EXTRACTING USEFUL ADVICE rFROMCONFLICTING EXPERTISE1

Rent Reboh.
Artificial Intelligence Center
SRI International
Menlo Park, CA 94025

Dedicated To
John Gaschnig

ABSTRACT

A method for automatically identifying areas of disagreement and their sources is presented for multiexpert knowledge-based systems in the context of the Prospector consultation system. It employs performance evaluation techniques in combination with the explanatory facilities present in many expert systems to assist the user of an expert system in deciding which among several possibly conflicting expert opinions he should choose.

1. INTRODUCTION

An important issue encountered in the construction of expert problem-solving programs is how to deal with conflicting expert advice. This question arises even more tangibly in real-life decision-making—whether in the courts, the financial world, or in the context of our political and social institutions. Typically, groups supporting different sides of an issue enlist the services of highly respected experts who, as often as not, arrive at diametrically opposing conclusions or suggest conflicting courses of action. Even when experts do arrive at the same conclusion, how do we know they reached it for the same reasons7 The process by which the decision-maker, who presumably is not an expert himself (otherwise we might, witness another conflicting opinion), finally decides whose advice to follow is at best highly subjective.

A significant consequence of the development of knowledge-based expert systems is that they render the reasoning processes of the expert more explicit and therefore open to scrutiny, evaluation, and testing. Most expert systems provide explanatory facilities that can communicate these reasoning processes to the user, thus enabling him to accept the system's opinion with greater confidenee--or at least providing him with a tool to assess their applicability to his problem. The question then, is: how can we take advantage of this explicit representation of expertise to design more objective multiexpert decision support tools?

The knowledge bases of many existing expert systems are each the product of contributions from several experts, and could therefore be termed multiexpert knowledge bases. Typically, the section of the knowledge base contributed by one expert encodes his particular area of expertise. For the purpose of this paper, however, we shall refer to these knowledge bases as nonconflicting multiexpert knowledge bases, restricting the term multiexpert knowledge, base to refer to the encoding by two or more experts of the same area of expertise.

Because a multiexpert knowledge base encodes expertise obtained from several expert sources, segments of that expertise will undoubtedly be duplicated, while other segments might vary in diverse ways. For instance, source A could believe more strongly than source B in the association between evidence E1 and hypothesis H; source C, on the other hand, might not consider E1 important at all, but might ascribe more relevance to some entirely different observation, E2- Obviously, there will be many interactions and inconsistencies in the resulting knowledge base. The problems encountered in representing such interactions, however, are not peculiar to conflicting multiexpert knowledge. In fact, similar interactions are encountered in nonconflicting knowledge bases, where the problem of maintaining the consistency of the knowledge base as it grows must be dealt with in a similar fashion. In [7] and [8| we explain in detail how such conflicts are discovered and resolved in the Prospector environment.

For simplicity, and without loss of generality, we shall assume in the ensuing discussions that we have encoded in some suitable expert system the opinions of just two experts, A and 13, about a particular subject. The conclusions arrived at in the course of consultation with such a system might correspond, essentially, to the following:

> If you were to consult with expert A,
> be would suggest ...
> If jou were to consult with expert B,
> he would suggest ...

For most user categories these conclusions are unsatisfactory because the user still has to make the final decision. If an appropriate explanatory facility is provided in the expert system, however, he might be able to probe the rationale underlying each expert's conclusions, identify the sources of possible disagreement, and perhaps derive a consensus opinion.

This research was supported in part by the United States Geological Survey under Contract No. 14-18-001-20717.

There are several techniques that can be readily implemented in most expert system environments to assist the user in arriving automatically at such a consensus opinion.[2] Examples of such techniques include the following:

- Assigning weights to the various sources of expertise to reflect the user's belief in the respective competence of each source, then having the system combine opinions obtained from each source according to those weights. For example, if at some installation the expert system is to be employed primarily to solve a particular class of problems, the weights can be distributed among selected segments of the knowledge base to reflect the experience of the people at that site and their belief in the applicability of such segments to the problems and situations most frequently considered.

- Adding a decision analysis interface (employing rule-based techniques or traditional decision analysis methods) to serve as a back end to assist the user in filtering and synthesizing the opinions of the various experts.

In both these approaches, specialized (meta) knowledge must be encoded in the decision support tool. For instance, the knowledge base of the decision analysis interface in the second approach must encode strategies for recognizing (perhaps through additional interaction with the user) the situational context in which a given opinion has been rendered. Furthermore, the knowledge base must know in which contexts one opinion is more valid than another, as well as take into account such special user preferences as degree of risk aversion, company policy, and so on.

The remainder of this paper describes an approach that does not require such special knowledge, but relies instead solely on the contents of the expert system's knowledge base The method assumes a representation of the knowledge base that supports an explanatory facility and allows performance and sensitivity analyses to be performed on selected segments of the knowledge base. We shall outline this technique in the context of Prospector, a knowledge-based consultant for mineral exploration [2, 3, 1). Although we have not applied this methodology directly to the reconciliation of conflicting expert advice in actual consultntions with Prospector, similar techniques were used during knowledge base construction to detect and resolve the contradictions that often arise when new expertise is merged with an existing knowledge base In order to elucidate important aspects of the proposed approach, we shall in the next section briefly review the representational formalisms, explanatory facility, and performance evaluation techniques employed in the Prospector environment; we shall then show how these features can be combined to automatically provide the user with impartial assistance when he is confronted with disparate expert opinions.

2. PROSPECTOR'S TECHNIQUES FOR KNOWLEDGE REPRESENTATION, EXPLANATION, AND PERFORMANCE EVALUATION

2.1. Inference Network Representation

Prospector's knowledge base consists primarily of models that describe important classes of ore deposits. The models encode experts' knowledge of the associations between field-observable evidence and relevant geological hypotheses. A hierarchical (acyclic) network structure called an inference network is used to group these associations in ways that make explicit the judgmental reasoning processes employed by the expert to solve geological problems Inference networks can be viewed as providing a simple language that can be utilized by a knowledge engineer for communicating with a domain expert. This language allows the expert to specify (a) the factors that are relevant for solving a problem, (b) the paths for propagating information from a given factor to a hypothesis by explicitly indicating which factors affect others, and (c) how degrees of belief in these factors should be combined to reach a conclusion.

Prospector's inference network language provides a set of standard primitive procedures for updating a probabilistic measure of belief in a given factor by combining the belief measures computed for the factors that affect it. The XYZ model of Figure 1 illustrates some of the representational constructs employed in Prospector. Without fully explaining the manner of interpreting these inference diagrams, let us simply observe that, in effect, various factors (represented by boxes) are combined by plausible inferences (arrows with which two rule strength numbers are associated), logical inferences (arrows to boxes containing AND. OR, or NOT), or contextual relations (dashed arrows). Numbers on the right corners of boxes are prior probabilities associated with the corresponding factors. The strength of the rule labeled L(CPY) varies over the range of values (i.e., pyrite concentrations) associated with the node CPY. P(CPY) defines the prior probability distribution associated with cpy.[3]

Once obtained, an inference network can be interpreted in varying ways to perform different tasks. For example, during a consultation the interpreter is concerned primarily with selecting a suitable line of reasoning and determining which would be the most important data to ask the user about. The interpretation thus results in an interaction with the nscrto ascertain which of the models encoded best matches the prospect described by the user's answers.

[2] We know of no expert system, however, in which such techniques have been attempted directly.

[3] The inference network language supports several additional constructs. In particular, in developing Hydro, a knowledge-based interface to a hydrological modeling system [6, 9|, several constructs were added to extend Prospector's probabilistic inference mechanisms to reasoning about numerical quantities.

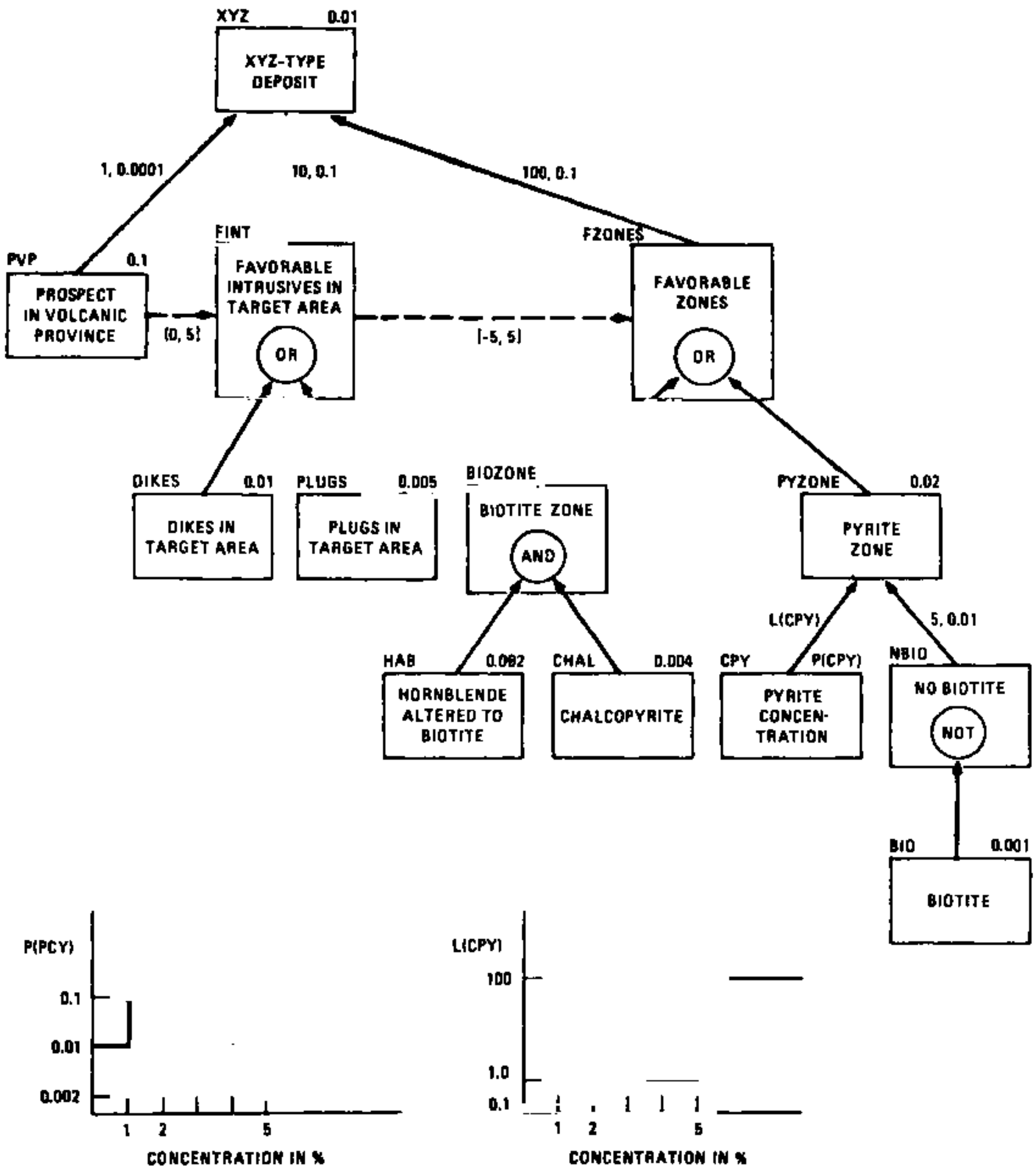

Figure 1: A PROSPECTOR INFERENCE NETWORK

2.2. Explanatory Facility

In addition to the simple rule-backchaining employed by many expert systems to explain their line of reasoning, Prospector is capable of a sophisticated interpretation of the state of its knowledge base for the purpose of furnishing more useful explanations anytime during a consultation. For instance, by performing both a best- and a worst-case analysis of the possible effects of rules invoked to establish a hypothesis, Prospector can inform the user of the most critical factors that contributed to its conclusions or that have the potential to change them. This often suggests to the user what actions he can take if, in fact, lie should wish to alter Prospector's conclusions or at least increase their reliability. This situation is particularly common in the early stages of exploration when, because the user supplies relatively little information to the system, very uncertain conclusions are generated. By displaying the reasons for its uncertainty, Prospector is in effect suggesting to the user that, to achieve a more certain conclusion, he should consider taking some proposed action—such as performing geochemical analyses, drilling a hole, and so on. Other features of the explanatory system stem from Prospector's ability to relate user-volunteered information efficiently to its knowledge base (using a semantic network matcher; see [7, 8j). For instance, the explanatory system can distinguish between evidence that has been inferred through the application of expert rules and evidence that has been directly or indirectly supplied by the user. It is also able to explain why the system's control strategy asks about a particular factor as well as why it is focusing on establishing a particular global hypothesis.

2.3. Performance Evaluation Techniques in Prospector

The exact methodology and tools employed are discussed in detail in [2, 5, 9]. A brief outline of the most important aspects of our approach is given below.

2.3.1. Relative Comparison of Prospector with the Expert

Because there exist no objective quantitative measures for evaluating the performance of human geologists, few experiments have been attempted to determine just how good Prospector's conclusions are in absolute terms (see [l], however, for one such successful experiment). We have concentrated instead on assessing the validity of Prospector's conclusions, as compared with those of the specialists who provided the models, when the human experts are presented with the same situation and input data. Selected test cases typically include not only regions that are good exemplars of the model being evaluated, but also some poor matches and false-alarm situations. For each test case, Prospector's conclusions arc compared with the expert's assessment (the *target* conclusion). Comparisons are performed for the top-level hypothesis in each model (usually corresponding to the likelihood that a particular type of ore deposit will be present in the considered location), as well as for several intermediate hypotheses (i.e., lower-level sections of the inference network).

2.3.2. Sensitivity Analysis

Because a degree of certainty can be associated with input data, we also investigate the sensitivity of Prospectors conclusions to perturbations in the certainties of the inputs. To perform this analysis, Prospector is applied (in a batch mode fashion) to each of the test cases while the input data are systematically modified. At least two runs are executed in which input data are modified towards increased and decreased certainty, respectively. On the basis of Prospector's certainly scale of -5 to 5, all inputs are moved towards 0 in the *less certain* run and towards the endpoints -5 or 5 in the *more certain* run. Conclusions from these runs are compared with the original *standard* run and the differences then analyzed. Here too the analysis is performed both for the top-level hypothesis and for several intermediate hypotheses.

2.3.3. Performance Evaluation and Sensitivity Analysis as Diagnostic Tools

Several tools (e.g., one for generating scattergrarns automatically on a bit map display) have been implemented in the Prospector environment for analyzing the result* of performance evaluations and sensitivity analyses As a consequence of these experiments, areas of disagreement between the expert and Prospector can be easily and accurately identified and their sources isolated. Furthermore, a careful interpretation of both the performance evaluation and the sensitivity analysis results makes it possible to establish a priority ordering to determine which sections of a model would most benefit from further revision and refinement—and to ensure that parallel conclusions of Prospector and the expert are indeed based on the same considerations.

3. APPLYING EVALUATION AND EXPLANATORY FACILITIES TO GENERATE CONSENSUS

In the foregoing section we outlined how, in Prospector, a model is compared with its human author. If, instead of being elicited from the human expert, the target conclusions were derived from another model encoding divergent expertise, it is easy to see how performance evaluation techniques similar to those outlined above can be applied to compare the two models. Because two models encoding opinions of two different experts might differ entirely as to which factors are considered to solve a particular problem and as to the manner in which the relevant knowledge and data are structured, the system's explanatory facility can be utilized to pinpoint areas of disagreement (as well as of agreement); sensitivity analysis techniques can be used to assess the criticality of the differences. More importantly, the source of such disagreement can be traced through the inference network to nodes that are shared by both models (i.e., factors considered by both experts but to which different effects are attributed) or to nodes contained in at least one of the models (i.e., factors considered relevant by some expert), and that correspond to observable evidence.

Conclusions generated in consultation with a system containing these features might look like the following:

```
If you were to consult with expert A,
      he would suggest ...
If you were to consult with expert B,
      he would suggest ...

However,

   IF you believe that observation E1 has
      a strong positive effect on
      intermediate hypothesis H1
        THEN I suggest you follow
            A's advice.
   IF, on the other hand, you don't
      believe THAT, and you believe
      strongly that observation E2
         (which, by the way, has not
          been considered by A)
      has anegative effect on intermediate
      hypothesis H2,
        THEN you should follow
            B's advice.
   OTHERWISE supply me with the following
      additional information: ...
```

If the user (i.e., the decision-maker) confronted with the above recommendation has no idea whatsoever regarding the relationships between observation E1 and hypothesis II, or between E2, and H2, he might conceivably probe further into *both sides of the issue* by enlisting additional experts. This time, however, they will be concerned with a much smaller subset of the problem: the relationships between E1, and H1 and between E2 and H2- It is possible that a finer-*grade* knowledge base with expertise relevant to that problem subset will have already been encoded, so that the same methodology could be applied recursively to the models encoding the smaller domain. Ultimately, because each subsequent reformulation of the sources of disagreement entails an even more constricted domain of expertise, the user will be capable of deciding the issue himself.

Finally, let us briefly describe how Prospector can represent and propagate the effects of conflicting rules. Let us first observe that no additional mechanism is needed to handle nonconflicting sections (i.e., identical) of both models or sections that occur in only one of the models. We must, however, be prepared to deal with conflicting rules (i.e., having conflicting rule strengths) that link the same evidence to the same hypothesis in different models. The case in which the conflicting rules have left-hand sides (or right-hand sides) that are different, yet semantically related, can be reformulated into the simpler case of identical left-hand sides (or right-hand sides). A method for performing this reformulation in Prospector is described in [7, 8]. It uses partial matching techniques to discover inference network nodes with overlapping semantic content and, if necessary, additional rules are automatically generated to interconnect the related concepts in a numerically consistent manner.

At present, when Prospector discovers conflicting rules with identical left- and right-hand sides, (e.g., in the process of adding a new model to the existing knowledge base), it declares an inconsistency; the knowledge engineer must then resolve the situation by talking to the experts and modifying the inference networks. In a system that tolerates the coexistence of nonconcuring expertise, we can leave the conflicting rules--as long as the paths for propagating their effects through the inference network are kept distinct. Prospector has a mechanism for maintaining such a distinction when necessary. Called the P*ROC* mechanism [2], it allows Prospector to be used recursively when, in the process of establishing some hypothesis, it must solve another classification suhproblem. The sections of the inference network that encode the alternative hypotheses concerning the subproblem can be executed as often as necessary to arrive at a solution for the snbproblem that can then be propagated through the network in the usual manner. To allow for the recursive use of sections of inference networks, a simple property list structure—rather than a single slot-is associated with certain fields (e.g., probability, value, etc) that, in turn, are attached to nodes in the networks. Values stored in these fields can thus be maintained (i.e., modified, propagated, examined, and so on) relative to distinct environments. In a multiexpert system, each expert can be regarded as constituting such a distinct environment.

4. CONCLUSION

When we construct a knowledge-based system, we attempt to encode approximations of an expert's problem-solving skills in some domain. There are limits, of course, to what any particular representation of such expertise in machine-interpretable form can express; some subtleties of specialized human know-how remain invariably unaccounted for. For these and other reasons it will be a long time before an expert system is developed whose advice will actually be superior to that of the experts who supplied its knowledge base. At best, by virtue of the systematic-analysis capabilities of computer programs and the fact that a knowledge base can contain expertise obtained from a number of experts, we can affirm that such a system indeed has the potential to become a better *overall* expert than any of the individuals who helped build its knowledge base. In any event, by employing techniques such as those summarized here, we can show that an expert system, in addition to its ability to exhibit occasionally impressive competence in a subject area and explicate the reasoning behind its conclusions, can offer even greater utility as an objective and impartial decision-support tool.

We believe that these techniques are sufficiently general to be readily implementable in other expert system environments.

Rather than display a complete analysis such as that illustrated here, Prospector's explanatory system allows the user to interactively explore only selected paths of the inference network pursuing each path to any desired depth.

ACKNOWLEDGMENTS

I have had many discussions with John Reiter in preparing this paper. He also conducted a substantial number of the performance assessment experiments for both Prospector and Hydro, implementing many evaluation tools. Savcl Kliachko read several draft versions of this paper and his helpful comments are greatly appreciated.

Many computer scientists and consulting geologists have contributed to the development of the Prospector system. The contributions over many years of Richard O. Duda, Vter *F. Hart*, and the late John G. Gaschnig are particularly relevant to the work described here. Jonathan Slocum made significant improvements in Prospector's explanatory system.

References

[1] Campbell, A. N., V. F. Hollister, R. O. Duda, and P. E. Hart, "Recognition of a Hidden Mineral Deposit by an Artificial Intelligence Program," *Science,* Vol. 217, pp. 027-929 (September 1982).

[2] Duda, R.O., P. E. Hart, P. Barrett, J. G. Gaschnig, K. G. Konolige, R. Reboh, and J. Slocum, "Development of the Prospector Consultation System for Mineral Exploration." Technical Report Final Report, SRI Projects 5821 and 6115, Artificial Intelligence Center, SRI International, Menlo Park, California (October 1978).

[3] Duda, R.O , P. E. Hart, K. G. Konolige, and R. Reboh, "A Computer-Rased Consultant for Mineral Exploration." Technical Report Final Report, SRI Project 6415, Artificial Intelligence Center, SRI International, Menlo Park, California (September 1979).

[4] Duda, R. O., "The Prospector System for Mineral Exploration." Technical Report Final Report, SRI Project 8172, Artificial Intelligence Center, SRI International, Menlo Park, California (April 1980).

[5] Gaschnig, J. G., "Development of Uranium Exploration Models for the Prospector Consultant System." Technical Report Final Report, SRI Project 7856, Artificial Intelligence Center, SRI International, Menlo Park, California (March 1980).

[0] Gaschnig .J. G.. R. Reboh and J. Reiter, "Development of a Know ledge- Based Expert System for Water Resource Problems." Technical Report Final Report, SRI Project 1619, Artificial Intelligence Center, SRI International, Menlo Park, California (August 1981).

[7) Rehoh, R., "Using a Matcher to make an Expert Consultation System Behave Intelligently," *Proceedings, First Annual National Conference on Artificial Intelligence,* pp. 231-234, Stanford, California (August 1980).

[8] Reboh, R., "Knowledge Engineering Techniques and Tools in the Prospector Environment." Technical Report Technical Note 243, Artificial Intelligence Center, SRI International, Menlo Park, California (June 1981).

[9] Reboh, R., J. Reiter and J. G. Gaschnig, "Development of a Know ledge-Based Interface to a Hydrological Simulation Program." Technical Report Final Report, SRI Project 3477, Artificial Intelligence Center, SRI International, Menlo Park, California (May 1982).