y) (absolute form)
+/-	$\frac{\delta (x + y)}{(x + y)}$	$\delta (x) + \delta (y)$
*	$\rho (x) + \rho (y)$	$\delta (x) * y + x * \delta (y)$
/	$\rho (x) + \rho (y)$	$\frac{\delta(x)}{y} + \frac{x * \delta(y)}{y^2}$

Table 1. Formulas for the computation of accuracy

Concerning the trigonometric functions (we deal here with cosine and sine only), there are no formulas giving directly the uncertainty of the result of applying the function, given the uncertainty of the argument. We must therefore compute separately the uncertainty of each value, using the following principle :

Let x_0 be a value with the absolute uncertainty $\delta(x_0)$. Given a mathematical function **f**, the absolute uncertainty of $f(x_0)$ is the maximal difference observed between $f(x_0)$ and the values f takes on the extreme values of x_0. This can be written as :

$$\delta(f(x0)) = \max \{ | f(x0 + \delta(x0)) - f(x0) | , | f(x0 - \delta(x0)) - f(x0) | \}$$

Note that this formula can, from now on, be easily used for any new operator which could be added to the system such as exponential, logarithm, etc.

4.3 Consequences on the final partitions and laws

ABACUS aims at splitting the example set into a partition such that the set of all the laws describing each group best describes the whole of the example set. One way to discover this partition is to find a variable which takes constant values on the subsets of a partition of the examples. It is then important to accurately compute these partitions that are "induced" by a variable, in order to find the right final partition.

The idea is therefore to compute these groups by comparing the values of a variable using the δ-equality we have defined above. Given the initial accuracy rates, we can now compute the partitions with more accuracy. This allows for instance the discovery of new partitions where all data were gathered together before, because the accuracy was not high enough to allow a distinction to be made between some values.

We present in the sections below two aspects of the improvements due to our better conception of accuracy. The first one illustrates and justifies more precisely the influence of the accuracy rate on the resulting partitions and laws. The second emphasizes the role of the iterative computations of the uncertainty rates of the variables. In conclusion, these two improvements must be absolutely taken into account altogether in order to have the right reasoning.

4.3.1 influence of the accuracy rate

Table 2 illustrates the influence of accuracy on both the final laws and the final partitions. Experiments have been performed on the same example set changing accuracy rate (artificial one and assumed the same for all parameters for the sake of clearness and comprehension). The experiments are relative to the law of the speed of sound in air. There are fifteen examples described by four parameters : the speed V, the time T, the frequency FREQ and the atmospheric pressure P. Let us recall that the real law is : $V^2 = 401.32\ T$ (thus, the last two variables are not relevant) but small errors have been introduced. The results have been obtained with our new version of ABACUS, computing the uncertainty of each new variable.

ρ	**partitions and laws**
5%	$T = 0.836\ V$ ($\pm$ 10%)
2%	$T = 250$ or 273 $V = 1.232\ T$ $T = 330$ $V = 1.107\ T$ ($\pm$ 4%)
0.5%	$T = 250$ $V = 1.267\ T$ $T = 273$ $V = 1.212\ T$ $T = 330$ $V = 1.107\ T$ ($\pm$ 1%)
0.1%	$V^2 = 401.32\ T$ ($\pm\ 3.10^{-1}$ %)
5.10^{-5}%	$T = 0.00249\ V^2$ ($\pm\ 2.10^{-3}$ %) (after simplification of V + T/V divided by V)

Table 2. Influence of the accuracy on the laws and the partitions.

In order to understand these results, it must not be forgotten that each solution corresponds to a constant variable. The first result (for $\rho = 5\%$) means that the values of the variable $\frac{T}{V}$ on the 15 examples are found δ-equal. In the two following cases, where the accuracy rate ρ is decreased, the 15 are no more δ-equal altogether but are δ-equal when they are splitted into 2 or 3 groups. In the two last cases, the accuracy rate was obviously too high to allow all the values of $\frac{T}{V}$ to be δ-equal, even by intervals. The system was able by generating the new variable $\frac{V^2}{T}$, to find an approximation of the right law. However, in the last case, $\frac{V^2}{T}$ was not found constant with regard of the error rate of $2\ 10^{-3}$% but found it all the same by simplification of a formula obtained by combining next variables, according to the new improvement we will see in section 4.

Moreover, if we had increased the accuracy rate, there exists some value of the accuracy for which $\frac{V^2}{T}$ like $\frac{T}{V}$ will no more be found constant, nor the right law will be found. Generally, increasing accuracy does not necessarily improve the solution : a too high accuracy may lead to a solution other than the one desired, or even none at all.

The results shown in table 2 have been performed by varying the uncertainty rate. It must be pointed out that the user cannot in reality choose the accuracy rates, since they are provided by the measurements devices, and that each parameter has only one given accuracy that cannot be changed anymore. Actually, all solutions proposed by the system are correct, and one uncertainty is not better than another. The estimation of a result can then be done only according to the user's point of view. We assume then that in the user's interest it is better to give effectively each accuracy rate of the initial parameters than to choose randomly one accuracy rate.

In the law of the speed of sound, for instance, if we give the following right uncertainty rate: $\rho(T) = 0.3\%$, $\rho(V) = 0.001\%$, $\rho(P) = 1\%$ and $\rho(FREQ) = 3\%$, the law $V^2 = 401.206 * T$ is correctly found with an approximation of 0.202%.

4.3.2 influence of the accuracy computation

We have shown how the uncertainty rate can have an important influence on the final partitions. Furthermore, this influence is not relied to the treatment of the accuracy along the search : the good relationship can be pointed out using the initial version of ABACUS with an accuracy rate well chosen. However, in a different way, automatic computation of the accuracies on all variables, by influencing the variable accuracy, also influences how the values of the examples on this variable are δ-equal and thus the final partition. The most important is not therefore to choose the right initial accuracies but also to update the accuracies in the course of the generation of the new variables.

For instance, let us compare the percentage of values constant for the variable $\frac{V^2}{T}$ according to the uncertainty rate chosen. The minimum observed value is 400.69612 and the maximum one is 401.32234. With an uncertainty rate of 0.09%, all the values are found δ-equal, whereas it is no more true with an uncertainty rate of 0.03% for which only 60% of the values are found δ-equal. Therefore, if the uncertainty rate is fixed by the user at 0.03%, in the initial version in which the accuracy rate stays constant, the law will not be found. On the contrary, by updating the uncertainty rate according to the number and type of the operations involved, the variable $\frac{V^2}{T}$ will be found constant and the law will be discovered.

In our new version of the algorithm, high accuracy rates clearly may occur on the initial parameters, but we know in this case that it corresponds effectively to the noise in the experiments, and we are insured to be right when comparing the values of this parameter. Moreover, high accuracy rates stay being a problem just when the law to discover is quite simple and does not need many computations. Indeed, as the accuracy rates are decreasing proportionally to the number of steps of the algorithm, if the law is quite complex, the variable representing it will never in fact now take this large accuracy rate. Even in this case, the problem (high accuracy preventing discovering a solution) that appeared in the initial version have therefore been eliminated.

5 Dealing with tautologies

Tautologies are the source of new difficulties. Since at each step the system does not memorize the functions already applied but it keeps only the result under the form of a sum of products, there exits a definite chance to be stopped discovering a formula which is by simplification (the user must do) equivalent to a tautology. We present in the following two important ways we have added to the previous version of ABACUS in order to avoid such uninteresting solutions.

5.1 Pruning negligible terms

Since δ-equality must take the place of the standard equality, we must exchange the standard definition of negligibility with a new one taking into account the interval of definition of the values. We consider now that x_0, with $\delta(x_0)$ as its absolute accuracy is negligible with regard to y_0 if $\forall\, x_i \in [x_0 - \delta(x_0), x_0 + \delta(x_0)]$, $y_0 + x_i \in [y_0 - \delta(y_0), y_0 + \delta(y_0)]$. Thus, we obtain this theorem :

x_0 is negligible with regard to y_0
if and only if
$| x_0 | \leq \delta(y_0) - \delta(x_0)$

This theorem is then used to determine if a given variable is negligible with regard to another thanks to the definition :

x is negligible with regard to y
if and only if **for all** examples E,
the value of x at E is negligible with regard to the value of y at E.

This notion of negligibility between two variables is from now on used to control the generation of new variables. It is forbidden now to add or subtract two variables when one is negligible regarding the other. Consequently, when x is negligible regarding y, we exclude the possibility of finding a solution such as $y = y + x$, or even of the type $1 + x/y = 1$ (obtained by dividing y + x by y), those are unfortunately sometimes proposed as solutions by the initial version of ABACUS. In this way, we reduce the search graph significantly by pruning numerous uninteresting paths.

We have introduced a new concept which allows us to control this kind of tautology and even to avoid logical errors (or that may be seen as logical errors by the user). Moreover, although this notion of negligibility can be considered as an improvement independent of the one due to accuracy, it is unsafe to use it without knowing the correct accuracy of the variables. For instance, pruning the space search according to this concept within the initial version of ABACUS where the uncertainty can be 2% (default value) for all values can lead to eliminating interesting variables. The robustness of the pruning is thus increased by taking into account the accuracy of the parameters.

5.2 Simplifying formulas

An intermediary step in the building of equations is relative to their simplification, or numerical cancellation. As the desired law becomes more complex, more numerous variables are generated and thus, higher is the chance for ABACUS to fail by finding a tautology. Thus, this phase of simplification is needed not only in the user's interest but essentially to eliminate the tautologies.

5.2.1 the problem

In the ABACUS system, the form used to express the equations is a sum-of-products. For instance, $\frac{x}{y} * (a-b)$ is expressed in the system as $\frac{ax}{y} - \frac{bx}{y}$. An algorithm based on this notation to test whether a new term contains a partial cancellation already exists in the initial version of ABACUS and when the test is fulfilled, the system does not create the candidate term. ABACUS thus already deals with numerical cancellations such as : $\frac{a}{b} * (bc)$ simplified to ac.

Unfortunately, numerous tautologies remain that the initial version of ABACUS cannot detect. One family of them can be represented by : $\frac{X_1.A}{X} + \frac{X_2.A}{X}$ with $X_1 + X_2 = X$.
For instance, $\frac{x}{x-y} - \frac{y}{x-y} = 1$ is obviously a tautology.

The reason why the initial version of ABACUS cannot recognize it is the following. In order to follow the sum-of-products format, when it subtracts two quotients $\frac{x}{x-y}$ and $\frac{y}{x-y}$, it seems sufficient to generate directly the form $\frac{x}{x-y} - \frac{y}{x-y}$, and therefore, the system does not test if any cancellation is possible.

More generally, let us compare ABACUS' way of dealing with formulas and computer algebra rules indicated by Davenport, Siret and Tournier (1986) in order to have a canonic representation. There are four of these rules :

1- no rational in the expression of a quotient.
2- no integer must divide both the numerator and the denominator.
3- the main coefficient of the denominator must be positive.
4- no common divisor between the numerator and the denominator.

Unfortunately, in the initial version of ABACUS, just the first condition is verified.

5.2.2 a solution

Our work has therefore consisted in improving the system such that all formulas also verify the last three rules. The solution we propose is to apply to quotients well-known arithmetical rules :

- In all cases, we execute if needed the self-evident simplification, like when subtracting or dividing two identical terms.

If there is no quotient in all the involved terms, go to the third thereafter step. Otherwise,

- firstly, we transform all the terms such that they get the same quotient. In general, that needs to compute the smaller common multiple of all the quotients.
- Secondly, we simplify the dividend if needed, and if the result is a constant, we note that it is indeed a tautology, else, we simplify the quotient.
- Thirdly, if it is not a tautology, we reconstruct the equation as a sum-of-products.

Example : let us perform $\frac{a}{x+y} - \frac{a}{y}$.

We start with computing the common denominator : $(x+y).y = xy + y^2$, then, we compute the dividend : $a.y - a(x + y) = a.y - a.x - a.y$, we simplify into $-a.x$, and finally we return the final quotient $\frac{-a.x}{xy + y2}$. Since it is already in the sum-of-products form, the simplification stops.

The last rule is unfortunately not completely verified since we do not deal with factorization. That implies that $\frac{x^2 - 2x + 1}{x^2 - 1}$ and $\frac{x-1}{x+1}$ are not still recognized as been equal. But since our aim is only not to encounter tautologies, the main important point to make sure is that none solution of the form $\frac{x^2 - 2x + 1}{x^2 - 1} = \frac{x-1}{x+1}$ will be found. To obtain this solution, the system must perform the subtraction between the two terms. Now, the system must transform the expression by putting the them on the same denominator, we know that it will obtain 0 and thus it will not create any tautology.

5.2.3 additional improvement

In ABACUS initial, when two variables are combined in order to attempt the generation of a third one, a simplification check is performed, and as soon as a first simplification occurs, the generation of the third variable is disowned.

In our approach, we do not use this heuristic for the mean reason that it either forbids or slowly delays the generation of powerful variables. Furthermore, another interesting aspect of this improvement is the possible creation of squares of variables. Indeed, if the initial parameters are x and y, ABACUS has just two ways to create the square of x : firstly when there exists two groups such that the monotonic dependency between x and y is different for each of them. As these groups do not always appear, it may sometimes be impossible to generate the square of x.

A second way to obtain x^2 is doing a product of the form (x+y) *(x+z). By development, x^2 + xz + xy + yz will be found. Unfortunately, by this way and if the system does not allow simplifications, it is impossible to obtain the square of x alone. Now, by simplifying equations, it is possible to create in an additional way, the square of x (for instance, by subtracting xy from x (x+y)). An illustration of the improvement is given by the law : $x^2 + y + z$, which can be discovered only applying simplifications. Indeed, with the initial version of ABACUS, a solution is obtained, but only at the 998th variable and is a tautology :

$$x * y^2 / (x^2 * y^2 / (x + y) + x^4 / (x + y) + x * y^3 / (x + y) + x^3 * y / (x + y)$$
$$+ x^3 / (x^2 * y^2 / (x + y) + x^4 / (x + y) + x * y^3 / (x + y) + x^3 * y / (x + y)) = 1.0$$

which can be simplified to : $\frac{(xy^2 + y^3) * (x + y)}{(x^2y^2 + x^4 + xy^3 + x^3y)}$.

Now, by simplifying x(x + y) - xy, we obtain the square of x alone and the solution is obtained when the 86th variable is generated. This shows a huge improvement on the original version of ABACUS.

6 Improving the search algorithm

The search is done through a directed graph where each generation is the result of all the possible combinations of all the couples of all the previous generations. Since for each couple of variables two functions are created (according to the monotonic dependencies), if **n** is the number of initial parameters, each generation can be composed at most of twice the number of combinations of two elements between **n**, which corresponds to **2n(n - 1)**.

In average, only half of the possible combinations are realized, but the complexity of the search remains equal to $O(n^{2^p})$ if p is the maximal depth of the graph. The problem is then how to decrease the size of this graph.

One way to limit combinatorial explosion would be to find an evaluation function predicting which variable leads the quickest to the solution. Unfortunately, there are no known functions able to perform this test. We will describe the function chosen by ABACUS, its drawbacks and the principal improvement we have added which help to reduce each time the complexity.

One of the most important heuristic used by ABACUS is to split (according to a given evaluation function) at each generation the resulting nodes (or variables) into the a-nodes set containing the most "interesting" nodes or active nodes and the s-nodes set containing the others which can be viewed as "sleeping" nodes. The exploration starts then developing the a-nodes, returning to develop the s-nodes only when no solution has been found. A maximum depth can be fixed by the user and helps to prevent combinatory explosion. The order of development of the nodes is illustrated the figure 1 below by the numbers above the nodes.

6.1 Evaluation function of ABACUS

ABACUS' choice of evaluation function is the "degree of constancy" which represents for a function the percentage of the data for which the function evaluates to a constant. This choice is justified by the construction of the data which consists in gathering the examples by doing successive experiments for which the variations of two parameters are examined while all the others parameters are held constant.

For instance, if the law is $ab/c = cte$, a group of examples will be built to examine the dependencies between a and b. When the system computes the new variables, it finds that the values of a increase when those of b decrease, and it will build $a+b$ and $a*b$ according to the basic heuristics.

Once the new variables being generated, the system will have to sort them according to their degree of constancy and to cut the ordered set in half to obtain the a-nodes set and s-nodes set. But here $a*b$ will be found constant on at least one group while $a+b$ has a small chance to be constant on some examples. We can see here that the choice is directly dependant on the groups built by the user.

6.2 The problem

Unfortunately, the user does not always provide all the groups characterizing all the couples of parameters. Furthermore, especially if there are numerous data, it can happen that variables not leading to the solution can present a degree of constancy greater than those of "good" variables. All these conditions can lead the system to split the variables badly.

Another instance of splitting the variables badly is presented in figure 1. This figure represents the initial tree built by the system on a set containing 13 examples which describes the conservation of momentum. Eight parameters M1, V1, M2, V2, M1P, V1P, M2P, V2P, representing the mass and the velocity of the two objects before and after a collision.

We recall that the real law is : M1V1 + M2V2 = M1PV1P + M2PV2P.

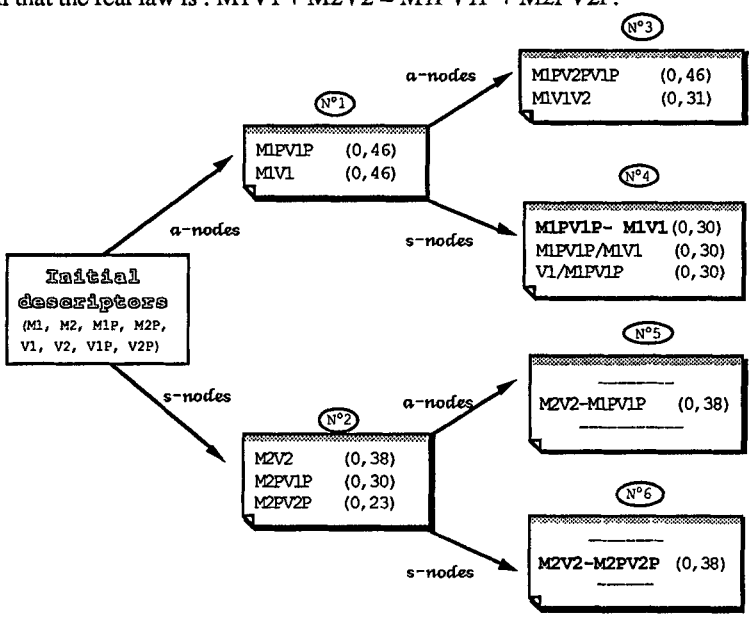

Figure 1. Variables splitting according to their degree of constancy
(numbers in parentheses give the accuracy of the variables)

We can note two important points :

- in the example, four among five variables are needed to create the equation and they must be gathered in the same set to be considered at the same time. Here, as the system is constrained to choose, it necessary splits the variables into two subsets, separating all the necessary variables. For this reason, the solution will be discovered only after the system has developed the subtree the root of which is the a-node n°1, after have exploring almost a hundredth of nodes. It is clear that systematically splitting the variables is not a good solution, especially when there are few of them.

- we can see an instance of a variable (M2PV1P in s-node n°2) which is not useful for discovering the solution which has however a better degree of constancy than a "good" variable (M2PV2P). That illustrates the well-known characteristic of an heuristic, that is to say, that a heuristic has a definite chance to fail in helping to solve a problem. Here, that implies that the known best nodes (necessary for building the solution) are not always those estimated with the higher degrees of constancy.

6.3 A new heuristic as a solution

To overcome these drawbacks of the algorithm, we restrict the heuristic of splitting the variables according to their degree of constancy by the following condition :

"If the variables of the generation are not too numerous,
do not split them into a-nodes and s-nodes."

We will estimate when the variables are too numerous not in absolute way, but with respect to the number of variables in the antecedent generation. For instance, hundred variables in the first generation may be considered as a great amount if there are only 20 parameters, but not in the case of 60 parameters. In this aim, we choose to set the minimum threshold for splitting to twice the number of variables in the previous generation. In this way, we are sure to reduce significantly the combinatorial explosion.

Furthermore, when there are few initial parameters, 10 or 15 for instance, even twice the number of them may be easily tractable. It is thus necessary to represent by a parameter the absolute minimum of variables under which it is not necessary to split. Let us call it *min-split*. The splitting of a generation gen_K will be controlled by the two hereunder rules :

" if count $(gen_K) \leq$ *min-split => do not split "*

and

" if count $(gen_K) >$ *min-split & count* $(gen_K) <$ *2 * count* (gen_{K-1}) *=> do not split "*

The results this new heuristic are shown in the table 3, which compares the number of nodes created when it is used with the number of nodes generated when it is not used.

discovered law	initial version	with new heuristic
$m1.v1 + m2.v2 = m1p.v1p + m2p.v2p$	250	107
$N1.\sin(\theta 1) = N2.\sin(\theta 2)$	47	82
$E = I.\cos(i) / r^2$	152	37
$I = V0.\cos(\omega t) / r$	25	35
$F = mv^2 \cos(\theta) / r$	146	121
$S = (x+y).(z+w)$	out of bound	65

Table 3. Results of the heuristic "do not split if not too numerous"

In the previous example of the conservation of momentum, the number of new variables (5) is even less than the number of initial parameters (8), thus the system will not split them. This corresponds to delaying the choice between these variables. Applying this heuristic to the example, we observe that the number of nodes created until the law is found is decreased from 250 to 183.

We can note that the importance of the addition of this new heuristic cannot be evaluated from the number of nodes. The last example is the most representative case : on one hand, a bad choice without heuristic leads to the impossibility to find a solution inside the boundary of thousand nodes, and on the other hand, the new heuristic has allowed a delayed choice which then leads to the solution within a hundredth nodes.

On the other extreme, two examples in this table show that when the choice was already the best one, the added heuristic does not improve the rapidity but on the contrary delays the discovery of the law. As we are interested in improving the system to diminish the search such that the number of nodes and the time spent stays reasonable, we consider that the addition of research the heuristic sometimes lead to is negligible compared to the large improvement it adds when the laws are complex and when the evaluation function is inefficient to sort the variables correctly.

With this heuristic, we have automatically eliminated a high proportion of failure due to a bad initial choice. Since these bad choices avoid a law to be found in the threshold of hundred nodes, the efficiency of this heuristic is self evident.

7 Conclusion and future work

We have presented here three complementary aspects of our improvement of the system ABACUS. These improvements do not change the basic principles used in the system but integrate simple and useful notions independent of the application domain, quality which is one of the most principal aim of our system. Moreover, these improvements can not only be applied to ABACUS, but to all systems belonging to the BACON family like FAHRENHEIT or IDS (Nordhausen & Langley, 1990). Unfortunately, we have not resolved completely the problem of finding the law in only one step. However, we think that numerous improvements can be still added. Therefore, we present some different aspects which worth been further analyzed and improved :

1- The research axis we have presented here concerning accuracy seems worth exploring further. Firstly, some problems are still not resolved : for example, the problem of dealing with trigonometric values, always belonging to the interval [-1, 1] and easy to confuse even with a small degree of accuracy.

An additional point remains to be studied : as the graph search is explored, variables appear which have more and more values such that they have an absolute accuracy rate higher than the value itself. Intuitively, when we are provided a value like 10 ± 20, we decide that it does not merit much attention. Let us then consider a variable such that all its values verify $\delta(x_0) > x_0$. Even if it leads to a solution, we are practically guaranteed that the solution proposed will verify a very low accuracy rate, and if another solution is provided with more reasonable accuracies, the former will be rejected.

We propose to fix a percentage T (it could be a parameter of the system) which would indicate that when the accuracies becomes higher than T times x_0, it is not worth keeping the variable. This heuristic could reduce in an important way the search graph, but needs to be carefully examined to avoid errors.

2- In the same way that we have introduced accuracy, we will continue examining which kind of knowledge could be integrated without restricting the application field. For instance, an interesting knowledge used in ABACUS is constituted by the measure units. Their importance is such that when they are not used, the complexity is increased in such a way that generally, the system cannot find a solution within the threshold of thousand nodes. It would therefore be interesting to find analogical information so much powerful.

Now, when looking at qualitative physics, we can verify that a lot of knowledge are also available in every domain : for instance, the relations or laws already known between the parameters, or even incompatibilities between different types of variables, others than the compatibility rules. For instance, the expert knows whether it is worth multiply the length of the cube and the temperature or not. Dealing with this kind of information, we would in some way relate qualitative physics to scientific discovery. Now, the problem is an acquisition one : how to gather all this information?

Acknowledgments

I would like to thank the following people who helped me a lot during this work : my thesis adviser Y. Kodratoff, M. Sebag and M. Schoenauer working at Polytechnic, H. Ralambondrainy who has worked at the INRIA in Diday's team, and all the mebers in the team Inférence et Apprentissage who helped me in the correction this paper, G. Bisson, K. Causse and A. Gordon.

References

Davenport J., Siret Y., Tournier E. *Calcul formel. Systèmes et algorithmes de manipulations algebriques*. Collection etudes et recherche en informatique, Eds Masson, Paris, 1986.
Eurin M., Guimiot H. *Physique*, Classiques HACHETTE, 1953.
Falkenhainer B.C., Michalski R.S. Integrating Quantitative and Qualitative Discovery: The ABACUS system. *Machine Learning Journal*, vol. 3, 1986.
Falkenhainer B.C., Michalski R.S. Integrating Quantitative and Qualitative Discovery. *Machine Learning: An Artificial Intelligence Approach*, vol III, R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), 1990.
Greene G.H. The ABACUS.2 system for quantitative discovery : Using dependencies to discover non-linear terms, MLI 88-17 TR-11-88, 1988.
Joyal M. *Cours de physique*, Vol. 3 Electricite, Eds Masson & Cie, 1956.
Langley P., Bradshaw G.L., Simon H. BACON.5: the discovery of conservation laws. *Proceedings of the seventh International Joint Conference on Artificial Intelligence*, p 121-126, 1985.
Langley P., Zytkow J., Simon H.and Bradshaw G.L. The search for regularity : Four aspects of scientific discovery in *Machine Learning: An Artificial Intelligence Approach*, volume II, Michalski R.S., Carbonell J.G., Mitchell T.M.(Eds.), Tioga, Palo Alto, Calif., 1986.
Langley P., Zytkow J., Şimon H. and Bradshaw G.L. *Scientific discovery . Computational explorations of the creative process*. MIT press, Cambridge, MA, 1987.
Nordhausen B., Langley P. A robust approach to Numeric Discovery", *Proceedings of the seventh International Conference on Machine Learning*, p 411-418, edited by B.W. Porter and R.J.Mooney, Morgan Kauffman Publishers, Austin, 1990.
Zytkow J. M. Combining many searches in the FAHRENHEIT discovery system. *Proceedings of the fourth International Workshop on Machine Learning*, p 281-287, Morgan Kauffman Publishers, Irvine, 1987.
Zytkow J. M., Zhu J and Hussam, A., Automated discovery in a chemistry laboratory. *Proceedings of the AAAI-90*, AAAI Press, p 889-894, 1990.