

## COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

13 Settembre 2007 (Punteggio su 30/30; Tempo 2h )

### Esercizio 1 (punti 8)

Dato il seguente training set S:

Distanza	Uguali	Stessa entità
1	Si	Si
2	No	Si
3	No	No
1	?	Si
2	Si	No
2	No	Si
3	Si	No
2	No	No
1	Si	Si
1	No	No
2	No	Si
?	Si	No
3	Si	Si
3	No	No
3	Si	No
1	Si	Si
3	?	No

- Si calcoli l'entropia del training set rispetto all'attributo Rilevante
- Si calcoli il rapporto di guadagno dei due attributi rispetto a questi esempi di training
- si costruisca un albero decisionale ad un solo livello per il training set dato, indicando le etichette delle foglie (numero di esempi finiti nella foglia/numero di esempi finiti nella foglia non appartenenti alla classe della foglia).
- si classifichi l'istanza:

?	Si
---	----

### Esercizio 2 (punti 8)

Si devono definire i turni per 10 infermieri su 6 giorni. In ogni giorno i turni sono Mattina, Pomeriggio, Notte

- I turni di mattina e pomeriggio devono essere coperti da 2 infermieri, mentre per il turno di notte deve essere presente solo 1 infermiere.
- Ogni infermiere non può fare due turni nello stesso giorno
- Se un infermiere fa il turno di notte non può fare nessun turno nel giorno successivo.
- L'infermiere 3 è in ferie i primi due giorni
- L'infermiere 5 non può fare più di 4 turni nei 6 giorni.
- Tutti gli infermieri devono comunque avere almeno un turno.

Si modelli il problema a vincoli usando possibilmente vincoli globali. Si identifichino alcune simmetrie del problema e si delinei un metodo per eliminarle.

### Esercizio 3 (punti 8)

Si consideri il seguente problema: un carrello ('c') è nella posizione '2' con dei sacchi di cemento ('s') caricati. In posizione '3' si trovano dei mattoni ('m'). La posizione '2' e la posizione '3' sono connesse ad '1'.

Il goal che si vuole raggiungere è che il carrello sia in '1' e mattoni e cemento scaricati in posizione '1', avendo a disposizione le seguenti azioni

Scaricamento materiale

**unload(C,M)**

**PREC: on(C,M), in(C,X)**

**EFFECT: ¬on(C,M), in(M,X)**

Caricamento materiale

**load(C,M)**

**PREC: in(C,X), in(M,X)**

**EFFECT: on(C,M), ¬in(M,X)**

Spostamento del carrello tra due posizioni connesse

**move(C,Loc1,Loc2)**

**PREC: in(C,Loc1), connected(Loc1,Loc2)**

**EFFECT: in(C,Loc2), ¬in(C,Loc1),**

Stato iniziale:

**in(c,2), connected(1,2), connected(2,1), connected(1,3), connected(3,1),  
in(s,2), on(c,s), in(m,3)**

Stato goal: **in(c,1), in(s,1), in(m,1)**

Si mostrino i passi compiuti dall'algoritmo STRIPS per risolvere il problema.

### Esercizio 4

- Spiegare brevemente le principali differenze tra la pianificazione automatica lineare alla STRIP e il Partial Order Planning.
- Si calcoli la least general generalization delle seguenti clausole:  
C1={p(a,f(a,c)), ~ p(a,f(a,d)), ~ p(b,f(b,c)), ~ q(a,b)}.  
C2={p(a, g(a, c)), p(b, f(b,c)), ~ p(a,f(b,c)),~ p(a, b, c)}.
- Si descriva cosa è Inductive Logic Programming
- Cosa è una rete neurale multistrato e in quali applicazioni può essere utilizzata.
- Si descriva il vincolo cumulative e si dia un esempio del suo uso.

## SOLUZIONE

### Esercizio 1:

a)  $\text{info}(S) = -8/17 * \log_2 8/17 - 9/17 * \log_2 9/17 = 0.998$  8 #pos, 9 #neg, 17 #tot)

b)

Per calcolare il guadagno dell'attributo Distanza non si usa l'entropia calcolata su tutto il training set ma solo sugli esempi che hanno Distanza noto (insieme F):

$$\text{info}(F) = -8/16 * \log_2 8/16 - 8/16 * \log_2 8/16 = 1$$

$$\text{info}_{\text{Distanza}}(F) = 5/16 * (-4/5 * \log_2 4/5 - 1/5 * \log_2 1/5) + 5/16 * (-3/5 * \log_2 3/5 - 2/5 * \log_2 2/5) + 6/16 * (-1/6 * \log_2 1/6 - 5/6 * \log_2 5/6) = // \text{tre entropie calcolate sui tre sottoinsiemi con diverso valore dell'attr. Distanza} \\ = 0.312 * 0.722 + 0.312 * 0.971 + 0.375 * 0.650 = 0.772$$

$$\text{gain}(\text{Distanza}) = 16/17 * (1 - 0.772) = 0.215$$

Entropia del partizionamento:

$$\text{splitinfo}(\text{Distanza}) = -5/17 * \log_2(5/17) - 5/17 * \log_2(5/17) - 6/17 * \log_2(6/17) - 1/17 * \log_2(1/17) = 1.809$$

// calcolato sui tre sottoinsiemi con Diverso valore Distanza e il sottoinsieme con Distanza non noto

$$\text{gainratio}(\text{Distanza}) = 0.215 / 1.809 = 0.119$$

Per calcolare il guadagno dell'attributo Uguali non si usa l'entropia calcolata su tutto il training set ma solo sugli esempi che hanno Uguali noto (insieme F):

$$\text{info}(F) = -7/15 * \log_2 7/15 - 8/15 * \log_2 8/15 = 0.997$$

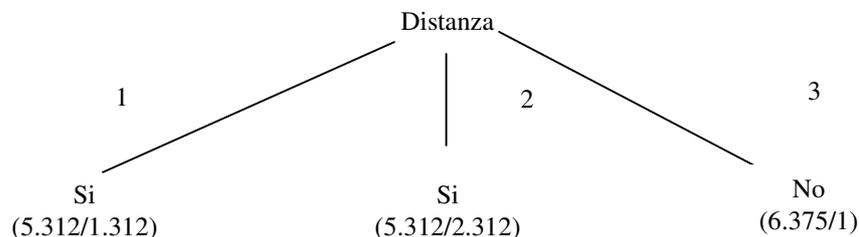
$$\text{info}_{\text{Uguali}}(F) = 8/15 * (-4/8 * \log_2 4/8 - 4/8 * \log_2 4/8) + 7/15 * (-3/7 * \log_2 3/7 - 4/7 * \log_2 4/7) = \\ = 0.533 * 1 + 0.467 * 0.985 = 0.993$$

$$\text{gain}(\text{Uguali}) = 15/17 * (0.997 - 0.993) = 0.004$$

$$\text{splitinfo}(\text{Uguali}) = -7/17 * \log_2(7/17) - 8/17 * \log_2(8/17) - 2/17 * \log_2(2/17) = 1.402$$

$$\text{gainratio}(\text{Uguali}) = 0.004 / 1.402 = 0.003$$

c) L'attributo scelto per la radice dell'albero è Distanza (maggiore gainratio).



d) l'istanza viene divisa in tre parti, di peso rispettivamente  $5.312/17=0.312$ ,  $5.312/17=0.312$  e  $6.375/17=0.375$ . La prima parte viene mandata lungo il ramo 1 e viene classificata come Si con probabilità  $4/5.312=75.3\%$  e come No con probabilità  $1-75.3\%=24.7\%$ . La seconda parte viene mandata lungo il ramo 2 e viene classificata come Si con probabilità  $3/5.312=56.5\%$  e come No con probabilità  $1-56.5\%=43.5\%$ . La terza parte viene mandata lungo il ramo 3 e viene classificata come No con probabilità  $5.375/6.375=84.3\%$  e come Si con probabilità  $1-84.3\%=15.7\%$ . Quindi in totale la classificazione dell'istanza è

$$\text{Si: } 0.312 * 75.3\% + 0.312 * 56.5\% + 0.375 * 15.7\% = 47.0\%$$

$$\text{No: } 0.312 * 24.7\% + 0.312 * 43.5\% + 0.375 * 84.3\% = 52.9\%$$

## ESERCIZIO 2

$M_{i1}$  primo infermiere di turno la mattina del giorno  $i$  per  $i = 1..6$

$M_{i2}$  secondo infermiere di turno la mattina del giorno  $i$  per  $i = 1..6$

$P_{i1}$  primo infermiere di turno il pomeriggio del giorno  $i$  per  $i = 1..6$

$P_{i2}$  secondo infermiere di turno il pomeriggio del giorno  $i$  per  $i = 1..6$

$N_i$  infermiere di turno la notte del giorno  $i$  per  $i = 1..6$

Inizialmente i domini sono  $[1..10]$

Ogni infermiere non può fare due turni nello stesso giorno

$\text{alldifferent}([M_{i1}, M_{i2}, P_{i1}, P_{i2}, N_i])$  per ogni  $i = 1..6$

Se un infermiere fa il turno di notte non può fare nessun turno nel giorno successivo.

$\text{alldifferent}([N_{i-1}, M_{i1}, M_{i2}, P_{i1}, P_{i2}, N_i])$  per ogni  $i = 2..6$

Questo vincolo sussume il precedente che deve essere imposto solo per il primo giorno.

L'infermiere 3 è in ferie i primi due giorni

$M_{i1} \neq 3, M_{i2} \neq 3, P_{i1} \neq 3, P_{i2} \neq 3, N_i \neq 3$  per  $i = 1,2$

L'infermiere 5 non può fare più di 4 turni nei 6 giorni.

$\text{gcc}([M_{11}, M_{12}, P_{11}, P_{12}, N_1, M_{21}, M_{22}, P_{21}, P_{22}, N_2, \dots, M_{61}, M_{62}, P_{61}, P_{62}, N_6], 5, 1,4)$

Tutti gli infermieri devono comunque avere almeno un turno.

$\text{gcc}([M_{11}, M_{12}, P_{11}, P_{12}, N_1, M_{21}, M_{22}, P_{21}, P_{22}, N_2, \dots, M_{61}, M_{62}, P_{61}, P_{62}, N_6], j, 1,6)$  per ogni  $j = 1..10$

### Esercizio 3

GOAL STACK	STATO
in(c,1), in(s,1), in(m,1)	in(c,2), connected(1,2), connected(2,1), connected(1,3), connected(3,1), on(c,s), in(m,3)
in(c,1) in(s,1) in(m,1) in(c,1), in(s,1), in(m,1)	Goal regression con l'azione: <b>move(c, X, 1)</b>
in(c,X), connected(X,1) <b>move(c, X, 1)</b> in(s,1) in(m,1) in(c,1), in(s,1), in(m,1)	Goal affiorante vero nello stato corrente con X legato a 2
<b>move(c, 2, 1)</b> in(s,1) in(m,1) in(c,1), in(s,1), in(m,1)	Esecuzione dell'azione: <b>move(c, 2, 1)</b>
in(s,1) in(m,1) in(c,1), in(s,1), in(m,1)	in(c,1), connected(1,2), connected(2,1), connected(1,3), connected(3,1), on(c,s), in(m,3)
in(s,1) in(m,1) in(c,1), in(s,1), in(m,1)	Goal regression con l'azione: <b>unload(c, s)</b>
on(c,s), in(c,1) <b>unload(c, s)</b> in(m,1) in(c,1), in(s,1), in(m,1)	Goal affiorante vero nello stato corrente
<b>unload(c, s)</b> in(m,1) in(c,1), in(s,1), in(m,1)	Esecuzione dell'azione: <b>unload(c, s)</b>
in(m,1) in(c,1), in(s,1), in(m,1)	in(c,1), connected(1,2), connected(2,1), connected(1,3), connected(3,1), in(s,1), in(m,3)
in(m,1) in(c,1), in(s,1), in(m,1)	Goal regression con l'azione: <b>unload(c, m)</b>
on(c,m), in(c,1) <b>unload(c, m)</b> in(m,1) in(c,1), in(s,1), in(m,1)	
on(c,m) in(c,1) on(c,m), in(c,1) <b>unload(c, m)</b> in(m,1) in(c,1), in(s,1), in(m,1)	Goal regression con l'azione: <b>load(c, m)</b>

<p> <code>in(c,X), in(m,X)</code>  <b>load(c,m)</b>  <code>in(c,1)</code>  <code>on(c,m), in(c,1)</code>  <b>unload(c,m)</b>  <code>in(m,1)</code>  <code>in(c,1), in(s,1), in(m,1)</code> </p>	
<p> <code>in(m,X)</code>  <code>in(c,X)</code>  <code>in(c,X), in(m,X)</code>  <b>load(c,m)</b>  <code>in(c,1)</code>  <code>on(c,m), in(c,1)</code>  <b>unload(c,m)</b>  <code>in(m,1)</code>  <code>in(c,1), in(s,1), in(m,1)</code> </p>	<p>Goal affiorante vero nello stato corrente per X/3</p>
<p> <code>in(c,3)</code>  <code>in(c,3), in(m,3)</code>  <b>load(c,m)</b>  <code>in(c,1)</code>  <code>on(c,m), in(c,1)</code>  <b>unload(c,m)</b>  <code>in(m,1)</code>  <code>in(c,1), in(s,1), in(m,1)</code> </p>	<p>Goal regression con l'azione:  <b>move(c, Loc, 3)</b></p>
<p> <code>in(c,Loc), connected(Loc,3)</code>  <b>move(c, Loc, 3)</b>  <code>in(c,3)</code>  <code>in(c,3), in(m,3)</code>  <b>load(c,m)</b>  <code>in(c,1)</code>  <code>on(c,m), in(c,1)</code>  <b>unload(c,m)</b>  <code>in(m,1)</code>  <code>in(c,1), in(s,1), in(m,1)</code> </p>	<p>Goal affiorante vero nello stato corrente per Loc/1</p>
<p> <b>move(c, 1, 3)</b>  <code>in(c,3)</code>  <code>in(c,3), in(m,3)</code>  <b>load(c,m)</b>  <code>in(c,1)</code>  <code>on(c,m), in(c,1)</code>  <b>unload(c,m)</b>  <code>in(m,1)</code>  <code>in(c,1), in(s,1), in(m,1)</code> </p>	<p>Esecuzione dell'azione:  <b>move(c, 1, 3)</b></p>
<p> <code>in(c,3)</code>  <code>in(c,3), in(m,3)</code>  <b>load(c,m)</b>  <code>in(c,1)</code>  <code>on(c,m), in(c,1)</code>  <b>unload(c,m)</b>  <code>in(m,1)</code>  <code>in(c,1), in(s,1), in(m,1)</code> </p>	<p> <code>in(c,3), connected(1,2),</code>  <code>connected(2,1), connected(1,3),</code>  <code>connected(3,1), in(s,1),</code>  <code>in(m,3)</code>    <p>Goal affiorante vero e goal and vero nello stato corrente</p> </p>

<b>load(c, m)</b> in(c, 1) on(c, m), in(c, 1) <b>unload(c, m)</b> in(m, 1) in(c, 1), in(s, 1), in(m, 1)	Esecuzione dell'azione: <b>load(c, m)</b>
in(c, 1) on(c, m), in(c, 1) <b>unload(c, m)</b> in(m, 1) in(c, 1), in(s, 1), in(m, 1)	in(c, 3), connected(1, 2), connected(2, 1), connected(1, 3), connected(3, 1), in(s, 1), on(c, m)  Goal affiorante vero nello stato corrente
in(c, 1) on(c, m), in(c, 1) <b>unload(c, m)</b> in(m, 1) in(c, 1), in(s, 1), in(m, 1)	Goal regression con l'azione: <b>move(c, X, 1)</b>
in(c, X), connected(X, 1) <b>move(c, X, 1)</b> on(c, m), in(c, 1) <b>unload(c, m)</b> in(m, 1) in(c, 1), in(s, 1), in(m, 1)	Goal affiorante vero nello stato corrente con X/3
<b>move(c, 3, 1)</b> in(c, 1) on(c, m), in(c, 1) <b>unload(c, m)</b> in(m, 1) in(c, 1), in(s, 1), in(m, 1)	Esecuzione dell'azione <b>move(c, 3, 1)</b>  in(c, 1), connected(1, 2), connected(2, 1), connected(1, 3), connected(3, 1), in(s, 1), on(c, m)
in(c, 1) on(c, m), in(c, 1) <b>unload(c, m)</b> in(m, 1) in(c, 1), in(s, 1), in(m, 1)	Goal affiorante vero e goal and vero nello stato corrente
<b>unload(c, m)</b> in(m, 1) in(c, 1), in(s, 1), in(m, 1)	Esecuzione dell'azione <b>unload(c, m)</b>  in(c, 1), connected(1, 2), connected(2, 1), connected(1, 3), connected(3, 1), in(s, 1), in(m, 1)
in(m, 1) in(c, 1), in(s, 1), in(m, 1)	Tutto vero nello stato corrente  (Stack goal vuoto)

#### Esercizio 4

lgg(C1, C2) = {p(a, X), p(Y, f(Y, c)), ~p(a, f(Y, Z)), ~p(W, f(b, c))}