Discovery of Quantified Hierarchical Production Rules from large set of Discovered Rules

Tamanna Siddiqui, and M. Afshar Alam

Abstract—Automated discovery of Rule is, due to its applicability, one of the most fundamental and important method in KDD. It has been an active research area in the recent past. Hierarchical representation allows us to easily manage the complexity of knowledge, to view the knowledge at different levels of details, and to focus our attention on the interesting aspects only. One of such efficient and easy to understand systems is Hierarchical Production rule (HPRs) system. A HPR, a standard production rule augmented with generality and specificity information, is of the following form:

Decision If <condition> Generality <general information> Specificity <specific information>.

HPRs systems are capable of handling taxonomical structures inherent in the knowledge about the real world. This paper focuses on the issue of mining Quantified rules with crisp hierarchical structure using Genetic Programming (GP) approach to knowledge discovery. The post-processing scheme presented in this work uses Quantified production rules as initial individuals of GP and discovers hierarchical structure. In proposed approach rules are quantified by using Dempster Shafer theory. Suitable genetic operators are proposed for the suggested encoding. Based on the Subsumption Matrix (SM), an appropriate fitness function is suggested. Finally, Quantified Hierarchical Production Rules (HPRs) are generated from the discovered hierarchy, using Dempster Shafer theory. Experimental results are presented to demonstrate the performance of the proposed algorithm.

Keywords—Knowledge Discovery in Database, Quantification, Dempster Shafer theory, Genetic Programming, Hierarchy, Subsumption Matrix.

I. INTRODUCTION

In KDD, the most predominant representation of the discovered knowledge is the standard production rules in the form If P Then D [12]. As the number of rules becomes significant, they are not comprehensible to people as meaningful knowledge, from which they can gain insight into the basis of decision-making. Much world knowledge is best expressed in the form of hierarchies. Hierarchies give comprehensible knowledge structure that allows us to manage the complexity of knowledge, to view the knowledge at different levels of details, and to focus our attention on the interesting aspects. Several efforts have been made in the recent past towards automated discovery of hierarchical structure in large databases [3]-[9]. Uncertainty pertains to information that is not definitely fixed, not precisely determined. Uncertainty must be quantified in order to use it systematically in decision-making processes. Uncertainty Quantification is the quantitative characterization and use of uncertainty in information applications In Genetic Programming (GP) [10][11] the basic idea is the evolution of a population of “program”, i.e., candidate solutions to the target problem. A program (an individual of the population) is usually represented as a tree, where the internal nodes are the functions (operators) and the leaf nodes are terminal symbols. Both the function set and terminal set must include symbols suitable for the target problem. Each individual of the population is assessed regarding its ability to solve the target problem. This evaluation is conducted by a fitness function, which is problem-dependent. Individuals undergo the action of genetic operators such as reproduction, crossover and mutation. Once genetic operators have been applied to the population based on given probabilities, a new generation of individuals is created. These newly created individuals are evaluated by the fitness function. The whole process is repeated iteratively for a fixed number of generations or until other termination criterion is met. The result of GP (the best solution found) is usually the fittest individual created along all the generations [11]. In the present work, a post-processing scheme based on GP and Dempster Shafer Theory is presented that takes quantified if then rules as input and discovers crisp hierarchical structure. Further, Quantified Hierarchical Production Rules (HPRs)[2], are generated using the discovered hierarchy. A concept of Subsumption Matrix (SM) is used to summarize the relationship between the classes. An appropriate encoding, suitable genetic operators and effective fitness function are suggested for the proposed scheme.

Tamanna Siddiqui is with the Department of Computer Science, Jamia Hamdard (Hamdard University), New Delhi 110 062, India (e-mail: ja_zu_siddiqui@hotmail.com).

M. Afshar Alam is with the Department of Computer Science, Jamia Hamdard (Hamdard University), New Delhi 110 062 India (e-mail: afshar@rediffmail.com).
II. BACKGROUND

The concept of CPR as suggested by Michalski and Winston has the following form:
If P \{premises/preconditions\}
Then D \{actions/decision\}
Unless C \{censor conditions\}

A censor is a low likelihood condition when hold will block the rule. So when the system is having low resources, it can skip checking the censor conditions. If the resources are available, the censor conditions are examined, increasing the certainty factor of making a high speed decision or reversing the decision itself. The above concept of CPR has been extended to HCPR [1] to incorporate both aspects of precision namely certainty and specificity. Two new operator added to CPR and we have the concept of HCPR having the general form as follows:

D \{decision/concept/action\}
If \( P \{p_1,p_2,p_3,\ldots,p_n\} \) \{preconditions\}
Unless C \{censor conditions\}
Generality \([G\%]\) \{general information\}
Specificity \([s_1,s_2,\ldots,s_k]\) \{mutually exclusive set of specific information\}

As a special case, dropping the unless operator due to time constraint, HPR takes the form

D\{decision/concept/action\}
If \( P\{p_1,p_2,\ldots,p_n\} \) \{preconditions\}
Generality \([G\%]\) \{general information\}
Specificity \([s_1,s_2,\ldots,s_k]\) \{mutually exclusive set of specific information\}

The root represents the most general concept in a HPR tree and any child in tree is more specific case of its parent. As the concept becomes more specific, the number of elements in its precondition part will increase obviously. However it is not required to list all such elements because total inheritance is an inherent feature of the HPRs tree structure; each HPR inherits the entire preconditions set of its parent HPR, and thus all of its ancestors. So the redundancy is minimized in the listing of preconditions in the child node. HPR system collects fragmented knowledge and represents these as collective one and hence significantly reducing the knowledge base. This representation scheme reduces the complexity of the discovered knowledge substantially, makes knowledge base easy to understand and efficient for future processing.

As an example, consider the following HPRs [1]:

level 0
change_car_status If [obstacle_ahead]
Generality \([\ ]\)
Specificity \([\ ]\)

level 1
use_breaks If [speed_distance_ratio_high]
Generality \([\ ]\)
Specificity \([\ ]\)

turn_off_road If [not_on_bridge]
Generality \([\ ]\)
Specificity \([\ ]\).

The above related HPRs form a tree (called HPR-tree) giving the following hierarchical structure Fig.1.

![HPR Tree Diagram](image_url)

III. DAMPSTER SHAFER THEORY

It provides a method for explicitly representing the ignorance inherent in knowledge and of reasoning in the absence of certain information. According to Haddawy the belief in a proposition A is represented by a shafer interval \([s(A), p(A)]\) which is a subinterval of the unit interval \([0, 1]\), where \(s(A)\) is called the support and \(p(A)\) is called the plausibility of the proposition A. These values can be thought of as an upper and lower bounds on the probability of A. The uncertainty of a proposition is then defined as the width of the interval: \(u(A) = p(A) - s(A)\). The precision of the probability estimate for the proposition A is then defined as: \(PRE(A) = 1 - u(A)\). The PRE of the decision is the measure, which will be used to characterize the certainty of an inference.

The variable precision logic (VLP) represents knowledge in terms of facts and rules. To apply Dempster Shafer theory to VLP, we must provide an interpretation of rules and facts in terms of shafer intervals. Dempster Shafer theory can be defined as:

\[ S(A) + s(\neg A) + 1 = 1, \quad \text{Where I stands for Ignorance.} \]

According to Dempster Shafer theory a belief factor is associated with each flat rule like \(P\rightarrow D\): \(\gamma\), which can be computed as [2]

\[ \gamma = \frac{P\rightarrow D}{|P|}. \quad (1) \]

This belief factor represents the support of the rule. Support of the decision can be obtained by using following formula:

\[ S(D) = s(P)^*s(P\rightarrow D) = s(P)^*\gamma \quad (2) \]

where \(s(P)\) is probability of occurrence of proposition A.

If rules can be expressed with uncertainty quantification using Shafer interval:

\[ \text{IF} \ P_1\{s(P_1), p(P_1)\} \text{and} \ P_2\{s(P_2), p(P_2)\} \text{THEN} \]

\[ \{s(P > D), p(P > D)\} \]

Dempster Shafer theory can be used for approximate reasoning in following manner:

a) \(\text{prob}\ (P_1 \text{and} P_2) \cdot [s(P_1) \cdot s(P_2), p(P_1) \cdot p(P_2)]\)

b) \(\text{prob}\ (P_1 \text{or} P_2) \cdot [s(P_1) + s(P_2) - (s(P_1) \cdot s(P_2)), p(P_1) + p(P_2) - (p(P_1) \cdot p(P_2))]\)

c) \(\text{prob}\ \left(\neg P_1 \right) \cdot \left[1 - p(P_1), 1 - s(P_1)\right]\)

d) \(\text{prob}\ (D) \cdot [s(P)^*s(P\rightarrow D), (1 - s(P)) + s(P) \cdot p(P\rightarrow D)]\)
IV. SUBSUMPTION MATRIX

A class $D_i$ can be defined by a set of properties (values of distinct attributes), $\text{class}_\text{prop}(D_i)$. Let $D_i$ and $D_j$ be any two classes with the set of properties $\text{class}_\text{prop}(D_i)$ and $\text{class}_\text{prop}(D_j)$, respectively. We can define the degree of subsumption ($\text{deg}_\text{sub}(D_i, D_j)$) as follows:

$$\text{deg}_\text{sub}(D_i, D_j) = \frac{|\text{class}_\text{prop}(D_i) \cap \text{class}_\text{prop}(D_j)|}{|\text{class}_\text{prop}(D_i)|}$$

(3)

where $\text{deg}_\text{sub}(D_i, D_j) \in [0, 1]$. A SM that summarizes the relationship between the classes, $D_1, D_2, \ldots, D_n$ is an $n \times n$ matrix defined as under:

$$\text{SM}[D_i, D_j] = \begin{cases} 1 & \text{if } \text{deg}_\text{sub}(D_i, D_j) = 1 \\ 0 & \text{i.e. } D_i \subseteq D_j \end{cases}$$

(4)

V. GENETIC PROGRAMMING APPROACH

As a post-processing scheme, we are using GP to discover crisp hierarchical production rules from the flat rules as input. The details of encoding, genetic operators and the fitness function for the proposed scheme are discussed in the following subsection:

A. Encoding

A hierarchical structure is encoded as a list representing a general tree:

Tree: (Root (sub-tree 1) (sub-tree 2) \ldots (sub-tree i) \ldots (sub-tree k)), where sub-tree i is either empty or has the same structure as Tree. For example, the hierarchy in Fig. 2.

![Fig 2 Hierarchy](image)

would be encoded as $(a (f (e (b (d) (k))))$.

An individual, as hierarchy must satisfy the following condition: $D_i \cap D_j = \emptyset$ for any two classes $D_i$ and $D_j$ at the same level in the hierarchy. During crossover/mutation operators, if any of the offspring or mutated individuals does not satisfy the above condition, then it will be rejected as an illegal individual.

B. Genetic Operators

The new elements in the population are generated by means of three operators: reproduction, crossover and mutation.

a) Reproduction

The reproduction operator selects one individual of the present population in proportion to its fitness value, so that the fitter an individual is the higher the probability that it will take part in the next generation of individuals. After selection, the individual is copied into the new generation without any modifications. Reproduction reflects the principle of natural selection and survival of the fittest [11].

b) Crossover

The crossover operator replaces a randomly selected sub-tree of an individual with a randomly chosen sub-tree from another individual and creates new offspring by exchanging sub-trees (i.e., sub-lists) between the two parents. The crossover point was chosen at random for both parents. For example, consider the following two individuals as parents (the “crossover point” is indicated by a tilted line and the sub-trees swapped by crossover are shown in bold):

Parent 1: \((a (b (c (h (n)))))\)

Parent 2: \((a (f) (e (b) (d) (k)))\)

with corresponding hierarchical structure is given in Fig. 3.

![Fig 3 Two parents before crossover](image)
The two offspring resulting from crossover are:
Offspring 1: (a(e (b) (d) (k)) (c(h) (n))) and
Offspring 2: (a(f) (b)) are shown below in Fig. 4.

![Fig. 4 Two offspring produced by crossover](image)

c) Mutation
For the tree mutation a sub-tree/leaf is replaced by randomly chosen sub-tree/leaf.

C. Fitness Function
The fitness function evaluates the quality of an individual in the population. For the proposed algorithm, the fitness measure of an individual is defined as:

\[
\text{Fitness} = \sum SM [Di, Dj] \quad (5)
\]

\[\forall D_i, D_j\]
The winning individual has the highest fitness such that \[\forall i, j \text{ Di} \rightarrow \text{ Dj}\].

VI EXPERIMENTAL RESULTS
Each GP run consisted of a population of 25 individuals evolving over generations. The probability of crossover, reproduction and mutation set to 0.8, 0.1 and 0.1, respectively, and the selection method used for both parents was fitness proportionate. For the practical reason of avoiding the expenditure of large amounts of computer time on occasional oversized programs, the depth of initial programs was limited to 8 and during the run the maximum tree depth was set to 10.

Example 1: By applying rule discovery algorithm and also using Dempster Shafer theory five flat rules along with belief factor are discovered:

1) If Employee_lives_in_city _y Then Employee _is_in_city_y : 0.2
2) If Employee _lives_in_city_ y \& time(night) Then Employee_is_at_home : 0.5
3) If Employee _lives_in_city_ y \& time(day) Then Employee_is_outside_home : 0.3
4) If Employee _lives_in_city_ y \& time(day) \& day(working) Then Employee_is_working_outdoor : 0.7
5) If Employee _lives_in_city_ y \& time(day) \& day(Sunday) Then Employee_is_entertaining_outdoor : 0.6
Support of the rules 1, 2, 3, 4, 5 is obtained by using formula (1). Values of $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ and $\gamma_5$ are 0.2, 0.5, 0.3, 0.7, and 0.6 respectively.

Using (1) and (2) the SM is constructed for the five classes 

D1= Employee _is_in_city_y;  
D2= Employee _is_at_home ;  
D3= Employee _is_outside_home ;  
D4= Employee _is_working_outdoor ;  
D5= Employee _is_entertaining_outdoor ,

Support for these decision classes can be obtained by using formula (2)

$S(D_1) = s(P) * \gamma_1 = 1.0 * 0.2 = 0.2$ 
$S(D_2) = s(P) * \gamma_2 = 0.4 * 0.5 = 0.2$ 
$S(D_3) = s(P) * \gamma_3 = 0.6 * 0.3 = 0.18$ 
$S(D_4) = s(P) * \gamma_4 = 0.28 * 0.7 = 0.19$ 
$S(D_5) = s(P) * \gamma_5 = 0.32 * 0.6 = 0.19$

Above five decision classes are mentioned in table

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The proposed algorithm produced the following individual with the highest fitness = 4:

(D1(D2)(D3(D4)(D5))).

The corresponding hierarchy is shown in Fig. 5.

![Fig. 5 Hierarchy-the individual with the highest fitness=4](image)

From the discovered hierarchy shown in Fig. 5, the following Quantified HPRs are generated:

level 0

(D1 If [Employee _lives_in_city_y] Generality [ ] Specificity [D2,D3 ]) [0.2 . . 1]

Dempster shafer intervals in the form $[s(P→D), p(P→D)]$ are obtained for each HPR. Further uncertainty of each rule can be obtained by using formula 

$u(A) = p(A) – s(A)$.

On the above discovered Quantified HCPR rules, approximate reasoning can also be applied according to Dempster Shafer theory explained in previous section.

Example 2: suppose we have 10 flat rules with 10 different classes as follows:

If $P_1 \& P_2 \& P_3$ Then $D_1 : \gamma_1$ 
If $P_1 \& P_2 \& P_4 \& P_7$ Then $D_2 : \gamma_2$ 
If $P_1 \& P_2 \& P_4$ Then $D_3 : \gamma_3$ 
If $P_1 \& P_2$ Then $D_4 : \gamma_4$ 
If $P_1 \& P_2 \& P_5 \& P_6$ Then $D_5 : \gamma_5$ 
If $P_1 \& P_2 \& P_3$ Then $D_6 : \gamma_6$ 
If $P_1 \& P_2 \& P_4 \& P_8 \& P_{11}$ Then $D_7 : \gamma_7$ 
If $P_1 \& P_2 \& P_4 \& P_8 \& P_9 \& P_{10}$ Then $D_8 : \gamma_8$ 
If $P_1 \& P_2 \& P_4 \& P_8$ Then $D_9 : \gamma_9$ 
If $P_1 \& P_2 \& P_4 \& P_8 \& P_{13}$ Then $D_{10} : \gamma_{10}$

Finally, the proposed algorithm produced the following individual with the highest fitness = 9:

(D4(D6)(D3(D7)(D8)(D10))(D1(D5))).

The corresponding hierarchy is shown in Fig. 6.
Fig. 6 Hierarchy—the individual with the highest fitness = 9
From the discovered hierarchy shown in Fig. 6, the following HPRs are generated:

- **Level 0**
  - (D4 If [P1, P2])
  - Generality [ ]
  - Specificity {[D6, D3, D1]} : [γ1 .. 1]

- **Level 1**
  - (D6 If [P3])
  - Generality [D4]
  - (D6 If [P3])
  - Generality [D4]
  - Specificity [ ] : [γ2 .. 1]

- (D3 If [P4])
  - Generality [D4]
  - Specificity [D2, D9] : [γ3 .. 1]
  - (D, [P3])
  - Generality [D4]
  - Specificity [D5] : [γ4 .. 1]
  - Specificity [ ] : [γ2 .. 1]

- (D3 If [P4])
  - Generality [D4]
  - Specificity [D2, D9] : [γ3 .. 1]
  - (D1 If [P5])
  - Generality [D4]
  - Specificity [D5] : [γ4 .. 1]

- **Level 2**
  - (D2 If [P7])
  - Generality [D3]
  - Specificity [ ] : [γ5 .. 1]
  - (D9 If [P8])
  - Generality [D3]
  - Specificity [D7, D8, D10] : [γ6 .. 1]
  - (D5 If [P6])

**VI. CONCLUSION**

As an attempt towards automated generation of hierarchies, a GP approach along with Dempster Shafer Theory is proposed to organize, summarize and present the discovered rules in the form of Quantified HPRs. Suitable genetic operators are proposed for the suggested encoding. Based on the SM, an appropriate fitness function is suggested. Performance of the proposed algorithm is demonstrated through experimental results, which are quite encouraging. Discovered Quantified HPR would facilitate quantitative reasoning and learning. One of the most important future research directions would be the discovery of Quantified HPRs with exceptions i.e. Quantified HCPR from large databases.

**REFERENCES**


