HINT FOR SQUINT: A COMPUTER RELIANT DIAGNOSTIC AID FOR STRABISMUS

by

Louis R. Gieszl, MS, Jacqueline Morris,*CO, and David L. Guyton,*MD

*Wilmer Institute, Johns Hopkins

Abstract

This paper provides an outline of methods used for the implementation of a computer reliant diagnostic aid in the medical specialty of Ophthalmology. Some problems have been associated with many previous diagnostic models. A careful review indicates that the most serious problems were:

- inability to handle:
  - multiple diseases coexistent in the patient
  - partially described diseases
  - extraneous input symptoms
  - quantitative signs along with symptoms

- inflexible and non-extendible prototype models

- lack of inclusive physician-approved data banks

The use of computer techniques developed for large scale wargaming has allowed the construction of an extendable prototype model. The hierarchical searching techniques used provided the necessary partial matching solution to the above set of computational problems. An active physician-analyst dialogue permitted the avoidance of the last problem, since the actual data and diagnostic logic was physician input.

Introduction

A prototype model of a simple diagnostic aid was made using some of the information management techniques developed for large scale wargaming (see Reference [1]). This model was revised with the aid of hash coded probing techniques (suggested in Reference [2]) to allow its implementation on a minicomputer.

The first model used a separate decision table to describe (diagnose) each disease in its data base. The decision tables were hierarchically interrogated by perturbing the set of input symptoms. The use of decision tables allowed for multiple disease descriptions and permitted easier physician-analyst communication.

A second prototype model made only implicit use of decision tables (see Reference [3]). A physician could use a decision table to initially describe a disease or to document an overall disease description currently residing in the data base; but, the actual symptom-to-disease causal logic was entered one rule at a time. The rules were not collected for processing in the computer - hence they formed an implicit decision table. The mechanics of handling information retrieval from implicit decision tables in an hierarchical manner are given in Reference [4].

This paper describes the mechanics of a model that is one step beyond prototype. That step is in-depth physician input and interaction. Strabismus, a subspecialty of ophthalmology is bounded enough to allow for rather complete coverage in a limited model, but complex enough to involve the mapping of 200 signs and symptoms in 100 different diagnoses - "diseases". The medical members of this team spent considerable time (on a volunteer basis) in clearly specifying the signs, symptoms, diseases, and their causal interrelationships (see Reference [5]). Any errors or misrepresentations that might be later found in this model will likely be due to a transcription mistake by the mathematician part of this team.

It is hoped that the initial use of this model will be to assist in teaching strabismus diagnosis. This is a far easier goal to aim for than clinical diagnosis.

General Model Characteristics

The diagnostic aid described here involves the use of interactive prompting to home in on one or more diseases that are associated with the specific sign/symptom complexes that are initially input. It will be seen that this is more than simply a listing of diseases tagged by any one of the input signs or symptoms. It will also be seen that this method will easily handle overlapping...
complexes of signs and symptoms (diagnosing several diseases simultaneously). In earlier prototype models a procedure was devised to allow the computer to hierarchically guess and try diagnoses until the correct one (or more) was found. The use of user/computer interaction in this implementation eliminates the faulty guesses sooner — and is more appropriate for a teaching model.

A major consideration for the acceptance of this system is the accuracy and sufficiency of the data base within the covered medical specialty. The heavy demand for physician input to assure this was minimized as much as possible in this project by using decision tables prepared by the physician that made use of the numbering scheme already in use in his clinic. In this application it was more desirable to stay with a familiar coding system (directly from the clinical worksheets) than to try to implement an English language user interface. Of course, the computer is expected to echo back the English term corresponding to a numeric entry to minimize user input errors.

**Model Dynamics — Using The System**

The actual data base is assumed to be subject to periodic revision; hence, expected physician interaction includes new or changed data. Since the entire knowledge base (the signs/symptoms, diseases, and their logical mappings) is contained in arrays that are directly derived from decision tables, updates are simple maintenance actions — not reprogramming.

A simple example of an actual decision table and its meaning will facilitate a description of model dynamics as well as illustrate this notion of a knowledge base in an array. A decision table and a corresponding and/or tree are given in Figure 1 (for three different rules for diagnosing partially accommodative esotropia). This shows that diagnosis is made by the observation of conditions 129 and 141; or 123 and 127 and 141; or 41 and 141.

The knowledge base is a collection of decision tables (equivalent to a forest of and/or trees), where each table contains the diagnostic rules for a specific "disease". For example, the user might request of the system a diagnosis from the input symptoms:

- 23 (intermittent diplopia)
- 41 (hyperopia)
- 122 (intermittent deviation)

This would result in the creation of a possible disease list containing diagnoses:

- 14, 31, 47 derived from 122
- 8, 15, 16 derived from 41
- 8, 24 derived from 23.

There are 26 rules in the data base associated with this list of diseases (such as the three shown in the decision table). However, for the above inputs, only eleven of these rules are actually collected for clarifying interaction with the user. Fifteen rules were eliminated (including the first two in the example above) because they did not involve any of the input signs/symptoms.

The rules that are collected for diagnosis and further user interaction are placed in a list that is ordered by number of conditions that match the inputs. The chaining of all equivalent conditions throughout this list eliminates the need for all but the minimum number of queries to the user before a diagnosis is retrieved. For example, condition 127 is required for both rules associated above with diagnosis 8 and one of the two rules associated with diagnosis 14 — the user's denial of condition 127, during interaction, would automatically eliminate these rules from any additional consideration.

The first rule that is found whose conditions are all matched (from some or all of the original input, supplemented if necessary with user responses) will be given as a diagnosis. If more rules are satisfied, they will also be output. If more rules remain with partial matches, the user may elect to continue to "browse among possible diagnoses".

The original HINT (Hierarchical INTERrogation) system used a knowledge base of decision tables that were "minimized". That is, the number of branches on the and/or tree equivalents was minimized by bringing as many conditions as possible to their highest position in the tree. Figure 2 is an example of this for the tree of Figure 1. This optimization with respect to fast interrogation failures (e.g., no condition 141 implies no disease 15) made it possible to guess a bigger and bigger set of additional needed conditions to suggest a diagnosis. This was equivalent to interrogating a much larger set of hierarchically structured decision tables.

The present program replaces the "guess and interrogate" protocol with user interaction. The ordering of rules to be considered in accordance with the number of known satisfied conditions is somewhat equivalent to the optimization mentioned above.

**System Structure**

The actual knowledge base is made up of two static arrays: S, the master list of symptoms; and D, the master list of diseases. S is an inverted list; that is, each entry of S carries a pointer to every disease that that entry may be associated with (one of its possible symptoms). D is a collection of decision tables, each of which is a collection of specific rules mapping appropriate signs/symptoms into a particular diagnosis (such as the example shown above).
A number of dynamic lists are generated during execution from the knowledge base and the user inputs (original and interactive). A list of original input symptoms is kept in addition to the two dynamic lists of currently known and unknown (potentially causal) conditions. A first cut of possible diagnoses is made in one list. Another list contains the rules extracted from D that are pertinent to the current validated conditions. It is the ordering, chaining, rule extraction, and user interaction with this list that forms the heart of this HINT for Squint system.

The actual software consists of a main executive and the various routines that it uses for initializing the dynamic lists, ordering and chaining the central rule list, performing the condition matching and user interaction, and outputting the diagnoses. The entire program consists of less than 2000 lines of Fortran code. The knowledge base consists of several hundred lines of "Data" statements.

**Search Logic**

The use of tree searching and even decision tables for medical diagnosis has been suggested for some time (see References [6], [7], [8], as well as [1], [3], and [4]). The difficulties usually encountered in this type of search, in non-trivial applications, are caused by the need to perform "fuzzy" or partial matching operations with "crisp" techniques and by problems of redundant or overlapping input condition complexen. In this model the first problem is bypassed through interaction with the user (to drive the hierarchical search), while the second problem is circumvented through the strict use of decision tables - derived from any of the input conditions.

The search used for HINT for Squint was shown by example above, but a closer look at that example may clarify the generality and extendibility of this technique. A diagram showing the actual decision tables extracted from the knowledge base when the system receives the three input conditions, 122, 41, and 23, is shown in Figure 1. This figure shows the 26 rules (involving 50 overlapping conditions) that are associated with 7 possible diagnoses. The eleven rules that are actually retained for user interaction are shown as blackened in circles.

The rules that have been retained for further search (e.g., the eleven rules above) are placed in a "chalkboard" dynamic rule list, RLIST. This list may contain as many as 100 rules (in the current model), but it quickly gets smaller through user interaction. The current ordering heuristic for RLIST is simply in decreasing order of number of matched symptoms. That is, the system tries for the disease it has the most symptoms for. In particular, with the small sign/symptom complexes found in strabismus most rules are either eliminated or specified as a diagnosis with only one query.

**Conclusion**

This model is still being validated and is subject to modifications - both for computational efficiency and for better tailoring to the heuristics of strabismus diagnosis. It must undergo extensive testing and physician operations before consideration as a diagnostic teaching aid. This program was written for operation on a minicomputer, but it is hoped that it will be adaptable to a personal computer. This will take some modifications in the use of storage. The use of disk storage for the knowledge base should not pose a serious threat to running time, since this knowledge base is only referred to once - in the initial set up of RLIST.

The ordering heuristics for the "chalkboard" rules may be modified after more operational experience is gained. This modification will involve any rules of thumb that can be found that would weight one symptom complex (partially validated) above another within the field of strabismus. In addition, the actual interrogation of specific conditions in the "top" rule, and subsequent validation, diagnosis, or rule elimination may be altered to better fit the more restrictive environment of personal computer implementation.

**References**


Rules

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>129 (esotropia)</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>123 (constant deviation)</td>
<td>-</td>
<td>YES</td>
</tr>
<tr>
<td>127 (eso deviation)</td>
<td>-</td>
<td>YES</td>
</tr>
<tr>
<td>41 (hyperopia)</td>
<td>YES</td>
<td>-</td>
</tr>
<tr>
<td>141 (angle reduced with RX but not eliminated)</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

Diagnosis 15 (esotropia - partially accommodative) | YES | YES | YES

Figure 1
Example Decision Table and Corresponding Tree

Figure 2
A "Minimized" Diagnostic Tree

Figure 3
Diagnostic Rules Triggered by Input Signs/Symptoms