A Hybrid Case-Based Medical Diagnosis System*

Chien-Chang Hsu[1] and Cheng-Seen Ho[2]

[1]cch@ailab2.et.ntust.edu.tw

[2] cheng-seen.ho@acm.org

Department of Electronic Engineering, National Taiwan University of Science and Technology

43 Keelung Road, Section 4, Taipei, TAIWAN 106

Abstract

This paper proposes a hybrid case-based system to help the physician. It includes a hypermedia human-machine interface and a hybrid case-based reasoner. The hypermedia human-machine interface provides a friendly human body image map for the clinician to easily enter a given consultation. It utilizes a medicine-related commonsense knowledge base to help complete the input data during the consultation. The hybrid case-based reasoner is responsible for selecting and adapting relevant cases from the case library into a diagnosis for the consultation. This reasoner does those jobs by hybridizing many techniques. Basically it uses a distributed fuzzy neural network for case retrieval. It employs decision theory, constrained induction trees, and relevance theory for case adaptation involving case combination. The technique is also used for learning new cases into the case library. Hybridizing these techniques together can effectively produce a high quality diagnosis for a given medical consultation.

Keywords: Hybrid Systems, Case-Based Reasoning

1. Introduction

Diagnosis is a process that determines the cause of a problem based on some observed symptoms. In clinic medical diagnosis, there exist tremendous types of diseases. Some of them have similar symptoms and many of them can cause complications. A clinician has to carefully investigate a patient's symptoms, chief complaints, as well as pathology examination to decide possible diseases. It takes years of training and practice for a physician to make correct decisions. This worsens when the related etiology is hard to discern. This diagnosis process may become easier and reliable if equipped with a system that provides past diagnosis cases of different morbidity, since the clinician can benefit a lot from these prior cases.

Case-based reasoning (CBR) is a method that uses experience to solve new problems [1, 12, 13, 15]. It comes up with solutions by adapting old ones that have successfully solved previous problems similar to the given ones. Employing this technique in the above diagnosis process, it works as follows. First CBR retrieves previous

* This work was support in part by the National Science Council of ROC under grant number NSC-85-2213-E-011-058.

similar diagnoses from a medical case library using the significant features of the given case as matching metrics. It then evaluates and compares those diagnoses to identify the most suitable ones for adaptation to the given cases. It will strive to store the newly adapted cases in the case library to improve its performance after the given problem is successfully solved. This problem-solving paradigm looks quite straightforward for medical diagnosis. However, diseases may manifest themselves in a huge number of symptoms and laboratory data. Which of them are significant enough for recognizing relevant previous cases poses the first problem. How to identify most relevant features and how to handle incomplete data in the previous cases during adaptation are even harder problems. Finally, maintaining a case library so that it can be efficiently and effectively used is not easy at all either.

In this paper, we propose a hybrid case-based system (HCBS) that solves the above problems in applying CBR to medical diagnosis. It integrates a medicine-related commonsense knowledge base [10], a hybridized fuzzy symbolic and sub-symbolic reasoning technique, an induction-based learning technique, and a medical-diagnosis-based case library to support the diagnosis task. This integration is interesting at several perspectives. First, with the help of the medicine-related commonsense knowledge base, it can find the most important features serving as effective clues to find similar cases. This knowledge base also helps verify the rationale underlying the input data. Second, By utilizing a parallel subsymbolic matching mechanism and an adaptability-guided retrieval technique it can retrieve past experiences in much the same way as human beings [13]. Third, by incorporating fuzzy processing it can properly handle the approximate matching problem, which resembles human's behavior and makes case-retrieval better noise-tolerant. This improves the efficiency of traditional symbol-based case retrieval. Fourth, decision theory and induction techniques allow it to create better solutions by discovering missing features. Finally, it can learn new cases by carefully examining whether they are subsumed in the case library, which not only reduces the growing speed of the case library but also reduces the frequency of re-training the subsymbolic systems.

2. System Architecture

Fig. 1 is the HCBS architecture for medical disease diagnosis.

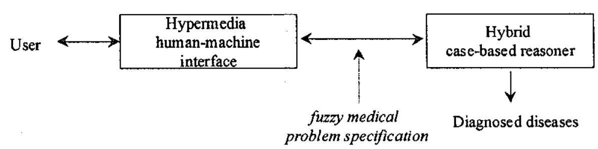

Fig. 1 Architecture of the hybrid case-based medical diagnosis system

It contains two modules, i.e., a hypermedia human-machine interface and a hybrid case-based reasoner. The hypermedia human-machine interface allows the clinician to interact with the system via a friendly human body image. It also improves the interaction quality by utilizing the medicine-related commonsense knowledge base to determine the rationale of the input data and to select the most important features from them. The knowledge base stores medicine-related commonsense of an average doctor. The hybrid case-based reasoner integrates a variety of techniques around the case-based reasoning paradigm to produce a better diagnosis for the given patient. Integrated techniques include the use of a fuzzy neural network to perform parallel matching against previous experiences stored in the case library, the use of induction heuristics to derive important features from the most similar cases, the use of decision theory to select the most adaptable features from all candidate cases, and the use of rule-based reasoning to perform case adaptation. The following sections describe these two modules in more detail.

3. Hypermedia Human-Machine Interface

Fig 2 elaborates the architecture of the hypermedia human-machine interface.

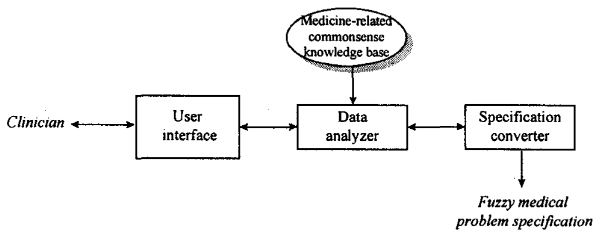

Fig. 2 Hypermedia human-machine interface

First, the user interface displays a human body image with visible organs for entering patient data (Fig. 3). The image classifies diseases into fourteen categories, named after proper organs, i.e., cardiology, respiratory chest, hematology, proctology, cardiovascular, gastric and intestine, hepatology, nephorology, urology, immune diseases, infectious diseases, endocrine, neurology, metabolism, and general medicine [5, 7]. By clicking on an interested organ, a clinician can properly respond to the associated pop-up checklists. Detailed data items are closely related to the case structure to be described in next section.

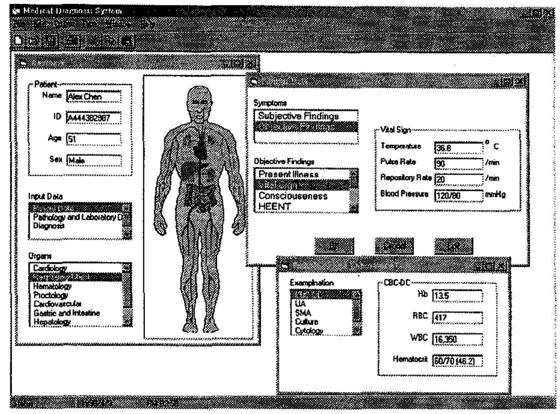

Fig. 3 User interface

The data analyzer checks the input data for completeness and consistency. If any input data is vague or in conflict with each other, it will ask the clinician for clarification. This can reduce the fuzziness of the patient data. The analyzer also analyzes the relationale of the input data with the help of the medicine-related commonsense. This helps to pinpoint the relevance of the input data items.

The medicine-related commonsense knowledge base contains the "commonsense about medicine", by which we mean the knowledge about the model of a human body, basic physical examination procedures, basic diagnosis methodology and procedures, basic morbidity of different diseases, medical decision procedures, and basic diagnosis-related principles and concepts. We treat this as the commonsense of an average doctor. This does a great help when the data analyzer is striving to prune unreasonable input data.

The specification converter is responsible for transforming the analyzed input data into a fuzzy specification. This is done by extracting those data marked as significant by the medicine-related commonsense knowledge base. The degrees of significance are then transformed into fuzzy degrees. It finally outputs the important patient data items with fuzzy degrees as a fuzzy medical specification to the hybrid case-based reasoner.

4. Hybrid Case-Based Reasoner

Fig. 4 shows the architecture of the hybrid case-based reasoner.

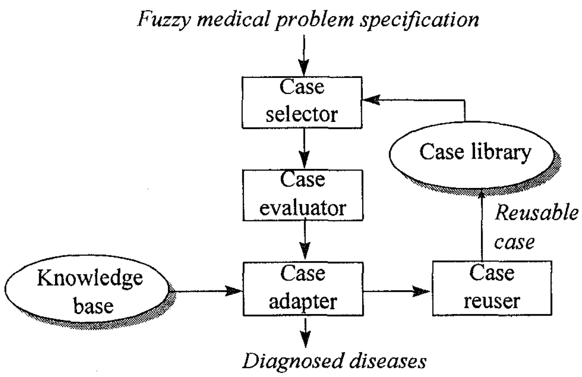

Fig. 4 Hybrid case-based reasoner

Its operation is illustrated in Fig. 5. Basically, the case selector takes the fuzzy medical specification as a pattern to select the cases that are mostly likely to be useful to solve this specification from the case library. The case evaluator then induces the selected cases into a constrained induction tree and employs decision theory to choose the most relevant features from the tree for adaptation. The case adapter does actual adaptation with the help of the adaptation knowledge. A successfully adapted case becomes the diagnosis result for the given patient. It will later be sent to the case reuser to see whether it deserves storage into the case library. The following goes through each module for details.

Selection (case selector)	Select most likely cases as candidate solutions from the case library for the input clinic chart using the distributed neural network.
Evaluation (case evaluator)	Induce important features from the candidate cases into a constrained induction tree and compute each node's expected utility using decision theory.
Adaptation (case adapter)	Prune the irrelevant features in constrained induction tree about the patient data. Adapt the candidate cases guided by the pruned constrained induction tree.
Reuse (case reuser)	Compare the new case with the original constrained induction tree to check any possible path from root to leaves. If a path exists, discard the new case, else compute similarities between the new case and all existing candidate cases. If they are below the threshold, add it to the case library and retrain the corresponding sub-neural networks. Else discard it.

Fig. 5 Hybrid case-based reasoning algorithm

First, the case library contains instances of medical diagnoses, each called a case. A case contains a patient model and a diagnosis model [2]. The patient model describes subjective findings, objective findings, and pathology and laboratory examinations about a patient. The subjective findings refer to the patient's family history, his personal history, and chief complaints. The objective findings contain the patient's physical examinations including vital sign, consciousness, HEENT (Head, Eye, Ear, Nose and Throat), neck, heart, thorax and lungs, breast, abdomen, extremity, urinary and neurology. The pathology and laboratory examinations refer to complete blood count-differential count (CBC-DC), urine analysis (UA), serum medical analysis (SMA), cultures, X-ray image, and electrocardiogram (EKG). The diagnosis model records the scenario of how a diagnosis is proceeded. In addition, a case may also contain domain-specific knowledge for adaptation. This knowledge is called case-specific adaptation knowledge.

The case selector is a distributed fuzzy neural network containing two layers of nets (Fig. 6).

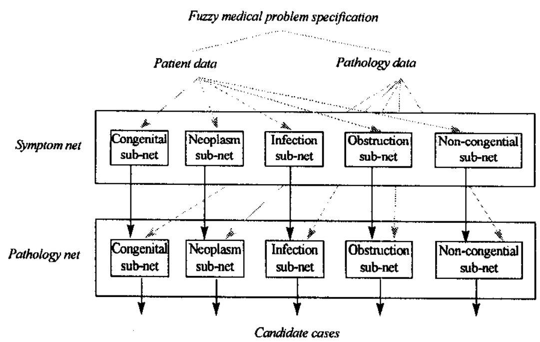

Fig. 6 Distributed fuzzy neural network

The first layer is the symptom net, which determines the similarity of the subjective and objective findings between the patient and the cases in the case library. The second layer is the pathology net, which does the similar job on the pathology and laboratory data. Each layer is further partitioned into sub-nets in accord with the pathology types of the cases, namely the congenital type, neoplasm type, infection type, obstructive type and non-congenital type [5]. The main advantage of this sub-net design is to alleviate retraining complexity; when a new case is to be introduced into the case library, we only need to re-train the corresponding sub-nets while keeping the others intact.

The output of the case selector may contain more than one candidate case. All of them contain features that are related to the morbidity of the patient. The case evaluator first induces these features into a constrained induction tree, i.e., an induction tree with constraint links to guide the selection of most possible diagnoses. The tree will help to select most appropriate features and cut down unnecessary search spaces. The construction algorithm for the constrained induction tree is shown in Fig. 7, which is based on the CLS algorithm [4, 11]. Fig. 9 exemplifies a constrained induction tree using pleural effusion and staphylococci pneumonia [6, 10] as two candidate cases (Fig. 8). The root of the induction tree, i.e., Disease Category (DC), represents the most likely feature in this example.

1. If all the feature value v_i, $i=1..n$ in each case of training set S are the same, then create a same node and go to step 7.
2. Otherwise, select the attribute in the following sequence:
   A、 Objective findings
   B、 Subjective findings
   C、 Pathology and laboratory data
   D、 Diagnosis
   E、 Diagnosis procedures
3. Select an attribute A with values v_i, $i=1..n$ and create a decision node.
4. Partition the training features in training set S into subsets s_i, $i=1..n$ according to the value of v_i.
5. Compute the probability $P(A \rightarrow s_i)$ for each attribute value.
6. Apply the algorithm recursively to each of the set s_i.
7. Create a constraint node for each constraint c_i; connect a constraint link to each node in c_i.

Fig. 7 Construction algorithm for constrained induction trees

	Case #1	Case #2
Personal history 1 (PH1)	Pneumoria	Pneumoria
Personal history 2 (PH2)	Typhoid	Typhoid
Chief complaint 1 (CC1)	Chest pain	Chest pain
Practical illness 1 (PI1)	Cough	Cough
Practical illness 2 (PI2)	Dyspnea	Dyspnea
Thoracentesis (TH)	Effusion	Effusion
Effusion protein (EP)	high	high
Specific gravity (SG)	high	high
Cultures (CU)	Bacteria (pneumoccal)	Bacteria (staphylococcus)
Cytology (CY)	Liver tumor	Normal
x-ray (XR)	Pleural effusion	Pleural effusion
Disease category (DC)	Respiratory	Respiratory
Disease organ (DO)	Pulmonary	Pulmonary
Diagnostic 1 (DI1)	Pleural effusion	Pleural effusion
Diagnostic 2 (DI2)	pneumoccal	Streptococcic pneumonia
Diagnostic procedure 1(DP1)	Pleural biopsy	Pleural biopsy
Diagnostic procedure 2 (DP2)	Pleurosocopy	Pleurosocopy
Diagnostic procedure 3 (DP3)	Thoracentesis	Thoecentesis

Fig. 8 Example candidate cases

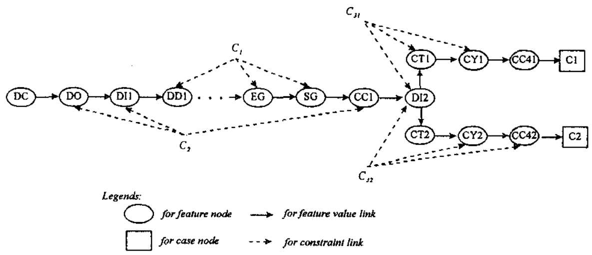

Fig. 9 Example constrained induction tree

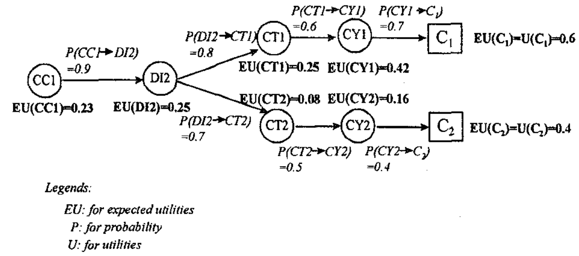

Fig. 10 Illustration of expected utilities computation

The case evaluator does a second thing, i.e., computing the expected utility for each node in the constrained induction tree. It uses Formula (1) for computation. Fig. 10 exemplifies this computation process. It starts by setting $EU(C_i)=U(C_i)$, when $U(C_i)$ is the utility of case i computed by taking into accounts its adaptability [7]. This value is then backed up to the father node of C_i by Formula (1) to compute its expected utility. This process repeats until it reaches the root node.

$$EU(A_j) = \sum_{k=1}^{n} EU(A_{jk}) * P(A_j \rightarrow A_{jk}) \quad \ldots\ldots(1)$$

where A_{jk}: *kth child node of feature node* A_j,

n: numbers of children of feature node A_j, *and*

$P(A_j \rightarrow A_{jk})$: *probability of feature value represented by link* $A_j \rightarrow A_{jk}$

The probability $P(A_j \rightarrow A_{jk})$ is assessed subjectively [14]. It represents the occurrence of the feature value $A_j \rightarrow A_{jk}$ in all categories of the case library [9]. This design essentially says that the more frequently a feature value occurs in the case base, the higher the probability that the value is relevant to the new problem. In short, the expected utility $EU(A_j)$ estimates how useful it would be to employ the subtree rooted at A_j for adaptation. In other words, higher expected utility means higher relevance [3, 8].

1. Travel the pruned constrained induction tree from root to leaves based on the patient data.
2. If a high relevant feature is not in the patient data then copy it and its value to the target case and goto step 7.
3. Else if the feature relevance is greater than 95% then go to step 4.

   Else if case-specific adaptation rule exist then adapt the feature value.

   Else search the general adaptation rules to adapt it.
4. Check the constraint links attached in the feature node.

   If violate, discard the feature node.
5. Find the sibling feature values with the highest adaptability and go to step 1.
6. Copy the matched or adapted feature to the target case.
7. If not a leaf node then follow the feature link one level down and goto step 2.

Fig. 11 Case adaptation algorithm

Given the indication of attribute relevance, now the case adapter can take turn to do case adaptation. Fig. 11 depicts the adaptation algorithm. It first prunes irrelevant features from the constrained induction tree based on the computed relevance. It then travels the pruned constrained induction tree from the root to the leaves based on the given patient data and selects the most relevant features into a target case. It finally uses the case-specific adaptation knowledge associated with the selected features to turn the target case into a diagnosis for the patient. If none of the case-specific adaptation rules can apply, it will turn to the general adaptation knowledge base, which uses general rules to do specialization, generalization, and transformation. Separation of case-specific knowledge from general knowledge to do adaptation generally improves solution quality. The case adapter is also responsible for inducing case-specific adaptation rules from the pruned constrained tree and puts them into the target case.

Finally, the case reuser maintains the case library whenever a new case is found. It uses the original constrained induction tree to check whether there already exist analogous or subsumed cases in the case library. This approach of case learning features the comparison of all existing candidate cases (represented by the induction tree) at the same time, which reduces the growing speed of the case library and the retraining necessity of the distributed neural network.

5. Conclusions

We have described a hybrid case-based system for medical diagnosis. Its hypermedia human-machine interface provides a friendly user interface for clinicians. It employs a medical-related commonsense base to help pinpoint the important information from the input data. The hybrid case-based reasoner hybridizes case-based reasoning, neural networks, fuzzy theory, induction, and knowledge-based technology to facilitate medical diagnosis. It accumulates experiences as cases in the case library and diagnoses new patients by adapting the old cases that have successfully diagnosed previous similar diseases. The distributed neural network performs approximate matching to tolerate potential noise in case retrieval. The induction technology along with relevance theory in case selection, adaptation and learning helps a lot in hammering out valuable features for the target case from existent ones and in pruning unnecessary search space. It also incorporates both case-specific and general knowledge for most robust case adaptation. This is an architecture that tightly couples case selection, adaptation and learning.

This hybrid system invites several advantages. It helps less-experienced doctors in performing efficient clinical diagnosis. The commonsense knowledge helps solve the problem of unreasonable input data and saves hard work of data analysis. It matches cases in parallel and approximately with the help of fuzzy theory. It helps identify relevant and/or lost features with the help of relevance theory and induction technique. It tightly employs the concept of constrained induction trees to integrate decision theory, induction, and relevance theory into a successful technique for case selection, adaptation and learning. Thus, it only stores those newly generated cases that deserve storage, which reduces the growing speed of the case library as well as the frequency of re-training the neural network.

References

1. B. Baphael and B. Kumar, "Indexing and Retrieval of Cases in a Case-Based Design System," *Applied Intelligence for Engineering Design, Analysis and Manufacturing,* vol. 10, no. 1, pp. 47-63, 1996.
2. B. Bates, *A Guide to Physical Examination and History Taking,* J. B. Lippincott, Philadelphi, PA, 1987.
3. A. L. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," *Artificial Intelligence,*

vol., no., pp. 245-271, 1997.

4. P. R. Cohen and E. A. Feigenbaum, *Handbook of Artificial Intelligence,* vol. III, Willian Kaufmann Pub., Los Altos, CA., 1982.

5. R. S. Cotran, V. Kumar, and S. L. Robbins, *Robins Pathologic Basis of Diseases,* W. B. Saunders Company, Philadelphia, PA, 1989.

6. M. R. Dambro, *Griffith's 5 Minute Clinical Consult,* Williams & Wilkins, Baltimore, MD, 1997.

7. A. S. Fauui, E. Brauwald, K. J. Isselbacker, J. D. Wilson, J. B. Martin, D. L. Kasper, S. L. Hauser, and D. L. Longo, *Harrison's Principles of Internal Medicine,* 14th ed., McGraw-Hill, NY, 1998.

8. M. Hadzikadic, B. F. Bohren, "Learning to Predict: INC2.5," *IEEE Transactions on Knowledge and Data Engineering,* vol. 9, no. 1, 1997.

9. C. C. Hsu and C. S. Ho, *An Intelligent Hybrid Case-Based Reasoning System,* Technique Report, Department of Electronic Engineering, National Taiwan University of Science and Technology, Taiwan, 1998.

10. W. S. J. Lin (Ed.), *Clinicopathological Conference: National Taiwan University Hospital,* Current Medicine, Taipei, Taiwan, 1979.

11. J. R. Quinlan, "Introduction of Decision Trees," *Machine Learning,* vol. 1, pp. 81-106, 1986.

12. E. Simoudis, "Using Case-Based Retrieval for Customer Technical Support," *IEEE Expert,* vol. 7, no. 5, pp. 7-12, 1992.

13. B. Smyth and M. Keane, "Experiments on Adaptation-Guided Retrieval in Case-Based Design," International conference on Case-Based Reasoning (ICCBR-97), pp.312-324, Providence, RI, 1997.

14. C. Tsatsoulis, Q. Cheng and H. Y. Wei, "Integrating Case-Based Reasoning and Decision Theory," *IEEE Expert,* vol. 12, no. 3, pp 46-55, 1997.

15. I. Watson and S. Perera, "Case-based Design: A Review and Analysis of Building Design Applications," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* vol. 11, no. 59-87, 1997.