Internet-Based System for Diagnosis of Coronary Artery Disease

JJ Lovelace, KJ Cios, DM Sala, LS Goodenday*

University of Toledo, Toledo, Ohio, USA
*Medical College of Ohio, Toledo, Ohio, USA

Abstract

An intelligent Internet-based coronary artery disease (CAD) diagnostic system based on Tl-201 planar imaging data was developed. The system can self-generate rules to correctly diagnose obstructions in the three main arteries, namely right coronary artery(RCA), left anterior descending (LAD), circumflex (CCX), or a patient being normal (with no obstructed arteries).

The rule formation engine is based on the CLIP3 (Cover Learning using Integer Programming version 3) *machine learning algorithm. The system stores all data (training data, fuzzy sets and rules) into a relational database. The primary consideration of our design was to keep the system easy to use, thus the database operations are transparent to the user.*

The architecture of the system is a multithreaded structure that allows multiple operations to be carried out in parallel. The biggest advantage of the system is that a diagnosis can be carried out over the Internet via HTML forms.

1. Introduction

Although SPECT imaging is currently more popular there still exists circumstances where Tl-201 planar imaging is the only option. Tl-201 planar imaging has the following advantages: low cost, it is portable, and can be used on all patients. Its biggest disadvantage, however, is that the images are fuzzy and hard to read.

Given the University of Toledo's Intelligent Systems Laboratory (ISL) past experience with planar Tl-201 data, Medical College of Ohio (MCO), the supplier of the data, was interested in the ISL developing an automated system for diagnosis of the CAD. Past projects have been successful in generating rules to properly diagnose the Tl-201 planar images [1-3]. MCO already had a successful product to perform statistical analysis of the Tl-201 planar images and wanted the ISL to create a knowledge-based system to provide the diagnosis based on the results of the statistical analysis. The name of the MCO Cardiology program is Ticker Tester [4]. Ticker Tester analyzes three images acquired from Tl-201 planar imaging.

1.1 Ticker Tester

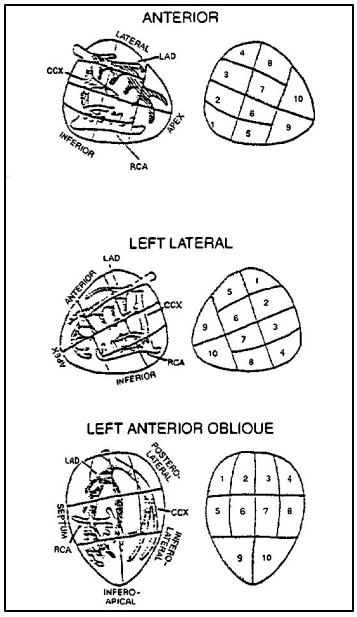

Figure 1. Tl-201 Imaging of 3 views showing 30 regions [3]

Ticker Tester is designed to perform statistical analysis on Tl-201 planar images. It accomplishes this task by dividing each of the three images into 10 sections and performing a wide variety of statistical analyses on each of the regions. The regions are shown in Figure 1.

Of interest for diagnosis is the percent of pixels in each region that are more than 2.5 standard deviations from the normal range for a region. Values this far from the normal are associated with blockage in the arteries. For this reason only these values are used for generation of the rules in the system proposed below.

1.2 System Specification

It was decided that the system should be able to autonomously deduce rules to properly diagnose patients. This was taken as a trade-off versus the time it would take to get the needed collection of rules from a group of physicians. This was to be accomplished by providing a database containing correctly diagnosed patients. New

patients were to be easily added to the database to improve the accuracy of the generated (and constantly being updated) rules. Since the ISL uses Windows NT machines and the MCO Cardiology Department uses Macintosh machines, it was decided that the application would be created to run under NT but would allow addition of patients to the training set and diagnosis to be carried out over the Internet.

Since the Internet was used, security precautions were to be implemented to protect patient confidentiality as well as to protect the database from malicious examples.

The last aspect was that the entire system should be easy to use with a minimum amount of training involved to keep the system running. The system layout is shown in figure 2.

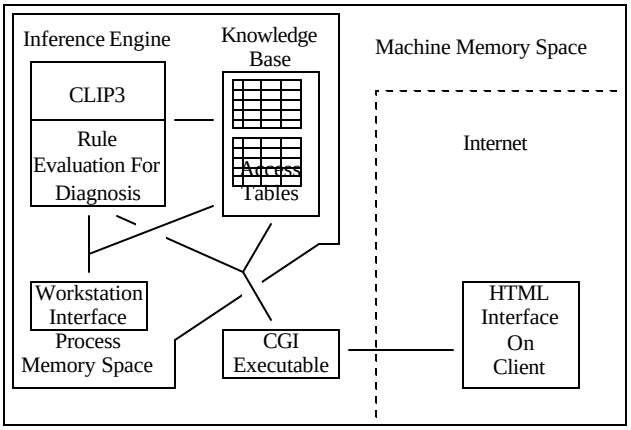

Figure 2. System Layout

2. The Knowledge-Based System

Knowledge-based systems, also called expert systems, are generally composed of three components. They are the user interface, the inference engine, and the knowledge base. The user interface provides the mechanism by which the user enters information into the knowledge base and interacts with the inference engine. The inference engine deduces facts from the knowledge base.

2.1 Knowledge Base

The knowledge base for this system is contained within an Access database. This file stores the training sets, rules and the discrete level conversion table. The system accesses the database through an ODBC driver. Through the use of ODBC it is possible to use almost any database system to store this information. Microsoft Access was chosen because it was familiar to the Cardiology Department at MCO. Each of these elements is contained in its own table.

The fields in the training patient table contain the pixel data for each of the 30 fields. There is also a record for the diagnosis of a particular patient. The diagnosis is coded as follows:

0 Normal (no stenosis of the arteries)
1 CCX stenosed
2 LAD stenosed
4 RCA stenosed

If a patient is diagnosed as having multiple arteries stenosed, adding the appropriate numbers together indicates the diagnosis. For example, if a patient has CCX and LAD the diagnosis would read 3. Additionally, there are fields for patient ID and user ID. The patient ID is a number representing the order in which a patient was added to the database. The user ID indicates who added this patient to the database.

The CLIP3 algorithm operates on discrete data. Therefore, we digitize the data into 9 levels such that an approximately equal number of the examples in the training set fall within each range. This allows areas where more examples are concentrated to be divided more precisely. Areas where not much activity is occurring are covered less precisely. A discrete level conversion table contains the boundary of each range.

The rules table contains a field for the rule and a field for the result. It also contains a field indicating if the rule is a user-entered rule (not learned) or has been modified by the user. During learning, rules that have been modified by the user can be kept and are applied before new rules are learned. Rules take on the following form:

Typical Rule: 5(4,5,7)7(1,2)

The rule reads as “if region 5 does not equal level 4,5, or 7 and region 7 does not equal level 1 or 2 then” the result is that shown in the result field for the particular rule. The notation for the rule is a result of how the CLIP3 algorithm generates the rules.

The user-rights table contains information on what users are allowed to do. The fields it contains are the user name, allowed IP’s, and user rights. The user name field is the name of an authorized user. The allowed IP’s are the IP addresses this user will connect to the sever from. The user rights indicate what this user is allowed to do, such as adding a patient or diagnosing a patient. Windows NT provides server side security.

The knowledge base contains complete information needed to perform a diagnosis and create rules. Information in Access files can be read by a wide variety of other applications for use in future projects beyond the scope of this application.

2.2 Inference Engine

The inference engine generates all of the rules from the training set. It requires no guidance from the user other than proper diagnosis for patients in the training set. The inference engine for this system is based entirely on the CLIP3 algorithm [5]. The inference engine runs in its own thread. This allows the program to do other tasks while new rules are being generated. The most recent set of rules is used until the entire new set is available.

The CLIP3 algorithm operates in three stages. First, data is presented to it in the form of two matrices. One matrix consisting of positive examples and the other consists of negative examples. In this case the positive examples would be a diagnosis we wish to find the rules for. The negative matrix would consist of all of the other examples. Since there are 4 different diagnosis we need rules for, the algorithm must be run 4 times.

In the first stage, positive examples are broken down into smaller subsets that represent collections of records that may form good rules.

The next stage is the formation of a template matrix. The number of sub-matrices generated from stage 1 determines the size of the template matrix. The matrix is converted into an IP problem and solved. The solution indicates which matrices should be used to generate the rules.

The final stage is the generation of rules. The matrices that survived stage 2 are back-projected against the matrix of negative examples. The resulting matrix is converted into a new IP problem and solved. The solution indicates which features should be used to create rules from. The rule that covers the most positive examples is selected and the examples it covers are removed from the positive example matrix. The algorithm is run again on the remaining positive examples. It continues until no positive examples remain uncovered.

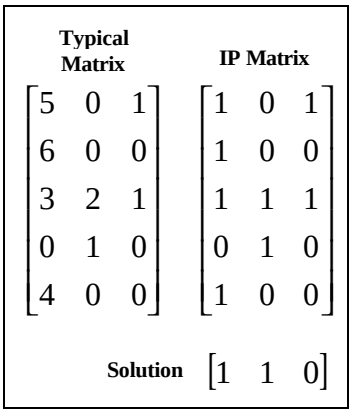

Figure 3. Typical Matrix solved as an IP problem.

Throughout the CLIP3 algorithm, the solution of IP matrix is key. This process can be repeated many hundreds of times for data sets. Figure 3 shows a typical matrix after stage 1, its binary equivalent and the IP solution. In our case, the matrices were bigger than 30 by 60 and would be too large to show in this paper. A solution to this type of a problem is a subset of columns from the original matrix that when logically AND-ed together generate a column consisting entirely of 1's. The optimal solution is the one that uses the minimum amount of columns. It would take far too long to search for the optimal solution in a matrix of 30 features by 300 patients. The binary matrix solver implemented searches for an optimal solution for three features and then takes the best three-feature solution and adds features that get it closest to the desired solution until a solution is found. Next, it attempts to remove a feature and still remain at the desired solution. The solution found is typically within about two features of the ideal solution.

2.3 User Interface

The final component of the system is the user interface. In the context of this system there are two user interfaces. One is available at the workstation running this application. The other is available over the Internet.

2.3.1 Workstation Interface

The workstation interface consists of a tabbed notebook style form. There is one tab for each of the workstation interfaces primary duties. The workstation interface has complete control of over the system. The visibility of the tabs is based on the user's access level.

The first tab is for Diagnosis. Selecting this tab brings up a display where the user can enter data for diagnosis by the system. The results are displayed as well as the initial patient information. The user has the option of entering the data manually, or as a file. The data in a file is scanned for numbers. It assumes the first ten numbers are for the ANTerior view. The next ten are for the LATeral view. The final ten are for the LAO (left anterior oblique) view of the gamma camera.

The next tab is the training patient tab. Selecting this view brings up a table showing all of the current patients. Patients can be added, modified, or deleted from this view. When a patient is added the user is shown a form which is identical the form used by the WEB interface. The user has the option to fill out the form from data contained in a file.

The next tab is the discretization tab. When selected it shows the user the discretization levels for each region. The levels can be regenerated from here. Regions are selected via a scroll bar. Additionally, the user for more precise control can modify each level setting. A graph shows how the levels are digitized.

The next tab is the rule tab. When selected the rule tab shows the current rules in the database. The user can change to any rule by way of a scroll bar. The user can modify any rule. Doing so causes the system to flag the rule as a custom rule. The user can add their own custom rule(s) that will be applied before the system implements

the CLIP3 algorithm to generate more rules. The user can also instruct the system to regenerate the entire rule set.

The last tab is where security information is set for WEB users and workstation users. Each user is assigned a user name and a password. The password is only needed at the workstation side. The valid IP addresses serve as the password for web users. All security information is set from this tab.

There is also a menu at the top of the interface. Parameters to modify CLIP3 performance and enabling the Internet features are available from this menu.

2.3.2 Internet Interface

On the server side, the Internet portion runs in its own thread allowing the Workstation users and Intemet users simultaneous access without any notice in performance change. The Internet interface consists of HTML forms. Currently, there are two forms. One form allows a user to generate a diagnosis. The other allows a user to submit a new patient into the database. Both forms submit information to the server with the GET method.

The GET method appends information contained in the form to the HTML reference as a string. This method was selected because every WEB browser on the market supports it and it is the only method of submitting data that can safely pass through a firewall. The information contained in the form has no information that can link it to a specific patient keeping patient confidentiality secure.

The information is transmitted via CGI. In Windows NT, this means that the WEB server takes the query string sent from the client form and executes an instance of the CGI executable. The executable breaks down the query string and passes the information to the heart diagnosis program via a pipe. A pipe is a conduit that transfers information between processes in the Windows NT environment. A process is any application that is running in memory. The CAD diagnosis program determines the user request, performs the appropriate action, and sends the results back through the pipe to the CGI executable. Based on the results, the CGI executable outputs the appropriate result in the HTML format to the WEB server, which passes this information back to the user.

3. Application

As described earlier the region values were digitized into 1 of 9 levels. A histogram equalization process determined the ranges for these levels. The process was accomplished in the following manner. Level 1 was assigned to values of 0.00. The remaining values were divided into 8 divisions containing equal numbers of patients. The results for lateral region 1 are shown in Table 1.

Table 1

Level	Range	Total
1	0.00-0.00	47
2	0.01-2.20	10
3	2.21-5.15	11
4	5.16-6.75	11
5	6.76-8.75	11
6	8.76-12.75	11
7	12.76-19.35	11
8	19.36-38.20	11
9	38.21-100.00	11

This process was repeated for each of the 30 regions. Using the results, the training set patients were digitized and fed into the CLIP3 engine for rule generation. It took 30 seconds for a 133 MHz system to generate 8 rules to successfully classify the 134 patient training set in to the proper diagnosis.

4. Conclusions

We have built an expert system, which can autonomously generate rules to correctly diagnose patients and can be operated across the Internet.

Further improvements to the system may include fuzzification of the rules, and better tuning of the parameters associated with the CLIP3 algorithm. For example, the membership value could indicate a relative certainty in the diagnosis result. It should also allow the machine to diagnose patients with multiple ailments.

References

[1] Cios, Shin, and Goodenday. Using Fuzzy Sets to Diagnose Coronary Artery Disease, IEEE Computer Magazine, 1991:57-63.

[2] Cios and Liu. CLIP2: A Machine Learning Algorithm Using Integer Linear Programming Part 1 & 2. Kybernetics;24(2)29-50&24(3):28-40.

[3] Cios, Goodenday, and Sztandera. Hybrid Intelligence System for Diagnosing Coronary Stenosis, 1994:723-29.

[4] Nelson, Leighton, Andrews, Goodenday, Yonovitz,, Thekdi. "A Comparison of Methods for the Analysis of Stress Thallium-201 Scintigrams," Computers in Cardiology, September 1979:315-18.

[5] Cios, Wedding, and Liu N. CLIP3: Cover Learning Using Integer Programming. Kybernetes:26(4-5):513-536.

Address for Correspondence:

Jeff Lovelace
Bioengineering Department, University of Toledo
2801 W. Bancroft St.
Toledo, OH 43606
jlovelac@eng.utoledo.edu