1991

# Learning: An Experiment with Five Selection Forms

John R. Rice
*Purdue University*, jrr@cs.purdue.edu

Report Number:
91-043

**LEARNING: AN EXPERIMENT WITH FIVE SELECTION FORMS**

John R. Rice

CSD-TR-91-043

June 1991

# LEARNING: AN EXPERIMENT WITH FIVE SELECTION FORMS

J.R. Rice*
Department of Computer Science
Purdue University
West Lafayette IN 47907
Technical Report CSD-TR-91-043
CAPO Technical Report CER-91-18

June 17, 1991

**Abstract**

Several approaches to learning to solve problems are identified in a previous paper: *Learning, Teaching, Optimization and Approximations* to appear in the book *Intelligent Scientific Software Systems*. Here we consider a simple, hypothetical problem related to Example 4.4, *Identification of underwater objects*, of the previous paper. Two scenarios are considered for solving this problem, one where two key problem features are known and one where they are not. The effectiveness of five selection forms, mathematical, rules, decision, neural nets, and exemplars, are studied for this hypothetical problem. The objective is to illustrate the nature and power of these selection forms in a simple context.

*Work supported in part by NSF grant CCR-8619817.

## Contents

## 1 INTRODUCTION

We consider a hypothetical problem abstracted from Example 4.4 of [1]. The problem is to identify underwater objects from sonar data and water conditions. Twenty-five "raw" problem variables are measured from which two problem features are computed. The problem is to identify the signals as one of "Rock", "Cylinder", "Other", and "Impossible". The latter means equipment malfunction while "Other" might be several rocks, a large school of fish, etc. We assign equal weight to knowing each solution value.

Two scenarios are considered, one where the features are known and one where they are not. Further, for the purpose of evaluating the effectiveness of different selection forms for learning, we assume we know exactly how the solution depends on the two features as shown in Figure 1. Selection forms of five types are considered: mathematical, rules, decision trees, neural nets and exemplars.

The final section presents three other feature spaces; the reader is invited to consider how well the selection forms would perform for them. All these feature spaces were created simply by free hand drawings and have no connection to actual features.

## 2 SCENARIO 1: TWO FEATURES KNOWN

We assume that two features, $q$ and $r$, are computed in some way from the 25 problem properties. The solution depends on $q$ and $r$ as seen in Figure 1, so the task for each selection form is to approximate the function displayed there which associates one of the four outcomes with each pair of feature values. We consider the selection forms in order and illustrate them with approximately optimal parameters, using 25 parameters in each case.

**Mathematical.** We need a function of two variables which takes on one of four values. Forms such as polynomials, exponentials, etc., are not suitable for this. The generic selection form, splines, are so suited and the splines of degree 0 are actually step functions, just what is needed here. The parameters of these splines are the "knots" which partition the feature space. Only 6 spline pieces are needed to define the exact selection form for this problem. However, current methodology does not include using such general domain shapes and would only allow one to partition the $(q, r)$ space into a tensor product grid of squares or something similar. We may use equispaced knots, so that a 5 by 5 grid has 25 parameters as shown in Figure 2. We may also use the knots as parameters, a well chosen 4 by 4 grid and corresponding values is shown in Figure 3, it has only 22 parameters.

**Rules.** Again, in theory, rules can define the exact selection form for this problem. However, again, current methodology does not include using such general domain shapes. Normal practice would allow rules linear in $(q, r)$ which, in turn, allows one to define domains in the feature space with straight line boundaries. Figure 4 shows the domains from a well chosen set of rules allowing for the composition by "and" or "or" of basic linear rules. The six lines shown use $6 * 2 + 13 = 25$ parameters.

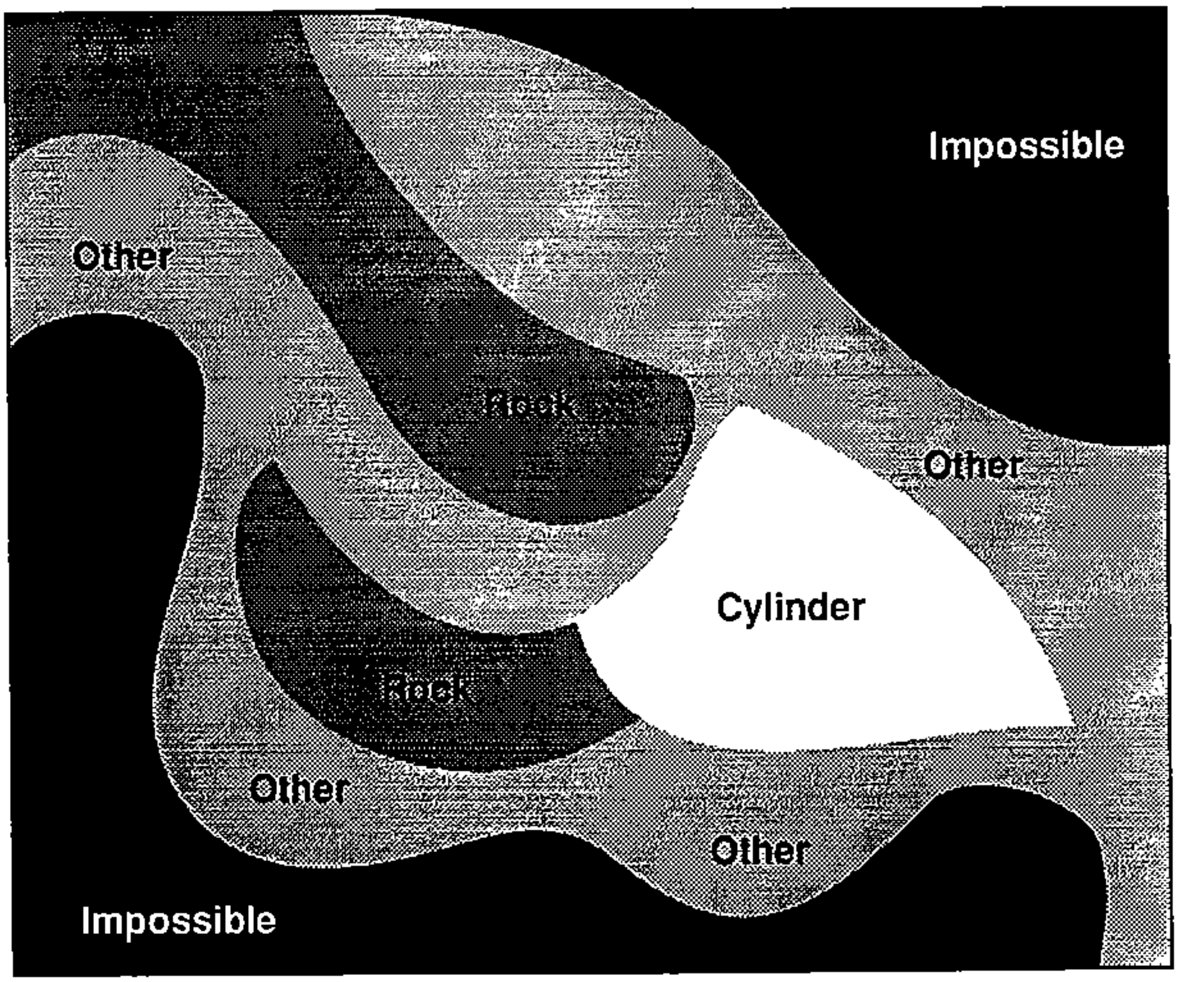



**Figure 1:** Feature space $(q, r)$ and solution values for a hypothetical underwater identification problem. The domains of "Other" are for collections of objects that are neither rocks nor cylinders and the "Impossible" domains are feature combinations that cannot be generated by any object.

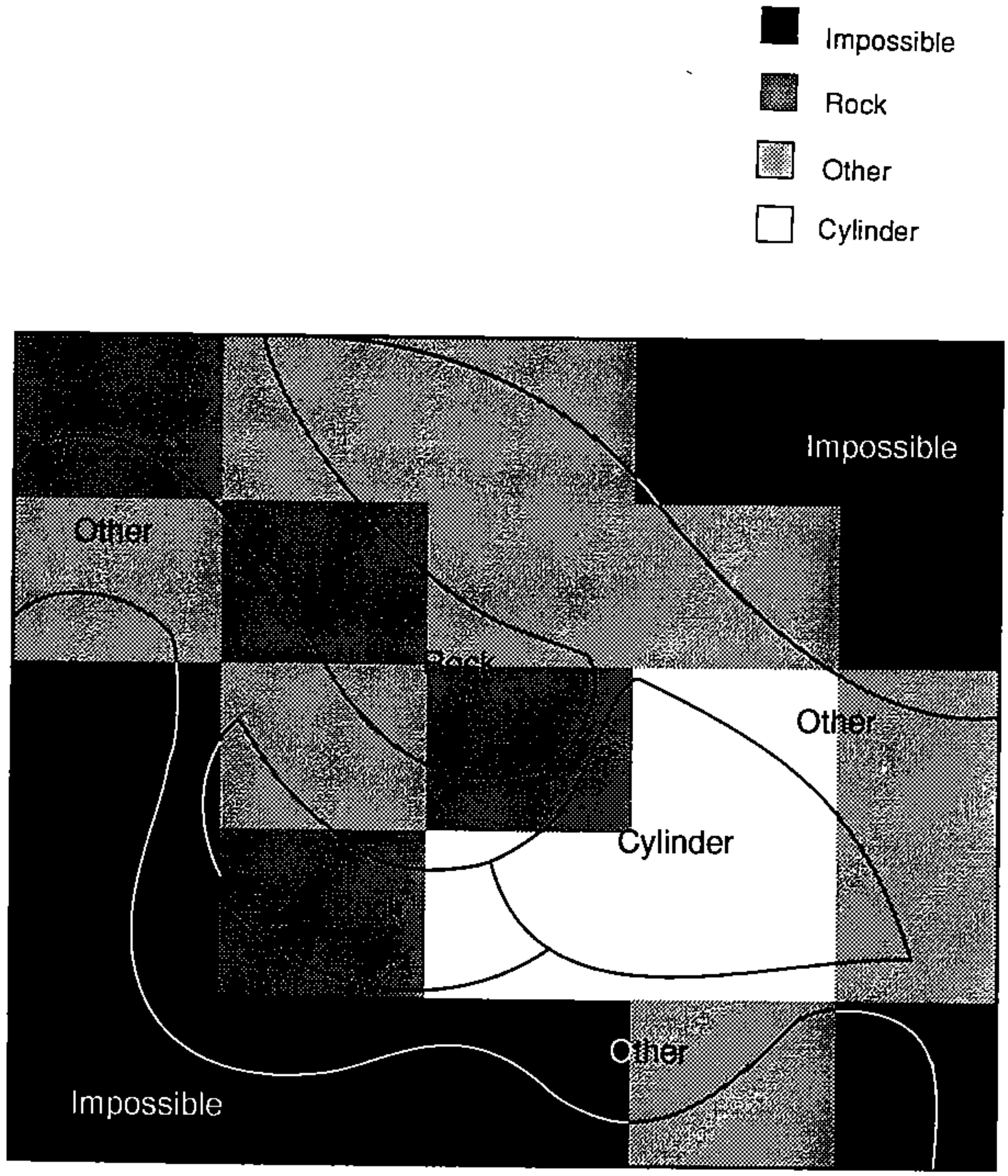



**Figure 2**: The solutions obtained from a selection form using zero degree splines with 25 uniformly spaced pieces. The best solution domains are outlined for comparison.

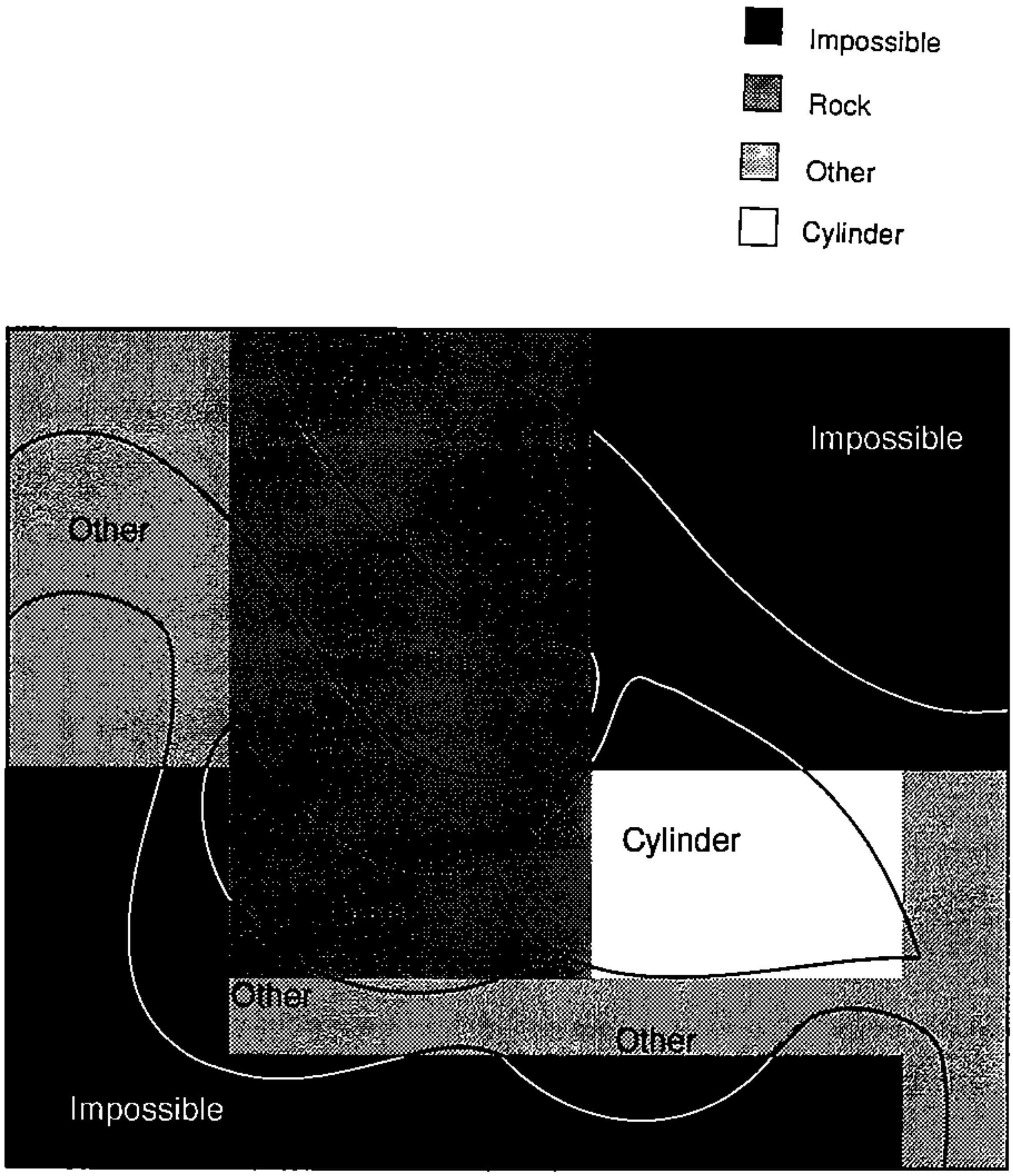
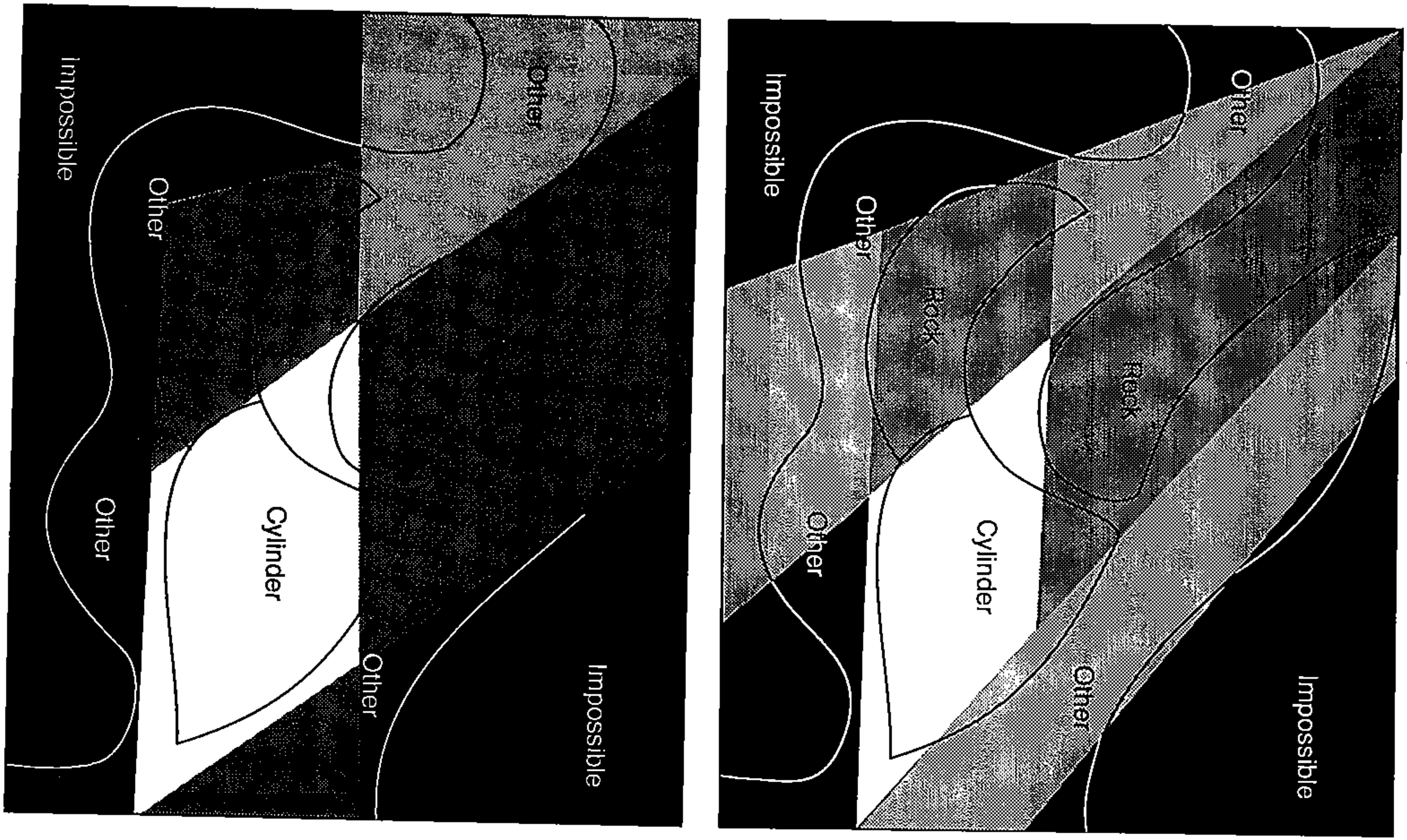



**Figure 3:** The solutions obtained from a selection form using zero degree splines with 16 variably spaced pieces. This form has 22 parameters, 6 for the dividing lines and 16 for the values in each square. The best solution domains are outlined for comparison.



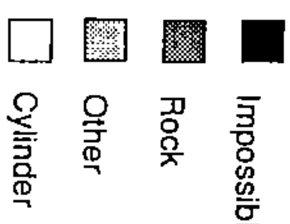




**Figure 4:** The solutions obtained from a selection form using 6 rules with 30 parameters in total (bottom-left) and 5 rule swith 26 parameters in total (top-right). The best solution domains are outlined for comparison.

**Decision Trees.** Again, in theory, decision trees can define the exact selection form for this problem. However, again, current methodology does not include such general domain shapes. If we assume, consistent with the previous approaches, that tests for decisions are linear functions of $q$ and $r$ then we see that: (a) anything that can be done by a zero degree spline can be done with decision trees, (b) anything that can be done with rules can be done with decision trees, (c) the converse of (a) and (b) are true, and (d) the number of parameters and their definitions may vary between the three selection forms even for identical functionality. Figure 6 shows the domains from a well chosen decision tree (Figure 5) with 25 parameters. Note that the tree is inverted from that Example 4.2, Figure 2 of [1] as an expansionistic rather than contractionistic approach is taken. A fan in tree like Figure 2 could be obtained by pairing ranges of $q$ and $r$ values at the top of the tree.

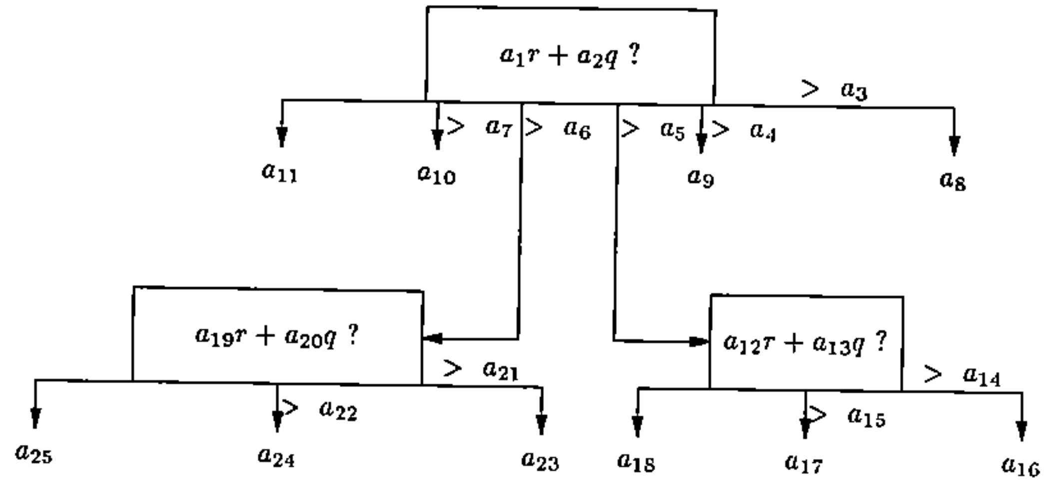



**Figure 5:** Decision tree with 25 parameters for Scenario 1 used for selection form of Figure 6.

**Neural Nets.** It is not obvious how to apply neural networks to the feature space directly. One could have a first level of two nodes, one for $q$ and $r$, and then one or two more levels. This would involve only a few parameters and would be contrary to the "broad and shallow" philosophy of neural networks. The identification of the 2 key features seems to have it difficult to apply the neural net approach.

**Exemplars.** Here we have a set of points in $(q, r)$ space with the correct solution known at each point. There are two obvious approaches to use for this method. (1) The 25 regularly spaced points with the 25 parameters being the values at these points. (2) Use 8 "optimally" spaced points with three parameters each ($q$ and $r$ coordinates plus value). Figure 5 shows reasonable results of the second of these two approaches. The first approach is identical to that of Figure 2 using zero degree splines.

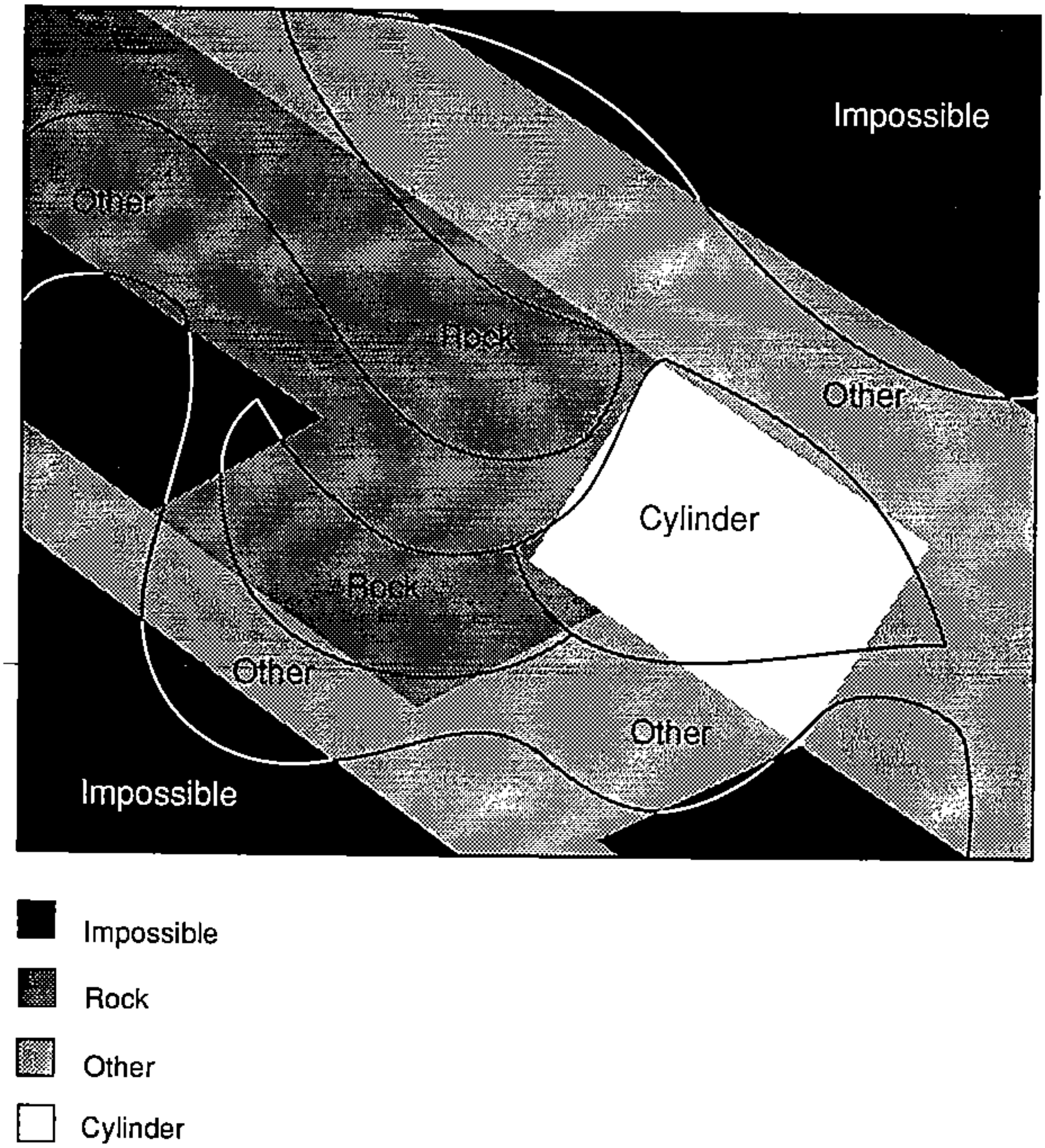



**Figure 6**: The solutions obtained from a selection form using a decision tree (see Figure 5) with 25 parameters. The best solution domains are outlined for comparison.

## 3 SCENARIO 2: ONLY 25 PROBLEM PROPERTIES KNOWN

We next turn to the scenario where these features are unknown and the 25 problem properties must be used directly. In this scenario we assume the 25 problem variables to be

| | |
|---|---|
| Sample of 7 readings from sonar of type 1, | $x_1, \ldots, x_7$ |
| Sample of 8 readings from sonar of type 2, | $x_8, \ldots, x_{15}$ |
| Sample of 8 readings from sonar of type 3, | $x_{11}, \ldots, x_{23}$ |
| Water temperature/30 | $x_{24}$ |
| Water depth/10,000 | $x_{25}$ |

and the features are computed by

$$\begin{aligned} q &= \alpha_1 \textstyle\sum_{i=1}^{7} x_i \cdot \cos(3i\pi/2)(1 + x_{24}i) \\ &\quad +\alpha_2 \textstyle\sum_{i=8}^{15} x_i \cdot e^{-1-ix_{25}} - (i-8)/16 \\ &\quad +\alpha_3 \textstyle\sum_{i=16}^{23} x_i(1 - (x_{i-15})(x_{i-8}) \sin(2i\pi/3) \\ r &= \beta_1 \textstyle\sum_{i=1}^{6} (x_i - x_{i+1}) \cos(i\pi/4)\, e^{-x_i x_{24}} \\ &\quad +\beta_2 \textstyle\sum_{i=8}^{15} \left[ 1 + (x_i - x_{i+8}) \left( \frac{1+x_{25}}{4+5x_{25}} \right) \right] (\gamma_1 + \gamma_2(x_{24}) + \gamma_3(x_{24})^2) \end{aligned}$$

In the first scenario we used selection forms with 25 parameters, now we have much more data so we consider forms with 25 parameters and 250 parameters.

**Mathematical.** There are very few choices for a selection function using just 25 parameters for a function of 25 variables $x_i$. A linear form (integer part of $\sum_i a_i x_i$) is easy to define but it would be extraordinary luck to have much success. With 250 parameters one cannot even have a general quadratic function of 25 variables (it takes 350 coefficients).

The zero degree splines are also difficult to define. If one has just one knot per variable, one obtains $2^{25}$ basis functions. An interesting alternative is to use a space filling curve, a function that maps the unit interval $0 \le t \le 1$ into a 25 dimensional cube. There are several easily computable such functions. One then divides the $t$-interval into 25 parts and assigns one of the four values to each interval. Or one can use a variable partition with 12 knots (in $t$) and 13 values for a total of 25 parameters. This approach is directly extendible to obtain a selection form with 250 parameters.

**Rules.** It is difficult to visualize a meaningful set of rules for 25 variables which involves only 25 parameters. The problem variable space could be subdivided into perhaps 10 or 12 domains since the simplest rule usually involves 2 parameters (e.g., if $x_{13} > 0.54$ then "rock"). With 250 parameters one could define about 10 general planes in the problem variable space.

**Decision Trees.** As with rules, decision trees of 25 parameters are hard to visualize for a problem with 25 variables. With 250 parameters and assuming that a decision involves an average of 5 parameters, then about 50 domains could be defined. If more general combinations of the 25 variables were used, then about 10 "generally oriented" domains could be defined in the problem

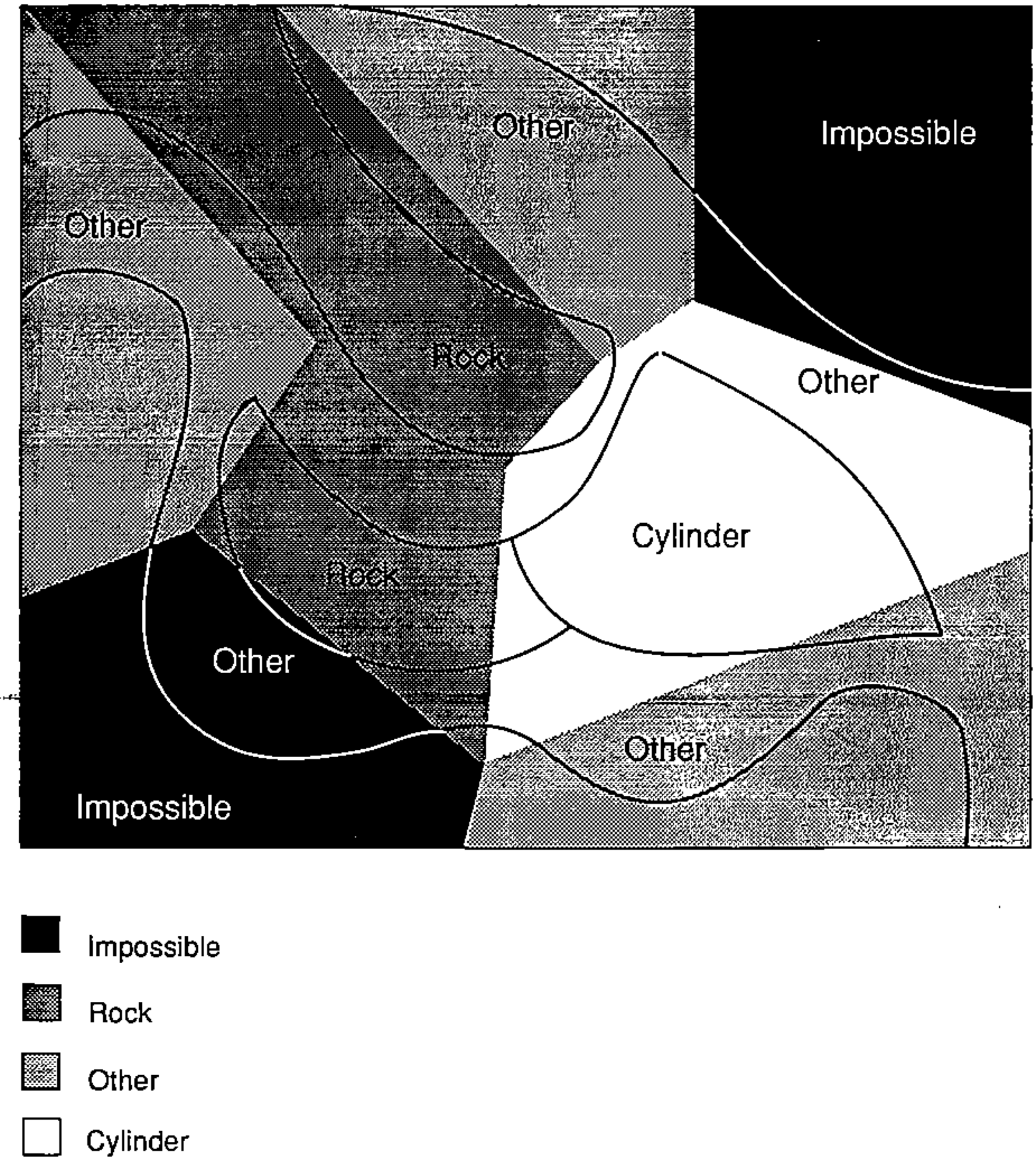



**Figure 7:** The solutions obtained from a selection form with 8 optimally spaced exemplars in the $(q, r)$ space. It has 24 parameters, each exemplar has 2 coordinate parameters plus a value. The best solution domains are outlined for comparison.

variable space. The combinatorial properties of rules and decision trees are quite similar but, as comparing Figures 4 and 5 shows, the actual selection possible may be quite different.

**Neural Nets.** The 25 problem variables can be inputs to the neural nets, but then the first level of the net involves 25 parameters. Thus a normal neural network for this problem would use about 60-80 parameters in 3 or 4 levels of the network. The use of 250 parameters would require about 10 or 12 levels in the network, an unusual number. We see that the neural network size is more closely tied to the number of problem variables (or features used) then the other selection forms.

**Exemplars.** The 25 exemplars selection for the 25 problem variables might be ones that map into the best 25 exemplars for the features $q$ and $r$. This does not, however, mean that the problem variables examplars are as good as the feature exemplars. The reason is the the pre-image of one examplar in the two-dimensional feature space is a subset of dimension 23 in the 25 dimensional space of the problem parameters. Using 250 exemplars could sample these best places 10 times each, but it is unlikely that even that would give a selection form of comparative performance as the 25 best exemplars in the feature space.

Examining the second scenario strongly suggests that using the 25 "raw" problem variables is unlikely to provide as good a selection form as the 2 problem features even if 10 times as many parameters are used in the selection form for the problem variables. Of course, it all depends on the mapping from the problem variables to the features.

## 4 THREE OTHER FEATURE SPACES

Figures 8, 9 and 10 present three other feature spaces for this hypothetical problem. These were created by free hand drawings as was that of Figure 1 are not constructed by any scientific method related to an actual problem. The readers are invited to consider how well the selection forms would do for these feature spaces.

## References

[1] John R. Rice, Learning, Teaching, Optimization and Approximation, in *Intelligent Scientific Software Systems* (Houstis, Rice and Vichnevetsky, eds.), North-Holland, Amsterdam (1991), to appear. Also CSD-TR-91-032, Computer Science Department, Purdue University, May, 1991.

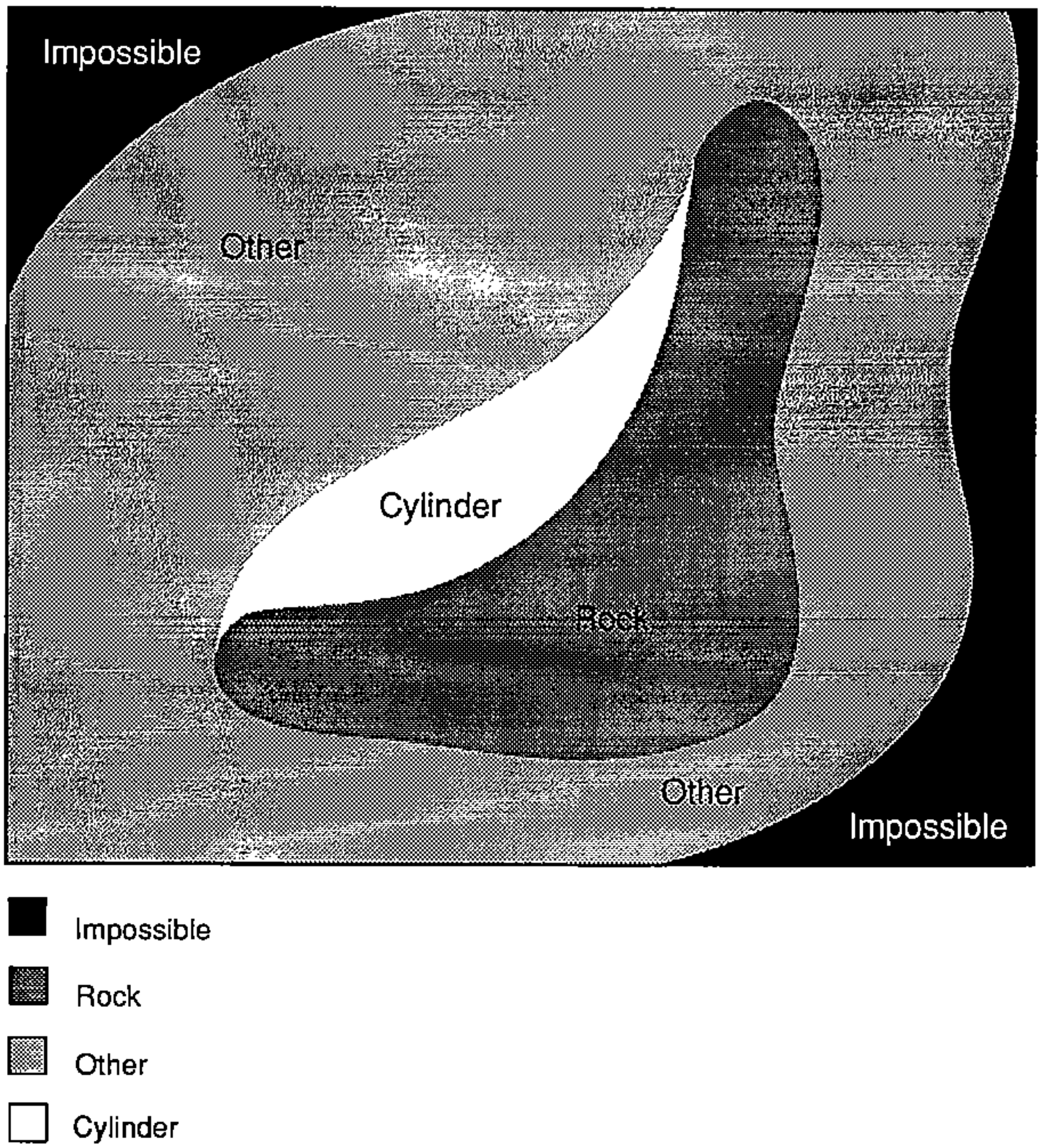



**Figure 8**: Second feature space $(q, r)$ and solution values for a hypethetical underwater identification problem.

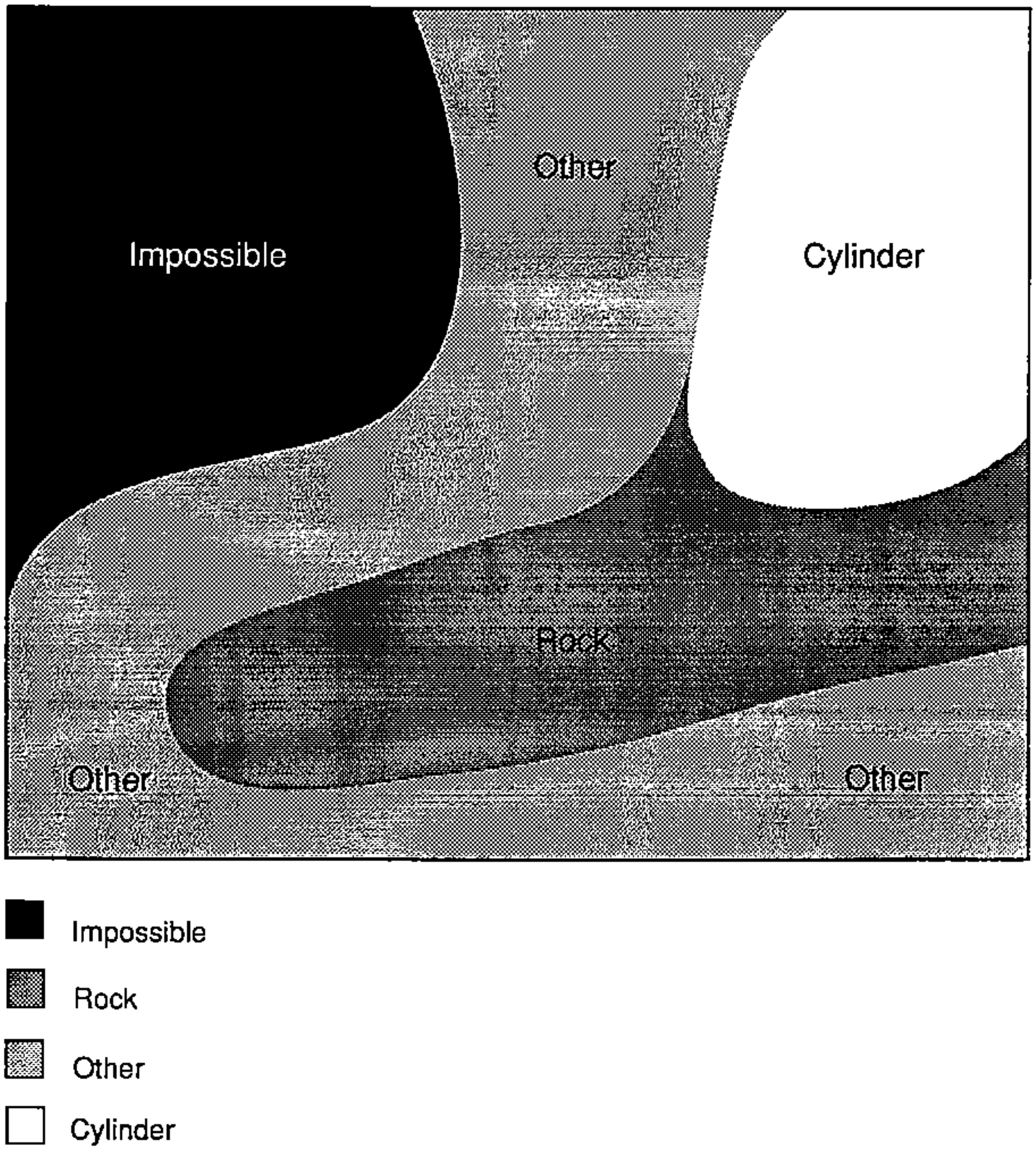



**Figure 9**: Third feature space $(q, r)$ and solution values for a hypethetical underwater identification problem.

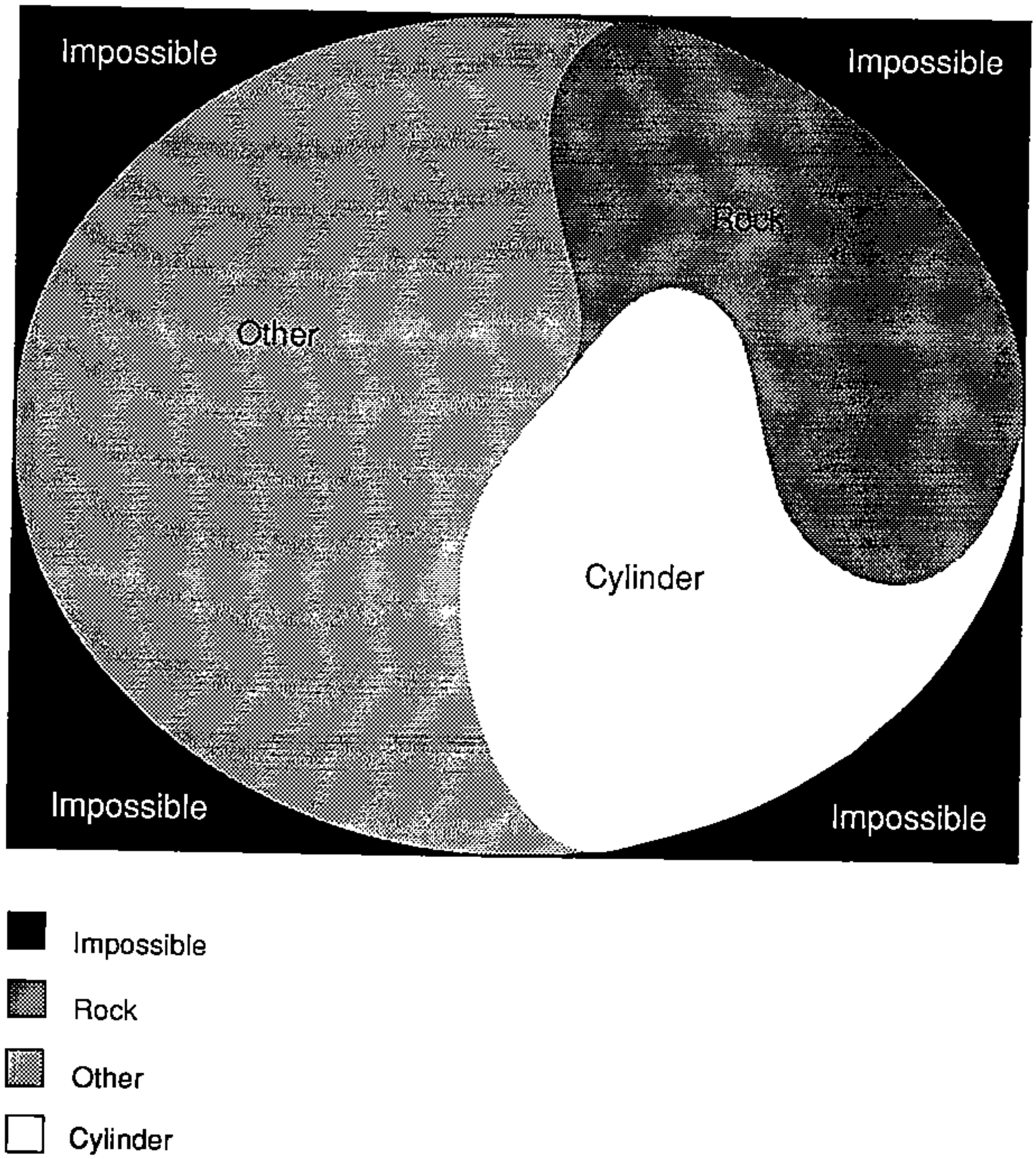



**Figure 10:** Fourth feature space $(q, r)$ and solution values for a hypethetical underwater identification problem.