

COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

19 gennaio 2011 (Punteggio su 30/30; Tempo 2h)

Esercizio 1 (punti 8)

Dato il seguente training set S:

Hop	Bandwith	Classe
1	10	Sì
1	10	No
1	10	No
2	100	Sì
1	10	No
?	100	Sì
2	10	No
1	100	Sì
1	10	No
2	100	Sì
2	100	Sì
1	10	No
2	100	Sì

- Si calcoli l'entropia del training set rispetto all'attributo Classe
- Si calcoli il 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:

1	?
---	---

Esercizio 2 (punti 8)

L'esercizio su CLP si svolge alle ore 14.30 in Laboratorio di Informatica Piccolo.

Esercizio 3 (punti 8)

Si consideri il seguente problema di pianificazione. Ci sia un robot in grado di muoversi, di afferrare e di depositare oggetti all'interno di un ambiente composto da due stanze comunicanti $s1$ e $s2$. Lo stato iniziale, l'obiettivo, le precondizioni e gli effetti di ogni azione sono formulati in STRIPS:

INIT (At(robot,s1) \wedge At(object2,s1) \wedge At(object1,s2) \wedge Room(s1) \wedge Room(s2) \wedge HandEmpty)

GOAL (At(object1,s1) \wedge At(object2,s1))

ACTION (Go(X,Y),

PRECOND: Room(X) \wedge Room(Y) \wedge At(robot,X) \wedge X \neq Y

EFFECT: \neg At(robot,X) \wedge At(robot,Y))

ACTION (Pick(O),

PRECOND: Room(X) \wedge At(robot,X) \wedge At(O,X) \wedge HandEmpty \wedge O \neq robot

EFFECT: \neg HandEmpty \wedge \neg At(O,X) \wedge Holding(O))

ACTION (Drop(O),

PRECOND: Room(X) \wedge At(robot,X) \wedge Holding(O)

EFFECT: HandEmpty \wedge At(O,X) \wedge \neg Holding(O))

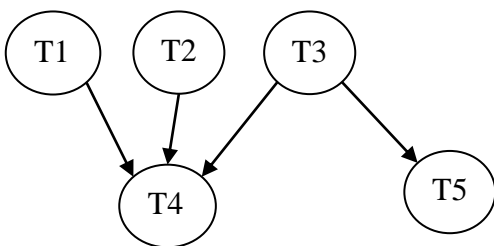
Si risolva il problema utilizzando l'algoritmo STRIPS mostrando una sola strada verso la costruzione del piano (i goals si risolvano in ordine left most). Si indichino eventuali punti di scelta aperti individuati durante la ricerca. Per semplicità si mostra già una parte della soluzione che andrà completata fino al raggiungimento del successo. Si indichino poi le azioni risultanti.

Stato:	Stack di goal e azioni:
At(robot,s1) At(object2,s1) At(object1,s2) Room(s1) Room(s2) HandEmpty	At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	At(object1,s1) At(object2,s1) At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	Room(s1) \wedge At(robot,s1) \wedge Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	At(robot,s1) Holding(object1) Room(s1) \wedge At(robot,s1) \wedge Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	Holding(object1) Room(s1) \wedge At(robot,s1) \wedge Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) \wedge At(object2,s1)

Si prosegue.....

Esercizio 4 (punti 6) –per chi ha seguito nell’A.A. 2006/07 o successivi

Sia data la seguente rete bayesiana



Dove tutte le variabili assumono i valori vero e falso.

Le tabelle di probabilità condizionata sono

per T1:

	T1=Falso	T1=Vero
	0.2	0.8

per T2:

	T2=Falso	T2=Vero
	0.5	0.5

per T3:

	T3=Falso	T3=Vero
	0.4	0.6

per T4:

T1	T2	T3	T4=falso	T4=vero
Falso	Falso	Falso	0.8	0.2
Falso	Falso	Vero	0.6	0.4
Falso	Vero	Falso	0.1	0.9
Falso	Vero	Vero	0.3	0.7
Vero	Falso	Falso	0.7	0.3
Vero	Falso	Vero	0.9	0.1
Vero	Vero	Falso	0.1	0.9
Vero	Vero	Vero	0.2	0.8

per T5:

T5	T3=falso	T3=vero
Falso	0.3	0.7
Vero	0.7	0.3

Si calcoli la probabilità $P(T1, T2 | \sim T3, T5)$.

Esercizio 4a (punti 3) –per chi ha seguito nell’A.A. 2004/05 o precedenti

Si costruisca un meta-interprete per Prolog (in Prolog) che chieda all'utente i goal che non è in grado di dimostrare con le clausole del programma, ma solo se essi sono *ground* (si supponga dato un predicato `ground/1` che è vero se il suo argomento è *ground*) e con la struttura `askable(X)` dove X è una variabile o un atomo..

Se l'utente risponde `true`, il goal deve essere poi asserito come fatto *ground*.

Esercizio 4b (punti 3) –per chi ha seguito nell’A.A. 2004/05 o precedenti

Spiegare qual è la differenza fra la formulazione di Green e la formulazione di Kowalski nel *planning automatico mediante deduzione*.

SOLUZIONE

Esercizio 1

a) $\text{info}(S) = -7/13 \cdot \log_2 7/13 - 6/13 \cdot \log_2 6/13 = 0.996$

b)

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

$$\text{info}(F) = -6/12 \cdot \log_2 6/12 - 6/12 \cdot \log_2 6/12 = 1$$

$$\text{info}_{\text{Hop}}(F) = 7/12 \cdot (-2/7 \cdot \log_2 2/7 - 5/7 \cdot \log_2 5/7) + 5/12 \cdot (-4/5 \cdot \log_2 4/5 - 1/5 \cdot \log_2 1/5) = 0.583 \cdot 0.863 + 0.417 \cdot 0.722 = 0.804$$

$$\text{gain}(\text{Hop}) = 12/13 \cdot (1 - 0.804) = 0.180$$

$$\text{splitinfo}(\text{Hop}) = -6/13 \cdot \log_2(7/13) - 6/13 \cdot \log_2(6/13) - 1/13 \cdot \log_2(1/13) = 1.314$$

$$\text{gainratio}(\text{Hop}) = 0.180 / 1.314 = 0.137$$

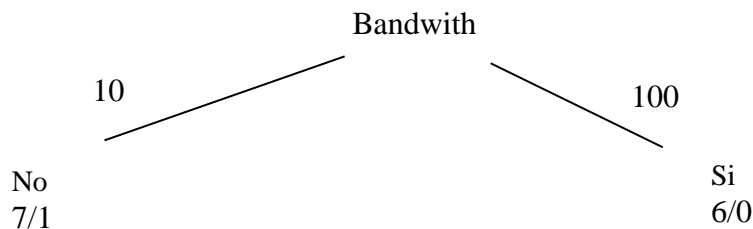
$$\text{info}_{\text{Bandwith}}(S) = 7/13 \cdot (-1/7 \cdot \log_2 1/7 - 6/7 \cdot \log_2 6/7) + 6/13 \cdot (-6/6 \cdot \log_2 6/6 - 0/6 \cdot \log_2 0/6) = 0.538 \cdot 0.592 + 0.462 \cdot 0 = 0.318$$

$$\text{gain}(\text{Bandwith}) = 0.996 - 0.318 = 0.678$$

$$\text{splitinfo}(\text{Bandwith}) = -7/13 \cdot \log_2(7/13) - 6/13 \cdot \log_2(6/13) = 0.996$$

$$\text{gainratio}(\text{Bandwith}) = 0.678 / 0.996 = 0.681$$

c) L'attributo scelto per la radice dell'albero è Bandwith



d) l'istanza viene divisa in due parti, di peso rispettivamente $7/13=0.538$ e $6/13=0.462$. La prima parte viene mandata lungo il ramo 10 e classificata come No con probabilità $6/7=85.7.5\%$ e come si con probabilità $1/7=14.3\%$. La seconda parte viene mandata lungo il ramo 100 e classificata come Si con probabilità 100%. Quindi in totale la classificazione dell'istanza è

$$P(\text{Si}) = 0.538 \cdot 14.3\% + 0.462 \cdot 100\% = 53.9\%$$

$$P(\text{No}) = 0.538 \cdot 85.7\% + 0.462 \cdot 0\% = 46.1\%$$

Esercizio 3

La soluzione trovata è $Go(s1,s2), Pick(object1), Go(s2,s1), Drop(object1)$.

INIT ($At(robot,s1) \wedge At(object2,s1) \wedge At(object1,s2) \wedge Room(s1) \wedge Room(s2) \wedge HandEmpty$)

GOAL ($At(object1,s1) \wedge At(object2,s1)$)

ACTION ($Go(X,Y)$,

PRECOND: $Room(X) \wedge Room(Y) \wedge At(robot,X) \wedge X \neq Y$

EFFECT: $\neg At(robot,X) \wedge At(robot,Y)$)

ACTION ($Pick(O)$,

PRECOND: $Room(X) \wedge At(robot,X) \wedge At(O,X) \wedge HandEmpty \wedge O \neq robot$

EFFECT: $\neg HandEmpty \wedge \neg At(O,X) \wedge Holding(O)$)

ACTION ($Drop(O)$,

PRECOND: $Room(X) \wedge At(robot,X) \wedge Holding(O)$

EFFECT: $HandEmpty \wedge At(O,X) \wedge \neg Holding(O)$)

Stato:	Stack di goal e azioni:
$At(robot,s1)$ $At(object2,s1)$ $At(object1,s2)$ $Room(s1)$ $Room(s2)$ $HandEmpty$	$At(object1,s1) \wedge At(object2,s1)$
<i>idem</i>	$At(object1,s1)$ $At(object2,s1)$ $At(object1,s1) \wedge At(object2,s1)$
<i>idem</i>	$Room(s1) \wedge At(robot,s1) \wedge Holding(object1)$ Drop(object1) $At(object2,s1)$ $At(object1,s1) \wedge At(object2,s1)$
<i>idem</i>	$At(robot,s1)$ $Holding(object1)$ $Room(s1) \wedge At(robot,s1) \wedge Holding(object1)$ Drop(object1) $At(object2,s1)$ $At(object1,s1) \wedge At(object2,s1)$
<i>idem</i>	$Holding(object1)$ $Room(s1) \wedge At(robot,s1) \wedge Holding(object1)$ Drop(object1) $At(object2,s1)$ $At(object1,s1) \wedge At(object2,s1)$
<i>idem</i>	$Room(X) \wedge At(robot,X) \wedge At(object1,X) \wedge HandEmpty \wedge object1 \neq robot$ Pick(object1) $Room(s1) \wedge At(robot,s1) \wedge Holding(object1)$ Drop(object1) $At(object2,s1)$ $At(object1,s1) \wedge At(object2,s1)$
<i>idem</i>	$Room(X)$

	<p>At(robot,X) At(object1,X) HandEmpty object1≠robot Room(X) ∧ At(robot,X) ∧ At(object1,X) ∧ HandEmpty ∧ object1≠robot Pick(object1) Room(s1) ∧ At(robot,s1) ∧ Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) ∧ At(object2,s1)</p>
<i>idem</i>	<p>At(robot,s2) % X/s2 At(object1,s2) HandEmpty object1≠robot Room(s2) ∧ At(robot,s2) ∧ At(object1,s2) ∧ HandEmpty ∧ object1≠robot Pick(object1) Room(s1) ∧ At(robot,s1) ∧ Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) ∧ At(object2,s1)</p>
<i>idem</i>	<p>Room(X) ∧ Room(s2) ∧ At(robot,X) ∧ X≠s2 Go(X,s2) At(object1,s2) HandEmpty object1≠robot Room(s2) ∧ At(robot,s2) ∧ At(object1,s2) ∧ HandEmpty ∧ object1≠robot Pick(object1) Room(s1) ∧ At(robot,s1) ∧ Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) ∧ At(object2,s1)</p>
<i>idem</i>	<p>Go(s1,s2) %X/s1, la eseguo At(object1,s2) HandEmpty ∧ object1≠robot Room(s2) ∧ At(robot,s2) ∧ At(object1,s2) ∧ HandEmpty ∧ object1≠robot Pick(object1) Room(s1) ∧ At(robot,s1) ∧ Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) ∧ At(object2,s1)</p>
<p>At(robot,s2) At(object2,s1) At(object1,s2) Room(s1)</p>	<p>Pick(object1) % la eseguo Room(s1) ∧ At(robot,s1) ∧ Holding(object1) Drop(object1) At(object2,s1)</p>

Room(s2) HandEmpty	At(object1,s1) \wedge At(object2,s1)
At(robot,s2) At(object2,s1) Room(s1) Room(s2) Holding(object1)	Room(s1) \wedge At(robot,s1) \wedge Holding(object1) Drop(object1) At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	At(robot,s1) Holding(object1) Room(s1) \wedge At(robot,s1) \wedge Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	Room(X) \wedge Room(s1) \wedge At(robot,X) \wedge X \neq s1 Go(X,s1) Holding(object1) Room(s1) \wedge At(robot,s1) \wedge Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	Go(s2,s1) % X/s2, la eseguo Holding(object1) Room(s1) \wedge At(robot,s1) \wedge Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) \wedge At(object2,s1)
At(robot,s1) At(object2,s1) Room(s1) Room(s2) Holding(object1)	Holding(object1) Holding(object1) Room(s1) \wedge At(robot,s1) \wedge Holding(object1) Drop(object1) At(object2,s1) At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	Drop(object1) % la eseguo At(object2,s1) At(object1,s1) \wedge At(object2,s1)
At(robot,s1) At(object2,s1) At(object1,s1) Room(s1) Room(s2) Handempty	At(object2,s1) At(object1,s1) \wedge At(object2,s1)
<i>idem</i>	VUOTO

L'and dei goal e' soddisfatto: stack dei goal vuoto.

Esercizio 4 A.A. 2006/07 e successivi

$$P(T1,T2|\sim T3, T5) = P(T1,T2,\sim T3, T5) / P(\sim T3, T5)$$

$$P(T1,T2,\sim T3, T5) = P(T1,T2,\sim T3, \sim T4, T5) + P(T1,T2,\sim T3, T4,T5)$$

$$P(\sim T3, T5) = P(T1, T2, \sim T3, T5) + P(T1, \sim T2, \sim T3, T5) + P(\sim T1, T2, \sim T3, T5) + P(\sim T1, \sim T2, \sim T3, T5) = \\ P(T1, T2, \sim T3, T5) + P(T1, \sim T2, \sim T3, \sim T4, T5) + P(T1, \sim T2, \sim T3, T4, T5) + P(\sim T1, T2, \sim T3, \sim T4, T5) + \\ P(\sim T1, T2, \sim T3, T4, T5) + P(\sim T1, \sim T2, \sim T3, \sim T4, T5) + P(\sim T1, \sim T2, \sim T3, T4, T5)$$

$$P(T1, T2, \sim T3, \sim T4, T5) = P(T1)P(T2)P(\sim T3)P(\sim T4|T1, T2, \sim T3)P(T5|\sim T3) \\ = 0.8 * 0.5 * 0.4 * 0.1 * 0.7 = 0.0112$$

$$P(T1, T2, \sim T3, T4, T5) = P(T1)P(T2)P(\sim T3)P(T4|T1, T2, \sim T3)P(T5|\sim T3) \\ = 0.8 * 0.5 * 0.4 * 0.9 * 0.7 = 0.1008$$

$$P(T1, \sim T2, \sim T3, \sim T4, T5) = P(T1)P(\sim T2)P(\sim T3)P(\sim T4|T1, \sim T2, \sim T3)P(T5|\sim T3) \\ = 0.8 * 0.5 * 0.4 * 0.7 * 0.7 = 0.0784$$

$$P(T1, \sim T2, \sim T3, T4, T5) = P(T1)P(\sim T2)P(\sim T3)P(T4|T1, \sim T2, \sim T3)P(T5|\sim T3) \\ = 0.8 * 0.5 * 0.4 * 0.3 * 0.7 = 0.0336$$

$$P(\sim T1, T2, \sim T3, \sim T4, T5) = P(\sim T1)P(T2)P(\sim T3)P(\sim T4|\sim T1, T2, \sim T3)P(T5|\sim T3) \\ = 0.2 * 0.5 * 0.4 * 0.1 * 0.7 = 0.0028$$

$$P(\sim T1, T2, \sim T3, T4, T5) = P(\sim T1)P(T2)P(\sim T3)P(T4|\sim T1, T2, \sim T3)P(T5|\sim T3) \\ = 0.2 * 0.5 * 0.4 * 0.9 * 0.7 = 0.0252$$

$$P(\sim T1, \sim T2, \sim T3, \sim T4, T5) = P(\sim T1)P(\sim T2)P(\sim T3)P(\sim T4|\sim T1, \sim T2, \sim T3)P(T5|\sim T3) \\ = 0.2 * 0.5 * 0.4 * 0.8 * 0.7 = 0.0224$$

$$P(\sim T1, \sim T2, \sim T3, T4, T5) = P(\sim T1)P(\sim T2)P(\sim T3)P(T4|\sim T1, \sim T2, \sim T3)P(T5|\sim T3) \\ = 0.2 * 0.5 * 0.4 * 0.2 * 0.7 = 0.0056$$

$$P(T1, T2, \sim T3, T5) = 0.0112 + 0.1008 = 0.112$$

$$P(\sim T3, T5) = 0.112 + 0.0784 + 0.0336 + 0.0028 + 0.0252 + 0.0224 + 0.0056 = 0.28$$

$$P(T1, T2|\sim T3, T5) = 0.112 / 0.28 = 0.4$$

/

Esercizio 4a A.A. 2004/05 e precedenti

```
solve(true) :- !.
solve((X, Y)) :- !, solve(X), solve(Y).
solve(X) :- clause(X, Body), solve(Body), !.
solve(X) :-
    ground(X),
    ask(X, Risp), Risp == true, asserta(X).
```

```
ask(X, Risp) :-
    write(X),
    write(" true or false ? "),
    read(Risp).
```

```
ground(askable(X)) :- atomic(X).
```

```
% oppure:
solve(true) :- !.
```



```
solve(A,B):-!,solve(A),solve(B).  
solve(A):-clause(A,B),solve(B),!.  
solve(A):-ground(A),write(A),write(" true or false ? "),  
            read(R),R==true,asserta(A),!.
```