Monograph:
Zhe , Ma., Harrison, R.F. and Kennedy, R. Lee. (1996) Dynamic Query Algorithms for Human-Computer Interaction Based on Information Gain and the Multi-Layer Perceptron. Research Report. ACSE Research Report 614 . Department of Automatic Control and Systems Engineering

---

Dynamic Query Algorithms for Human-Computer Interactio Based on Information Gain and the Multi-layer Perceptron

Zhe Ma and Robert F Harrison, The University of Sheffield, Department of Automatic Control and Systems Engineering, Sheffield, UK

R Lee Kennedy, The University of Edinburgh, Department of Medicine, Edinburgh, UK.

7 February 1996

Research Report 614

Corresponding Author: Robert F Harrison

email: r.f.harrison@sheffield.ac.uk tel: +44 (0)114 2825139 fax: +44 (0)114 2780409

Abstract

Algorithms are presented for an "intelligent" human machine interface for efficient on-line decision making and pattern recognition. The algorithms structure the data input process dynamically, asking the user the next most "informative" question based on its current state of knowledge, to reach a conclusion as quickly as possible.

Using the information gain principle in attribute selection, IQA and IQA1 dynamically generate a query process without the construction of the decision tree. A further development, IQA2, generalises IQA1 by including a Multi-layer Perceptron (MLP) to mitigate the effects of noise and ambiguity, and to establish incremental learning. The IQA algorithms perform well on noisy and incomplete data-sets. This is demonstrated by an example from an artificial domain and two from medical diagnosis.

Key Words: Intelligent Human-Machine Interface, On-line Decision Support, Clinical Decision Support, Decision Tree, Neural Network, Heuristic Search.

1. Introduction

How to ask the next best question? How to reach a conclusion quickly before all the facts have been given? In designing an efficient computer-aided decision support system, these questions should be uppermost in the designer's mind. This is especially true when there are many data to be input or when obtaining all the facts is difficult. In the first instance, the usability of a system can be severely compromised if formal verification of each data item is demanded, or when a user knows that for a particular case only a few salient data items will lead to an irrefutable conclusion. Such irritations are unlikely to assist the widespread adoption of a system. In the second case, it is clearly preferable to make decisions based upon the "cheapest" information available. Calling for expensive, time consuming or difficult laboratory tests, for instance, when other information can be used to arrive at an equally sound decision is again likely to militate against widespread use. Thus there is a motivation to devise a human computer interface and inference system which requests no more information than is required to make an unambiguous decision, *on a case-by-case basis*.

This paper introduces three such "intelligent" query cum inference algorithms which we call informative query algorithms (IQAs), in deference to their use of the information gain heuristic. These can be used to speed data entry in computer aided decision support systems for pattern recognition and diagnostic applications. They provide some important potential advantages such as human labour efficiency, effectiveness in handling unknown facts, and robustness to noisy, ambiguous and incomplete data environments. We consider binary domains in this paper, although the algorithms presented here are suited to more general domains.

We review the idea of the decision tree induced via "information gain" in Section 2. This is followed in Section 3 by the proposed algorithms: the basic algorithm, IQA, using the idea of information gain, and a more general version, IQA1. In a further refinement, a Multi-layer Perceptron (MLP) (Rumelhart, Hinton et al. (1986)) is introduced to derive the IQA2 algorithm. The IQA algorithms are investigated in Section 4 and are evaluated on an artificial domain and on two data-sets from medical domains. The first uses information derived from pathology slides for the diagnosis of breast cancer and the

second, patient data for the early diagnosis of suspected heart attack. The methods are summarised in the last section, where further developments are proposed.

2. Decision Trees At The User Interface

In the machine learning community, Top Down Induction of Decision Trees (TDIDT) is a successful and influential methodology (Cestnlk, Kononenko et al. (1987); Quinlan (1990)). However, when applied to a dynamic query process where a complete set of attribute values is not provided in advance, it has some inherent limitations, since a decision tree, once constructed, remains fixed.

First, it is difficult to classify patterns containing unknown attribute values because the order of the decision nodes, once the decision tree has been constructed, is fixed. If an attribute whose value is missing or unknown is located at a high level (i.e. near to the root) of the decision tree, all lower sub-trees relevant to this attribute must be taken into account throughout the rest of the classification process. Since the orders of decision nodes in different sub-trees are possibly different, it is difficult to decide the next most informative attribute to be queried across all relevant sub-trees. In fact, if the value of an attribute is unknown, no information exists to partition the current training set, thus the next most informative attribute must be selected from the same training set instead of its partitioned subsets. The unknown value represents three classes of input datum in this paper: a) the datum is missing, b) the datum is not known or not recognised, c) the datum is not important or difficult to get, and is therefore ignored.

Second, incremental learning is relatively expensive because the reconstruction of the decision tree is usually required.

In contrast, in on-line operation, a computer-aided decision support system needs only a single branch of decision nodes to classify a sequence of attribute values one at a time, rather than the whole decision tree. This branch represents a dialogue process between the computer system and the user. This process should be able to request the best attribute (in the sense of being the next most informative) according to the status of the previously assigned attribute values. If the user gives the value "unknown", the process should simply prompt for the next most informative attribute according to the previously computed result, without any further treatment. Incremental learning is achieved simply by adding the new patterns to the training set, because the attribute is selected directly

from the training set instead of from a previously constructed structure such as the decision tree.

The main disadvantage of the dialogue process compared with TDIDT is that the average computation time in the former may be more than that in the latter. However, the time consumed in each step of the dialogue is likely to be too little to be detected by human users.

The founding heuristic for the induction of decision trees is the idea of information gain (sometimes called transmitted information) introduced by Quinlan in the ID3 and the C4.5 algorithms (Quinlan (1986); Quinlan (1993)), which is also used in the algorithms in this paper.

3. The Algorithms

Suppose each pattern in the training set, S, consists of a vector of attribute (or input) values, I_j, j=1,2,...N, and an output which indicates its class membership. The patterns belong exclusively to M classes, C_1, C_2,... C_M. The proportion of patterns belonging to class C_i in S can be taken as the prior probability, $P(C_i)$, of C_i occurring in the population. According to information theory, the information for class C_i conveyed by an attribute vector depends on its probability and can be measured as $-\log_2 P(C_i)$, and the expected information for class C_i is $-P(C_i)\log_2 P(C_i)$. The average expected information of S, also called the entropy, is

$$H(S) = -\sum_{i=1}^{M} P(C_i)\log_2 P(C_i)$$

Giving the values of attribute I_j in all patterns of S, the conditional probability of class C_i given $I_j = a_\ell$ is $P(C_i | I_j = a_\ell)$, here a_ℓ is a value assigned to I_j, say either 0 or 1 in the binary case, and the conditional entropy for the distribution of the j^{th} attribute's values is

$$H(S|I_j) = -\sum_{\ell=1}^{m} P(I_j = a_\ell)\sum_{i=1}^{M} P(C_i | I_j = a_\ell)\log_2 P(C_i | I_j = a_\ell)$$

where m is the number of possible values for I_j. $H(S) - H(S|I_j)$ is the information gain as the training set is partitioned according to I_j's possible values. Selection of I_j whose

information gain is maximal, or whose $H(S|I_j)$ is minimal, since $H(S)$ is a constant, is the heuristic of attribute selection.

3.1 Basic Informative Query Algorithms, IQA and IQA1

The first algorithm, IQA, supports a sequential dialogue with the user by prompting for the next, potentially most informative question, step by step. At each step of the dialogue, an attribute is selected by the algorithm to elicit from the user the answer which has the *potential* to provide maximal information gain, or the minimal conditional entropy $H(S^*|I_j)$, where S^* is the relevant subset of the training set, or the current training subset, which is initially the entire training set, S. Once a question is answered, the relevant subset of the current subset S^* is reduced by the most recently instantiated input, I_j, so that the new S^* only contains those patterns which belong to the old S^* and also have the same instantiated I_j corresponding to the preceding answer. Given a training set of N attributes and M classes, IQA is used to decide the class to which a sequence of instantiated attributes, whether complete or not, belongs by requesting as few attribute values as possible.

3.1.1 The IQA algorithm

(1) Assign the current training subset, S^*, as the training set S, $S^*=S$; initialise the set of attributes to be requested, $T^* = \{I_1, I_2, \ldots, I_N\}$.

(2)

(i) If all patterns in S^* are of one class, the conclusion is that class; stop.

(ii) Else if $T^* = \{\}$ but the current S^* contains patterns of different classes, the training set contains conflicts, or some significant attribute values have not been acquired; report a failure and stop.

(3) Else select the next most informative attribute $I_j \in T^*$ so that

$$H(S^*|I_j) \leq H(S^*|I_k) \forall I_k \in T^*, j \neq k.$$

(4) Issue a query "What is the value of I_j?"

(5) Replace T^* with $T^* - \{I_j\}$.

(6)

(i) If the user's answer is "unknown"; go to (2)(ii).

(ii) Else the answer is $I_j = a_\ell$, and S_1* = the subset of patterns in S* with $I_j = a_\ell$.

(7)

(i) If S_1* = { } , issue a warning; "No pattern contains the attribute value $I_j = a_\ell$", go to (2)(ii).

(ii) Else S* = S_1*, go to (2).

The algorithm terminates at step (2). It successfully classifies a pattern at (2)(i) and fails to classify one at (2)(ii). The loop from step (2) to (i) of step (6) or to step (7) is expected to repeat a maximum of N times. If an unknown value is answered at (i) of step (6), S* will not be reduced, but the corresponding attribute will be ignored in the future since it has been excluded in step (5). The temporary set, is reduced from S* at (ii) of step (6), but S* is only reduced if the reduced set, S_1* is not empty, as checked in step (7).

The IQA algorithm induces a sequence of instantiated attributes which is equivalent to one branch of the ID3 decision tree if all attributes are known. But if the value "unknown" is answered, IQA generally reduces the current subset S* along with the sequence, while ID3 has to collect all relevant branches of the decision tree.

IQA is very rigid. A success and exit at (i) of step (2) is restricted to a unique class in S*. IQA is possibly not able to recognise some patterns in S which include conflicting training patterns. Therefore IQA is only presented here for discussion and to motivate the generalisations, IQA1 and IQA2. It is not used in the experiments of Section 4.

3.1.2 The IQA1 algorithm

The algorithm IQA1 is introduced to improve IQA by relaxing the failure condition at (ii) of step (2). We also introduce a control parameter called the Early Stop Tolerance (EST) to accelerate the classification process when the data set contains noise. It is used to decide on the termination of further queries, if a particular class is prevalent in the current training subset, S*. Denoting the size of the i^{th} class by $\psi^*(C_i)$, C_i is said to be prevalent in the current training subset S* if the following heuristic holds.

$\Phi(\psi^*(C_i)) > EST$ where

$$\Phi(\psi^*(C_i)) = \frac{\psi^*(C_i)}{\psi^*(C_i) + \sum_{j \neq i} \psi^*(C_j)^2} = \frac{1}{1 + \sum_{j \neq i} \psi^*(C_j)^2 \Big/ \psi^*(C_i)}$$

and EST $\in$ [0.5, 1]

$\Phi(\psi^*(C_i))$ is concerned not only with the number of patterns belonging to the i^{th} class, but also with the numbers of patterns belonging to all other classes, $\psi^*(C_j)$. If the latter are small, they can be treated as noise and will have little affect on $\Phi(\psi^*(C_i))$. On the other hand, if the size of a class other than the i^{th} is not small, it should not be treated as noise, and the value of $\Phi(.)$ is reduced significantly, owing to the square-law operation. Certainly there is only one C_i in S* that can satisfy the condition.

IQA1 differs from IQA only at steps (1) and (2) as follows,

(1) Initialisation: $S^*=S$; $T^* = \{I_1, I_2, \ldots, I_N\}$; set the EST in the recommended range [0.5,1].

(2)

(i) If $\max_i(\Phi(\psi^*(C_i))) = \Phi(\psi^*(C_j)) >$ EST, conclude class C_j; stop.

(ii) Else if $T^* = \{\}$ and there exists $\psi^*(C_j) > \psi^*(C_k)\ \forall k,\ k \neq j$, conclude class C_j; stop.

(iii) Else if $T^* = \{\}$ and $\psi^*(C_j) = \psi^*(C_k)\ \forall k,\ k \neq j$, report failure; stop.

Essentially, we introduce $\Phi(.)$ rather than the (preferable) probability, owing to the problem of estimating this quantity from a shrinking sample, S*.

Selection of the EST threshold must trade-off between efficiency and accuracy of classification. If the EST is very high (close to 1) many unnecessary queries may be made before any further information for classification is gained. This will be especially severe in cases with conflicting outputs in the training set. If the EST is very low, some patterns which are consistent in S will be misclassified because those attribute values necessary for their discrimination will not have been requested before termination. Note that the EST should always be equal to one when the data set is noise-free and classification accuracy is of primary concern.

Furthermore, assuming the effort of answering an unknown value is negligible, IQA and IQA1 perform well when the training set contains some "unknown" values. This assumption is reasonable in practice because evaluating certain attributes in some situations is costly, risky or time consuming, answering "unknown" provides a means for the user to avoid such cost, risk or time. This is analysed in Section 4.1.

3.2 An Informative Query Algorithm Including An MLP (IQA2)

Artificial neural networks, such as the MLP, can learn effectively in noisy, incomplete and ambiguous environments (Richard & Lippman (1991); Rumelhart, Hinton et al. (1986)). We now propose a further improvement to IQA/IQA1, denoted IQA2, which makes use of an MLP, a further user defined tolerance, the Final Classification Tolerance (FCT), and an incremental operation. Here we assume that an MLP has been trained to give adequate classification performance beforehand.

3.2.1 The IQA2 algorithm

(1) Initialisation:

(i) $S^*=S$; then unify conflicting cases in S* by replacing the pattern outputs in S^* by the result of recalling the trained MLP on the input vectors of the patterns in S^*.

(ii) $T^* = \{I_1, I_2, \ldots, I_N\}$.

(iii) Set the EST in the recommended range [0.5,1].

(iv) Set the FCT, in the recommended range [0, 0.5].

(2)

(i) If $\max_i\left(\Phi\left(\psi^*(C_i)\right)\right) = \Phi\left(\psi^*(C_j)\right) >$ EST conclude class C_j; stop.

(ii) Else if $T^* = \{\}$ then

if there is a C_j so that $(\psi^*(C_j) - \psi^*(C_k))/|S^*| >$ FCT $\forall k \neq j$, conclude class C_j; stop.

Otherwise go to (8) for incremental learning.

(3) Select the next most informative attribute $I_j \in T^*$ so that

$H(S^* | I_j) \leq H(S^* | I_k)\ \forall I_k \in T^*, j \neq k$.

(4) Issue a query "What is the value of I_j?".

(5) Replace T^* with T^*-{Ij}.

(6)

(i) If the user's answer is "unknown"; go to (2)(ii).

(ii) Else the answer is $I_j = a_\ell$, and S_1^* = the subset of patterns in S^* with $I_j = a_\ell$.

(7)

(i) If $S_1^* = \{\}$, issue a warning; "No pattern contains the attribute value $I_j = a_\ell$", go to (2)(ii).

(ii) Else $S^* = S_1^*$, go to (2).

(8) Recall the MLP on the attribute value vector to obtain the classification result; add a pattern into S, including this attribute value vector and the MLP recall result; stop.

The major differences between IQA1 and IQA2 are two-fold. First, the initial current training subset, S^*, is the training set with pattern outputs modified by recalling the MLP at (i) of step (1) in IQA2. Second, step (8) in IQA2 is the incremental learning operation in the sense that new information is incorporated into the decision process, for future use. A minor change is the Final Classification Tolerance used in (ii) of step (2) in IQA2. Because the MLP is available for further classification, and all attributes have been requested at (ii) of step (2), raising the FCT can tighten the criterion of the conclusion and increase the reliability of the classification.

Now, in contrast to IQA and IQA1, IQA2 has three places of exit:

a) the conclusion is drawn from the current training subset, S^*, and only some of the attributes are queried, exit at (i) of step (2);

b) the conclusion is drawn from the current training subset, S^*, where all the attributes have been requested, exit at (ii) of step (2);

c) the conclusion is drawn by the MLP instead of from S^*, exit at step (8).

IQA2 takes advantage of three learning modes.

a) If the attribute value vector to be requested is included in some pattern(s) in the training set S, it is classified as accurately as the MLP is able. As demonstrated in (Shavlik, Mooney et al. (1991)), the MLP performs better than ID3 when the data are noisy or incompletely specified.

b) If the attribute vector is not included in the training set, but can be distinctly classified in the initial S^* (S modified by the MLP outputs), the classification results are almost as accurate as those of the MLP, because S^* does not contain conflicting cases. Thus ID3 performs similarly well owing to the non-conflicting training set. We assume that applying the MLP twice on the same training set is functionally equivalent to applying the MLP once on the training set. This is because the first application of the MLP always results in a consistent pattern set. Case (b) can be written symbolically as:

$$\text{ID3}(\text{MLP}(S)) \approx \text{MLP}(\text{MLP}(S)) \approx \text{MLP}(S)$$

In neither case (a) nor (b) is recall of the MLP required, nor are conflicting data encountered in the training set. IQA2 is more efficient than IQA1 in these cases in terms of the average query length and therefore, of computation time.

c) The attribute vector is not able to be classified by the information gain in the set S^*, which is the last situation at (ii) of step (2) in IQA2. The MLP is recalled. Both IQA1 and IQA2 have requested the full length of the attribute vector because $T^* = \{\}$. Again, we expect the MLP to give the classification result in IQA2, instead of the failure report in IQA1. Furthermore, the incremental learning operation in IQA2 should reduce the opportunity for case (c) in subsequent operation.

4. Experimental Results

In this section, experimental results are presented from an artificial domain called "Go-to-beach?", one using patterns derived from pathology slides of fine-needle aspirates of the breast for the diagnosis of breast cancer and one from the early diagnosis of acute myocardial infarction (AMI) or heart attack. IQA1 and IQA2 are compared on the "Go-to-beach?" example as unknown attribute values are introduced in different proportions. Both IQA1 and IQA2 are evaluated on the breast cancer diagnosis data, which contain ambiguous classifications, i.e. a proportion of conflicting diagnoses exists for identical

attribute vectors and on the AMI problem which has an unambiguous training set, but with a conflicting test set. The former has a maximum query length of 10 and is therefore considered to be a small problem. The latter, having a potential query length of 54, might be expected to demonstrate the benefits of IQA more clearly.

4.1 The "Go-to-beach?" Example

We have constructed a simple example, called "Go-to-beach?", to illustrate the efficiency of the IQA algorithms. The scenario is to decide whether or not to go to the beach to indulge in some form of beach activity. In the example, there are four binary attributes under consideration:

Attribute A: Weekday: is it a weekday or not?

Attribute B: Temperature: is it hot or not?

Attribute C: Dry: is it dry or not?

Attribute D: Windy: is it windy or not?

To simplify things further, the symbol A indicates that the attribute A is true, or A=1; ~A indicates that A is false, or A=0. And the same is used for B, C and D. The final decision is represented thus:

E: go to the beach; ~E: do not go to the beach

The decision to go to the beach is made *only* under the following conditions:

a) if it is not a weekday (the weekend), *and* it is any of the following

not hot, dry and windy (e.g. for sailing),

hot and not windy (e.g. for swimming);

hot and dry (e.g. for sunbathing);

b) if it is a weekday, and it is at least hot and dry.

This scenario is designed to demonstrate a decision tree with different node orders in its sub-trees and the complete pattern set (truth table) is listed in Table 1.

Case #	A	B	C	D	E
1	0	0	0	0	0
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	1
5	0	1	0	0	1
6	0	1	0	1	0
7	0	1	1	0	1
8	0	1	1	1	0
9	1	0	0	0	0
10	1	0	0	1	0
11	1	0	1	0	0
12	1	0	1	1	0
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	1
16	1	1	1	1	1

Table 1 The "Go-to-beach?" truth table

This pattern set can be used to generate an optimal decision tree as shown in Figure 1 according to the ID3 information gain principle.

Let us simulate a query process based on the decision tree. When attribute B is first requested, if the answer is "unknown", both sub-trees led by B=0 and B=1 must be taken into account for further attribute selection. There is no straightforward method for selecting the next most informative attribute because the order of the nodes in the leftmost sub-tree is C, A and D, but in the right sub-tree it is A, C and D. Besides, the

next attribute should be selected from S^*, instead of from the subsets corresponding to the sub-trees in the decision tree.

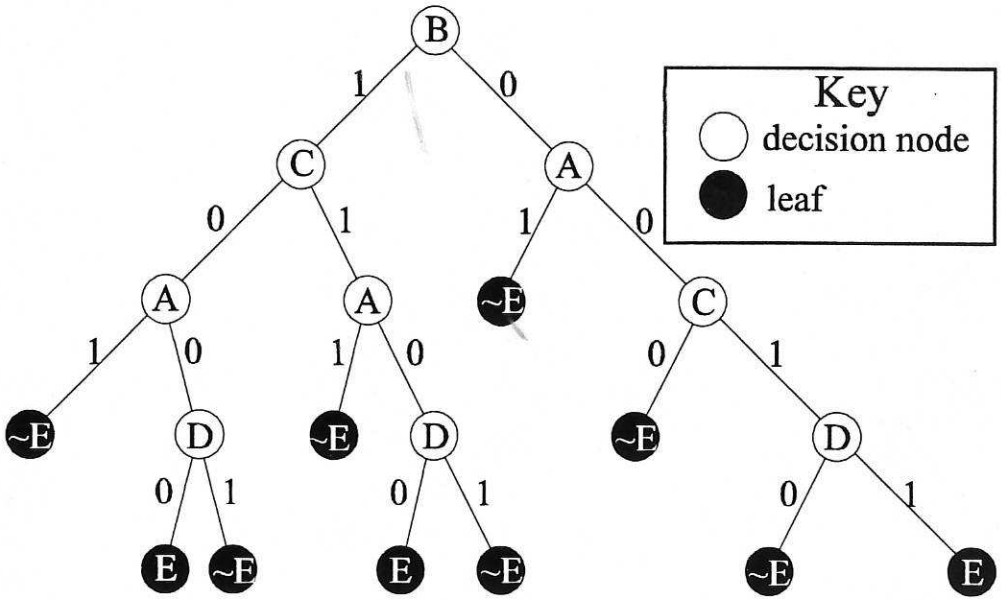

Figure 1 C4.5 induced decision tree for the "Go-to-beach?" example.

Being unrestricted by the decision tree, the three IQA algorithms simply ignore the "unknown" attributes and leave the current test set, S^*, unchanged. The next most informative attribute will be selected, whose information gain was computed on S^* when the first attribute was selected. Here we examine the IQA algorithm on a simple query process. The user-given answers will be A=1, B is "unknown", C=0 and D=1 if the attributes are requested. The conclusion of this case is not to go to the beach.

Before requesting any attribute, the conditional entropies are:

$H(S^*|A)=0.882856$

$H(S^*|B)=0.771782$

$H(S^*|C)=0.771782$

$H(S^*|D)=0.882856$

Attribute B is selected because $H(S^*|B)$ is one of the minima, i.e. the information gain will be maximal if the training set is partitioned according to the distribution of B's possible values in the training set. Because B's value is given as "unknown", S^* will not be partitioned, and C will be requested since $H(S^*|C)$ is the next minimum. Note that the list order of attributes is used here, arbitrarily, to break a tie.

Because C=1 is answered, there are only 8 patterns containing C=1, remaining in S^*. The conditional entropies for the two remaining attributes A and D are re-calculated thus:

$H(S^*|A)=0.405639$

$H(S^*|D)=0.405639$

Here A is selected for instantiation, and because A=1 is answered, S^* is reduced to the 4 patterns containing A=1. Until the last attribute D is queried, and D=1 is answered, all the patterns in S^* have the conclusion E=0, the conclusion of the query process is E=0: "do not go to the beach". Although some advanced TDIDT approaches, such as C4.5, can suggest the same optimal query sequence, the query selection process is much more complex.

Now we consider the average path length, or the average number of questions in a dialogue on the training set itself, which is a complete, consistent and non-redundant set. The ID3 decision tree can be used for this investigation since there are no unknown values in the set. There is one leaf with path length 2, covering 4 patterns; three leaves with path length 3, each covering 2 patterns; and six leaves with path length 4, each covering 1 pattern. For example, the left most path in the tree is B=1, C=0 and A=1 to a leaf E=0, which covers two possible patterns:

If(B=1, C=0, A=1, D=0) Then (E=0)
If(B=1, C=0, A=1, D=1) Then (E=0).

Only the first three attributes are necessary to the conclusion ~E. Therefore the necessary query length is 3 for each of the two patterns. The average path length to classify all 16 patterns is therefore:

$$(1 \times 2 \times 4 + 3 \times 3 \times 2 + 6 \times 4 \times 1)/16 = 3.125$$

By setting EST=1, all the IQA algorithms result in the same average path length with 100% accuracy on the original "go-to-beach?" training set. (Note that the use of an MLP in IQA2 is unnecessary for this simple, noise-free example.) This is not so when the values of some attributes in the patterns are "unknown".

In the next experiment, we assume that answering "unknown" to a query is trivial and not counted, both in terms of computation and of human effort. The former

(computational triviality) is observed in (i) of the step 6 in all IQA algorithms, the latter is discussed in Section 3. Both IQA1 and IQA2 are applied to the "go-to-beach?" example as increasing proportions of "unknown" attribute values are introduced. IQA is not applied because IQA1 is more general and more suitable in practice. The average path length is computed when all of the patterns have been used, while the answer when a value "unknown" is not counted.

Figure 2 and 3 shows the performance of IQA1 and IQA2 on the "go-to-beach?" domain when the unknown values occur in the attributes at proportions in the range [0, 40%]. The EST is 1.0, which is the maximum so that the dialogue does not stop until either all patterns in S^* are of one class or all attributes have been requested. Because "unknown" values occurring at different positions affect the classification results to very different extents, the results are averaged over 20 runs of the algorithms, for each proportion of the "unknown" values.

When there are no "unknown" value included, the accuracy is 100% and the average query length is 3.125 by both IQA1 and IQA2: identical to the ID3 decision tree based approach. As the proportion of "unknown" values increased, the average accuracy generally declines from 100% to 85%. IQA2 appears to be slightly more accurate than IQA1 but this is probably due to statistical variation. The average query length declines from 3.125 to 1.4625, while IQA2 requires slightly shorter queries than does IQA1. Again this difference may not be significant.

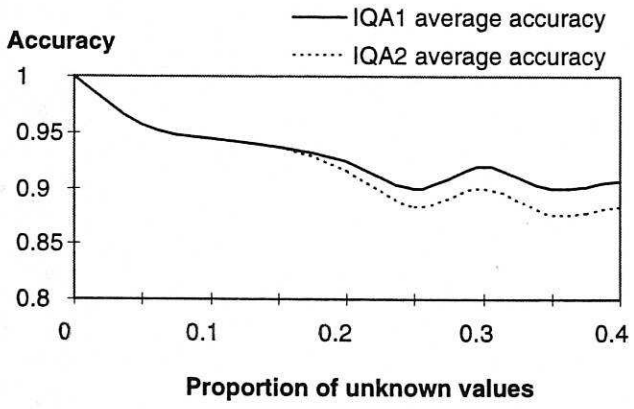

Figure 2 IQA1 and IQA2 accuracies for the "Go-to-beach?" example as proportions of "unknowns" range from 0 to 0.4..

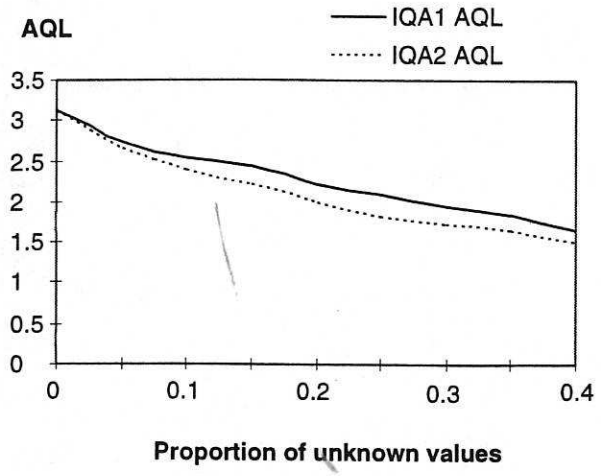

Figure 3 IQA1 and IQA2 average query lengths for the "Go-to-beach?" example as proportions of "unknowns" range from 0 to 0.4.

Evidently, accuracy in classification can be preserved to a reasonable extent in the face of up to 40% "unknown" data with a reduction of almost 50% in labour.

4.2 Breast Cancer Diagnosis

The data-set consisted of 413 patient records, each comprising ten binary-valued features recorded from human observation of breast tissue samples, together with the actual outcome representing whether or not a lesion had proved to be malignant or benign. Although the data-set was claimed to have predictive value for the diagnosis task (Trott (1991); Koss (1992)), there are only 92 distinct input vectors in the set and 12 of them correspond to conflicting conclusions appearing in many places. 38 of the 92 distinct input vectors occur more than once in the data-set. Among the conflicting occurrences, 17 patterns belong to the minority class while most patterns having the same input vectors do not.

The data set were divided into three randomly selected sets: 100 cases for the training set, the next 100 cases for the verification set, and the remaining 213 cases for the test set. A number of conventional MLPs with one hidden layer was trained on the training set. These MLPs with different numbers of hidden units were then tested on the verification set to determine an adequate MLP structure, which was found to be one with 10 input units and 5 hidden units. The previously unseen test set was then classified by the trained MLP.

There are three performance indicators widely used in medical decision making. Sensitivity: defined as the ratio of the number of correct positive diagnoses to the number of positive outcomes. Specificity: defined as the ratio of the number of correct negative diagnoses to the number of negative outcomes. Accuracy: defined as the ratio of the number of correct diagnoses to the total number of patients. In this domain, taking malignancy as a "positive" outcome, specificity must be high (approaching 100%), to avoid unnecessary surgery being carried out. This is because a patient diagnosed as having a malignant tumour will go straight for surgery, whereas a benign diagnosis will be referred to the surgeon and the patient is reviewed. Thus some false negatives are acceptable whilst false positives are not.

Based on the same training set, IQA1 and IQA2 were applied to the training set itself, to the verification set and to the test set respectively. Because the final MLP is dependent on both the training set (for weight adjustment) and the verification set (for structure selection) performance of the MLP and IQA2 (which uses the MLP) on these data sets may not be indicative of prospective use in the field. Thus results are reported for the test set only, as this represents prospective behaviour.

C4.5 is used on the training set to generate a decision tree as shown in Figure 4. The tree is then used for classification on the verification set and test set respectively. The attribute labels refer to histopathological features whose definition is unimportant to the work presented here.

Platform(EST)	Sensitivity (%)	Specificity (%)	Accuracy (%)	AQL
MLP	94.6	95.2	94.9	10
C4.5	92.7	95.1	93.9	1.66
IQA1(1.00)	92.7	96.1	94.4	5.12
IQA1(0.85)	92.7	96.1	94.4	4.14
IQA1(0.75)	92.7	96.1	94.4	1.79
IQA2(1.00)	94.5	94.2	94.4	2.11
IQA2(0.85)	92.7	96.1	94.4	1.54
IQA2(0.75)	94.5	93.2	93.9	1.54

Table 2 Results for the test set (213 cases) of the breast cancer diagnosis records

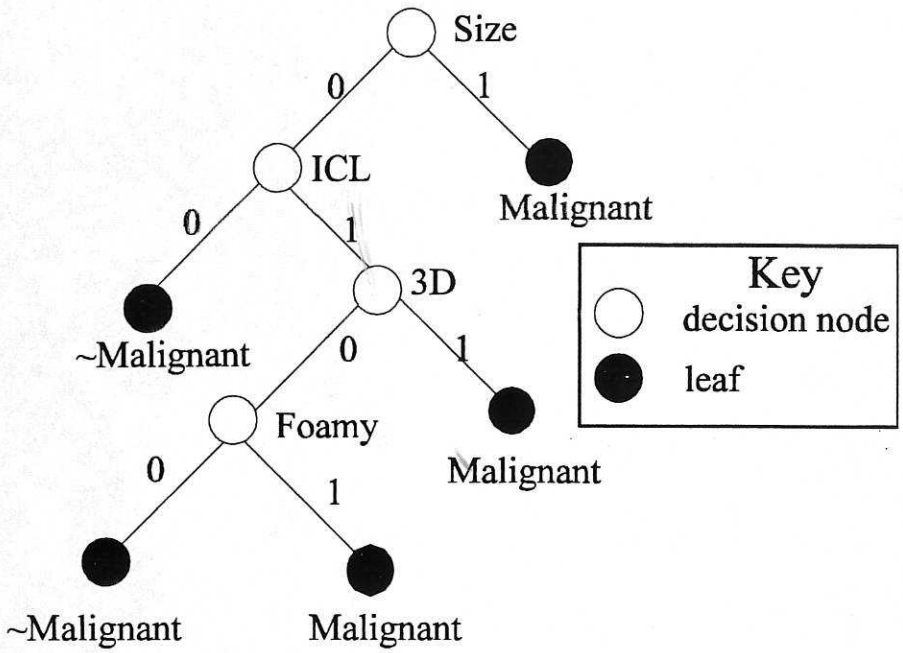

Figure 4 C4.5 decision tree derived from 100 cases (the training set) of the breast cancer diagnosis records

Clearly the MLP performs best but requires a full input vector for each case. C4.5 gives satisfactory results and significantly reduces the average query length to 1.66. The generated decision tree is tolerant to noisy data and some leaf nodes correspond to conflicting patterns in the ratios of 54/4 or 3/1, suggesting EST values for the IQA algorithms of 0.77 and 0.75 respectively. The performance of the IQA algorithms is influenced by the EST value and lies between that of the MLP and of C4.5. Average query lengths are also influenced by the choice of EST. IQA2 always outperforms IQA1 and for EST=0.85 shows a slight improvement in both performance and query length over C4.5. At this point, the performance of the MLP is almost recovered but with, on average, approximately an eight-fold decrease in query length.

For this example, the performance of IQA2 is robust to the choice of FCT. Indeed, its value was found to have no effect on either performance or query length until set at its upper limit of 0.5, which forces a full query. The average computation time per query on a Sun Sparc station 10 is 25 milli-seconds for both the IQA1 and the IQA2 algorithms.

4.3 Diagnosis of acute myocardial infarction

The early identification of patients with acute ischaemic heart disease remains one of the greatest challenges in emergency medicine. The electrocardiograph (ECG) only shows diagnostic changes in about half of AMI patients at presentation (Stark & Vacek (1987)). None of the available biochemical tests becomes positive until at least three hours after symptoms begin, making such measurements of limited use for the early triage of patients with suspected AMI (Adams, Abendschein et al. (1993)). The early diagnosis of AMI relies, therefore, on an analysis of clinical features along with ECG data. A variety of statistical and computer-based algorithms has been developed to assist with the analysis of these factors (Kennedy, Harrison et al. (1993)). Although none of these has yet found widespread usage in clinical practice, this remains an important area of research not only because of its clear potential to improve triage practices for the commonest of all medical problems, but also because of the light it may shed on techniques for the development of decision aids for use in other areas of medicine.

The data set consisted of records from 970 consecutive referrals to a major UK teaching hospital with a major complaint of chest pain. The final diagnosis in the patient cohort comprised 191 AMI and 779 non-AMI (angina, musculo-skeletal pain and other diagnoses). These diagnoses were assigned independently by three experts and a majority decision was taken in cases of disagreement. The input patterns were derived from 38 items of information recorded at presentation coded as 54 binary inputs. The training set comprised 279 records reflecting the statistics of the known outcomes in the entire sample, and the test set comprised a further 520 with the same statistical properties. The remaining 171 patterns were ignored (Kennedy, Harrison et al. (1994)). An MLP with 54 inputs, 18 hidden units and a single output unit was trained to give perfect performance on the training set. Its test set performance is given in table 3.

Platform	Sensitivity(%)	Specificity(%)	Accuracy(%)	AQL
MLP	89.9	88.8	89.4	54
C4.5	88.1	82.2	83.5	7.45
IQA1(1.0)	89.0	85.9	86.5	6.18
IQA1(0.9)	89.0	86.1	86.7	5.39
IQA1(0.8)	87.2	85.4	85.8	4.49
IQA1(0.7)	86.2	87.6	87.3	3.82
IQA1(0.6)	83.5	88.3	87.3	3.70

Table 3 Results for the test set (520 cases) of the AMI diagnosis records

Because the training set here is free from conflicting diagnoses, the FCT becomes redundant and the MLP will not be recalled (unless FCT=0.5). Thus IQA1 and IQA2 become functionally identical and we report results for IQA1 only.

While the AQL of C4.5 is a great improvement on that of the MLP its relative performance is disappointing especially in its ability to exclude the diagnosis of AMI (specificity). IQA1, in contrast, maintains good performance over a considerable range of values for EST (between 0.7 and 1.0) while reducing the AQL still further.

5. Conclusions And Recommendations

Efficient human machine dialogue for decision support should be a dynamical classification process involving the user in providing the least amount of data concomitant with reaching an unambiguous and correct conclusion. In a machine learning context an induced decision tree such as one derived through the ID3 algorithm may provide good diagnostic performance, however, the decision tree remains static and depends entirely on the initial training set. This may lead to sub-optimal performance in the sense (a) that lengthy queries will in general be required even if the data gathered are not needed to reach a particular conclusion or complicated processes will be needed to handle "unknowns", and (b) new information can only be incorporated by re-derivation of the tree.

We have presented algorithms of increasing complexity and generality which are based on the same principle as ID3—the idea of information gain—which address these two shortcomings. They provide a rationale for selecting the next most informative attribute to be queried and lead to a query path which is only as long as it needs to be. For this reason we label our methods IQA—Informative Query Algorithms.

The algorithm IQA makes use of information gain at each step in the query process to select the attribute next most likely to maximally increase information in the system. This provides the underlying principle for the more general methods, IQA1 and IQA2.

IQA1 is a simple, relatively efficient procedure which builds on IQA to handle noisy, redundant and incomplete data-sets. In itself this provides a useful heuristic for dynamic query in a computerised decision support system, but does not optimally handle noise, nor admit incremental "learning".

The IQA2 procedure includes a trained artificial neural network—an MLP—to pre-process the training data into a uniform set (all input vectors assigned unambiguously) and an incremental learning facility.

The IQA approach relies on up to two user supplied control parameters, the EST and the FCT. The effects of the former have been reported and it has been seen that EST can assist in reducing the average query length while maintaining good classification performance. The results given here have been found to be robust with respect to the value of FCT, but this may not be the case for other problems.

Experiments in an artificial domain and two using clinical data indicate that IQA2 is on average more efficient and more accurate than IQA1 in uncertain environments and that more benefit is to be gained the larger the potential attribute set.

This work forms a part of the hybrid neural network / knowledge-based system, GR2 (Ma & Harrison (1995)), which integrates a rule-based system and an MLP via rule extraction from the neural network. The query algorithms presented here will be used to compare with another approach to creating an "intelligent" user interface based upon a rule base generated by rule extraction from an MLP.

Acknowledgement

This research work has been supported by the Science and Engineering Research Council, UK, Grant Number GR/J29916. Thanks are due to Dr. Simon Cross of the Department of Pathology, University of Sheffield Medical School for supplying and interpreting the breast cancer data.

References

Adams, J.E., Abendschein, D.R. et al. (1993) Biochemical markers of myocardial injury. Is MB creatine kinase the choice for the 1990s?. *Circulation,* **88,** 750-763.

Cestnlk, B., Kononenko, I. et al. (1987) Assistant 86: a knowledge elicitation tool for sophisticated users. In Bratko, I. & Lavrac, N. (Eds.), *Progress in Machine Learning.*

Kennedy, R.L., Harrison, R.F. et al. (1993) Do we need computer-based decision support for the diagnosis of acute chest pain?. *Journal of the Royal Society of Medicine,* **86,** 31-34.

Kennedy, R.L., Harrison, R.F. et al. (1994) A comparison of logistic regression and artificial neural network models for the early diagnosis of acute myocardial infarction (AMI). **539**, The University of Sheffield, Sheffield.

Koss, L.G. (1992) *Diagnostic cytology and its histopathologic basis.*

Ma, Z. & Harrison, R.F. (1995) *GR2: a hybrid knowledge-based system using general rules. Proceedings of the* International Joint Conference on Artificial Intelligence. (pp.488-493) Montreal.

Quinlan, J.R. (1986) Induction of decision trees. *Machine Learning,* **1,** 81-106.

Quinlan, J.R. (1990) Decision trees and decision making. *IEEE Transactions on Systems, Man and Cybernetics,* **20,** 339-346.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*. Morgan Kauffman.

Richard, M. & Lippman, R. (1991) Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation,* **3,** 461-483.

Rumelhart, D., Hinton, G. et al. (1986) Learning representations by back-propagating errors. *Nature,* **323,** 533-536.

Shavlik, J.W., Mooney, R.J. et al. (1991) Symbolic and neural learning algorithms: an experimental comparison. *Machine Learning,* **6,** 111-143.

Stark, C.M.E. & Vacek, J.L. (1987) The initial electrocardiogram during admission for myocardial infarction. *Archives of Internal Medicine,* **147,** 843-847.

Trott, P.A. (1991) Aspiration cytodiagnosis of the breast. *Diagnostic Oncology,* **1,** 79-87.