

Appendice C

La teoria del ragionamento sfumato

La Fuzzy Logic è una generalizzazione della logica tradizionale, per intenderci, quella fondata sul Vero/Falso, On/Off, 0/1. Nella logica fuzzy gli insiemi cessano di essere chiusi ed un generico elemento può contemporaneamente appartenere (con diverso grado di verità o probabilità) a più insiemi. Vediamone in dettaglio il funzionamento.

La teoria degli insiemi sfumati o fuzzy è stata introdotta nel 1965 da L. Zadeh e si basa sulla generalizzazione della funzione di appartenenza di un elemento ad un insieme, che invece di essere la funzione binaria della teoria degli insiemi classici, viene assunta essere una funzione qualsiasi. In tal modo risultano trattabili in maniera formale anche i concetti sfumati (come "caldo", "freddo", "alto", "medio", ecc.) che gli esseri umani trattano continuamente, al fine di prendere decisioni. La logica fuzzy formalizza il ragionamento "approssimato" utilizzando gli insiemi sfumati. La tecnica utilizzata per realizzare un sistema fuzzy si basa su regole del tipo "if...then...else", le stesse su cui si basano i sistemi decisionali o più in generale quei sistemi basati sulla conoscenza (Knowledge Based Processing) come i cosiddetti Sistemi Esperti. La base di conoscenza di un sistema esperto è come una specie di database che anziché essere composto di dati è composto di relazioni tra i dati intese come associazioni di conseguenze determinate da particolari condizioni di ipotesi: tutto ciò viene chiaramente espresso nelle regole tipo "if then else". La determinazione delle regole è fatta in base alla conoscenza del sistema fisico e attraverso apprendimento sui dati disponibili. Il programma che "aziona le regole" contenute nel "knowledge base" viene spesso chiamato motore inferenziale o "reasoning engine" e può essere basato su diverse tecniche di supporto alla decisione. In Fig. 1 potete vedere un tipico protocollo di ragionamento per un sistema esperto ibrido con la estensione delle ultime tecnologie come reti neurali e algoritmi genetici.

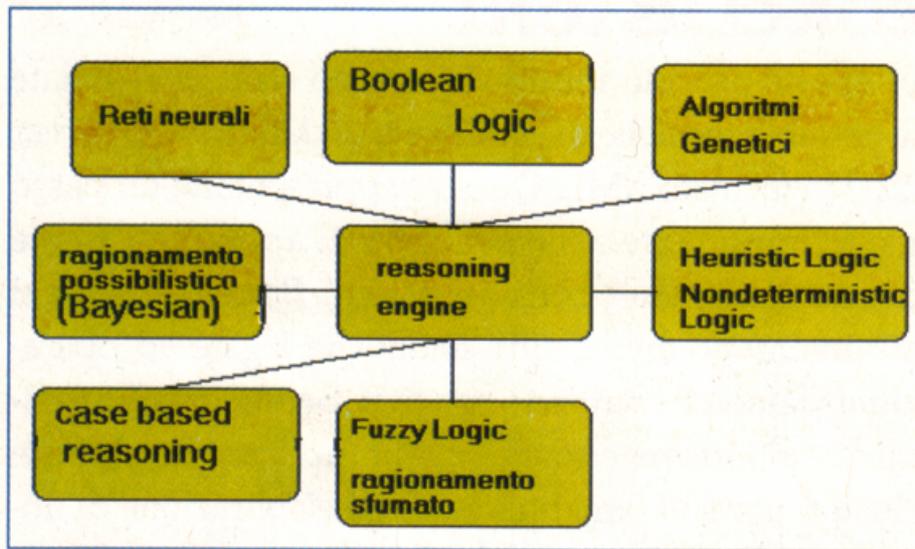


Fig. 1: Protocollo di Ragionamento per un Sistema Esperto Ibrido.

Come si può immaginare, un sistema esperto può essere basato anche solamente su regole fuzzy e quindi è il caso di capire cosa significa ragionamento fuzzy o sfumato. Se prendiamo in esame una regola del tipo ¹ "if (acqua bolle) then (butto la pasta)", non vi sono sicuramente incertezze a decidere quando buttare la pasta dato che (acqua bolle) è sicuramente vero o sicuramente falso. Se prendiamo in esame invece la regola "if (acqua è calda) then (abbassa la fiamma)" ci troviamo di fronte a dati incerti: che cosa significa calda? E quanto devo abbassare la fiamma? Supponendo di poter misurare la temperatura con un termometro, come faremo a decidere se quel valore rientrerà nella definizione "calda" Dovremo iniziare a definire un range di valori in cui la temperatura può variare e dei subranges che rappresentino definizioni come "calda" o "molto calda". Appare quindi evidente che non sono i dati input a essere imprecisi ma bensì le relazioni che esistono tra input e output: nei problemi trattati con fuzzy logic infatti non esiste un facile modello matematico che collega input e output. In pratica con un motore inferenziale fuzzy si può empiricamente sintonizzare una funzione matematica che vincola input e output, tramite regole tipo "if...then...".

Fuzzificazione dell' input

Il ragionamento cosiddetto sfumato dei sistemi fuzzy è dovuto al fatto che un'ipotesi non è mai completamente vera ne completamente falsa ma ha un suo "grado di verità" che inciderà sulla forza con cui verrà eseguita la regola e quindi applicata la conseguenza. Il grado di verità dell'ipotesi è in realtà il grado di appartenenza del valore di input a quel determinato range (es: calda) e ciò è dovuto al fatto che questi ranges non sono nettamente separati ma

¹ Una Variabile Linguistica è una quintupla (X,T(X),U,G,M_s), in cui X è il nome della variabile, T(X) è il suo insieme di definizione, cioè dei termini associabili alla variabile X, U è l'universo del discorso, G è la grammatica da usare per generare i nomi ed M è l'insieme delle regole semantiche che associano ad ogni valore di X il suo significato.

si sovrappongono con funzioni di appartenenza normalmente di tipo triangolare/trapezoidale, per cui un valore di una variabile di input potrebbe appartenere a due ranges con due differenti gradi di credibilità (Fig. 2). Il valore 25 *gradi* della temperatura appartiene alla classe tiepida con un degree of membership (grado di appartenenza) pari a 0.8 e alla classe fredda con un grado di appartenenza uguale 0.4. Come vedete, definendo delle classi triangolari o trapezoidali (le più usate) si hanno dei gradi di appartenenza che diminuiscono verso gli estremi delle classi costruendo così un inizio di quello che è un ragionamento sfumato. Quello che rappresenta la Fig. 2 è la operazione che viene comunemente definita "FUZZIFICAZIONE DELL'INPUT", cioè trasformazione del dato preciso in dato tipo fuzzy:

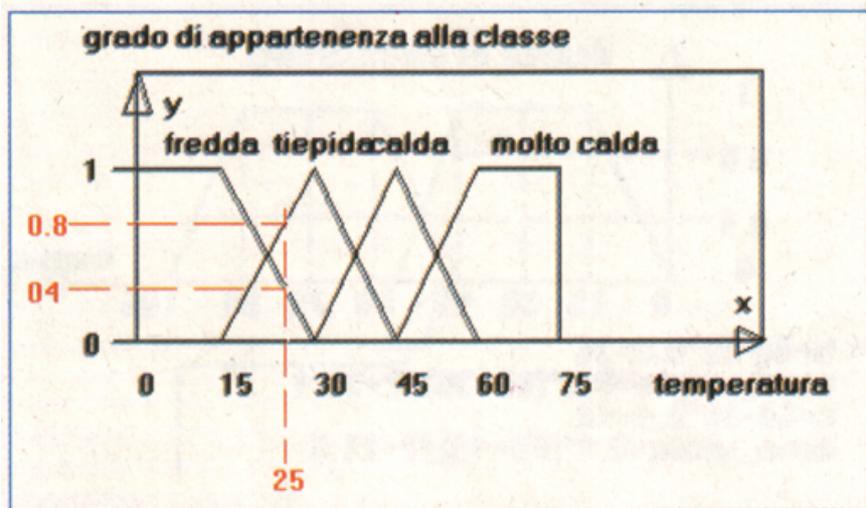


Fig. 2: Esempio di diverso grado di appartenenza di una variabile di input a due classi.

dato preciso

- *temp=25 gradi centigradi dato fuzzy*
- *temp = tiepida con credibilità 0.8*
- *temp = fredda con credibilità 0.4*

Per realizzare matematicamente il processo di fuzzificazione bisogna effettuare alcuni semplici calcoli sulle funzioni triangolari/trapezoidali. Il primo passo da fare nel processo di fuzzificazione è uno scanning dei valori estremi di ogni classe per cui quando si ha che $limite_inf_class < x < limite_sup_class$ allora x appartiene alla classe e bisogna calcolare il grado di appartenenza. Bisogna definire uno "slope" cioè una pendenza, sui lati dei triangoli o dei trapezi che definiscono le classi, che sia fisso per ogni classe e che ci consenta di calcolare il grado di appartenenza di un input x alla classe y . A questo punto bisogna vedere se il valore x cade nella parte bassa della classe (pendenza positiva) o nella parte alta (pendenza negativa): per far questo è sufficiente verificare se: $x > (lim_sup - lim_inf)/2$ oppure $x < (lim_sup - lim_inf)/2$ che rappresenta il centro della classe. Nel primo caso si effettuerà il calcolo con lo slope positivo e nel secondo con lo slope negativo (Fig. 1).

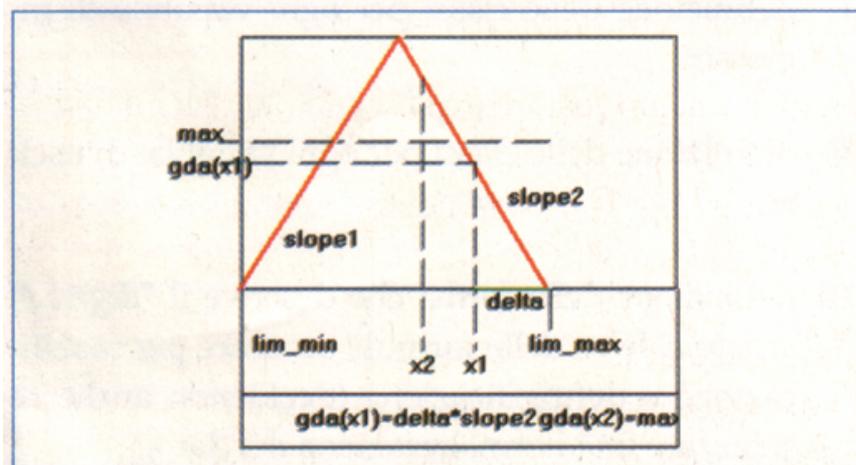


Fig. 3: Il grado di appartenenza ad una classe si calcola mettendo in relazione la variabile di input con la pendenza della classe (FUZZIFICAZIONE).

1. $gda = (x - lim_inf) * slope$
(se $gda > max$ allora $gda = max$);
2. $gda = (lim_sup - x) * slope$
(se $gda > max$ allora $gda = max$);

(*gda=grado di appartenenza o "degree of membership"*)

Bisogna precisare che questo è solamente un possibile approccio alla soluzione del problema di fuzzificazione, infatti è possibile implementare funzioni di qualunque tipo per definire il grado di appartenenza di un input ad una classe; si tratta solamente di calcolare il valore di una funzione in un particolare punto definito dal valore dell'input. Quello che va tenuto presente è che, normalmente, il range dei valori di appartenenza deve essere normalizzato tra il valore 0 ed il valore 1. Ciò previene evidenti problemi tra variabili del sistema che hanno ranges di valori completamente differenti e sarebbe comunque auspicabile una soluzione che preveda la normalizzazione delle variabili di input su scala [0, 1]. Rimanendo comunque nelle classiche funzioni trapezoidali/triangolari si possono effettuare moltissime varianti. Ad esempio si può fare in modo che la pendenza (slope) dei triangoli sia differenziabile per ogni classe e all'interno di ogni classe quella di salita da quella di discesa. Si può anche effettuare il calcolo di fuzzificazione in modo differente senza calcolare il centro della classe (Fig. 4).

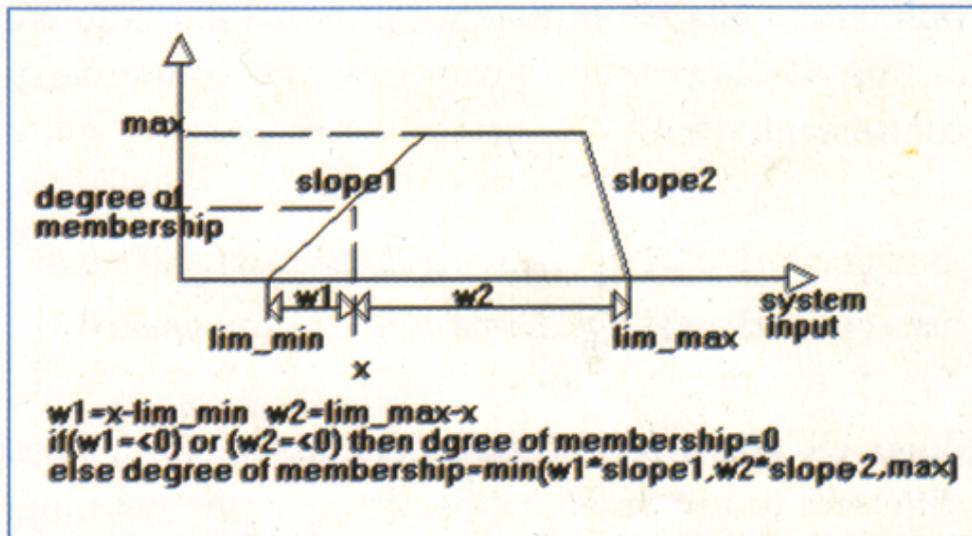


Fig. 4: Si possono definire classi sempre più complesse, ma aumenterà la quantità di informazione da memorizzare.

Risulta evidente che quanto più complesse sono le forme delle funzioni scelte per definire le classi, tanto maggiore è la quantità di informazione che deve essere memorizzata per definire una classe e la quantità di calcoli da effettuare per ottenere un valore di "membership degree". Ad esempio la semplice scelta di differenziare le pendenze dei lati dei triangoli o dei trapezi obbliga a memorizzare tale informazione per ogni classe presente.

La valutazione delle regole

Quando siamo in possesso di dati fuzzy provenienti dal processo di fuzzificazione dobbiamo inserire nel motore decisionale delle regole che ci diano degli output fuzzy particolari per particolari situazioni (fuzzy) degli input. Una di queste regole può avere la forma: "if (input n appartiene a classe k) then (output m appartiene a classe j con forza pari al grado di appartenenza di n a k)". Spesso nella applicazione delle regole alcune di esse portano alla medesima conseguenza con livelli di forza differenti: in questi casi è pratica comune scegliere il valore maggiore. Avere dei dati fuzzy in uscita ci servirebbe a ben poco però, perciò si rende necessario trasformare i dati che derivano dalla valutazione delle regole in dati numerici reali: questo processo è il processo opposto alla fuzzificazione dell'input e infatti si chiama defuzzificazione dell'output. In Fig. 5 vedete rappresentato lo schema di un sistema fuzzy completo.

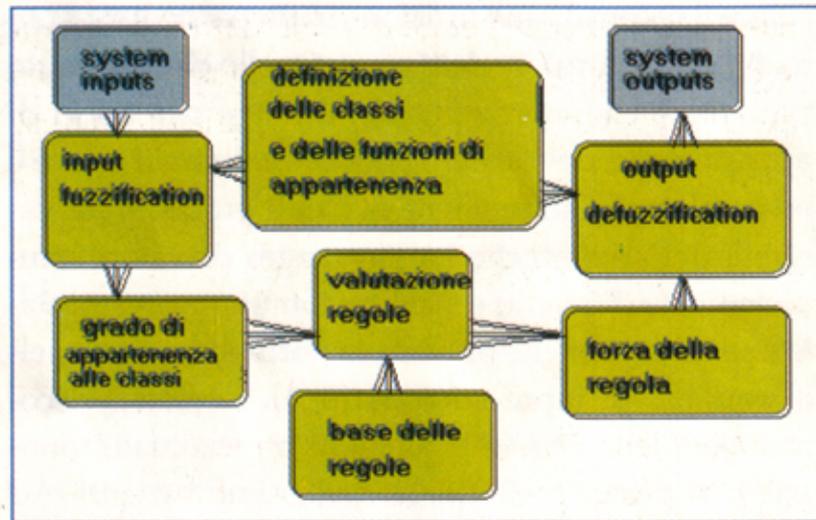


Fig. 5: Schema completo di un sistema Fuzzy.

Defuzzificazione dell'output

Una volta che abbiamo fuzzificato gli input e abbiamo fornito le informazioni in formato fuzzy al motore decisionale del nostro sistema (il programma che fa girare le regole) e che questo ci ha fornito in risposta degli output in formato fuzzy del tipo "output 3 appartiene a classe 2 con grado 0.6", dobbiamo trovare un sistema che ci consenta di estrarre dati numerici precisi da applicare alle uscite. Questa operazione, detta defuzzificazione dell'output, deve risolvere il problema della conflittualità che nasce dal fatto che alcune regole possono avere generato conseguenze contrastanti tipo:

conseguenza 1: olit 1 appartiene a classe 2 con grado 0.6

conseguenza 2: out 1 appartiene a classe 3 con grado 0.3

dove, cioè lo stesso output viene assegnato a classi differenti (normalmente adiacenti o vicine nel range della variabile). Viene comunemente utilizzato un metodo chiamato "center of graviti/ method" che consiste nel calcolare le aree "attive" dei trapezi o triangoli che definiscono le classi e, una volta posizionato il valore a metà tra i centri delle classi in conflitto, farlo "attrarre" dai rispettivi centri di gravita in modo proporzionale alle aree attive associate, come mostrato in Fig. 6.

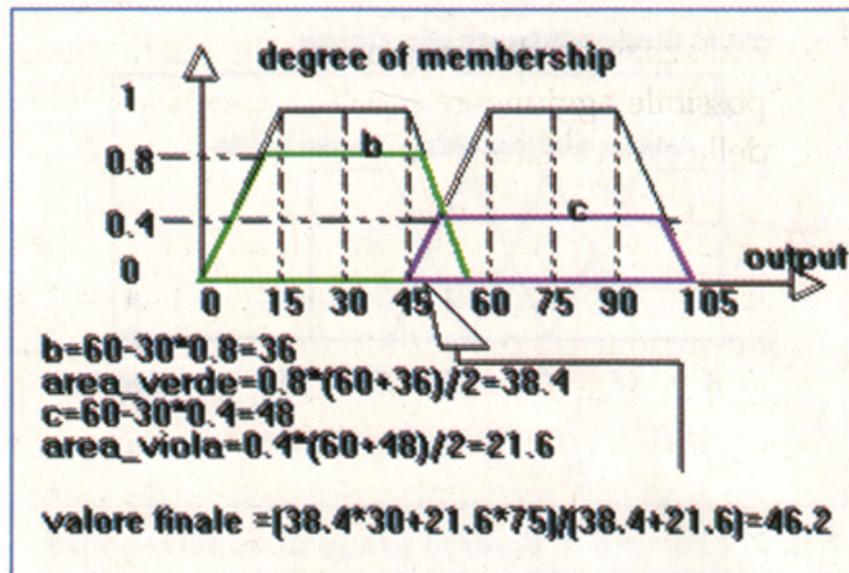


Fig. 6: Il metodo del Centro di Gravità ci consente di ottenere un output DEFUZZIFICATO.

Una possibile semplificazione può essere quella di prendere i centri di tutte le classi in conflitto e far attrarre il valore numerico finale da tutti i centri in modo proporzionale alla forza delle regole che hanno determinato le conseguenze di appartenenza ad ogni classe:

$$valore\ finale = (f_1*c_1 + f_2*c_2 + f_3*c_3... + f_n*c_n)/(f_1 + f_2 + f_3... + f_n)$$

dove c_n rappresenta il centro numerico della classe n ed f_n è il grado di appartenenza (o forza della regola).