

# A Hierarchical Structure for Representing and Learning

## Fuzzy Rules

Ronald R. Yager

Machine Intelligence Institute

Iona College

New Rochelle, NY 10801, USA

N 93-23358

60277

p-5

## 1. Introduction

In [1] Yager provides an example in which the flat representation [2] of fuzzy *if-then* rules leads to unsatisfactory results. Consider a rule base consisting to two rules

if U is 12 the V is 29 I.

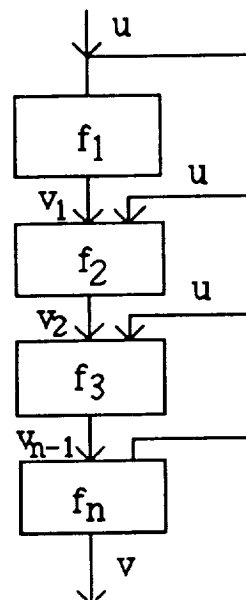
if U is [10-15] the V is [25-30] II.

If  $U = 12$  we would get  $V$  is  $G$  where  $G = [25 - 30]$ . The application of the defuzzification process leads to a selection of  $V = 27.5$ . Thus we see that the very specific instruction was not followed.

The problem with the technique used is that the most specific information was swamped by the less specific information. In this paper we shall provide for a new structure for the representation of *fuzzy if-then rules*. The representational form introduced here is called a **Hierarchical Prioritized Structure (HPS)** representation. Most importantly in addition to overcoming the problem illustrated in the previous example this HPS representation has an inherent capability to emulate the learning of general rules and provides a reasonable accurate cognitive mapping of how human beings store information.

## 2. Hierarchical Prioritized Structure

Figure 1, shows in a systematic view the of representation of the function  $V = f(U)$  by this new HPS representation. The overall function  $f$ , relating the input  $U$  to the output  $V$ , is comprised of the whole collection of subboxes, denoted  $f_i$ . Each subbox is a collection of rules relating the system input,  $U$ , and the current iteration of the output,  $V_{i-1}$ , to a new iteration of the output. The output of the  $n^{\text{th}}$  subsystem,  $V_n$ , becomes the overall output of the system,  $V$ . In the HPS the higher priority boxes, for  $i < j$  we say that  $f_i$  has a higher priority than  $f_j$ , would have less general information, consist of rules with more specific antecedents than those of lower priority. As we envision this system working an input value for  $U$  is provided, if it matches one or more of the rules in the first (highest priority) level then it doesn't bother to fire any of the less specific rules in the lower priority levels.



**Figure 1 Hierarchical Prioritized Structure**

In the following we describe the formal operation of this HPS. As we indicated  $V_j$  denotes

the output of the  $j^{\text{th}}$  level. We shall assume  $V_0 = \Phi$ . In the HPS we shall use the variable  $\widehat{V}_j$  to indicate the **maximum membership grade** associated with the output of the  $j^{\text{th}}$  level,  $V_j$ .

In the HPS each  $f_j$  (accept for the lowest level,  $j = n$ ) is a collection of  $n_j$  rules

**When  $U$  is  $A_{ji}$  is certain and  $\widehat{V}_{j-1}$  is low then  $V_j$  is  $B_{ji}$**  **I**

The representation and aggregation of rules at each level is of the standard Mamdani type[2], disjunction of the individual rules. If  $\widetilde{B}_j$  is the value obtained from the aggregation of the outputs of the collection of individual rules in I then the output of this subbox is

$$V_j = V_{j-1} \cup \widetilde{B}_j.$$

In I a rule fires if we are certain that the input  $U$  lies in  $A_{ji}$  and  $\widehat{V}_{j-1}$  is low. Since  $\widehat{V}_{j-1}$  is the maximum membership grade of  $V_{j-1}$  it can be seen as a measure of how much matching we have up to this point. Essentially this term is saying that if the higher priority rules are relevant,  $\widehat{V}_{j-1}$  is not low, then don't bother using this information. On the other hand if the higher priority rules are not relevant, not to much matching  $\widehat{V}_{j-1}$  is low, then try using this information.

The representation of the box  $f_n$  is a collection of rules

**When  $U$  is  $A_{ni}$  and  $\widehat{V}_{n-1}$  is low then  $V$  is  $B_{ni}$**  **II**

plus the rule  $V = V_n = V_{n-1} \cup \widetilde{B}_n$ . The notable difference between the lowest priority box and the other ones is that the antecedent regarding  $U$  is certainly quality in the higher boxes. The need for this becomes apparent when the input is not a singleton.

In the HPS structure  $\widehat{V}_{j-1}$  is the highest membership grade in  $V_{j-1}$  and as such the term  $\widehat{V}_{j-1}$  is *low* is used to measure the degree to which the higher prioritized information have matched the input data. We note that low is a fuzzy subset on the unit interval. One definition for low [1] is

$$\text{low}(x) = 1 - x.$$

In [1] Yager looks at the formal operation of this kind of HPS we shall present the results obtained in [1]. We shall denote  $\mathfrak{S}_{ij}$  as the degree of firing (or relevancy) of  $A_{ij}$  under the input, if the input is  $U = x^*$  then  $\mathfrak{S}_{ij} = A_{ij}(x^*)$ . We shall denote  $g_i = \text{Max}_y G_i(y) = \text{Poss}[G_i]$ . We let

$T_i = \bigcup_{j=1}^{n_i} \mathfrak{S}_{ij} \wedge B_{ij}$ , the aggregation of the rules in the  $i^{\text{th}}$  level for input  $U$ , it is essentially the contribution of the  $i^{\text{th}}$  subsystem using the Mamdani type reasoning.. We shall let  $G_i$  be the output of the  $i^{\text{th}}$  subbox, that is  $V_i = G_i$ . In [1] it is shown that

$$G_i(y) = (T_i(y) \wedge (1 - g_{i-1})) \vee G_{i-1}(y). \quad \text{III}$$

We notice that the term  $(1 - g_{i-1})$  bounds the allowable contribution of the  $i^{\text{th}}$  subsystem to the overall output. We see that as we get at least one element  $y$  to be, a good answer (an element in  $G_{i-1}$ ) we limit the contribution of the lower priority subsystems. It is this characteristic of a kind of saturation along with the prioritization that allows us to avoid the problem described earlier.

In the following we suggest a modification of the above that leads to a more suitable formulation to the aggregation between the levels of the HPS [1]. We can replace  $\wedge$  by another t-norm operator product  $*$  and replace  $\vee$  by another t-conorm, bounded sum, a  $\boxed{+}$   $b = \text{Min}(1, a+b)$ [3]. Thus we get

$$G_i(y) = T_i(y) * (1 - g_{i-1}) \boxed{+} G_{i-1}(y).$$

However we note that since  $g_{i-1} = \text{Max}_y G_{i-1}$  then  $T_i(y) * (1 - g_{i-1}) \leq G_{i-1}(y)$  hence  $T_i(y) * (1 - g_{i-1}) + G_{i-1}(y) \leq 1$  thus we can replace  $\boxed{+}$  by  $+$ . This gives us the formulation

$$G_i(y) = T_i(y) * (1 - g_{i-1}) + G_{i-1}(y) \quad \text{IV}$$

$$G_i(y) = T_i(y) * (1 - \text{Poss}[G_{i-1}]) + G_{i-1}(y)$$

What is happening in this structure is that as long as we have not found one  $y$  with

membership grade 1 in  $G_{i-1}$ ,  $\text{Poss}[G_{i-1}] \neq 1$ , we add some of the output of the current subbox to what we already have. Each element  $y$ , gets  $1 - \text{Poss}[G_{i-1}]$  portion of the contribution at that level,  $T_i(y)$

We should point out that the aggregation performed in the hierarchical structure, whether we use III or IV, is not a pointwise operation. This means that the value of  $G_i(y)$  doesn't only depend on the membership grade of  $y$  in  $G_{i-1}$  and  $T_i$  but on membership grades at other points. In particular through the term  $\bar{g}_{i-1} = 1 - \text{Max}_y G_i(y)$  it depends upon the membership grade of all elements from  $Y$  in  $G_{i-1}$ .

We should note that implicit in this structure is a new kind of aggregation. Assume  $A$  and  $B$  are two fuzzy subsets we define the combination of these sets as the fuzzy subset  $D$ , denoted  $D = \gamma(A, B)$  where

$$D(x) = (1 - \text{Poss}(A)) * B(x) + A(x).$$

### 3. Representation and Operation of the HPS

In the previous part we have described the formal mechanism used for the reasoning and aggregation process in the HPS. While the formal properties of the new aggregation structure are important a key to the usefulness of the HPS in fuzzy modeling is the semantics used in the representation of the information via this structure.

In constructing an HPS representation to model a system we envision that the knowledge of the relationship contained in the HPS structure be stored in the following manner. At the highest level of priority,  $i = 1$ , we would have the most specific precise knowledge. In particular we would have point to point relationships,

*When U is 3 then V is 7*  
*When U is 9 then V is 13*

This would be information we know with the greatest certainty.

At the next level of priority the specificity of the antecedent linguistic variables, the  $A_2$ 's, would decrease. Thus the second level would contain slightly more general knowledge.

Essentially what we envision is that at the highest level we have specific point information. The next level encompass these points and in addition provides a more general and perhaps fuzzy knowledge. We note that the lowest most level can be used to tell us what to do if we have no knowledge up to this point. In some sense the lowest level is a default value.

**Example:** Assume we are using an HPS representation to model a function  $V = f(U)$ , where the base set for  $U$  is  $[0, 100]$ . A typical HPS representation could be as follows.

#### LEVEL #1

R11            When U is 5 then V is 13  
R12            When U is 75 then V is 180  
R13            When U is 85 then V is 100

#### LEVEL #2

R21            When U is "about 10" then V is "about 20"  
R22            When U is "about 30" then V is "about 50"  
R23            When U is "about 60" then V is "about 90"  
R24            When U is "about 80" then V is "about 120"  
R25            When U is "about 100" then V is "about 150"  
                  (we assume triangular fuzzy subsets)

#### LEVEL #3

R31            When U is "low then V is "about 40"  
R32            When U is "meet" then V is "about 85"  
R31            When U is "high" then V is "about 130"

#### LEVEL #4

R41            U is anything the V is 2u.

Having defined our knowledge base we now look at the performance of this system for various inputs;

Case 1:  $U = 75$ . At level one we get  $T_1 = \left\{ \frac{1}{180} \right\}$  hence since

$$G_1(y) = \bar{g}_0 * T_1(y) + G_0(y).$$

Since  $G_0 = \Phi$  then  $g_1 = 1$  which give us  $G_1 = T_1 = \left\{ \frac{1}{180} \right\}$ . We now see that  $\bar{g}_1 = 0$  and hence no other rules will fire lower in the hierarchy. This system provides as its output for  $U = 75$  that  $V$  is 180.

Case 2:  $U = 80$ . At level no rules fire,  $\mathfrak{S}_{ij} = 0$  for all  $j$ . Thus  $T_1 = \Phi$  hence

$$G_1 = \bar{g}_0 * T_1 + G_0 = \Phi$$

and therefore  $\bar{g}_1 = 1$ . At level two

$$G_2 = \bar{g}_1 * T_2 + \Phi = T_2.$$

For  $U = 80$  we assume that  $R_{24}$  fires completely,  $\mathfrak{S}_{24} = 1$  and that all other rules don't fire,  $\mathfrak{S}_{2j} = 0$ , for  $j \neq 2$ . Thus  $T_2 = \text{"about 120"}$  and  $G_2 = \text{"about 120"}$ . Since  $g_2 = 1$  then  $\bar{g}_2 = 0$  and no rules at lower priority will fire thus  $G_2$ , "about 120", is the output of the system for  $U = 80$ .

Case 3:  $U = 20$ . No rule at level one will fire, hence  $G_1 = G_0 = \Phi$ . At level two we shall assume that  $R_{21}$  fires to degree .3 and  $R_{22}$  also fires to degree .3. Thus

$$T_2 = .3 \wedge B_1 \cup .3 \wedge B_2 = .3 \wedge (B_1 \cup B_2)$$

$$T_2(y) = .3 \wedge (B_1(y) \vee B_2(y)).$$

We note  $B_1$  and  $B_2$  are "about 20" and "about 30" respectively. Hence

$$G_2(y) = (1 - g_1) * T_2(y) + G_1(y) = T_2(y)$$

At level three  $R_{31}$  fires to degree 1 while  $R_{32}$  and  $R_{33}$  don't fire at all. Hence

$$T_3 = \text{"about 40"}$$

Since  $\text{Max } [G_2] = .3$  thus  $1 - g_2 = .7$  and therefore

$$G_3(y) = .7 * T_3(y) + G_2(y)$$

Since  $\text{Max } T_3(y) = 1$  we see that the process stops here and  $G_3$  is the output of the system.

What we see with this HPS representation is that we have our most general rule stored at the *lowest* level of priority and we store exceptions to this rule at higher levels of priority. In some cases the exceptions to general rules may themselves be rules, we would then store exceptions to these rules at still higher levels of priority. As the previous example illustrates in the HPS system for a given input we first look to see if the input is an exception, that is what we are essentially doing by looking at the high priority levels.

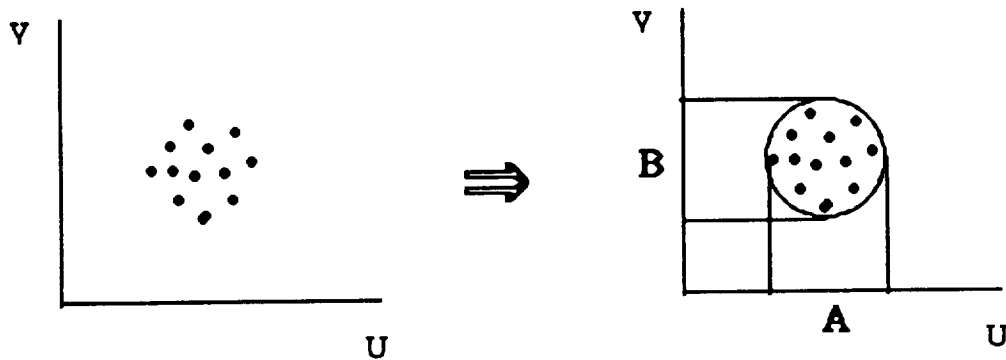
#### 4. Learning in the HPS

The HPS representation is a formulation that has an inherent structure for a natural human like learning mechanism. We shall briefly describe the type of learning that is associated with this structure.

Information comes into the system in terms of point by point knowledge, data pairs between input and output. We store these points at the highest level of priority. Each input/output pair corresponds to a rule at the highest level. If enough of these points cluster in a neighborhood in the input/output space we can replace these points by a general rule (see figure 2).

Thus from the dots, input/output pairs, we get a relationship that says *if U is in A then V is B*. We can now forget about the dots and only save the new relationship. We save this at the next lowest priority in the system, in subbox 2.

We note that the introduction of the rule essentially extends the information contained in the dots by now providing information about spaces between the dots. We can also save storage because we have eliminated many dots and replaced them by one circle. One downside to this formulation is that in generalizing we have lost some of the specificity carried by the dots.



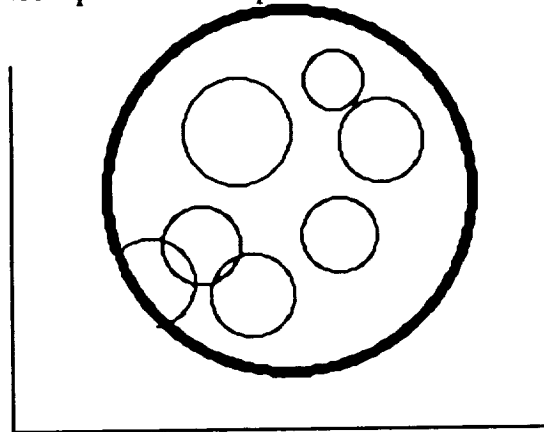
**Figure 2 Formulation of Rules for Input/Output Pairs**

It may occur that there are some notable exceptions to this new general rule. We are able to capture this exception by storing them as high level points.

We further note that new information enters the system in terms of points. Thus we see that the points are either new information or exceptions to more general rules. Thus specific information enters as points it filters its way up the system in rules.

We see that next that it may be possible for a group of these second level rules to be clustered to form new rules at the third level.

In figure #3 the large bold circle is seen as a rule which encompasses the higher level rules to provide a more general rule. The necessity to keep these more specific rules, thus in level 2, depends upon how good the less specific rule captures the situation.



**Figure 3 Aggregation of Rules into More General Rules**

## 5. References

- [1]. Yager, R. R., "On structures for fuzzy modeling and control," Technical Report MII#1213, Machine Intelligence Institute, Iona College, 1992.
- [2]. Mamdani, E. H. and Assilian, S., "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. of Man-Machine Studies* 7, 1-13, 1975.
- [3]. Dubois, D. and Prade, H., "A review of fuzzy sets aggregation connectives," *Information Sciences* 36, 85 - 121, 1985.