Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 36 (2009) 65–71
Expert Systems with Applications
www.elsevier.com/locate/eswa

Hybrid approach using case-based reasoning and rule-based reasoning for domain independent clinical decision support in ICU

K. Ashwin Kumar *, Yashwardhan Singh, Sudip Sanyal

Indian Institute of Information Technology, Allahabad, Uttar Pradesh 211012, India

Abstract

This paper presents a hybrid approach of case-based reasoning and rule-based reasoning, as an alternative to the purely rule-based method, to build a clinical decision support system for ICU. This enables the system to tackle problems like high complexity, low experienced new staff and changing medical conditions. The purely rule-based method has its limitations since it requires explicit knowledge of the details of each domain of ICU, such as cardiac domain hence takes years to build knowledge base. Case-based reasoning uses knowledge in the form of specific cases to solve a new problem, and the solution is based on the similarities between the new problem and the available cases. This paper presents a case-based reasoning and rule-based reasoning based model which can provide clinical decision support for all domains of ICU unlike rule-based inference models which are highly domain knowledge specific. Experiments with real ICU data as well as simulated data clearly demonstrate the efficacy of the proposed method.

Keywords: Rule-based reasoning; Case-based reasoning; Clinical decision support system; Expert systems; Artificial intelligence

1. Introduction

Expert systems (Bobrow, Mittal, & Stefik, 1986; Ignizio, 1991; Jackson, 1998) (ES) are powerful tools that serve as adjuncts to decision making and have found wide applicability in a wide variety of areas. An expert system, also known as a knowledge based system, is a computer program that contains some of the subject-specific knowledge of one or more human experts. Expert systems can explain why data is needed and how conclusions were reached. The range of problems that can be handled by expert systems is vast. Integrating expert systems with clinical decision support systems has the potential to enhance the quality and efficiency of treatment in ICU.

ICU's maintain huge amount of data related to a single patient. Deriving conclusions out of such large data was never an easy task. Moreover domains of problems which ICU's tackle are very vast, for example Cardiology, Poisoning, Neurotrauma, Cancer and Accident etc. There are hundreds of such domains which ICU's are related to. Currently, the most common form of expert system structure is a rule-based reasoning (RBR) which deals with specific domains such as Cancer. It is apparent that there is some inflexibility in current expert system structure as decision making has become domain specific. Some of the reasons for this inflexibility in structure are

- Expertise completely depended on knowledge base.
- It takes years to build knowledge base for a single domain.

Subsequently, these shortcomings in RBR based ES have given rise to the idea of development of case-based reasoning (CBR) (Alterman, 1989) based ES. Some of the identified features of CBR which provide flexibility in ES structure are

- No need to maintain knowledge base for each domain.
- Can be easily extended to different domains.

* Corresponding author. Tel.: +43 1 4277 39673; Mobile: +91 9415235180.
E-mail addresses: kakumar_b03@iiita.ac.in (K.A. Kumar), ysingh_b03@iiita.ac.in (Y. Singh), ssanyal@iiita.ac.in (S. Sanyal).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.09.054

This paper is organized as follows: in Section 2, we review related work in the area. In Section 3, we discuss the system design. In Section 4, we discuss some algorithmic details of the system. In Section 5, we present some of the results and then conclude.

2. Background

2.1. Research in medical rule based ES

Some of the early AI/Decision Support Systems are mentioned in the following. AAPHelp: deDombal's system for acute abdominal pain (1972): An early attempt to implement automated reasoning under uncertainty. De Dombal's system, developed at Leeds University de Dombal, Leaper, Horrocks, Staniland, and McCann (1974), was designed to support the diagnosis of acute abdominal pain and, based on analysis, the need for surgery. The system's decision making was based on the naive Bayesian approach.

INTERNIST I (1974): Pople and Myers begin work on INTERNIST (Miller et al., 1982), one of the first clinical decision support systems, designed to support diagnosis, in 1970. INTERNIST-I was a rule-based expert system designed at the University of Pittsburgh in 1974 for the diagnosis of complex problems in general internal medicine. It uses patient observations to deduce a list of compatible disease states (based on a tree-structured database that links diseases with symptoms). By the early 1980s, it was recognized that the most valuable product of the system was its medical knowledge base. This was used as a basis for successor systems including CADUCEUS and Quick Medical Reference (QMR), a commercialized diagnostic DSS for internists.

MYCIN (1976): MYCIN was a rule-based expert system (Buchanan & Shortliffe, 1984) designed to diagnose and recommend treatment for certain blood infections (antimicrobial selection for patients with bacteremia or meningitis). It was later extended to handle other infectious diseases. Clinical knowledge in MYCIN is represented as a set of IF-THEN rules with certainty factors attached to diagnoses. It was a goal directed system, using a basic backward chaining reasoning strategy (resulting in exhaustive depth-first search of the rules base for relevant rules though with additional heuristic support to control the search for a proposed solution). MYCIN was developed in the mid-1970s by Ted Shortliffe and colleagues at Stanford University. It is probably the most famous early expert system, described by Mark Musen as being "the first convincing demonstration of the power of the rule-based approach in the development of robust clinical decision support systems" [Musen, 1999].

The EMYCIN (Essential MYCIN) expert system shell, employing MYCIN's control structures was developed at Stanford in 1980. This domain independent framework was used to build diagnostic rule-based expert systems such as PUFF, a system designed to interpret pulmonary function tests for patients with lung disease.

2.2. Research in integrating CBR with RBR

The CBR/RBR hybrids took one of two approaches to integration

- The first approach is to have independent CBR and RBR modules, each of which can solve the problem independently of the other.
- The second approach is to take an essentially RBR system, and add a CBR module to provide some portion of system's overall functionality.

Our system differs from these approaches, in that it enhances an essentially CBR system (Nilsson & Sollenborn, 2004) with an RBR system.

Montani et al. (2003), Riva, Bellazzi, and Stefanelli (1997) have attempted to integrate a CBR and RBR in a decision support system for Type I Diabetes patients' care and evaluated their results using simulated patients (Montani et al., 2003). We think that they were not able take full advantage of the fact that CBR are more general than RBR, as their system was domain specific i.e. Diabetes.

There have been several more attempts made to develop CDSS (Ehrhart, Hanson, Marshall, Marshall & Medskerl, 1999) earlier. However most of them are time consuming like RBR systems (Giarratano, Joseph, & Gary Riley, 2005) and are domain dependent like MYCIN which was for providing support for diagnosis of blood diseases. The need is to construct generic CDSS system.

3. System architecture

System architecture mainly consists of (1) CBR system, (2) RBR module, (3) data entry module, (4) system tuner, (5) ICU scoring expert and (6) knowledge miner. The system architecture is shown in Fig. 1, dotted arrow from one state to other shows that user can go back to previous state if he wants to make some changes in the previous state.

3.1. CBR system

The case base reasoning system is the heart of the entire system. It has three roles to play and therefore has three components that export various important functionalities.

3.1.1. CBR agent

The CBR agent component has the functionality of driving the system at beginning of each CBR cycle. Initially taking partial information of the new patients case as input, CBR agent searches the past cases and picks the most relevant matches for the given input case information. The CBR agent uses simple domain key for searching past cases and retrieving cases, if domain key unavailable then CBR agent takes help of bayesian classifier (Zelic,

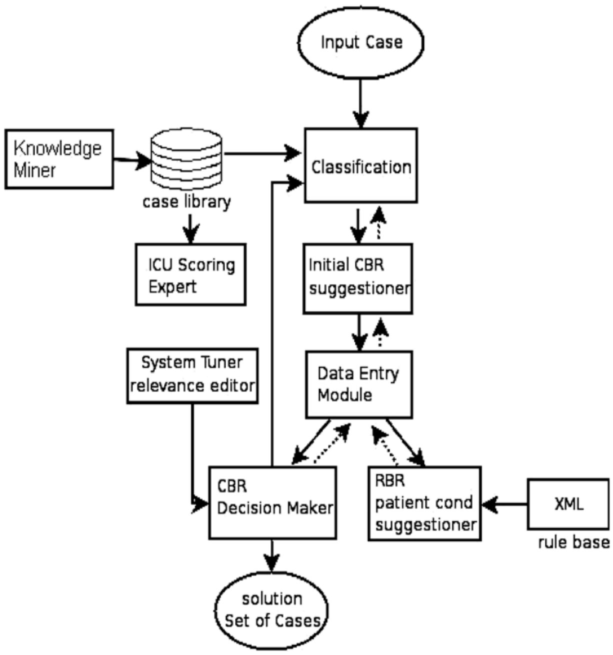

Fig. 1. System architecture.

Kononenko, Lavrac, & Vuga, 1997) to get the domain of the new case. It provides the retrieved cases to the CBR decision maker for further processing. There are some challenges in retrieving cases using partial case information of the patient. One of the challenges is dealing with uncertainty in data. It is the job of this component to handle these kinds of uncertainties in data.

3.1.2. CBR decision maker

This component takes as input the cases retrieved by the CBR agent. It analyzes these retrieved cases and decides the observations that are to be made and investigations that are to be performed with a certain level of confidence. After making decision, it suggests the observations that should be made and investigations that are to be performed to the user. This component is connected with the data entry system component for entering the values got after performing investigations suggested by this component.

3.1.3. Performance monitor

This component holds the functionality of monitoring the performance of the system. It archives certain rules or methods which were applied to the cases that were previously successfully solved with support of this system. These archived rules and methods can be adapted to solve other similar cases in future.

3.2. RBR module

The declarative knowledge collected from the domain experts' opinion is embedded into a taxonomy of production rules, fired through a forward chaining mechanism. For each rule this module performs an action. Rules are stored in XML format and care is taken that these rules are most general to all the existing domains handled by ICU. Sample rule base is shown in Fig. 2.

This module incorporates general knowledge of all observations and investigations made in ICU. Information about the local condition of the patient is shown to the user based on the values entered in the Data Entry System. It refers to the rule base to get information of the local condition of the patient. After this, it returns the control to CBR agent for further filtering of cases. This module also provides a Rule Base Editor for editing, adding or deleting rules from the rule base. This editor is password protected so that only domain expert can modify the rules. User-interface of Rule Base Editor is shown in Fig. 3.

3.3. Data entry system

This component is a user-interface in which the user can enter the data values of observations and investigations of the patient for 24 h. The data is temporarily stored in XML file. When the user is satisfied with the results, then data of that patient is entered into the database or case-library permanently. If the user feels difficulty in entering the values for 24 h, then this component can be configured such that it takes data automatically from the patient health monitoring devices. For better visualization of this module, UI is shown in Fig. 4.

3.4. System tuner

This module is used for tuning the system. Whenever the user feels that system is filtering wrong set of cases or moving in wrong direction, then he can tune the system through this module, so that the system can perform better. We have provided a *Relevance Editor* which allows the user to increase or decrease the relevancy of factors related to patient. For example: If user increases the relevancy of BP-systolic, then the system keeps this in consideration while filtering out the cases at the end of each CBR cycle. UI is shown in Fig. 5.

3.5. ICU scoring expert

This component can calculate ICU scores for given patient case. ICU scores like APACHEII (Acute Physiol-

```
<RuleBase>
<item>
<name>BP_Systolic</name>
<hint>Use diastolic and systolic pressures.</hint>
<case>
<if>BP_Systolic(less than)120</if>
<then>Normal Systolic Blood Pressure</then>
</case>
</item>
</RuleBase>
```

Fig. 2. Rule base in XML format.

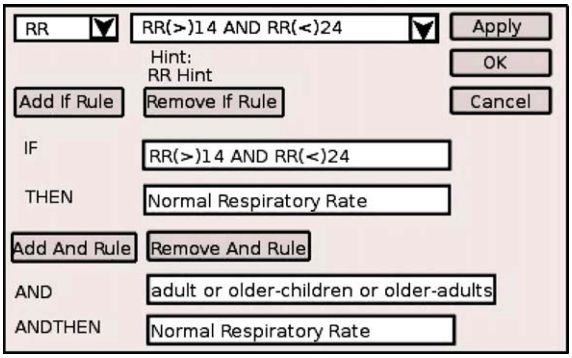

Fig. 3. Rule base editor.

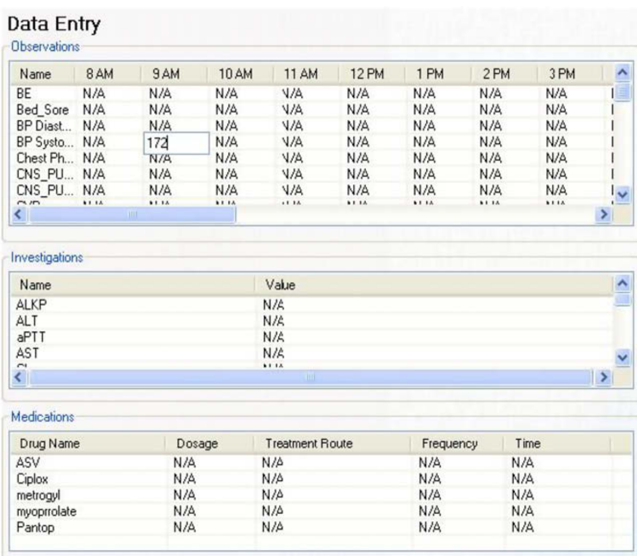

Fig. 4. Data entry system.

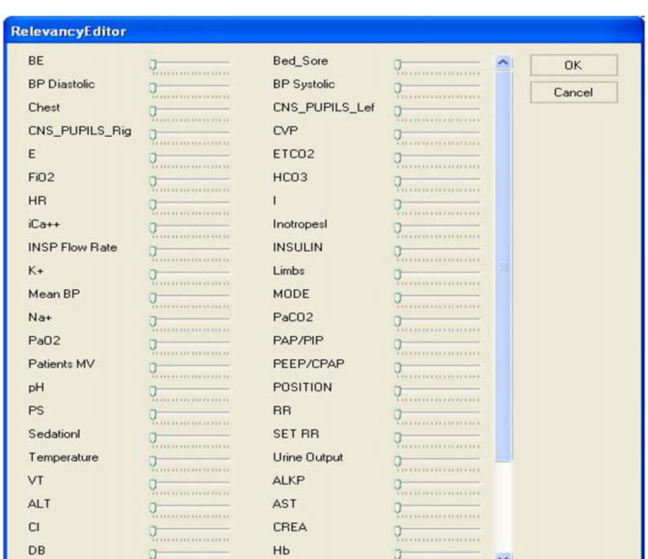

Fig. 5. Relevancy editor.

ogy and Chronic Health Evaluation II), APACHEIII, GCS (Glasgow Coma Scale), SOFA (Sequential Organ Failure Assessment), MODS and APS can be calculated using this module. It can show statistics of mortality rate of patients in hospital based on various grouping like age group, sex etc. This information is useful for the doctors to have an idea about trends in mortality rates in the hospital. The information provided by this component will be a kind of feedback to the hospital. For example, if patients whose age is greater than 35 and if mortality rate is higher than normal then this will indirectly point to the fact that the hospital was unable to provide proper medication support for this group of patients. By looking at the information provided by this component hospital medication techniques can be changed to increase quality of service.

3.6. Knowledge miner

ICU data contains many hidden information and pattern. When extracted, these can prove to be very useful for the doctors in effective treatment of the patients. We came up with the idea of *knowledge miner*, which takes the observations from case-base as input, mines them and retrieves hidden facts. For example: suppose doctor wants to know the relation between GCS and APACHEII, then this module will take GCS values of all the patients from case-base as input column and APACHEII values of all patients as output column and gives out rules which shows relationship between this two values. As far as implementation details of this module is concerned, the Iterative Dichotomizer Algorithm (ID3) is used to mine the data. Algorithmic details of the same can be found in next section.

4. Algorithmic detail of the system

4.1. CBR algorithm

At the heart of the system, case-based reasoning has been formalized for purposes of computer reasoning as a four-step process (Aamodt & Plaza, 1994).

(1) *Retrieve*: Given a target problem, retrieve cases from memory that are relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived.
(2) *Reuse*: Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation.
(3) *Revise*: Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.
(4) *Retain*: After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory.

4.2. ID3 algorithm for knowledge mining

ID3 (Iterative Dichotomiser 3) is an algorithm used to generate a decision tree (Quinlan, 1986). The algorithm is based on Occams razor: it prefers smaller decision trees (simpler theories) over larger ones. However, it does not always produce the smallest tree, and is therefore a heuristic. Occams razor is formalized using the concept of information entropy:

$$I_{\mathrm{E}}(i) = -\sum_{i=1}^{m} f(i,j)\log f(i,j) \tag{1}$$

The ID3 algorithm can be summarized as follows:

(1) Take all unused attributes and count their entropy concerning test samples.
(2) Choose attribute for which entropy is smallest.
(3) Create node containing that attribute.

4.3. Case retrieval approaches

Accurate retrieval of cases is very critical in CBR, without which the system may not give proper results. Our system uses two major retrieval approaches to retrieve cases.

4.3.1. Weighted Euclidean distance

The most common type of distance measure is based on the location of objects in Euclidean space (i.e., an ordered set of real numbers), where the distance is calculated as the square root of the sum of the squares of the arithmetical differences between the corresponding coordinates of two objects. More formally, the weighted Euclidean distance between cases can be expressed in the following manner. Let $\mathrm{CB} = \{e_1, e_2, e_3, \ldots, e_N\}$ denote a case library having N cases. Each case in this library can be identified by an index of the corresponding features. In addition, each case has an associated action. More formally, we use a collection of features $F = (j = 1, 2, \ldots, n)$ to index the cases and a variable V to denote the action. The ith case e_i in the library can be represented as an $(n+1)$-dimensional vector, that is, $e_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in}, \theta_i)$, where x_{ij} corresponds to the value of feature $F_j (1 \leqslant j \leqslant n)$ and θ_i corresponds to the value of action $V(i = 1, 2, \ldots, N)$.

Suppose that for each feature $F_j(1 \leqslant j \leqslant n)$, a weight $w_j(w_j \epsilon [0,1])$ has been assigned to the jth feature to indicate the importance of that feature. Then, for any pair of cases e_p and e_q in the library, a weighted distance metric (Pal, Shiu 2004) can be defined as:

$$d_{pq}^{(w)} = d^{(w)}(e_p, e_q) = \left[\sum_{j=1}^{n} w_j^2 (x_{pj} - x_{qj})^2\right]^{\frac{1}{2}} = \left(\sum_{j=1}^{n} w_j^2 \chi_j^2\right)^{\frac{1}{2}} \tag{2}$$

where $\chi_j^2 = (x_{p_j}^2 - x_{q_j}^2)$. Using the weighted distance, a similarity measure between two cases, SM(w) pq, can be defined as:

$$\mathrm{SM}_{pq}^{(w)} = \frac{1}{1 + \alpha d_{pq}^{(w)}} \tag{3}$$

where α is positive constant. The higher the value of $d_{pq}^{(w)}$, the lower the similarity between e_p and e_q. When all of the weights take a value of 1, the similarity measure is denoted by $\mathrm{SM}_{pq}^{(1)}, \mathrm{SM}_{pq}^{(1)} \epsilon [0,1]$.

4.3.2. Mahalanobis distance

Considering Mahalanobis distance for finding distance between two cases is a good idea because it is a useful way of determining similarity of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant, i.e. not dependent on the scale of measurements.

Formally, the Mahalanobis distance from a group of values with mean $\mu = (\mu_1, \mu_2, \mu_3, \ldots, \mu_p)$ and covariance matrix $\sum$ for a multivariate vector $x = (x_1, x_2, x_3, \ldots, x_p)$ is defined as:

$$D_{\mathrm{M}}(x) = \sqrt{(x-\mu)^T \sum^{-1} (x-\mu)} \tag{4}$$

Mahalanobis distance can also be defined as dissimilarity measure between two random vectors $\vec{x}$ and $\vec{y}$ of the same distribution with covariance matrix $\sum$

$$\mathrm{d}(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \sum^{-1} (\vec{x} - \vec{y})} \tag{5}$$

4.4. Decision making

At the end of each CBR cycle system decides the cases that need to be eliminated based on a particular score. In this system, as RBR supports the decision made by CBR system, we formulate this as:

$$\widetilde{E}_{\mathrm{score}} = \lambda \cdot E_{\mathrm{CBR}} + (1-\lambda) \cdot E_{\mathrm{RBR}} \tag{6}$$

where λ is a weighing parameter and is empirically set at an optimum value in this study. We have selected value of λ where precision was found to be high and value of λ we got is 0.25 as shown in Fig. 7.

E_{CBR} is calculated as

$$E_{\mathrm{CBR}} = E_{\mathrm{dist}}(R_{\mathrm{score}}) \tag{7}$$

R_{score} is the vector of relevancy scores obtained from the relevancy editor as an input from the doctor. E_{dist} is the distance between the new case and the old case calculated by weighted Euclidean distance or Mahalanobis distance. Steps to calculate $E_{\mathrm{dist}}(R_{\mathrm{score}})$ can be given as:

(1) If O_i where $i = (1, 2, \ldots, n)$ denote feature vectors of an old case and if N_j where $j = (1, 2, \ldots, n)$ denote feature vectors of the new case, then Euclidean distance between each feature vector is denoted by $\mathrm{Euc}_i(O_i, N_j)$.
(2) Multiplying individual elements of Euc_i and $R_{\mathrm{score}_i} + eps$ *where* $i = (1, 2, \ldots, n)$,$(0 < \mathrm{eps} < 1)$, we get a new vector and let this be D_i.
(3) Taking the square root of sum of squares of D_i to get $E_{\mathrm{dist}}(R_{\mathrm{score}})$.

As we have $E_{\mathrm{dist}}(R_{\mathrm{score}})$ we can calculate E_{CBR} using Eq. (7).

E_{RBR} is mainly calculated as score of satisfaction of the user. When RBR module gives suggestions based on the condition of the patient by referring to the rule-base, score is calculated mainly based on the number of suggestions accepted by the user and user satisfaction level, in our system the E_{RBR} ranged between 1–10.

$\widetilde{E}_{\mathrm{score}}$ can be calculated from Eq. 6. For filtering of cases we decide upon a optimal threshold, if $\widetilde{E}_{\mathrm{score}}$ is less than the threshold then we eliminate the case else the case is retained for next CBR cycle.

5. Results

We tested the system with real ICU data provided by Intensive Care Unit of Sir Sunderlal Hospital that is attached to the Institute of Medical Sciences of Banaras Hindu University. Our case-base had patient data which consisted of several domains like poisoning, accident, cancer, viral diseases etc. We tested with six ICU domains and tested the performance of our system. Here we will show some interesting result during evaluation of system for poisoning domain. We took a related old case of a patient and gave this as input to the system. Before that, we populated the database with five tagged cases which were very close to the case given as input to the system. These tagged cases were selected manually by doctors. We programmed and visualized all cases in the case library as points scattered in the space with the input case at the center of the space. At the end of each CBR cycle the points which are very far from the center point are eliminated and the points which have high confidence score come closer to the center as shown in the Fig. 6. In Fig. 4. of Fig. 6 the tagged cases in case-base came very close to the central input case and with CBR–RBR integration this was achieved in 4 CBR cycles whereas similar result was achieved in 8,9 cycles using CBR only. We repeated this experiments for all the 6 domains and obtained similar results.

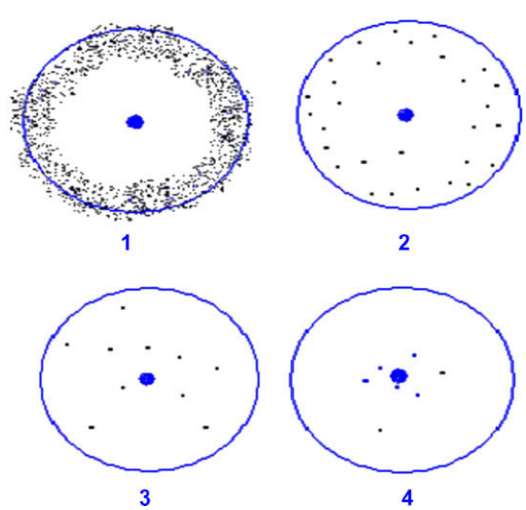

Fig. 6. Each figure showing the result at the end of each CBR cycle.

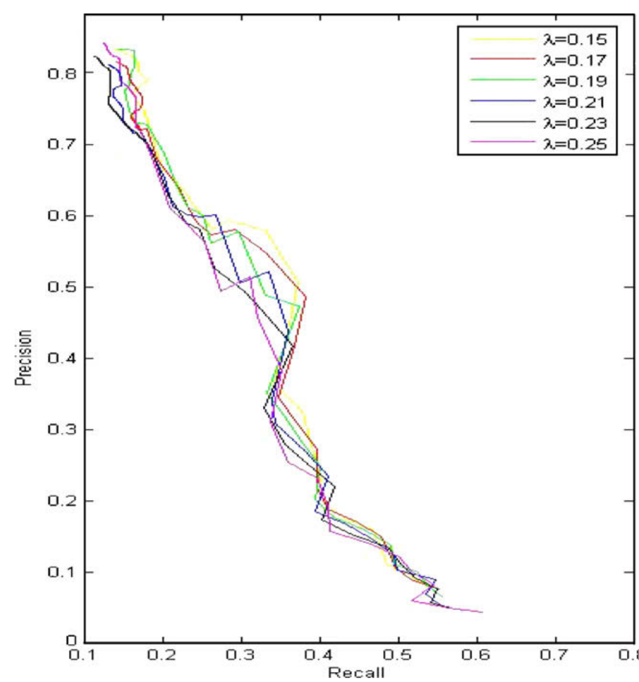

Fig. 7. Relation between precision and recall for 6 different ICU domains where max prec = 0.858, $\lambda = 0.25$ and max recall = 0.628.

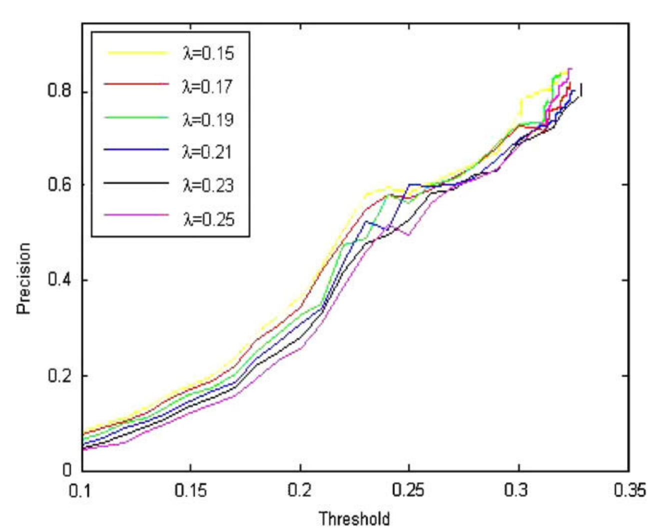

Fig. 8. Relation between precision and $\widetilde{E}_{\mathrm{score}}$ for 6 different ICU domains where max prec=0.858, max $\widetilde{E}_{\mathrm{score}} = 0.335$ and max $\lambda = 0.25$.

For effectiveness, we examined the relevance of the retrieved cases given a test case. The retrieval effectiveness can be defined in terms of precision and recall rates. A precision rate can be defined as the percent of retrieved cases similar to the new case among the total number of retrieved cases. A recall rate is defined as the percent of retrieved cases, which are similar to the new case, among the total number of cases similar to the new case which are tagged in the case-base. The recall and precision rates are computed using:

$$\text{precision} = \frac{N_{\text{correct}}}{N_{\text{correct}} + N_{\text{false}}} \tag{8}$$

$$\text{recall} = \frac{N_{\text{correct}}}{N_{\text{total}}}, N_{\text{total}} = N_{\text{correct}} + N_{\text{missed}} \tag{9}$$

where N_{total} denotes total number of tagged cases similar to the new case and N_{correct} denotes the number of retrieved cases similar to the new case. N_{false} is the number of retrieved cases dissimilar to the new case and N_{missed} is the number of tagged cases that are similar to the new case but not retrieved. In our experiments, we tested for 6 different domains of ICU and similar results were obtained for all the domains with max precision of 0.85, recall of 0.62, optimum threshold of 0.335 and λ of 0.25, as shown in the Figs. 7 and 8.

6. Conclusion

We believe that the systems having flexible architectures that can support large domains are very much needed and are more useful than the systems which are domain specific. So in this system we have tried to induce that flexibility by giving more importance to CBR technique and making sure that rule-base consists of rules which are common for all domains of ICU. We also believe that our approach would be useful in areas other than Medical.

Acknowledgements

This work represents research done in Indian Institute of Information Technology, Allahabad. We wish to thank Dr. P. Bhattacharya, Director of Intensive Care Unit BHU Hospital, for giving us opportunity for developing a system for their hospital and helping us with ICU data required for the system. The contribution and support of R. Amirdha Gopal, Dr. Raghuraj, and Junior doctors at Sir Sundarlal BHU Hospital are gratefully acknowledged.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications, 7*(i), 39–59.

Alterman, R. (1989). Panel discussion on case representation. In *Proceedings of the second workshop on case-based reasoning*, Pensacola beach. FL, US.

Bobrow, D. G., Mittal, S., & Stefik, M. J. (1986). Expert systems: Perils and promise. *Communications of the ACM, 29*(9), 880–894.

Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison-Wasley.

de Dombal, F. T., Leaper, D. J., Horrocks, J. C., Staniland, J. R., & McCann, A. P. (1974). Human and computer-aided diagnosis of abdominal pain: Further report with emphasis on performance of clinicians. *BMJ, 1*, 376–380.

Ehrhart, PhD, L. S., Hanson, MD, C. W., Marshall, MD, B. E., Marshall, PhD, C., Medskerl C. (1999). *Collaborative prototyping approaches for ICU decision aid design*.

Giarratano, Joseph, C., Gary Riley. (2005). *Expert systems, principles and programming*, ISBN 0-534-38447-1.

Ignizio, James (1991). *Introduction to expert system*, ISBN 0-07-909785-5.

Jackson, Peter (1998). *Introduction to expert systems*, ISBN 0-20-187686-8.

Miller, Randolph A. et al. (1982). INTERNIST-1: An experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine, 307*, 468–476.

Montani Magni, P., Bellazzi, R., Larizza, C., Roudsari, A. V., & Carson, E. R. (2003). Integrating model-based decision support in a multimodal reasoning system for managing type 1 diabetic patients. *Artificial Intelligence in Medicine, 29*, 131–151.

Markus Nilsson, Mikael Sollenborn. (2004). *Advancements and trends in medical case-based reasoning: An overview of systems and system development*.

Pal, S. K., Shiu S. C. K. (2004). *Foundations of soft case-base reasoning*. ISSN 1573-7497.

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

Riva, A., Bellazzi, R., & Stefanelli, M. (1997). A web-based system for the intelligent management of diabetic patients. *MD Computing, 14*, 360–364.

Zelic, I., Kononenko, I., Lavrac, N., Vuga, V. 1997. Induction of decision trees and Bayesian classification applied to diagnosis of sport injuries. In *Proceedings of IDAMAP 97 workshop (pp. 61–67)*. IJCAI 97, Nagoya.