A PROBABILISTIC INDUCTIVE LEARNING APPROACH 
TO THE ACQUISITION OF KNOWLEDGE IN 
MEDICAL EXPERT SYSTEMS

Keith C. C. Chan, John Y. Ching and Andrew K. C. Wong
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Abstract

In this paper, we present an inductive knowledge acquisition method based on 
the probabilistic inference technique. Our proposed system can be applied to 
generate decision rules automatically for certain medical expert systems. 
Given a patient database containing historical diagnosis and prognosis 
information, our method is capable of detecting the inherent probabilistic 
patterns in the data. Classification knowledge can be synthesized in the form 
of explicit production rules with associated probabilistic weight of evidence 
based on the patterns detected. With these rules, new patient cases can be 
quickly and accurately classified. Using real-world medical data, we show 
that our method performs better in terms of classification accuracy and 
computational efficiency than some of the major existing methods.

Introduction

Medical expert systems such as MYCIN [25] are examples of some of the most successful 
real-world applications using artificial intelligence techniques. Knowledge engineering is a 
key step in the development life cycle of a medical expert system. The most common 
traditional approach to knowledge acquisition is through interviews of human experts by 
knowledge engineers. Unfortunately, such a process can be very time and resource consuming 
due to the complexity of the knowledge involved as well as the limiting communications 
abilities of the knowledge engineers and experts. In order to effectively elicit useful 
knowledge from human experts, the knowledge engineers must first spend considerable time 
and effort to familiarize themselves with the particular areas of interest. They must be able to 
ask the right questions at the right time, and be able to interpret the information provided by 
the human experts. On the other hand, the success of the knowledge acquisition process also 
depends on the level of cooperation and communication skills of the experts. In many cases, 
despite the efforts of the knowledge engineers and the human experts, only certain types of 
knowledge, such as definitions, notations, terminologies and simple concepts can be obtained 
successfully through interviews. Other types of more complex declarative and procedural 
knowledge are much more difficult to articulate. As a result, the knowledge acquisition step is 
well known to be a bottleneck in the development of an expert system [10].

Inductive learning techniques have been shown as a feasible way to automate the process of 
knowledge acquisition [1, 2, 3, 13, 17, 23, 27]. The type of problems that inductive learning

572 0-8186-2742-5/92 $3.00 © 1992 IEEE
systems deal with is known as classification [4, 5]. Since the problem solving strategies of many medical expert systems are classificatory in nature (e.g., the classification of patients into their disease classes or types of therapy), the idea is that the classificatory knowledge can be detected and acquired from examples of human expert decision making. By giving an expert system the ability to automatically acquire and refine knowledge, the knowledge engineering bottleneck problem can be mostly avoided. Two families of inductive learners, based on the ID3 [19] and AQ [12] algorithms, are especially popular. ID3 or decision-tree based systems are so well established that a number of them have even been applied commercially to the building of medical expert systems [3, 9, 21]. The original AQ algorithm became well known after it was applied successfully to the diagnosis of soybean diseases [13]. More recent versions of the algorithm have been applied to medical knowledge acquisition [15]. Despite their success, both ID3 and AQ share one limitation: they are not designed to handle noisy data efficiently [6].

Background

The basic ID3 algorithm is deterministic in nature, and is inadequate for most real-world applications where uncertainties may be present in the data [6]. It is designed to include all positive instances while excluding all negative ones. This weakness of ID3 has recently been acknowledged and a number of modifications have been proposed to address this problem (e.g., [1, 2, 3, 16, 21, 22]). Most of the modifications focus on the pruning of the decision trees to reduce less significant branches, some of which may be due to noise. Depending on whether pruning occurs before or after the construction of the decision tree, they are called pre-pruning and post-pruning techniques, respectively. Pre-pruning methods such as the one described in [21] halt the branching process at a node in a decision tree when no attribute is able to increase the information gain for the classification task. A more interesting method based on the chi-square test for stochastic independence between an attribute and the classes has been adopted by Quinlan [20] to terminate a branch of a decision tree at a node when the attribute associated with the node is tested to be statistically independent of the classes. The problem is that the chi-square test can be used only when a large training set is available [20].

Pre-pruning of decision trees implies that the termination decision can only be made based on local information alone [1]. To address this problem, post-pruning of already-formed decision trees has been proposed. A recent version of ID3 constructs a set of production rules equivalent to a decision tree [22]. The terms on the condition side of each rule are examined to determine, according to the Fisher's exact test for independence, the ones that are relevant for the classification process. Conditions that are irrelevant are discarded from a rule. To further simplify the rules, each of them is omitted in turn to determine how well the rest of the rules perform on the training instances. If omission of a rule would not lead to more misclassified instances, then the least useful rule is discarded and the process is repeated until no such rule can be found. The disadvantage here is that the process is rather inefficient [22].

Like ID3, the basic AQ algorithm is not designed to handle uncertainties. To reduce overfitting caused by noisy training data, a number of pre-processing and post-processing techniques have been used in recent systems such as AQ11 [14] and AQ15 [15]. For the diagnosis of soybean diseases, an external system called ESEL was used to select the most
representative training examples before the application of AQ11 in order to reduce the effect of noise in the data [13, 14]. The best known AQ-based system for learning from noisy data is AQ15. Although still dependent on the basic deterministic AQ covering algorithm, AQ15 utilizes a built-in rule truncation and flexible matching procedure called TRUNC [15].

In the presence of uncertainty, some training events may be misclassified, and some of the rules generated using such noisy data may be incorrect. AQ15 uses a rule truncation technique to remove (truncate) portions of the rules that may be due to noise. Associated with each complex within a rule, there are two weights: \( t \) and \( u \), where \( t \) represents the total number of events covered by the complex, and \( u \) represents the number of events uniquely covered by the complex. The complex with the highest \( t \)-weight is viewed as the most representative description of events of the class, while the complex with lower \( u \)-weights is considered as describing exceptional cases, some of which may be due to noise. Therefore, the lower ranking complexes can be truncated from the cover to reduce the effect of noise.

With the rules truncated, a flexible match routine, as opposed to strict match, is needed [15]. Three outcomes are possible with flexible match. First, only one rule satisfies an event, and class implied by that rule is assigned to the event. This is the simplest case, and it is identical to the strict match paradigm. In the second case, more than one rule may cover the event, and there is a need to determine which class should be assigned. Finally, no rules may be found to cover the event, and again one needs to determine which class most closely represents the test event. In both cases, a simple probability related distance measure is used to determine the final classification result [15]. The chief weakness of AQ15 is that it is inefficient. In addition, its rule truncating procedure seems rather arbitrary and potentially important rules may be eliminated in some cases.

In this paper, we describe an inductive learning method designed specifically for the acquisition of knowledge from noisy domains. Based on the powerful probabilistic inference technique [4, 5], our method is particularly suitable for acquiring medical diagnostic rules from a set of noisy patient records. The method has been implemented and tested with real-life medical data [27]. Here, we are particularly interested in how our method compares with the other leading inductive learning systems in medical knowledge acquisition tasks.

A Robust Method for Medical Knowledge Acquisition

Our inductive learning method has been described in detail elsewhere [4, 5]. For completeness, we present a brief overview here. The approach consists of three phases. The first phase identifies the important attribute values that provide significant information for the assignment of each class. In the second phase, classification rules are constructed based on the characteristics detected for each class in phase one. Finally, the third phase involves the classification of new instances by partially matching the descriptions of each class.

Determine Relevant Features

Given a set of real-life medical data that may be noisy or incomplete and is characterized by \( M \) instances of patient records pre-classified into \( P \) known classes (e.g., types of diseases or
treatments), \( c_p, p = 1, \ldots, P \). Suppose each training instance is described by \( n \) distinct symptoms or attributes, \( A_1, \ldots, A_n \). For any distinct attribute \( A_j \), there is a domain of plausible values, \( \text{domain}(A_j) = \{v_{jk} | k = 1, \ldots, K\} \), where \( v_{jk} \) can be numeric, symbolic, or both.

To uncover the inherent associative heuristics between the attributes and classes, we first need to discover which attributes are important in determining class membership. To do so, for each unique attribute value relation, we determine the number of training events covered for each of the given classes. For each possible class, our method searches through the attribute value space to find the relevant features. A feature \( A_j = v_{jk} \) is relevant if the conditional probability of events belonging to some particular class \( c_p \), given \( A_j = v_{jk} \) is significantly different from the a priori probability of events belonging to \( c_p \). In other words, we need to determine whether or not the number of subjects in class \( c_p \) that are observed to have the feature \( v_{jk} \), denoted \( o_{pk} \), deviates significantly from \( e_{pk} \), the expected number of subjects in class \( c_p \) with the feature \( v_{jk} \) under the assumption that the attributes are randomly distributed among the subjects. We define \( e_{pk} = \frac{O_p \cdot o_{pk}}{M'} \), where \( O_p = \sum_{k=1}^{K} o_{pk} \) is the total number of objects in the training set that are classified into class \( c_p \), \( o_{pk} = \sum_{p=1}^{P} o_{pk} \) is the total number of objects with attributes \( v_{jk} \), and \( M' = \sum_{p=1}^{P} o_{pk} \leq M \) is the total number of objects in the training set with the possibility of missing data. To measure the significance of the difference between \( e_{pk} \) and \( o_{pk} \), we adopt an objective measure based on the techniques described in [4, 5]:

\[
Z_{pk} = \frac{o_{pk} - e_{pk}}{\sqrt{e_{pk}}}
\]  

(1)

The maximum likelihood estimate of its variance \( \gamma_{pk} \) is calculated as [4, 5]:

\[
\gamma_{pk} = \left(1 - \frac{O_p}{M}\right) \left(1 - \frac{o_{pk}}{M}ight)
\]  

(2)

And therefore,

\[
d_{pk} = \frac{Z_{pk}}{\sqrt{\gamma_{pk}}}
\]  

(3)

has an approximate standard normal distribution.

If \( d_{pk} \) is greater than 1.96, the 95th percentile of the normal distribution, we say that attribute value \( v_{jk} \) is a relevant characteristic of the objects belonging to \( c_p \) with a confidence level of 95%. The sign of the \( d_{pk} \) indicates whether the presence or the absence of \( v_{jk} \) is a relevant feature of \( c_p \). Values that do not affect the class assignment for this attribute are discarded from further process to minimize overfitting.
Generate Classification Rules

Decision rules that will be used to classify new objects can be generated using the relevant features that have been determined to affect class memberships. These rules all take the form: IF <condition> then <conclusion> with weight of evidence $W$. The condition side specifies the attribute and attribute values an object must possess in order to be classified into the object class indicated on the conclusion side. The weight of evidence $W$ is a measure of uncertainty in a noisy environment. It is calculated based on the information theoretic concept of mutual information:

$$I(\text{Class} = c_p; A_j = v_{jk}) = \log \frac{\Pr(\text{Class} = c_p | A_j = v_{jk})}{\Pr(\text{Class} = c_p)}$$  \hspace{1cm} (4)

Mutual information measures the change in uncertainty about the assignment of an object to a class $c_p$ given that it has attribute $v_{jk}$. The weight of evidence is calculated as \[4, 51:

$$W(\text{Class} = c_p | \text{Class} \neq c_p; A_j = v_{jk}) = I(\text{Class} = c_p; A_j = v_{jk}) - I(\text{Class} \neq c_p; A_j = v_{jk})$$  \hspace{1cm} (5)

Intuitively, $W$ can be interpreted as a measure of the difference in the gain in information when an instance characterized by $v_{jk}$ is assigned to $c_p$, and when it is assigned to other classes. The weight of evidence is positive if the attribute value $v_{jk}$ provides positive evidence supporting the class assignment of the instance to $c_p$; otherwise it is negative.

Determine Class Membership

Using the generated classification rules, a new event that was not in the training set can be classified. The rule base is searched sequentially to match the attributes of the given event to be classified with the left hand side (condition side) of the rules. If a match is found, we say there is positive or negative evidence supporting the classification of the new event into the class specified by the conclusion side of the rule, depending on the sign of the rule's weight of evidence. In fact, multiple rules are often found to match an event's attributes. Suppose $val_1, ..., val_{i_1}, ..., val_{i_m}$ are the $n$ attribute values associated with the event $e$ to be classified, then the total weight of evidence by all $n$ attributes of $e$ in favor of it being assigned to $c_p$, as opposed to being assigned to any other class is simply the sum of the weight of evidence provided by each individual attribute value of $e$ that is deemed relevant:

$$W(C_e = c_p / C_e \neq c_p, val_{i_1}, ..., val_{i_m}) = \sum_{i=1}^{m} W(C_e = c_p / C_e \neq c_p, val_{i})$$  \hspace{1cm} (6)

where $m$ attributes match one or more classification rules, and $m \leq n$. It is also possible that a particular attribute of an event indicates that it belongs to more than one class. Using the sum of the weight of evidence as a measure, we consider the class $c_p$ with the greatest total weight of evidence as the class to be assigned \[4, 51. That is:

$$W(C_e = c_p / C_e \neq c_p, val_{i_1}, ..., val_{i_m}) > W(C_e = c_h / C_e \neq c_h, val_{i_1}, ..., val_{i_m})$$  \hspace{1cm} (7)

where $h = 1, ..., P'$, and $h \neq p$. $P' \leq P$ is the number of classes matched by the attribute values according to the rules.
Performance Evaluation With Medical Data

The probabilistic inductive learning algorithm has been implemented in a knowledge acquisition tool called APACS [5]. To assess the performance of the proposed approach, we conducted a comparative study using real-world medical data with the systems based on the AQ and ID3 algorithms. We implemented a version of AQ11, AQ15 and three versions of ID3 for comparison tests. For the purpose of providing a reference, we have also implemented a simple Bayesian Classifier. We describe these implementations as follows:

1) The AQ11 implementation is based on the original AQ algorithm as described in [12].
2) The AQ15 implementation includes the post-processor TRUNC, as described in [15].
3) The ID3 implementation is based on the TDIDT algorithm described in [19].
4) The ID3 implementation with pre-pruning terminates the branching process using the chi-square test [20].
5) The ID3 implementation with post-pruning simplifies a completed tree using Fisher's Exact Test for statistical dependence [22].
6) The Bayesian Classifier implementation uses the class conditional probabilities to predict class assignment.
7) The APACS implementation is based on the inductive inference technique, without rejection (i.e., an event is assigned to the most common training class if there is not enough evidence supporting it to be assigned to any specific class) [5].

All systems were implemented in C, and all tests were conducted on a 80486 micro computer running at 33Mhz. We used three sets of real-world medical data with different degrees of noise for the comparative study.

Experiment 1

The data used for the first experiment was collected from the Forest City Hospital in Ohio [18]. It consisted of 120 patient records of an adult patient Intensive Care Unit (ICU). Each of the 120 patients was characterized by twelve attributes representing 12 different symptoms from the Central Nervous System, the Respiratory System, the Cardiovascular System, the Skin Signs and the Renal System of the patients. All 120 patients were classified into two groups: those required intensive care, and those that could be discharged to the main floor. The data set was distributed in the following way: 66 patients were classified into group 1, and 54 were classified into group 2. This set of data was moderately noisy with no missing values.

To evaluate the relative performance of APACS against the AQ and ID3 based implementations, we randomly selected a set of 84 patient records (70% of the available data) from the data set for training, and the rest (36 records) were used for testing. The concept to be learned was which patients should be placed in intensive care, and which ones should be discharged. The experiment is repeated 10 times using 10 different sets of randomly selected records for training. Each time, the records used for testing were not included in the training process. We used the same sets of test data for all of the learning systems we compared. The final comparative results summarized in Table 1 represent the average results of the 10 runs.
Table 1: Comparative Test Results from Experiment I

<table>
<thead>
<tr>
<th>Inductive Learning Systems</th>
<th>Accuracy</th>
<th>Learning Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic AQ11</td>
<td>76.4%</td>
<td>3.57 sec</td>
</tr>
<tr>
<td>AQ15 (with TRUNC)</td>
<td>78.9%</td>
<td>3.58 sec</td>
</tr>
<tr>
<td>ID3</td>
<td>85.6%</td>
<td>0.22 sec</td>
</tr>
<tr>
<td>ID3 with pre-pruning</td>
<td>89.2%</td>
<td>0.11 sec</td>
</tr>
<tr>
<td>ID3 with post-pruning</td>
<td>82.0%</td>
<td>1.93 sec</td>
</tr>
<tr>
<td>Bayesian Classifier</td>
<td>85.0%</td>
<td>0.11 sec</td>
</tr>
<tr>
<td>APACS</td>
<td>91.1%</td>
<td>0.19 sec</td>
</tr>
</tbody>
</table>

Experiment II

For the second experiment, we used another set of patient data from the Forest City Hospital [18]. This data set consisted of 99 patient records described in terms of the same 12 attributes as those in the data set we used for experiment I. Except this time, each patient was diagnosed by a group of physicians into one of the following four disease types: i) chest disease, ii) abdominal disease, iii) cardiac disease, and iv) neurological disease. 15 patients were available for group 1, 33 for group 2, 25 for group 3, and 26 for group 4.

To test the learning systems, 80% of the available records from each disease group were randomly selected for training and the rest for testing. The experiment was again repeated 10 times using ten different sets of randomly selected training and test data. The average results of the 10 runs using the different inductive learners are shown in Table 2.

Table 2: Comparative Test Results from Experiment II

<table>
<thead>
<tr>
<th>Inductive Learning Systems</th>
<th>Accuracy</th>
<th>Learning Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic AQ11</td>
<td>70.0%</td>
<td>6.37 sec</td>
</tr>
<tr>
<td>AQ15 (with TRUNC)</td>
<td>67.5%</td>
<td>5.55 sec</td>
</tr>
<tr>
<td>ID3</td>
<td>79.5%</td>
<td>0.22 sec</td>
</tr>
<tr>
<td>ID3 with pre-pruning</td>
<td>59.5%</td>
<td>0.12 sec</td>
</tr>
<tr>
<td>ID3 with post-pruning</td>
<td>87.0%</td>
<td>5.16 sec</td>
</tr>
<tr>
<td>Bayesian Classifier</td>
<td>81.0%</td>
<td>0.11 sec</td>
</tr>
<tr>
<td>APACS</td>
<td>94.0%</td>
<td>0.17 sec</td>
</tr>
</tbody>
</table>

Experiment III

For the third experiment, we wanted to use a set of data that is known to be very noisy and uncertain. A good candidate was a set of dermatoglyphics data. Ever since a 1936 report claiming that patients with Down's Syndrome showed characteristic features on palms and fingers [7], medical researchers have tied dermatoglyphic characteristics to congenital diseases [24]. Our set of data consisted of 126 finger-print patterns, and each instance was represented
in terms of the *ad4 angle* and the *ab ridge-count* of both the right and left hands of 126 different human subjects. These subjects had been divided into three groups. The first group includes 51 myelomeningocele patients drawn from the Spinal Dysfunction Clinic of the A. I. du Pont Institute. The second and the third groups consisted of, respectively, 40 and 35 normal subjects that show different dermatoglyphic patterns [26].

The experiment was conducted using 10 random sets of 86 training samples (representing 70% of the total numbers of subjects available). For each run, the remaining subjects (40) were used for testing. Classification results were compared with the given grouping, and the comparative results are summarized in Table 3.

<table>
<thead>
<tr>
<th>Inductive Learning Systems</th>
<th>Accuracy</th>
<th>Learning Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic AQ11</td>
<td>49.7%</td>
<td>15.93 sec</td>
</tr>
<tr>
<td>AQ15 (with TRUNC)</td>
<td>51.3%</td>
<td>15.15 sec</td>
</tr>
<tr>
<td>ID3</td>
<td>51.0%</td>
<td>0.22 sec</td>
</tr>
<tr>
<td>ID3 with pre-pruning</td>
<td>48.2%</td>
<td>0.12 sec</td>
</tr>
<tr>
<td>ID3 with post-pruning</td>
<td>55.0%</td>
<td>8.68 sec</td>
</tr>
<tr>
<td>Bayesian Classifier</td>
<td>44.7%</td>
<td>0.17 sec</td>
</tr>
<tr>
<td>APACS</td>
<td>68.2%</td>
<td>0.28 sec</td>
</tr>
</tbody>
</table>

**Discussion of Experimental Results**

These experiments demonstrated that our knowledge acquisition system APACS can effectively acquire medical knowledge to produce the best classification accuracy. Compared with the AQ-based systems, APACS consistently performed better both in terms of classification accuracy and computational efficiency. Our method was as much as 50 times faster than AQ-based systems while retaining at least a 10% accuracy edge. We also note that while AQ15 was marginally better than the basic AQ algorithm in two out of the three experiments, its TRUNC function can sometimes truncate useful classification information inadvertently causing it to perform worse than AQ11 in experiment II. Although slightly faster than APACS, the Bayesian Classifier was clearly outperformed in terms of accuracy especially in the third experiment where the data was particularly uncertain.

Our method also consistently provided better classification accuracy than all of the ID3-based learning systems. Although the learning time of ID3 without pruning was comparable to APACS, post-pruning resulted in a much slower training process. The extra processing cost in ID3 with post-pruning managed to improve its classification accuracy in two of the data sets but still lagged behind APACS in every case. Pre-pruning in ID3 reduced learning time because it had the effect of reducing the search space by terminating the branching process early. However, as the results indicate, ID3 with pre-pruning actually consistently performed worse than with no pruning in terms of accuracy. This was due to the fact that the chi-square test terminated branching pre-maturely due to the absence of a very large training set.
The computational efficiency difference of the methods can be easily explained. Let $M$ represent the size of the training set, $n$ the total number of attributes, and $w$ the maximum width of an AQ star. The key AQ operation is the generation of a partial star, and its overall time complexity for each negative example is $O(n \cdot w(M + \log(n \cdot w)))$ [6]. The time complexity of the main ID3 branching operation is $O(n \cdot m)$ [6]. However, ID3 must perform this basic operation for as many times as there are nodes in the tree. Therefore, its overall complexity is $O(n \cdot M \cdot \text{no. of nodes})$. With post-pruning, even more overhead is required. The time complexity of our method to search for the relevant attribute values based on significant adjusted residuals is $O(n \cdot m)$, and it is only performed once for the entire set of training data. It is therefore not surprising that our method was extremely fast in our experiments.

Conclusions

In conclusion, our proposed inductive learning method is a simple, fast, and powerful tool for medical diagnostic knowledge acquisition in terms of production rules. Our method is efficient and effective even when the training data contains inaccurate, incomplete and inconsistent values. Relative to ID3 and AQ based methods, our approach is more computationally efficient without sacrificing classification accuracy. For these reasons, we believe APACS can be used effectively to eliminate the knowledge engineering bottleneck problem in the construction of medical expert systems.

References