Toward a Theory of Intelligently Directed Aggregation IDA

Ronald R. Yager
Machine Intelligence Institute
Iona College
New Rochelle, NY 10801
yager@panix.com

In various applications we have introduced some specialized information aggregation and fusion methods in which the effect and importance of new information depends on earlier information provided, the current state of our knowledge.

IDA

We shall refer to these methods as **IDA**

Intelligently Directed Aggregation

Here we shall describe some examples of IDA

Human Learning

- Step by Step inclusion of new information
- We fuse new information with old
- We use our intelligence to control the learning process
- Provides a prototypical example of IDA

Understanding Mechanisms of Human Learning

- Enables the construction of more cognitively capable agents
- Build better algorithms for processing data
- Improve aspects of man-machine interaction

Participatory Learning

A Paradigm for More Human Like Learning

Basic Premise of Participatory Learning Paradigm

Learning takes place in the framework of what is already learned and believed

- **Every** aspect of the learning process is effected and guided by the current belief system

- Learning is a highly noncommutative Process

  First Impressions are Important

  The order of experiences matters

An Example Model of Participatory Learning

- Variables of Interest: $V = \begin{bmatrix} V_1 \\ . \\ V_n \end{bmatrix}$
- Current Belief of Variables: $X = \begin{bmatrix} x_1 \\ . \\ x_n \end{bmatrix}$
- Additional new data: $D = \begin{bmatrix} d_1 \\ . \\ d_n \end{bmatrix}$

Updation Process

- Updation Process:

$$\widehat{X} = F(X, D)$$

Updated Belief Structure

$$\widehat{X} = \begin{bmatrix} \widehat{x}_1 \\ . \\ \widehat{x}_n \end{bmatrix}$$

- Learning depends on difference between what we currently believe and new data

$$(D - X)$$

Participatory Learning Mechanism

$$\hat{X} = X + \beta\,(D - X)$$

↑

Openness to learning

$$\beta = \alpha\,\rho$$

Basic Learning Rate

Compatibility of observation with current belief

Compatibility Mechanism

- **Strongly Noncommutative**

  Measures compatibility of Observation with Belief

- **Possible Form for ρ**

$$\rho = 1 - \frac{1}{n} \sum |x_j - d_j|$$

- **Updation Algorithm:**

$$\widehat{X} = X + \beta (D - X)$$

$$\widehat{x}_j = x_j + \alpha \rho (d_j - x_j)$$

- **Provides Context** Uses Information about whole observation

- Features of Learning Algorithm

$$\hat{x}_j = x_j + \alpha\rho(d_j - x_j)$$

i. No learning when data **completely** agrees or disagrees with current belief

ii. Optimal learning when **most** of the d_j's agree with the x_j's

iii. Step by step

- Allows fast learning

High α

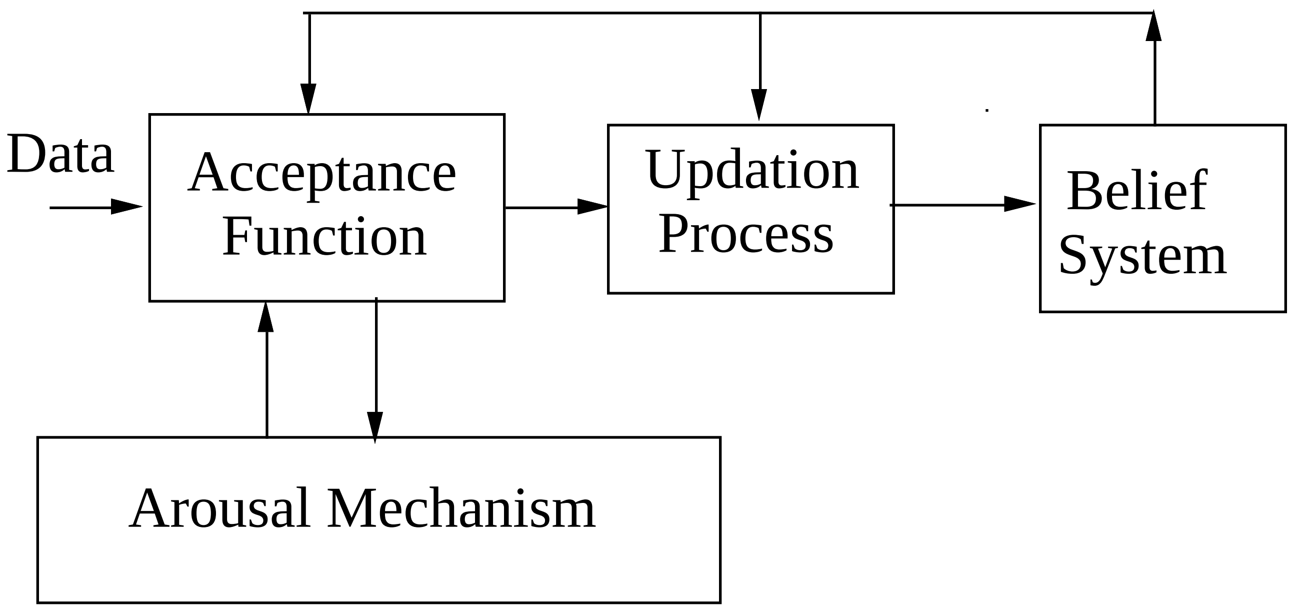
"Bad" Observations Choked by ρ

- ρ evaluates the acceptability of data in context of current belief system

- can be a very complex process involving a more complex inference process then in the above

Needs Mechanism to Avoid Being Locked into Wrong Beliefs !!!!

General Framework of Participatory Learning Paradigm



AROUSAL MECHANISM

Introduce arousal factor to monitor performance of beliefs

Modified Updation Rule:

$$\hat{X} = X + \alpha\, \rho^{(1-\delta)}\, (D - X)$$

$\uparrow$ arousal factor

$$\widehat{X} = X + \alpha\, \rho^{(1-\delta)} (D - X)$$

- δ **Near zero**

  Model blocks out incompatible data

  Past data are compatible with belief

- δ **Near One**

  Lets all data modify beliefs

  Historically Beliefs are Incompatible with observations

Arousal Factor Calculation

$$\hat{\delta} = \delta + \lambda ((1 - \rho) - \delta)$$

δ: Current Arousal factor

$\hat{\delta}$: New arousal factor

λ: Learning rate

ρ: Compatibility of current observation

- Incompatible Observation ($\rho = 0$) increases arousal
- Compatible Observation ($\rho = 1$) decreases arousal

Alternative Method for Inclusion of Arousal Factor

- $F(\rho, \delta) = \rho(1 - \delta)$

- $F(\rho, \delta) = S(\rho, \delta)$

  S is a t-conorm

- $F(\rho, \delta) = \text{Max}(\rho, \delta)$

- $F(\rho, \delta) = \rho + \delta - \rho\,\delta$

Basic Model

$$\widehat{X} = X + (\rho + \delta - \rho\,\delta)\,\alpha\,(D - X)$$

Overall Structure of Participatory Learning Paradigm

Two Level Model

I. Foreground: Learning Rule

- $\widehat{X} = X + \alpha \rho^{(1 - \delta)} (D - X)$
- Changes belief based on current observation
- Fast Learning: high α

II. Background: Arousal Factor Calculation

- $\widehat{\delta} = \delta + \lambda ((1 - \rho) - \delta)$
- Performance based on history of compatibility
- Slow change: low λ

One Lesson for teaching

Repeat things in different ways

Participatory Learning In Neural Networks

- Neural model with n output nodes
- Error at the k^{th} output neuron e_k
- Change in connection weight between node k and node j of the previous layer

$$\Delta w_{kj} = \eta \, x_j \, f'_k \, (net\ k) \, e_k$$

- Introduce term ρ^b : $\Delta w_{kj} = \rho^b \, \eta \, x_j \, f'_k \, (net\ k) \, e_k$
- $\rho = 1 - \frac{1}{n} \sum |e_j|$, compatibility of observation with model
- b is the measure of confidence

  necessitates introduction of a arousal component
- ρ^b term introduced at all layers in the back propagation

Rule Base Expression of PL Model

R1: If compatibility of observation is high then move current belief closer to current observation

R2: If compatibility of observation is low then don't change current belief

Fuzzy Systems Model of Rule Base

R1: If ρ is **HIGH** then Δ is α (D - X)

R2: If ρ is **LOW** then Δ is 0

Solution of Fuzzy Model

$$\Delta = \frac{\mathrm{HIGH}(\rho)\ \alpha\ (D - X) + \mathrm{LOW}(\rho)\ 0}{\mathrm{HIGH}(\rho) + \mathrm{LOW}(\rho)}$$

Using Linear definitions of Fuzzy Sets

$$\mathrm{HIGH}(\rho) = \rho$$

$$\mathrm{LOW}(\rho) = 1 - \rho$$

$$\Delta = \rho\ \alpha\ (D - X)$$

Obtain Original Rule Updation Rule:

$$\widehat{X} = X + \rho\ \alpha\ (D - X)$$

Inclusion of Arousal: $\widehat{X} = X + \rho^{(1-\delta)}\ \alpha\ (D - X)$

Allow Slow Learning in case of Incompatibility

R1: If ρ is **HIGH** then Δ is α (D - X)

R2: If ρ is **LOW** then Δ is β (D - X)

$$\alpha > \beta$$

Solution:

$$\Delta = (\beta + (\alpha - \beta)\rho)\ (D - X)$$

Updation Rule:

$$\widehat{X} = X + (\beta + (\alpha - \beta)\rho)\ (D - X)$$

Inclusion of Arousal Factor:

$$\widehat{X} = X + (\beta + (\alpha - \beta)\rho^{(1 - \delta)})\ (D - X)$$

Properties of Model: $\widehat{X} = X + (\beta + (\alpha - \beta)\rho^{(1-\delta)})\,(D - X)$

- $\beta = 0$ **Obtain Previous model**

- $\beta = \alpha$ $\widehat{X} = X + \alpha\,(D - X)$

  Usual learning model Nonparticipatory model

- $\beta \neq 0$

  Learning Occurs even if $\rho = 0$

- $\rho = 1$

  Learning Occurs at α rate

- $\rho = 0$

  Learning Occurs at β rate

General Participatory Learning Model

- **Rule Base:**

  If ρ is A_i then Δ is α_i (D - X)

- A_i fuzzy subsets partitioning unit interval:

(very low, low etc) $A_i < A_j$ for $i < j$

- α_i learning rates:

  $\alpha_i < \alpha_j$ for $i < j$

Updation Rule

$$\widehat{X} = X + \frac{\sum_{j=1}^{n} A_j(\rho)\alpha_j}{\sum_{j=1}^{n} A_j(\rho)}(D - X)$$

Inclusion of Arousal Component

$$\widehat{X} = X + \frac{\sum_{j=1}^{n} A_j(\rho^{(1-\delta)})\alpha_j}{\sum_{j=1}^{n} A_j(\rho^{(1-\delta)})}(D - X)$$

Learning Experience

- **Components**

  Content of the Experience

  Source of the Content

- **Acceptability of Experience depends on**

  Content compatibility current belief system

  Credibility of the source.

Information about both these components are contained in PL agent's current belief system

Normally compatible content is allowed into the system and is more valued if it is from a credible source rather than a non-credible source

Incompatible content is generally blocked, and more strongly blocked from a non-credible source than credible source

Including Source Credibility in PL Process

- Collection of possible sources $S = \{S_1, ..., S_n\}$

- $C(j) \in [0, 1]$ belief about credibility of source S_j

  Stored in the belief system of the Pl agent

- Updation when k^{th} observation from source S_j

$$V_k = V_{k-1} + \alpha\, C(j)\, \rho_k^{(1-a_k)}(d_k(i) - V_{k-1}(i))$$

- Arousal level updation algorithm

$$a_k = (1 - C(j)\beta)\, a_{k-1} + C(j)\, \beta\ \bar{\rho}_k$$

Updation of the Source Credibility

- PL agent learns source credibility from learning experiences

- $C_k(j)$ credibility of source S_j after k^{th} experience

- $M_{jk} = 1$ if S_j is the source of k^{th} experience
  $M_{jk} = 0$ if S_j is not source of the kth experience

- $C_k(j) = C_{k-1}(j) + M_{jk} \lambda \bar{a}_{k-1}(\rho_k - C_{k-1})$

  $\lambda \in [0, 1]$ is a base learning rate

- Can be different learning rates for each source

- If $\lambda(j) = 0$ fixed credibility for the j^{th} source

Classes of Learning Sources

- Direct sensory experiences

  Seeing an auto accident,

  Smelling alcohol on somebody's breath

  Hearing John tell Mary I love you.

  Observations made with our own sensory organs

- From an "authority"

  Being told by another person

  Reading something in a book

  Obtaining it from the internet

  Seeing it on T.V.

  Contents processed by some other cognitive agent

  Subject to "interpretation" by the processing agent

- From electro-mechanical sensor

  Speedometer on your car

  Thermometer

  Air traffic controllers screens

Contents processed by neutral physical device

- From reflection

  Deduction

  Induction

  Reasoning

Conscious **rational** manipulation of information

already in an agent's belief system

- From Beyond(Mystic)
  - Dreams
  - Hallucinations
  - Being told by God
  - Gut feeling

Useful in modeling terrorists many of whom are religious fundamentalists who construct their belief system using this type of source.

REFERENCES

• Yager, R. R., "A model of participatory learning," IEEE Transactions on Systems, Man and Cybernetics 20, 1229-1234, 1990.

• Yager, R. R. and Filev, D. P., "Modeling participatory learning as a control mechanism," International Journal of Intelligent Systems 8, 431-450, 1993.

• Yager, R. R., "Participatory learning: A paradigm for building better digital and human agents," Law, Probability and Risk 3, 133-145, 2004.

• da Silva, L. R. S., Gomide, F. and Yager, R. R., "Fuzzy clustering with participatory learning and applications," in Advances in Fuzzy Clustering and its Applications, edited by Oliveira, J. V. and Pedrycz, W., John Wiley & Sons: Hoboken, NJ, 139-153, 2007.

• Yager, R. R., "Extending the participatory learning paradigm to include source credibility," Fuzzy Optimization and Decision Making 6, 85-97, 2007.

• Yager, R. R., "Participatory learning with granular observations," IEEE Transactions on Fuzzy Sets and Systems 17, 1-13, 2009.

• Yager, R. R., "Using participatory learning to model human behavior," in Social Computing and Behavioral Modeling, edited by Liu, H., Salerno, J. and Young, M. J., Springer: New York, 244-253, 2009.

• Yager, R. R., "Participatory learning of propositional knowledge," IEEE Transactions on Fuzzy Systems, (To appear).

Hierarchical Representation for Fuzzy Systems Modeling

The Hierarchical Prioritized Structure

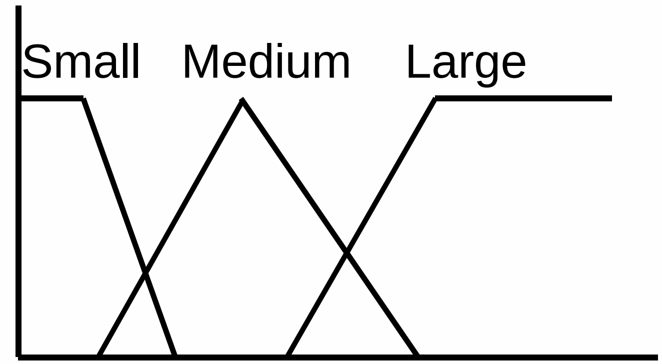
The HPS Model

Advantages of Fuzzy Systems Modeling

- Widely used technology for building intelligent systems with many applications
- Allows rapid and inexpensive development of systems by greatly reducing the number of rules needed in the modeling process
- Allows capture of expert information in a manner easy for the expert to articulate by providing a bridge between human cognition and formal models.

Basic Fuzzy Systems Modeling

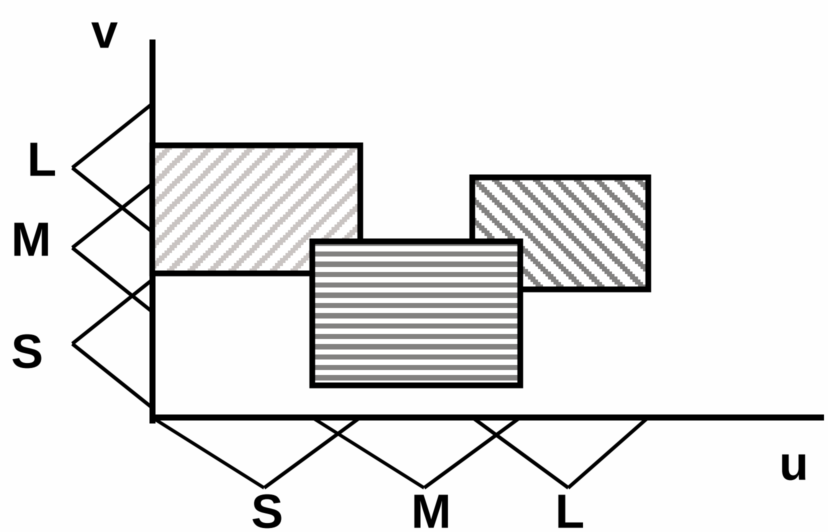
- Uses fuzzy sets to represent words



- Uses collection of fuzzy rules to represent relationships:

  If U is A_i then V is B_i

  A_i and B_i are fuzzy sets



- **Flat** representation of *if-then* rules can lead to unsatisfactory results.

- **Rule base:**
  If U *is* 12 the V *is* 29
  If U *is* [10-15] then V *is* [25–30].

- With U = 12 we get V is [25 - 30].
- Defuzzification leads to V = 27.5

- *Very specific instruction was not followed*

The more specific information was swamped by the less specific information

Difficulty Handling Conflicting Special Cases when Using Default Rules

Default Rule:
If the light is red then Cross.

Special Case:
If the light is red and a fire engine is coming then **Don’t** Cross.

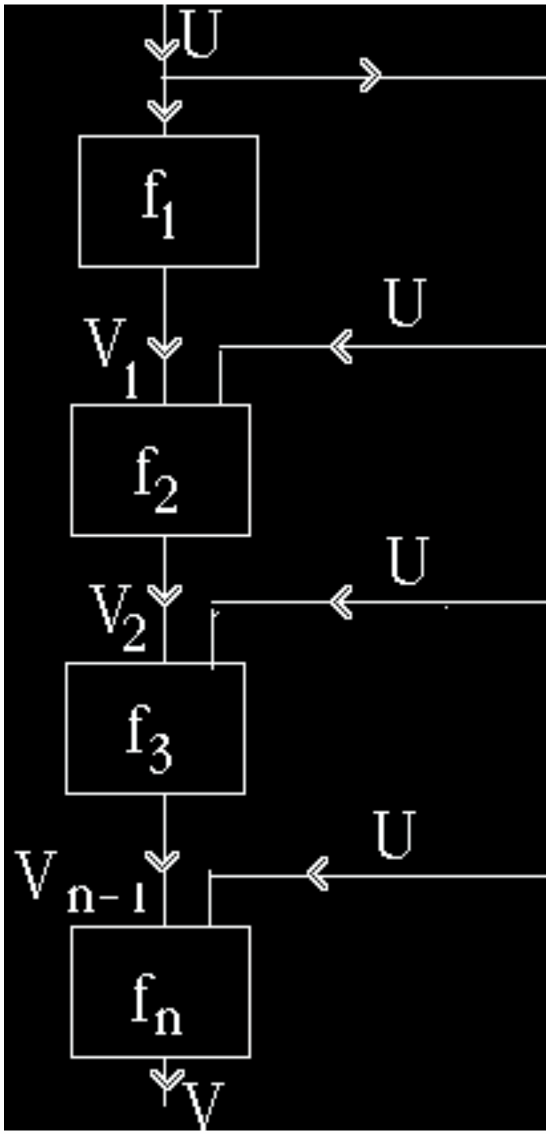
Systems needs to intelligently decide which rule to apply

Benefits of Including Default Rules in Models

- Gives applications robustness to operate in situations in which they have not been explicitly trained.
- Contributes to affordable systems development by further reducing number of rules needed
- Allows modularity by enabling modeling of commonsense knowledge which can used across applications

We provide the ability to include default rules in applications by using an extension of fuzzy systems modeling technology based on a hierarchical structure

Hierarchical Prioritized Structure (HPS)



- Each level consists of a collection of fuzzy rules
- More specific rules higher in structure
- Default rules at lower level
- Inference combines basic fuzzy model with hierarchical aggregation

Inference in Hierarchical Prioritized Structure

- Within each level uses basic fuzzy reasoning
- Between levels uses special hierarchical aggregation operator
- Combination allows preference to be given to rules higher in the structure

Functioning of the HPS

- 1. Initialize $V_0 = \varnothing$ and j = 1
- 2. Calculate potential contribution of level j, T_j
- 3.Using HUA **H**ierarchical **U**pdation **A**ggregation find output V_j of the j th level, denoted G_j
- 4. If j = n stop; Model output is V_n
- 5. Set j = j + 1 and go to step 2

Calculation of Output T_j

- Level j consists of fuzzy rules

  If U is A_{ji} then V is B_{ji}

- System input is U = x*

- Apply standard fuzzy reasoning calculate potential output at level j T_j

$\forall T_j(y) = \mathrm{Max}_i[\Upsilon_{ji} \wedge B_{ji}(y)]$ where $\Upsilon_{ji} = A_{ji}(x^*)$

Hierarchical Aggregation Operator

Formally the hierarchical aggregation method uses

$$G_j(y) = T_j\,(y) * (1 - g_{j-1}) + G_{j-1}(y)$$

→ G_j is output of level j

→ g_{j-1}= $Max_y[G_{j-1}(y)]$ measures how much matching up to this point.

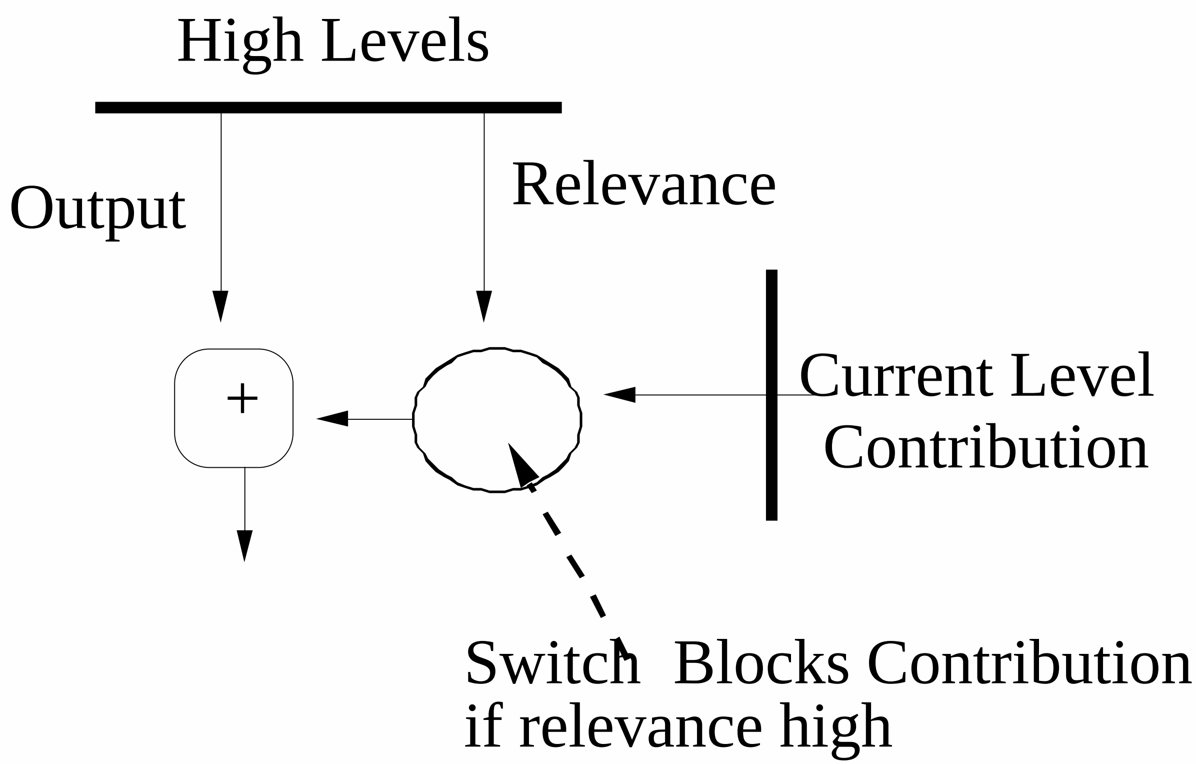
→ T_j is potential contribution of level j

Properties of HAU

$$G_j(y) = T_j\,(y) * (1 - g_{j-1}) + G_{j-1}(y)$$

- Not pointwise
- g_{j-1} acts as choke on effect of T_j
- If $g_{j-1} = 1$ no effect of new level
- Effect of new level inversely proportional to strength of solution already found

How Hierarchical Aggregation Works



Allows contribution from current level only if no solution found at higher levels

HUA Operator

- If no y with $G_{j-1}(y) = 1$ we add to output
- Each y gets $1 - g_{j-1}$ portion of the potential contribution at that level
- HPS looks for the most relevant rule relating to this input.
- If this rule completely matches x* then we stop our inference process

*The **HEU** function*

$$G_j(y) = T_j(y) * (1 - g_{j-1}) + G_{j-1}(y)$$

Maximizes the specificity *of the value of the output variable. If G is the final output of the system HEU operates so make G have a high specificity given the knowledge base*

HEU Is Nonmonotonic

$D(z) = A(z) * (1 - b_{max}) + B(z)$

- **Monotonicity:**

If $A_2 \subseteq A_1$ and $B_2 \subseteq B_1$ then $D_2 \subseteq D_1$

- **Counter Example:**

$A_1 = A_2 = \{1/x, 0.8/y\}$

$B_2 = \{0.2/x, 0.4/y\}$ $B_1 = \{0.2/x, 1/y\}$

$D_1 = \{0.2/x, 1/y\}$ $D_2 = \{0.8/x, 0.88/y\}$

$D_2 \not\subset D_1$

Constructing HPS Models

- Constructed from User Supplied Rule Base
- Learned from Observed Data
- Adaption
  Initialize with user knowledge
  Modify using Observations

Example

Using an HPS to model $V = f(U)$

LEVEL #4

Default knowledge

If U is anything the V is 2u

LEVEL #3

If U is *low* then V is "about 40"

If U is *med* then V is "about 85”

If U is *high* then V is "about 130"

LEVEL #2

If U is "about 10" then V is "about 20"

If U is "about 30" then V is "about 50"

If U is "about 60" then V is "about 90"

If U is "about 80" then V is "about 120

If U is "about 100" then V is "about 150

LEVEL #1

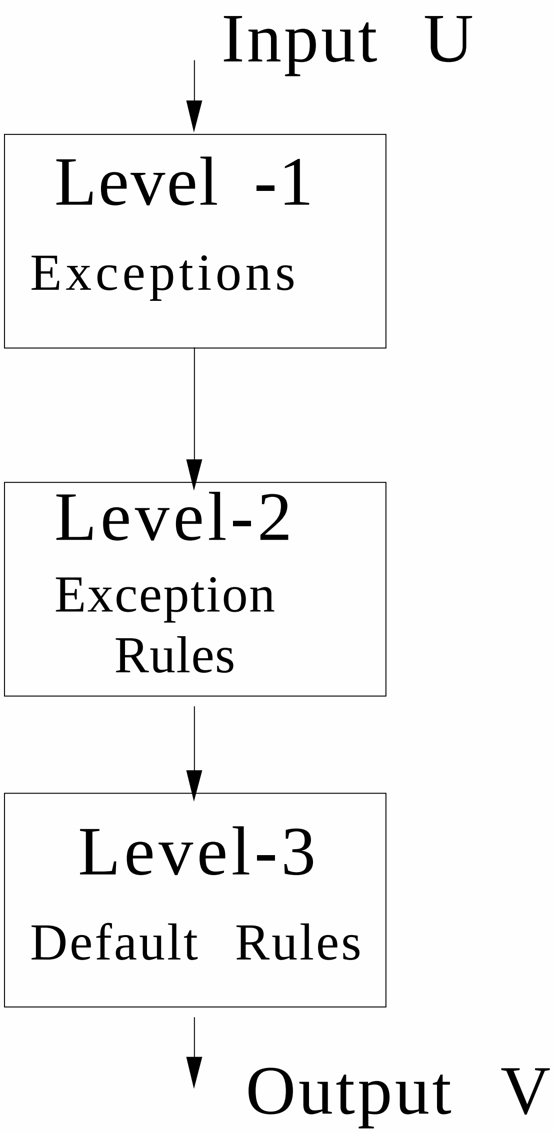
If U is 5 then V is 13

If U is 75 then V is 80

If U is 85 then V is 100

HPS has an inherent structure to enable natural human-like learning mechanism

Three Level Exception Driven HPS

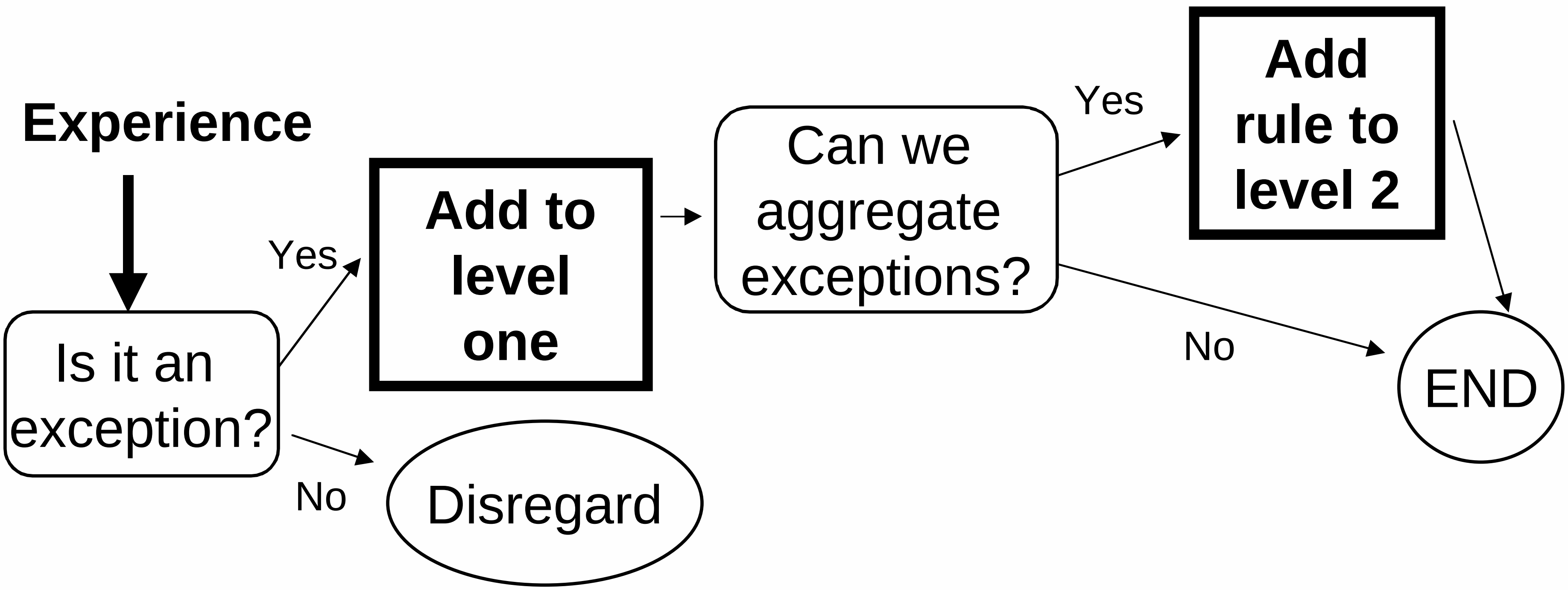


- Initialize with default and other rules at lowest level
- Only exceptions (observations deviating from model prediction) are remembered,
- Rules at level-2 based on clustering of exceptions
  - Patches to initial rules

Learning Paradigm in HPS

1. Information enters the system from observations and experiences.
2. Stored at highest level of hierarchy.
3. When enough observations cluster in a neighborhood, replace these by a rule at next lower level of hierarchy
4. If possible combine group of second level rules to form new “more general” rule at third level - resulting in rule reduction

Learning Mechanism



Three Level HPS

- Provides natural human-like learning
- Allows for inclusion of prior knowledge
- Uses observations to modify prior knowledge
- Rules filter down the model

Benefits of Hierarchical Prioritized Structure

- Allows inclusion of default rules
- Creates prioritized aggregation
- Retains modularity
- Capability to emulate learning of general rules
- Provides model allowing explanation of how and what has been learned

Lexicographically Prioritized Multicriteria Decisions Using Scoring Functions

Multi-Criteria Decision Problem

- Collection of criteria $C = \{C_1, ..., C_n\}$
- Set of alternatives $X = \{x_1, ..., x_m\}$.
- $C_i(x)$ as a value in the unit interval
- Overall satisfaction of alternative to criteria

$$C(x) = \sum_i w_i \, C_i(x)$$

- Weighted Aggregation of criteria satisfactions

Properties of Importance Weights

- $w_i \in [0, 1]$

- C(x) is called a **weighted scoring function**

- C(x) is monotonic in $C_i(x)$

- **Special case**: w_i sum to 1

  C(x) is called a **weighted averaging function**

  $\mathrm{Min}_i[C_i(x)] \leq C(x) \leq \mathrm{Max}_i[C_i(x)]$ (Bounded)

These weighted aggregation operators allow tradeoffs between criteria.

We can compensate for decrease of Δ in satisfaction to criteria C_i by gain $w_k/w_i \ \Delta$ in satisfaction to criteria C_k.

In some applications we may have a **lexicographic** ordering of the criteria and do not want to allow this kind of compensation between criteria.

Child Bicycle Selection Problem

- Selecting bicycle for child using criteria of **safety** and **cost**
- However any bicycle we select must be safe
- **We do not want poor safety to be compensated for by very low cost**.
- Before considering cost must be sure the bicycle is safe.
- A **lexicographic** induced prioritization ordering of criteria.
- Safety has a higher priority.

- In organizational decision making criteria desired by superiors generally, have a higher priority then those of their subordinates. The subordinate must select from among the solutions acceptable to the superior.

- Air traffic controller decisions involve a prioritization of considerations with passenger safety usually at the top.

WHAT IS NEEDED

An aggregation operator that can handle lexicographically induced priority between the criteria

Solution Imperative

- Use importance weights
- Importance weight of lower priority criteria based on satisfaction to higher priority criteria
- Effectively prevents satisfaction of lower priority criteria from compensating for poor satisfaction to higher priority criteria.

Prioritized Scoring

Operator

Problem Formulation

- Collection of criteria partitioned into q distinct categories

  $H_1, H_2, ..., H_q$

- $H_i = \{C_{i1}, C_{i2}, ..., C_{ini}\}$: C_{ij} are the criteria in category H_i
- A prioritization between these categories

  $H_1 > H_2, ... > H_q$

- Criteria in H_i have a higher priority than those in H_k if $i < k$
- Criteria in the same category have the same priority
- Total number of criteria is n

Prioritized Scoring Operator
PS Operator

- Alternative $x \in X$
- $C_{ij}(x) \in [0, 1]$ is x satisfaction to criteria C_{ij}.
- $C(x)$ overall score for alternative x
- **P**rioritized **S**coring (PS) operator

$$C(x) = \sum_{i=1}^{q} \left(\sum_{j=1}^{n_i} w_{ij} C_{ij}(x) \right)$$

- Weights used to enforce the priority relationship
- Weights will be dependent on x

Determination of Weights

- For each category H_i we calculate $S_i = \text{Min}_j[C_{ij}(x)]$
- S_i is the value of the least satisfied criteria in category H_i
- $S_0 = 1$ by convention
- Calculate

$$T_i = \prod_{k=1}^{i-1} S_k \qquad (T_3 = S_0 S_1 S_2)$$

- Set $u_{ij} = T_i$
- Use $w_{ij} = u_{ij}$

Properties of the weights

- Criteria in same category have same weight

$$w_{ij} = T_i$$

- Criteria in top category have weight 0ne

$T_1 = 1$ (Criteria in H_1 have weight 1)

- Lower priority criteria smaller weights

$$T_i \geq T_k \text{ for } i < k$$

- If $S_i = 0$ then $w_{kj} = 0$ for $k > i$ (Contribution blocked)

Effective Prioritized Scoring Operator

$$C(x) = \sum_{i=1}^{q} T_i\left(\sum_{j=1}^{n_i} C_{ij}(x)\right)$$

T_i decreases as i increases

Low satisfaction for higher priority criteria blocks contribution by low priority criteria

Manifests Fundamental Feature of the Prioritization

Poor satisfaction to any higher criteria reduces the ability for compensation by lower priority criteria.

.

Example

- $H_1 = \{C_{11}, C_{12}\}$, $H_2 = \{C_{21}\}$, $H_3 = \{C_{31}, C_{32}, C_{33}\}$

  $H_4 = \{C_{41}, C_{42}\}$
- $C_{11}(x) = 0.7$, $C_{12}(x) = 1$, $C_{21}(x) = 0.9$, $C_{31}(x) = 0.8$

  $C_{32}(x) = 1$, $C_{33}(x) = 0.2$, $C_{41}(x) = 1$, $C_{42}(x) = 0.9$
- $S_1 = \text{Min}[C_{11}(x), C_{12}(x)] = 0.7$

  $S_2 = \text{Min}[C_{21}(x)] = 0.9$

  $S_3 = \text{Min}[C_{31}(x), C_{32}(x), C_{33}(x)] = 0.2$

  $S_4 = \text{Min}[C_{41}(x), C_{42}(x)] = 0.9$
- $T_1 = 1$, $T_2 = S_1T_1 = 0.7$, $T_3 = S_2T_2 = 0.63$

  $T_4 = S_3T_3 = 0.12$
- $w_{11} = w_{12} = 1$, $w_{21} = 0.7$, $w_{31} = w_{32} = w_{33} = 0.63$

  $w_{41} = w_{42} = 0.12$
- $C(x) = \sum w_{ij}C_{ij}(x) = 3.82$

Basic Features of the PS Operator

- Importance weights of a criterion depend on the satisfaction to higher priority criteria
- Lower priority criteria only contribute to the score of alternatives satisfying higher priority criteria
- Lower priority criteria used to distinguish between alternatives satisfying higher priority criteria
- Importance weights will be different across alternatives.

Why have we chosen this scoring type operator rather then an averaging operator which simply requires that we normalize the weights ?

In this case of partial ordering of the criteria (more the one criteria in each category) performing this normalization does not always guarantee a monotonic aggregation

Example 1

- $H_1 = \{C_{11}, C_{12}, C_{13}, C_{14}\}$ $H_2 = \{C_{21}, C_{22}, C_{23}\}$
- $C_{11}(x) = C_{12}(x) = C_{13}(x) = 1, C_{14}(x) = 0$
  $C_{21}(x) = C_{22}(x) = C_{23}(x) = 0.$
- $S_1 = 0$ hence $T_1 = 1$ and $T_2 = 0$.
- $u_{1j} = T_1 =$ and $u_{2j} = T_2 = 0$ and hence $\sum_{ij} u_{ij} = 4$
- Applying Normalization

$w_{1j} = 1/4$ for $j = 1$ to 4 $w_{2j} = 0$ for $j = 1$ to 3

- $C(x) = 1/4(C_{11}(x) + C_{12}(x) + C_{13}(x) + C_{14}(x)) = 0.75$

Example 2

- $H_1 = \{C_{11}, C_{12}, C_{13}, C_{14}\}$ $H_2 = \{C_{21}, C_{22}, C_{23}\}$
- $C_{11}(x) = C_{12}(x) = C_{13}(x) = 1$, **$C_{14}(x) = 1$**
  $C_{21}(x) = C_{22}(x) = C_{23}(x) = 0$.
- **$S_1 = 1$** hence $T_1 = 1$ and **$T_2 = 1$**.
- $u_{1j} = T_1 = 1$ and **$u_{2j} = T_2 = 1$** and hence $\sum_{ij} u_{ij} =$ **7**
- Applying Normalization
  $w_{1j} =$ **1/7** for j = 1 to 4 $w_{2j} =$ **1/7** for j = 1 to 3
- $C(x) =$ **1/7**$(C_{11}(x) + C_{12}(x) + C_{13}(x) + C_{14}(x)) =$ **0.57**
- **$0.57 < 0.75$**

Prioritized Scoring Operator Respects the Monotonicity

For example 1

- $w_{1j} = u_{1j} = 1$ and $w_{2j} = u_{2j} = 0$
- $C(x) = 3$.

For example 2

- $w_{1j} = u_{1j} = 1$ and $w_{2j} = u_{2j} = 1$
- $C(x) = 4$

The monotonicity is respected.

If the priority relationship between the criteria is a linear ordering (one criteria in each category) then we can obtain a monotonic prioritized averaging (PA) operator

Prioritized Averaging

Operators

Problem Formulation

- Collection of criteria partitioned into q distinct categories

  $H_1, H_2, ..., H_q$

- $H_i = \{C_i\}$: **One criteria** in criteria in category H_i.
- A prioritization between these categories

  $C_1 > C_2, ... > C_q$.

- Criteria C_i has higher priority than C_k if $i < k$.

Prioritized Averaging Operators
PA Operator

- Alternative $x \in X$
- $C_i(x) \in [0, 1]$ is x satisfaction to criteria C_i
- C(x) overall score for alternative x
- **P**rioritized **A**veraging (PA) operator

$$C(x) = \sum_{i=1}^{q} w_i C_i(x)$$

The w_i depend on $C_k(x)$ for $k < i$

Determination of Weights

- For category H_i we calculate $S_i = C_i(x)$
- S_i is the value of the least satisfied criteria in category H_i
- $S_0 = 1$ by convention
- Calculate

$$T_i = \prod_{k=1}^{i-1} S_k \qquad (T_3 = S_0 S_1 S_2)$$

$$u_i = T_i \qquad \text{(pre-weights)}$$

$$w_i = \frac{T_i}{T} \qquad T = \sum_i T_i$$

Prioritized Averaging Operator

$$C(x) = \sum_{i=1}^{q} w_i C_i(x)$$

$$w_i = \frac{T_i}{T} \qquad T = \sum_i T_i$$

$$T_1 = 1$$

$$T_i = C_1(x) C_2(x) C_3(x) \ldots C_{i-1}(x) \qquad i > 1$$

Weights decrease as i increases

Lack of satisfaction to higher priority criteria blocks compensation by lower priority criteria

Illustration

$$C_1 > C_2 > C_3 > C_4$$

$$C_1(x) = 1 \quad C_2(x) = 0.5 \quad C_3(x) = 0.2 \quad C_4(x) = 1$$

$$T_1 = 1 \quad T_2 = 1 \quad T_3 = 0.5 \quad T_4 = 0.1$$ **T = 2.6**

$$w_1 = 0.38 \quad w_2 = 0.38 \quad w_3 = 0.2 \quad w_4 = 0.04$$

$$C(x) = (0.38)(1) + (0.38)(0.5) + (0.2)(0.2) + (0.04)(1) = 0.65$$

$$C_1(y) = 0.2 \quad C_2(y) = 0.5 \quad C_3(y) = 1 \quad C_4(y) = 1$$

$$T_1 = 1 \quad T_2 = 0.2 \quad T_3 = 0.1 \quad T_4 = 0.1$$ **T = 1.4**

$$w_1 = 0.72 \quad w_2 = 0.14 \quad w_3 = 0.07 \quad w_4 = 0.07$$

$$C(y) = (0.72)(0.2) + (0.14)(0.5) + (0.07)(1) + (0.07)(1) = 0.35$$

Alternative Determination of S_i

$$H_i = \{C_{i1}, C_{i2}, C_{i3}, \ldots\ldots, C_{in_i}\}$$

S_i is effective satisfaction of criteria in H_i

$$S_i = \mathrm{Min}_j[C_{ij}(x)]$$ (Least satisfied criteria)

$$S_i = \sum_{j=1}^{n_i} \frac{1}{n_i} C_{ij}(x)$$ (Average satisfaction in H_i)

$$S_i = \mathrm{OWA}(C_{i1}(x), C_{i2}(x), C_{i2}(x), \ldots, C_{in_i}(x))$$

REFERENCES

[1]. Yager, R. R., "Lexicographically prioritized multicriteria decisions using scoring functions," Proceedings of International Conference on Information Processing and Management of Uncertainty IPMU, 1438-1445, 2008.

[2]. Yager, R. R., "Lexicographic ordinal OWA aggregation of multiple criteria," Information Fusion 11, 374-380, 2010.

[3]. Yager, R. R., Reformat, M. and Ly, C., "Web PET: An on-line tool for lexicographically choosing purchases," IEEE Intelligent Systems Nov/Dec, 76-83, 2010.

[4]. Yager, R. R., Reformat, M. and Ly, C., "Using a web personal evaluation tool - PET for Lexicographic multi-criteria service selection," Knowledge-Based Systems 24, 929-942, 2011.

Prioritized Aggregation Operators and their Application

Ronald R. Yager
Machine Intelligence Institute
Iona College
New Rochelle, NY 10801
yager@panix.com

Non-Montonic Prioritized Set Operations

Basic Operations on Fuzzy Sets

- Assume A and B are two fuzzy subsets of X.
- Standard intersection operation: $E = A \cap B$

$$E(x) = \mathrm{Min}[A(x), B(x)] = A(x) \wedge B(x).$$

- Set inclusion for fuzzy sets $A \subseteq B$ if $A(x) \leq B(x)$ for all x

- A of X is called normal if there exists at least one $x \in X$ such that $A(x) = 1$.

- Possibility: $\mathrm{Poss}[A/B] = \mathrm{Max}_x[A(x) \wedge B(x)] = \mathrm{Max}_x[E(x)]$

- Certainty: $\mathrm{Cert}[A/B] = 1 - \mathrm{Poss}[\bar{A}/B]$

Properties of Intersection

- **Pointwiseness**: E(x) just depends on A(x) & B(X)
- **Commutativity**: $A \cap B = B \cap A$ (Equivalence of Arguments)
- **Associativity:** $A \cap (B \cap C) = (A \cap B) \cap C$
- **Monotonicity:** If $A_1 \subseteq A_2$ & $B_1 \subseteq B_2$ then $A_1 \cap B_1 \subseteq A_2 \cap B_2$
- $A \cap X = A$ $X \cap A = A$
- $\varnothing \cap A = \varnothing$ $A \cap \varnothing = \varnothing$
- $A \cap B \subseteq A$ $A \cap B \subseteq B$
- If $A \subseteq B$ then $A \cap B = A$ If $B \subseteq A$ then $A \cap B = B$

Significant Observation About Intersection

- Conflict between A and B results in subnormal E
- The larger the conflict the smaller E
- **Essentially conflict results in null intersection**
- 1 - Poss[A/B] measures degree of conflict

Non-Monotonic Intersection Operator

- Assume A and B are two fuzzy subsets of X.
- Define the non-monotonic intersection operator η

$$\eta(A, B) = D$$

- Here D is fuzzy subset of X such that

$$D(x) = ((1 - Poss[A/B]) \wedge A(x)) \vee (A(x) \wedge B(x))$$

- We can also express η as

$$D(x) = A(x) \wedge (B(x) \vee (1 - Poss[A/B]))$$

Essential Workings of $\eta(A, B)$

- Test whether A and B are not conflicting

- If a non-null intersection then this non-null

intersection becomes $\eta(A, B)$.

- If A and B don't intersect then we obtain A.

Important Distinction Between Standard Intersection and the η Intersection

- If $F = A \cap B$ then $F(x) = A(x) \wedge B(x)$
- $F(x)$ just depends upon $A(x)$ and $B(x)$

$\cap$ is a pointwise operator

- If $E = \eta(A, B)$ then

$$E(x) = ((1 - Poss[A/B]) \wedge A(x)) \vee A(x) \wedge B(x))$$

- $E(x)$ depends upon the whole sets A and B

η is not a pointwise operator

η is non-monotonic

Example: $A = \{x_1, x_2, x_3\}$ and $B = \{x_5\}$

Consider $\eta(A, B)$

Since Poss[A/B] = 0 we get

$$\eta(A, B) = A = \{x_1, x_2, x_3\}$$

Example: Consider $G = \{x_1, x_2, x_3, x_4\}$ and $H = \{x_1, x_5\}$.

We note that

$$A \subseteq G \text{ and } B \subseteq H$$

Consider $\eta(G, H)$.

Since Poss[G/H] = 1 then

$$\eta(G, H) = G \cap H = \{x_1\}$$

We see that $\eta(G, H) \subset \eta(A, B)$

The η operator is **Not** commutative,

$$\eta(A, B) \neq \eta\ (B, A).$$

Example:

$A = \{x_1, x_2, x_3\}$

$B = \{x_4, x_5\}$.

Poss[A/B] = 0

$$\eta(A, B)\ (x) = A(x) \vee (A(x) \wedge B(x)) = A(x).$$

$$\eta\ (B, A)\ (x) = B(x) \vee (A(x) \wedge B(x)) = B(x)$$

- Ordering of the arguments in η operator is of significance.
- There is distinction between the arguments
- In the face of conflict we revert to A
- In $\eta(A, B)$ we say that A has ***priority*** over B.
- Call A the primary argument and B the secondary argument

.

The non-monotonic intersection admits a concept of priority amongst its arguments.

Interaction with X and Null Set

- $\eta(A, X) = A$
- $\eta(X, A) = A$ if A is normal

  $\eta(X, A)(x) = (1 - \text{Max}_x[A(x)]) \vee A(x)$ if A is subnormal

- $\eta(\varnothing, A) = \varnothing$
- $\eta(A, \varnothing) = A$

- $\eta(A, B) = \varnothing$ iff $A = \varnothing$

Observations About Containment

- $\eta(A, B) \not\subset A$
- If Poss[A/B] = α, then $\eta(A, B)(x) \leq B(x)$ for all x where $B(x) \geq 1 - \alpha$

  In particular if $\alpha = 1$, then $\eta(A, B) \subseteq B$.

- If $A \subseteq B$ then $\eta(A, B) = A$
- If $B \subseteq A$ then $\eta(A, B)(x) = (1 - ß) \wedge A(x) \vee B(x)$, where $Max_x[B(x)] = ß$
- When $B \subseteq A$ the closer B is to being normal the closer $\eta(A, B)$ is to B while the further B is from normal the closer $\eta(A, B)$ is to A.

Idempotency is satisfied by this operation

$\eta(A, A)(x) = ((1 - Poss[A/A]) \wedge A(x) \vee (A(x) \wedge A(x)) = A(x).$

η Operator is Not Associative

Procedure for Aggregation Depends upon the Priorities

- A > B to indicate A has priority over B
- A $\equiv$ B to indicate A and B are of equal priority

The following are the structurally distinct manifestations of priorities among three elements

1. $A > B > C$ $\eta(A, B, C)$
2. $A \equiv B \equiv C$ $\eta((A, B, C))$
3. $A > B, B \equiv C$ $\eta(A, (B, C))$
4. $A \equiv B, B > C$ $\eta((A, B), C)$

A > B > C

Definition: $\eta(A, B, C) = \eta(\eta(A, B), C)$

More generally if $A_1 > A_2 > \ldots, > A_q$ we denote this as

$$\eta(A_1, A_2, \ldots, A_q)$$

and define it

$$\eta(\eta(\eta, \ldots, (\eta(A_1, A_2), A_3), \ldots, A_q)$$

if we let D_j be the result after including the j^{th} term then

$D_1 = A_1$

$D_j(x) = ((1 - \text{Poss}[D_{j-1}/A_j]) \wedge D_{j-1}(x)) \vee (D_{j-1}(x) \wedge A_j(x)) \qquad j > 1.$

$A \equiv B \equiv C$

- $\eta((A, B, C)) = D$
- $D = A \cap B \cap C$
- Implement as the standard intersection

$A > B, B \equiv C$

- $D = \eta(A, (B, C))$
- $D(x) = ((1 - \text{Poss}[B/A]) \wedge (1 - \text{Poss}[C/A]) \wedge A(x))$

$$\vee ((1 - \text{Poss}[B/(C \cap A)]) \wedge C(x) \wedge A(x))$$
$$\vee ((1 - \text{Poss}[C/(B \cap A)]) \wedge B(x) \wedge A(x))$$
$$\vee (A(x) \wedge B(x) \wedge C(x))$$

- **Aggregation Imperative**

We first try to use the intersection of all three, $A \cap B \cap C$.

If this results in conflict we then try to use the intersection of

A with either B or C, $(A \cap B) \cup (A \cap C)$

if both these fail then we use A.

$A \equiv B, B > C$

- $D = \eta((A, B), C)$
- $D(x) = ((1 - Poss[C/A \cap B]) \wedge A(x) \wedge B(x))$
  $\vee (A(x) \wedge B(x) \wedge C(x))$.

- **Aggregation Imperative**

We first try to use the intersection of all three, $A \cap B \cap C$.

If this results in conflict we then we use the intersection of A and B

A complete generalized approach is available in

Yager, R. R., "Non-monotonic set theoretic operations," Fuzzy Sets and Systems 42, 173-190, 1991

SOME APPLICATIONS

Information Fusion and Reasoning

- Conjunction or intersection plays a central role in the aggregation of information.

- Given a piece of information I_1. If we receive a second piece of information I_2 then the combined information is the intersection of these two pieces of information.

- If the two pieces of information conflict we left, as a result of their conjunction, with an inconsistency, the null set.

Commonsense or Default reasoning

- We start with a piece of knowledge called an ***assumption or default value***, I_3.
- If we receive a piece of information, I_4, we must combine this with our default value.
- In this case if there exists a conflict rather then being left with the inconsistency we realize that the original piece of information, I_3, was only an assumption and withdraw it in the face of conflict.
- Here then information I_4 has priority over I_3

Example of Commonsense Default Logic

- For example one could assume that a typical college student is in their twenties.

- If we find out that Mary is a college student who is fifty years old we want to withdraw the conjecture that she is in her twenties

- However if we get no other information about Mary's age we use the default value

Multi-Criteria Decision Making

Set of criteria: A_1 A_2 , ………, A_q

Priority of A_j over A_{j+1}

if $A_1 > A_2 > . . . . . , > A_q$ we denote this as

$$\eta(A_1, A_2, ... , A_q)$$

Information Retrieval

Modification of η operation.

- Consider

$$\eta(A, B; \alpha) = A(x) \wedge (B(x) \vee ((1 - Poss[A/B]) \wedge \alpha))$$

- When $\alpha = 0$ we get a non prioritized standard intersection

$$\eta(A, B; 0) = A \cap B$$

- While when $\alpha = 1$ we get the prioritized intersection

$$\eta(A, B; 1) = \eta(A, B)$$

- View α as specifying some degree of priority.
- When $\alpha = 0$, A and B are of equal priority
- When $\alpha = 1$, A has complete priority over B.
- Intermediate values indicate degrees of priority.

Application to Belief Revision

- View E = η(A, B) as a kind of Belief revision operation
- Using η(A, B) we are having A revised by B.

$$E(x) = A(x) \wedge (B(x) \vee (1 - Poss[A/B]))$$

- Allow A to be revised if B doesn't conflict too strongly with it.
- If degree of conflict (1 – Poss[A/B]) = 1 then no revision occurs
- If conflict equals zero maximal revision occurs.
- Viewing η(A, B; α) in this perspective we can see that α determines how strongly A is held.

Non-Monotonic Union Operator

- $\psi(A, B) = E$
- E is a fuzzy subset of X such that

$$E(x) = (Poss(\neg A/\neg B) \vee A(x)) \wedge (A(x) \vee B(x))$$

- Alternatively we can write it as

$$E(x) = A(x) \vee (B(x) \wedge Poss(\neg A/\neg B))$$

- We note that ψ and η are related by De Morgans Law.

$$\psi(A, B)(x) = (1 - \eta(\neg A/\neg B)(x))$$

The End