

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Intelligent Tutoring System: Hierarchical Rule as a Knowledge Representation and Adaptive Pedagogical Model

Nabil M. Hewahi

Department of Computer Science, Islamic University of Gaza, Palestine

Abstract: In this study, we present a new rule structure called Hierarchical Rule (HR) to be an effective knowledge representation in Intelligent Tutoring System (ITS) and in Intelligent Educational Systems (IES). The structure of the rule shall expedite the process of inference and allow the system to work in forward as well as backward chaining. The HR structure will help in putting the knowledge in a very systematic way, which will lead to well structured system. This representation can be used very effectively in the pedagogical model to inform the system, which method should be followed up with the current user. The pedagogical model shall benefit from the user 's model (history of the user) to choose the proper explanation method. We also present an algorithm in the novel form to represent the HRs using the neural networks to enhance the performance of the rule system. We call this algorithm HRANN. In addition to that, a general method concerning the adaptation of pedagogical model is introduced. This method is mainly depend on competitive learning (unsupervised learning) if enough number of examples is not given. The system 's performance will keep improving as long as the system is working for various users. The system shall benefit from its experience. In case enough number of examples are provided, the traditional method of backpropagation algorithm can be used.

Key words: Intelligent tutoring system, neural networks, knowledge representation

INTRODUCTION

Intelligent Tutoring Systems (ITS) are composed of three main models, The domain knowledge, user model and pedagogical model. The domain knowledge is concerned with the knowledge related to the problem to be considered which would include the teaching contents and the meta information about the subject to be taught. The user model is concerned with information related with the user abilities and subject understanding, it contains the history of the user (Hatzilygeroudis and Prentzas, 2005; Hatzilygeroudis and Prentzas, 2004; Simic and Deveddzic, 1996). The pedagogical model is the model where the system should decide the appropriate explanation style to be used with a certain user based on his history (user model). Some web based ITSs emerged (Brusilovski, 1999). There are two main factors which play a significant role in ITSs, these are knowledge representation and adaptation. Knowledge representation should have some features to be useful in ITS, those features might include (a) Simple representation, (b) Easy to reach to information, (c) Can work with appropriate inference mechanism, (d) Easy to modify and update, (e) Can be adapted, (f) Supports the explanation. Many knowledge representation techniques were tried with ITS, some of those are, Symbolic rules, Case-based representation (Gilbert, 2000), neural networks, belief

networks (probabilistic nets) (Vanlehn and Zhendog, 2001), fuzzy rules, connectionist rule-based representation, neurofuzzy representation, neurules (Hatzilygeroudis and Prentzas, 2005), neuronet (Tan *et al.*, 1996), VPNL (Bharadwaj and Silva, 1998), KBANN (Towell and Shavlik, 1991; Towell and Shavlik, 1994), GRSNL (Hewahi, 2004) and granularity hierarchy (knowledge structure) with Bayesian network (Pek and Poh, 2005).

Adaptation in ITSs is very important to allow the system to deal with several users in various techniques (this may include also the user interface itself). The adaptation is affected with the user (student) model.

There are two main objectives of this research:

- Developing a new knowledge representation for ITS.
- Identifying a neural network structure to help in adaptation necessary for pedagogical model.

Many knowledge representations were tried for ITSs. Most for those differ from one model to another. For example, in a pedagogical model a certain method of representation might be used, where another representation method is used for the domain knowledge. According to Hatzilygeroudis and Prentzas (2005), it is difficult to have a unique knowledge representation for

ITSs and the better way is to have a hybrid approach. We try to have a knowledge representation structure that can be plausible with all the ITS 's models and might be common for various ITSs. The underlying criteria to have such kind of knowledge representation, is to encompass most of the properties that should be available with a knowledge representation (those mentioned in section 1). The representation should also be easy to represent through neural networks, so that adaptation would be tangible.

The second goal is to help the pedagogical model to choose the appropriate teaching style (this may include the teaching content sequence) by using neural network. This for sure is based on the history of each of the users that can be obtained from the user model. Because, our proposed approach should be general as much as possible, we assume that we can't provide examples (input/output) to the system. This thinking yields us to go to unsupervised learning to decide about the level of the student. The main point is that, the more the system deals with students/users, the more experience it gains, therefore, changing its decisions.

THE PROPOSED KNOWLEDGE REPRESENTATION: HIERARCHICAL RULE (HR)

The standard rule structure is very well known in the area of expert systems and ITS. The structure of standard rule is (<IF condition THEN action>). The standard rule structure is very simple and can work well in many systems. One of the drawbacks of such a structure is that it is totally dependent of the used inference mechanism and its structure does not help in allowing the system to work either in forward chaining or backward chaining. Moreover, the standard rule structure does not help to guide the system to which rule to be tried if the current rule fails. For sure, solutions to such problems need extra complexity for the structure. Bharadwaj and Jain (1992) proposed a rule called Hierarchical Censored Production Rule (HCPR) as an extension of Censored Production Rule (CPR) proposed by Michalski and Winston (1986). The CPR is based on Variable Precision Logic (VPL) in which certainty varies, while specificity stays constant. A HCPR is a CPR augmented with specificity and generality information, which can be made to exhibit variable precision in the reasoning such that both certainty of belief in conclusion and its specificity may be controlled by reasoning process. Such a system has numerous applications in situation, where decision must be taken in real time and with uncertain information. Compton and

Richards (1998) developed a rule structure, which they call ripple down rules. Ripple down rules are considered to be highly succinct and comprehensible to human experts. In ripple down rule, if a rule fails the system can know its exception rule and if is fired the system can know the next rule to be fired. Hewahi (2002) utilized the advantages of both HCPR and Ripple down rules and proposed a rule structure called GRS. GRS has been developed to be used with various kinds of applications and to enable the application systems to use multiple reasoning processes. Following the GRS structure with neglect to variable precision portion, we propose Hierarchical Rule (HR) structure that can help in knowledge representation in many systems and specially in ITS. The proposed rule structure of HR is of the form

```
(Rule name : <rule name>           {each rule has a unique name}
(IF < condition>                    {preconditions (AND) conditions}
THEN <action/process/direction>    {concept/decision/action/head}
GENERALITY [G]                     {general Information/parent rule}
SPECIFICITY [S]                    {specific information/more specific rml}
EXCEPTION [E]                      {the rml to be tried in case of failure}
)
```

The parts of IF and THEN are the same as in the standard rule structure. The GENERALITY part contains the name/index/number of the parent rule. SPECIFICITY part contains the name/index/number of the next rule to be tried if the current rule is matched and fired. EXCEPTION part contains the name/index/umber of the rule to be tried if the current rule is failed (this could be due to some lack of information or none condition satisfaction due to some inference). As an example

```
( Rule (node) name : RAIN
  IF < It is cold AND It is cloudy>
  THEN < It will rain>
  GENERALIT < WINTER>
  SPECIFICITY <FLOOD>
  EXCEPTION <COLD> )
```

The sequence of HRs for a complete structure called Rule graph. To illustrate this, we continue explaining the previous example

```
( Rule name: WINTER
  IF < time between Nov.-March)
  THEN < It is winter>
  GENERALITY < >
  SPECIFICITY <RAIN>
  EXCEPTION <SUMMER/FALL/SPRING> )
```

```
( Rule name: FLOOD
  IF < it is windy >
  THEN <flood might occur>
  GENERALITY < RAIN>
  SPECIFICITY <>
  EXCEPTION <> )
```

The relation between the above rules is the rule graph as shown in Fig. 1.

The HR structure is very flexible structure, we must notice that we check the rule RAIN after being sure it is winter by checking first the rule WINTER. If the rule WINTER fails, then the system is redirected to check SUMMER/AUTUMN/SPRING rule. And if we are sure that it is raining, we can test for whether a flood is going to happen or not. This means, to take action about the flood, the system has to check the following conditions: It is winter, It cold and cloudy and it is also windy. If the system following the forward chaining of inference, it can easily know the next rule by using SPECIFICITY part and if the system is following the backward chaining of inference, the system can reach to parent rule through the GENERALITY part.

This structure would be very useful in ITSs in all of its models. It can be as the standard rule structure if empty values are assigned to GENERALITY, SPECIFICITY and EXCEPTION parts. Moreover, in pedagogical model, instead of using direct actions in the action part, it can be a calling of a certain process of teaching strategy. For example, consider the following set of HRs:

```
(
  Rule name : LEVEL-1
  IF < level-1 >
  THEN < call level-1 procedure >
  GENERALITY <>
  SPECIFICITY <>
  EXCEPTION < LEVEL-2>
)
( Rule name: LEVEL-2
  IF < level-2>
  THEN < call level-2 procedure>
  GENERALITY <>
  SPECIFICITY <>
  EXCEPTION < LEVEL-3> )
```

In the previous HRs, we notice that the condition 's inquiry is about the true level, which is obtained in the pedagogical model based on the history of the user registered in the user model. In the next section we shall explain the proposed strategy to identify the level of the user. Based on the level, the pedagogical model should

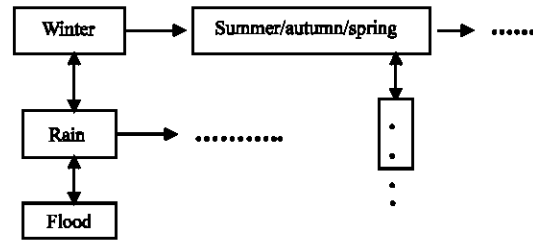


Fig. 1: A group of related HRs form rule graph

use/call the appropriate teaching method. Within each of the chosen level procedures, there are a sequence of transactions that should be followed in order to achieve good results (understanding of the user). To clarify this point, let us consider the following rules:

Level-1 rules

```
( rule name: LESSON-1
  IF < LESSON-1 = OK >
  THEN < LESSON-1 is understood>
  GENERALITY <>
  SPECIFICITY <LESSON-2>
  EXCEPTION <LESSON-1, LOW-LEVEL-CHANGE> )
( rule name: LESSON-2
  IF < LESSON-2 = OK >
  THEN < LESSON-1 is understood>
  GENERALITY <LESSON-1>
  SPEIFICITY < LESSON-3>
  EXCEPTION <LESSON-2, LOW-LEVEL-CHANGE, LESSON-1> )
```

From the previous rules, it is to be noticed that the EXCEPTION (s) of LESSON-1 rule are LESSON-1 again (which means, the lesson has to be explained to the user again with the same method), or the Chosen level of teaching explanation has to be changed to a lower level. The choosing of one of the two choices of the EXCEPTION part could be either based on the user 's request or based on the system s' evaluation for the user's ability. The condition part matching might also be based on the response of the user, or based on the system evaluation of the user through his interaction and understanding of the lesson.

TRAINING THE HR STRUCTURE

One of the main advantages of the proposed structure is that it can be represented using the neural network and train it with the given examples, therefore, ability to answer not known questions. We follow the

same procedure adopted by KBANN algorithm (Towell and Shavlik, 1991, 1994), which proved very high capabilities. KBANN works by translating domain theory consisting of a set of propositional rules directly into a neural network. KBANN approach brings the advantages of connectionism, that is learning, generalization, robustness, massive parallelism, etc., to the elegant and beautiful methods for symbolic processing, logical inferences and goal driven reasoning.

We propose a similar algorithm that can deal with HR structure. We shall call this algorithm HRANN. The steps of the algorithm are:

- Consider all the system inputs (used in the domain or not used) to be the input layer of the neural network.
- Classify the inputs into two types, the first type is those which are connected directly to the top level rules (their necessary inputs to fire). The second type are the inputs that are connected to the hidden layer. Hidden layer represent the specific rules and their exceptions.
- The layer which contains the top level rules contains also their exceptions.
- Connect with solid line the inputs with their rules either in the layer of the top rules or in the hidden layers, each node/rule based on its requirements.
- Connect the rules with the general rules with solid lines.
- Connect other non-used inputs with all the layers (top rules or hidden layer) with a very thin line.
- Every negated connection is linked to its corresponding rule (regardless of the layer) with dashed line.
- Each solid line is given a weight W .
- Each non-negated weight is given weight $-W$.
- Each thin line is given very low weight.
- The threshold weight is $-(n-0.5)W$, where n is number of non-negated antecedents/inputs.
- If the difference between the sum and the threshold is ≥ 0.5 , it means the output is true, otherwise it is false.
- Apply backpropagation algorithm.

To clarify the algorithm, let us consider the following set of rule

(Rule name : Rule-1
 IF <a> THEN <M> GENERALITY <> SPECIFICITY < Rule-3 > EXCEPTION <Rule-2>)

(Rule name : Rule-2
 IF <n and e> THEN <D > GENERALITY <Rule-1> SPECIFICITY < Rule-4 > EXCEPTION <>)

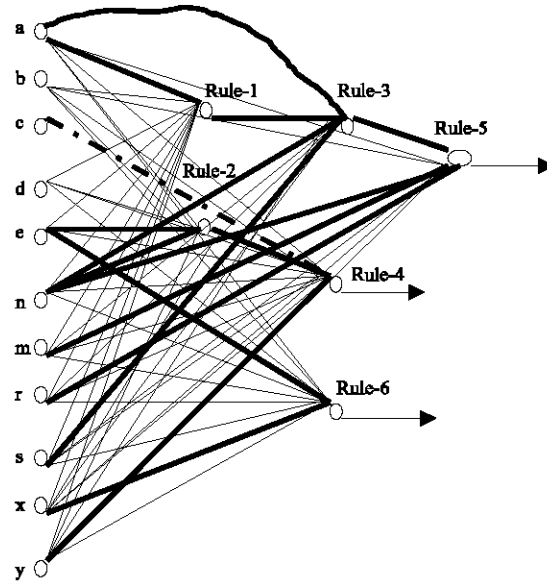


Fig. 2: The neural net that can be trained of the given set of rules Rectangular shape represents exceptional rule

(Rule name : Rule-3
 IF <n and a and s> THEN <M> GENERALITY <Rule-1> SPECIFICITY < Rule-5 > EXCEPTION <Rule-6>)

(Rule name : Rule-4
 IF <y and ~c> THEN <L > GENERALITY <Rule-2 > SPECIFICITY <> EXCEPTION <>)

(Rule name : Rule-5
 IF <m and r and n> THEN <F> GENERALITY <Rule-3 > SPECIFICITY <> EXCEPTION <>)

(Rule name : Rule-6
 IF <x and e > THEN <M> GENERALITY <Rule-1 > SPECIFICITY <> EXCEPTION <>)

Figure 2 shows the training structure. It is to be noticed that input a is the only input alone for the inputs of the top level rules, whereas inputs c, m, r, s, y are for the hidden and output layers. Inputs e, n are for both the top level rules and specific and exceptional rules of the hidden layers. As the KBANN algorithm proved to be very effective, it is expected that this algorithm will be of the same effectiveness.

TRAINING OF PEDAGOGICAL MODEL

Pedagogical model gets its inputs from the user model to be able to take a proper decision regarding the methods

and the approaches that should be followed with the user based on his abilities. One of the effective methods that can be used is the neural networking. With this concept, the two approaches, supervised learning and unsupervised learning can be used. In this section, we propose the following steps:

- Assign the neural network inputs which can be obtained from the user model such as, the user level in answering the questions, degree of examples understanding, user IQ, general observations and many other inputs related to the user's performance history.
- The output of the neural net is the set of user levels (say level 1, level 2 and so on). The neural net has to decide at which level is the student. Based on the level decision, the pedagogical model is going to specify the teaching strategy.
- If we have enough number of examples (inputs and outputs), we use the backpropagation algorithm for training the net.
- In case, we do not know how to decide the levels (output is not known) and there is no enough examples, we let the system do continuous learning whenever it is working with any user. This means, whenever the system works more, it learns more and might change its experiences. We do the following as below:
 - (a) Register the inputs of user x and using the competitive learning (unsupervised learning), find out the output.
 - (b) Keep the previous inputs in a database.
 - (c) If we fell that we still need training, go to a.
- If the number of cases is enough, stop keeping the inputs in the data bases. We only get the cluster (level) of the student, by applying the new inputs to the final network weights.

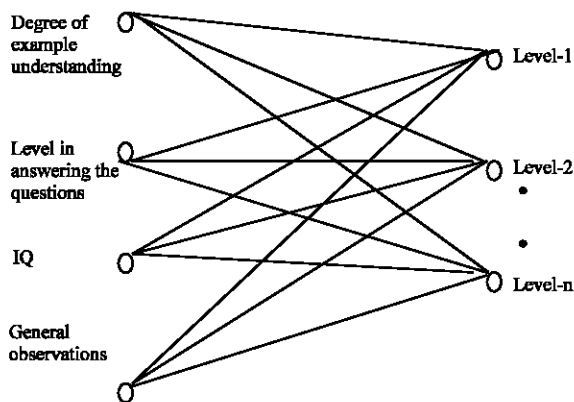


Fig. 3: The general network structure that can be used by pedagogical model using the competitive learning

A general form of the network structure that can be used in pedagogical model is shown in Fig. 3.

CONCLUSIONS

In this study, we presented a new knowledge representation structure called Hierarchical Rule (HR) which can be very well used in various rule based systems. This structure can be of good benefit to ITSs because it has the criteria to allow the system to go step by step in explanation or examination process based on the student's level. This structure helps the system to try other options in case the current option fails (could be in exams or explanation). A collection of HRs is called Rule graph. The main advantage of the structure is that, it can be represented using neural network, hence, the system rule based performance can be improved by answering questions not represented in the given set of rules. An algorithm called HRANN is proposed for this purpose. Another simple algorithm is proposed to improve the performance of the pedagogical model using competitive learning of unsupervised learning. This algorithm is very useful because whenever the system deals with more students, it learns more about their levels and what the appropriate action to be taken for them. This system is changing its behavior based on its accumulated experience. The future directions would be; to implementing and using the HR structure with real ITSs and other systems; to designing a tool to convert from standard rule structure to HR structure. This might be important because dealing with HR structure needs a person who is expertise in the domain.

REFERENCES

Bharadwaj, K. and K. Jain, 1992. Hierarchical Censored Production Rules (HCPR) system. *Data and Knowledge Eng.*, 8: 19-34.

Bharadwaj, K. and J. Silva, 1998. Towards Integrating Hierarchical Censored Production Rule (HCPR) Based System and Neural Networks. In: *Lecture Notes in Artificial Intelligence*. Oliveira, F. (Ed.), 1515: 121-130.

Brusilovski, P., 1999. Adaptive and Intelligent Technologies for Web-based Education. In: *Kustliche Intelligenz*. Rollinger, C. and C. Peylo (Eds.), Special Issue on Intelligent Systems and Teleteaching, 4: 19-25.

Compton, P. and D. Richards, 1998. Taking up the situated cognition challenge with ripple down rule. *Int. J. Hum. Comp. Studies*, Special Issue on Situated Cognition, 49: 895-926.

- Gilbert, J., 2000. Case-based reasoning applied to instruction method selection for intelligent tutoring systems. Proceeding of the ITS'2000 Workshop on Case-based Reasoning in Intelligent Training Systems, pp: 11-15.
- Hatzilygeroudis, I. and J. Prentzas, 2004. Using a hybrid rule-based approach in developing an intelligent tutoring systems with knowledge acquisition and update capabilities. *J. Exp. Syst. Applic.*, 26: 447-492.
- Hatzilygeroudis, I. and J. Prentzas, 2005. Knowledge Representation Intelligent Educational Systems. In: *Web-Based Intelligent E-learning Systems: Technologies and Applications*. Ma, Z. (Ed.), Idea Group Inc.
- Hewahi, N., 2002. A general rule structure. *J. Inform. Software Technol.*, 44: 451-457.
- Hewahi, N., 2004. Principles on integrating General Rule Structure (GRS) based systems and neural networks. Proceeding of the International Conference on Artificial Intelligence, IC-AI '04, pp: 174-180.
- Michalski, R. and P. Winston, 1986. Variable precision logic. *Artificial Intelligence*, 29: 121-145.
- Pek, P. and K. Poh, 2005. Making decisions in an intelligent tutoring system. *Int. J. Inform. Technol. Decision Making*, 4: 207-233.
- Simic, G. and V. Deveddzic, 2003. Building an intelligent systems using modern internet technologies. *Expert Syst. Applic.*, 25: 231-246.
- Tan, C., T. Quah and H. The, 1996. An artificial neural network that models human decision making. *Computer*, pp: 64-70.
- Towell, G. and J. Shavlik, 1991. Interpretation of Artificial Neural Networks: Mapping Knowledge Base Neural Networks into Rules. *Advances in Neural Information Processing Systems*. Vol. 4, Denver, Co: Morgan Kaufmann, pp: 977-984.
- Towel, G. and S. Shavlik, 1994. Knowledge base artificial neural networks. *Artificial Intelligence*, 70: 119-165.
- Vanlehn, K. and N. Zhenog, 2001. Bayesian student modeling, user interfaces and feedback: A sensitivity analysis. *Int. J. AI Educ.*, 12: 155-184.