DAVID, M., DIMITRAKOPOULOS, R. and MARCOTTE, D. GEOSTAT1: A prototype expert system for the explicit knowledge approach to geostatistics. APCOM 87. Proceedings of the Twentieth International Symposium on the Application of Computers and Mathematics in the Mineral Industries. Volume 3: *Geostatistics*. Johannesburg, SAIMM, 1987. pp. 121 – 126.

GEOSTAT1: A Prototype Expert System for the Explicit Knowledge Approach to Geostatistics

M. DAVID, R. DIMITRAKOPOULOS and D. MARCOTTE

Ecole Polytechnique, University of Montreal, Montreal, Canada

This paper addresses the problem of technological transfer in geostatistics. Expert systems have become popular in certain fields because the needs of the industry are not provided for by traditional concepts and systems of education. The explicit knowledge approach in developing expert systems is presented through two experiments. The first describes a dedicated system written in GOLDEN LISP and the second a shell, using INSIGHT2, which has more general applications.

Using expert systems forces on the user a new approach to parameter definition, and as an example an original method for the research of any anisotropy axes is developed which might well be useful in traditional programming. Two examples of application to a coal and a gold deposit are given.

Introduction

Geostatistics as we know it today came into existence some twenty years ago. From an esoteric and obscure theory,[1] it has evolved into a working technique now used worldwide.[2] Scores of universities are teaching it, in Europe, North and South America, Australia, the Far East and South Africa. The majority of papers in the *Journal of Mathematical Geology* concern geostatistics, and several sessions are devoted to geostatistics at each APCOM meeting. Geostatistics can thus be considered a successful tool.

However, we constantly find that when the industry makes the decision to use it, very often it does not have the properly trained personnel to do so. Goodwill is not sufficient to do a good job. After twenty years of geostatistics, it can be stated that from the technological transfer point of view, it is a quasi-total failure. Good geostatisticians have often spent three or more years of graduate studies, but undergraduate students may be exposed to only a few hours of the subject. In the industry, very large companies can afford to send one of their staff away for two or three years, but the average mining company cannot. Developing countries are catching up, experts can be sent to leading seminars for a few weeks, but afterwards people are left on their own.

Hence, there seems to be a need for a method which will allow the end user to perform a decent geostatistical study with only a very short period of training. This is certainly not the ideal, but in the economic system we live in, this is making the best of our resources and opportunities.

Expert systems have been around for about a decade. They have been popularized by medical systems like MYCIN,[3] and the general public even reads about PROSPECTOR[4] in *Time* magazine and *Newsweek*. Other expert systems exist in the earth sciences, mostly in petroleum engineering, such as the Dipmeter advisor project.[5] These expert systems are hardly commercial, but trying to develop them generates a lot of thinking which ultimately should be beneficial.

The term 'expert system' was brand-new when the first version of PROSPECTOR appeared, and it may seem that a whole new vocabulary has emerged to represent what could have been developed within traditional frameworks. We will see that, in fact, it helps in generating new ideas and that the development of commercial expert systems oscillates in its focus.[5] On the one hand, one has to show the feasibility of the project and worthwhile use of expert systems tools, and on the other, one has to refine the domain knowledge, get deeper into the field of knowledge engineering and conclude a better inference engine to handle this new knowledge. For instance, we will study two inference engines, a shell, INSIGHT2 and a specially designed system written in LISP. We will present an overview of how we see an expert system and present two attempts at formalizing these concepts.

What is an expert system?

An expert system in geostatistics will start as an advisory system which interfaces with existing accepted programs

to help a user reach a correct and useful answer; hopefully, it will expand into a reservoir of new expertise and new knowledge which can be useful to a large number of users. To use a variogram program, for instance, one only has to be able to read and type to fill in a menu asking for the choice of several parameters.

To obtain a meaningful result, however, is another story. Some large companies have invested in the development of a 3-D variogram program to compute variograms in 64 directions, while other direct efforts at computing variograms on 20 000 blastholes at a time, and students spend hours to obtain a horizontal variogram with three drillholes. The advisory part of the system will try to establish the parameters to use by asking a number of questions, using a number of rules which have been embedded in a knowledge base after questioning experts, reasoning and solving contradictions.'

The different parts of an expert system are thus revealed: the knowledge base, established by questioning one or several experts; the inference engine, which may not be specific to a certain field; and the user interface with existing programs. The latter is a matter of traditional programming and although important, it will not be discussed here. Later we will see that the system may be weak on two points: the knowledge base may be insufficient and the inference engine may be inadequate. Different representations of knowledge may require different inference engines – INSIGHT2, which can only process rules, and our own system in LISP have already been mentioned.

Knowledge

It has been suggested[6] that in a universe of discourse, such as geostatistics, everything is represented in terms of symbols and processes implemented with physical relationships. For instance, a number is a symbol and only has a meaning when compared to other numbers. If we say that it is the grade of a sample at a location, it is a physical relationship, and the two together comprise knowledge. Knowledge is a collection of definitions, special facts, strategies, heuristics, relationships and algorithms.

The implicit knowledge formalism: the standard solution of present programs

Conventional programs are all based on the same scheme. Parameters are input to a method to obtain an output. Knowledge is present in the sense that a procedure will be carried out, but it may or may not succeed. The program has no consciousness of the input or what the output represents. We say that the program possesses implicit knowledge.

Making the program smarter: explicit knowledge

It is possible to improve the program by making it able to create, modify, reproduce or destroy knowledge. These are four characteristics of intelligence. The program can be made to decide the appropriate input and also evaluate its output because it contains knowledge. As it contains knowledge, it can also reason. The knowledge of the domain is explicitly declared in the program. A variogram algorithm is implicit knowledge, while a variogram conditioned by certain rules is explicit knowledge. For the program to possess explicit knowledge, we have to be able to make symbolic calculations. For instance, in variogram modelling, if we are considering a molybdenum deposit, it should not have a zero nugget effect. We must be able to process facts and make deductions. Listing all the facts which have to be considered is part of the knowledge engineering job, interviewing the expert to find out all the elements he intuitively processes before making a decision.

If we take as an example of a task the calculation of a variogram, we see that we will need several program modules. One is needed to select the parameters, one to run existing programs, one to evaluate and interpret the results and decide whether the job is finished or whether another run is required and, finally, one to fit a model.

Integrating geostatistical programs and domain knowledge

We will first have to acquire knowledge, then represent it functionally in the form of antecedent-consequent pairs drawn from symbolic objects or, in the case of a more involved inference engine such as the one developed in LISP, frames or networks. This will be the knowledge base. Then we will need an inference engine: we will use one especially developed for geostatistics, written in GOLDEN LISP[7] and we will consider a second general one, a shell, which has been equally successful in other applications, for wine selection with a meal, for example. Finally, there will be the interface with conventional subroutines for the execution of external tasks, result evaluation and interpretation.

GEOSTAT1: The prototype experiment

At present the system is limited to variogram calculations; this has been actually implemented and it works. We will see that this small example is sufficient to point to the areas where more work is needed and where we should go next.

The system is written in GOLDEN LISP which is available on PCs. LISP has been chosen because of its flexibility and relative speed for symbolic manipulations. It is modular, rules and data are part of the knowledge base, and it is possible to run only part of the program both at the top level of execution or lower level of consultation. For instance, we can break the program at any time to change a parameter we do not like.

As a knowledge-based system, GEOSTAT1 schematically consists of

(a) the knowledge base which is the program's storage of facts and the associations it 'knows' about geostatistical domain subjects, namely calculation of experimental variograms;
(b) the inference mechanism which structurally controls the processing of the knowledge base;
(c) the user interface for the collection of information

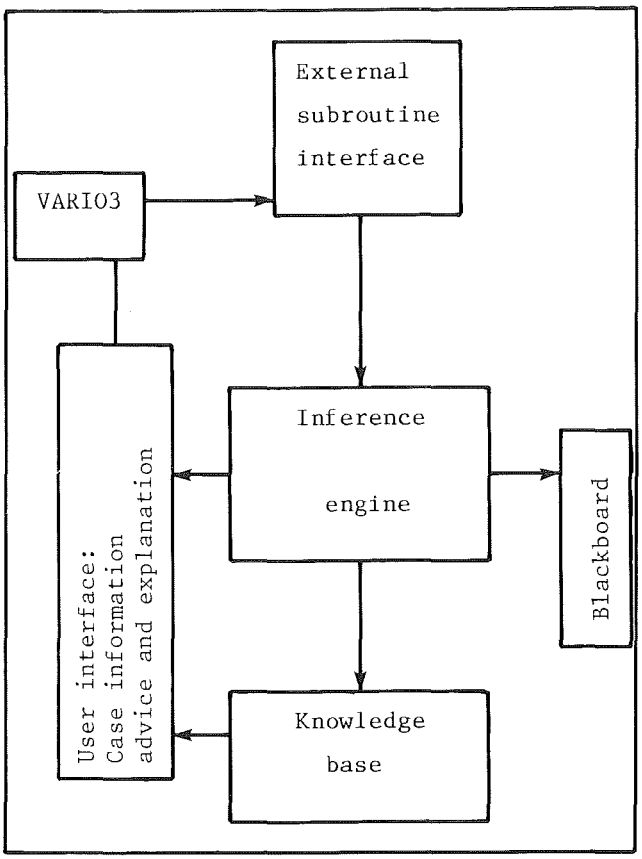

FIGURE 1. Main parts of GEOSTAT1. Arrows indicate information flow

which is transformed to facts related to the particular case;

(d) the interface with subroutine VARIO3[8] for variogram calculation, which at present is indirect owing to hardware memory limitations; and

(e) a 'blackboard' where numerical values of global variables are stored and accessed during consultations.

GEOSTAT1 is required to decide upon the values of certain parameters, such as the number of directions in which variograms will be calculated. In addition, the evaluation of the results of the variogram calculation is based on the evaluation of the numerical values of different parameters, as for example, the number of sample pairs used to calculate the first experimental point in a variogram. Deciding and evaluating numerical values has a controlling effect on the design of both the knowledge base and the inference engine of the system, as discussed below.

The knowledge base

Knowledge is represented by rules, frames and networks.

Rules are antecedent-consequent conditional statements, IF-THEN. For instance, a rule concerning the number of directions into which calculations should be made can be:

number of directions (IF (no. of samples is less than 75)
(THEN (no. of direction – NDIR is 1)

It can be seen already on such a simple rule that expert opinions may vary and that a better rule should be sought. Antecedents are statements about possible states of the world and are responsible for examining items in the database and testing them for matches. If a match is obtained, a consequence is triggered. Consequents, like antecedents, are symbolic objects. A first type of consequent contains geostatistical inferential knowledge, and a second controls the flow of a program, for instance, FINISH-PART2. After a rule is satisfied it will not be used further.

Frames[8] are of two kinds, one to store and retrieve a parameter and a second which contains the definition of geostatistical entities. Examples are shown in Figure 2.

Networks[9] can be seen as equivalent to complex rules of the type IF/A THEN B, possibly cascading.

Already we have 40 rules, 11 frames and 10 networks – all this for a rudimentary experiment which has been actually implemented! One can easily imagine the work

```
(INPUT-FOR-VARIO3    (No-OF-DIRECTIONS-NDIE    (MUST-BE-SET-TO   4 ))
                     (MAX-STEP-STEP            (MUST-BE-SET-TO   10 )


|
|
|
(FPUT  '2D-GEOMETRICALLY-ANISOTROPIC-3DV  'NUGGET-EFFECT  'IS  (*0.5
                                        (+ NUGGET-1 NUGGET-V))    )
(FPUT  '2D-GEOMETRICALLY-ANISOTROPIC-3DV  'SILL  'IS (*0.5  (+SILL-1
                                                    SILL-V))    )
(FPUT  '2D-GEOMETRICALLY-ANISOTROPIC-3DV  'RANGE  'IS  (MAX  RANGE-1
                                                   RANGE-V)     )
(FPUT   '2D-GEOMETRICALLY-ANISOTROPIC-3DV  'ANISOTROPY-RATIO  'IS (/
              (MAX RANGE-1 RANGE-V)/(MIN RANGE-1 RANGE-V)    )    )
(FPUT       '2D-GEOMETRIC-ANISOTROPY-3DV       'DISTANCE         'IS
                                  'FUNCTION - OF - ANISOTROPY-RATIO)
```

FIGURE 2. Example of the two types of frames. In type 1 a parameter is stored which can be retrieved; in type 2 a definition is given

required to create a full database for a real and complete system. This is where the art of interrogating the knowledge of the experts must be developed.

The inference system

This proceeds in a forward-chaining fashion. Antecedents are matched against facts (data), rules are triggered and consequents are added to the database. The process is repeated until no rule is triggered. Antecedents are matched to facts in the database through a pattern matching function which accepts only constants in the pattern; this allows no missing symbols. When all the antecedents in the IF part of a rule match the known facts, then the tag, antecedents and consequents are put into separate streams. These streams are then used to provide explanations on deductions made. This is the reasoning.

The program

The program contains three parts: the initialization of the parameters to run VARIO3, the evaluation of the results and finally the suggestion of a model, after the system has decided that the second part should not be repeated further.

The initialization stage

The first task is the collection of information from the user. Every piece of information is transformed to a fact (for instance, we are working in 2D) and stored in the database. Once all the information is collected, the inference starts. An overview of the consulting process is shown in Figure 3.

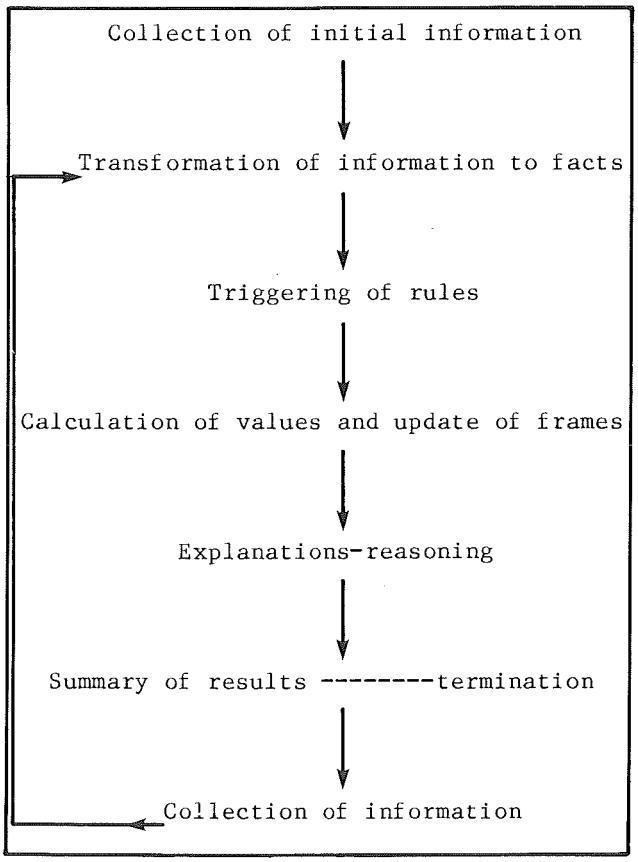

FIGURE 3. Overview of consultation process in GEOSTAT1

```
Give me the minimum Y coordinate:    -10
Give me the maximum Y coordinate:     10
Give me the minimum Z coordinate:    -10
Give me the maximum Z coordinate:     10
How many samples are involved?       282
Do you know that the data are isotropic?
                        (If don't know press N)
                        (Y or N) No
```

FIGURE 4. Example of questions asked in the initialization phase

An example

We have selected an example from a Pennsylvania coal field as provided by Consolidated Coal. This is an easy 2-dimensional situation with 112 data points fairly well distributed. The variable considered here is thickness × ash × density.

After a welcoming message, the program starts to ask simple questions to which one answers 'yes' or 'no' or certain numbers, as can be seen in Figure 4.

Then the system states its conclusions for the values it has found for the parameters and asks whether we want to know why these conclusions have been reached. If we answer in the affirmative, it will list the rules it has used in the reasoning.

The program then proceeds to evaluate the results of VARIO3 according to the number of pairs in the first point of the variogram. According to this value, it may decide to change the interval for the calculation of the variogram or the regularization angle.

Then, in a third part, it will ask for an estimation of the parameters of the variogram in the different calculated directions, and it will conclude with a final model (Figure 5). The final model adopted in this case is spherical, isotropic with range 17 400 feet, nugget effect 0,11 and sill 0,32 (Figure 6).

```
I have concluded PART-3 and now I will give
you a summary of my conclusions.

The variogram model is:  4D-ISOTROPIEC-2DV
    WHERE
the value of the NUGGET-EFFECT is:  0.11
the value of the SILL is:  0.32
the value of the RANGE is:  17.425
Well, I have done my best to help you.   Now
our session is over.
If  you want to have another round you  must
call me again.  BYE BYE ROUSSOS
```

FIGURE 5. Final result reached for the variogram

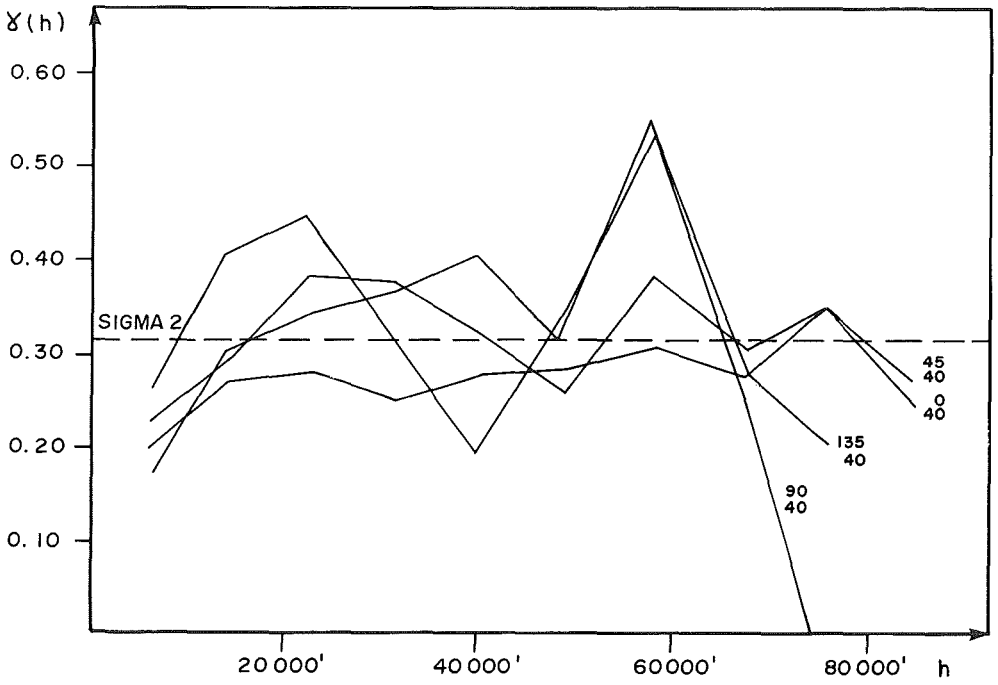

FIGURE 6. Variogram of ash × density × thickness

Second example using INSIGHT2

INSIGHT2 is a shell for expert systems. It will process rules and help reach a conclusion. It does not possess arithmetic functions other than +, −, ×, and does into accept general rules. All rules have to be specified; for instance if we want to find in a group whether x is the grandfather of y, we must have all the facts with actual names, like John is the father of Jim, Jim is the father of Bill, then John is the grandfather of Bill. In the small example of variogram calculation, this has not been found to be a limitation. It may later on. A subroutine has been written to approximate a square root. Speed of calculation is equivalent to LISP in our example, and the memory required is much less (512 K versus 1,2 Meg.).

The coal data example

A test was performed on thickness. An initial test showed slow convergence in the definition of the proper calculation interval, which led to a change in the choice of the initial interval from the minimum distance between two holes to an initial guess. With 100 samples the program will oscillate 5 or 6 times before finding an adequate interval.

If variograms are computed in four directions, tests will be made for the coherence of ranges. If there are contradictions, the variograms are recomputed after a rotation of /8.

SIMOR data

SIMOR is a simulated gold deposit used for teaching purposes. Students have $100 000 to drill this subvertical volcanogenic gold deposit. One student database containing 280 samples was used as a test. In this three-dimensional example, we have the question of identifying the principal directions of anisotropy. For this, an original technique was developed. We compiled the principal components of the coordinates of samples above a high cut-off (the median, for example). This method correctly identified the three axes as parallel to the usual coordinate axes when in fact all the drilling was made with a 45° dip.

The program then correctly computed an isotropic variogram in the vertical north-south direction (along strike) but failed to get a model in the east-west direction, stating that there were not enough pairs. This is what should be expected as only two drillholes per section were available.

Lessons learned

The variogram program used is basically ten years old, and few if any new ideas have been published on this subject. Running our expert systems, a number of problems were found which led to new ideas in standard programming. It was also found that the first rules gained from experts only tell part of the reasoning which they really perform. A better standard of teaching can result by forcing these rules to be spelled out.

On the 3-D example, one problem is to identify the principal axes of variations of grades. Sometimes grade trends follow the geometric shape of the deposit, sometimes not. A new approach is to look at the general trend of high grade values using a principal component analysis on the coordinate of samples above a high cut-off. An example on a simulated deposit containing only 280 samples correctly identified the main axes of variations. One could use the same idea to define homogeneous areas in a large deposit, computing the principal components on coordinates of high grade samples and mapping their variations. Also to perform kriging with a locally variable variogram, one could compute the orientation for each block from a PC analysis of the high grade samples available. As computers become faster, this may not be unfeasible.

Where there were a limited number of samples, problems were encountered with the definition of the step. They clearly showed first of all that a different step should be specified for each direction. For ten years, variograms were computed by some researchers using the same step in all directions. This should be changed. Such a step completely alters the architecture of the program. Also, it was realized that there is nothing which says that a variogram should be computed at regular intervals. In fact, it would be more logical to have short steps on short distances and larger steps at larger distances because of the diminishing number of pairs. This would make variograms look better without any loss of information, and it would ease the model fitting.

The lessons we are learning are the same as those which other expert systems builders have found. Smith and Baker, for example, realized that using the system, experts themselves change. This is exactly what happened to us in defining STEP. It is also traditional wisdom that a task should be carefully defined before the system is designed, but in fact it has been found that in order to advance, one should not be too rigorous, and the process will evolve iteratively and show its weaknesses when we try it. Similarly, one should not try to have a 'complete' knowledge base before starting, because again the voids will show up better when one starts running examples, and one then tries to squeeze the experts more or consult different experts.

Conclusion

Altogether, it is believed that the efforts outlined here can be beneficial in several directions. First, it should eventually fulfil our initial goal of speeding up technology transfer and help generate good quality fast studies. It also forces the experts to revise their assumptions and may generate better traditional programming. Whether one ends up with a true expert system or simply better interactive programs, progress will have been made.

Acknowledgement

This research has been financed by the National Research Council of Canada, grant NRC 7035. The financial help of Ecole Polytechnique de Montreal is also appreciated.

References

1. MATHERON, G. *La théorie des variables regionalisées*. Masson, Paris, 1964.
2. MOUSSET-JONES, S. Mineral reserve estimation of gold deposits. A survey of practices in ore reserve estimation. *Methods, Models and Reality*. C.I.M.M. Symposium, 1986. pp. 172 – 184.
3. BUCHANAN, B.G. and SHORTCLIFFE, E.H. *Rule Based Expert Systems: The MYCIN Experiments of the Heuristic Programming Project*. Reading, Mass., Addison Wesley, 1983.
4. DUDA, R.O. and REBOH, R. AI and decision making: the PROSPECTOR experience. In *Artificial Intelligence: Applications for Business. W. Reitman*. Norwood, N.J., Ablex, 1984.
5. SMITH, R.G. and BAKER, J.D. The Dipmeter advisor system. *Proc. Eighth Intl. Joint Conf. on Artificial Intelligence,* 1983. pp. 122 – 129.
6. NEWELL, A. and SIMON, H.A. A program that simulates human throught. *Computers and Thought. Feiberbaum, E.A. and Feldman, J.A.* New York, McGraw-Hill, 1963.
7. *GOLDEN LISP*. WINSTON, P.H. and HORN, B.K.P. Addison Wesley, Menlo Park, 1986.
8. GEOSTAT SYSTEMS INTERNATIONAL. *VARIO3 User's guide*. Geostat Systems International, Montreal, 1981.