

COMPITO DI APPLICAZIONI DI INTELLIGENZA ARTIFICIALE

7 Aprile 2005 (Punteggio su 30/30; Tempo 2h)

Esercizio 1 (punti 8)

Dato il seguente training set S:

Att1	Att2	Classe
A	Alto	Giovane
B	Basso	Vecchio
A	Alto	Giovane
B	Alto	Vecchio
C	Basso	Vecchio
A	Basso	Giovane
B	Alto	Giovane
C	Basso	Giovane
A	Basso	Vecchio
B	Basso	Vecchio
C	?	Giovane
C	Alto	Giovane

- a) Si calcoli l'entropia del training set rispetto all'attributo Classe (punti 1).
- b) Si calcoli il gain ratio dei due attributi rispetto a questi esempi di training (punti 4).
- c) 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) (punti 1).
- d) si classifichi l'istanza (punti 2):

A	?
---	---

Esercizio 2 (punti 8)

In una spiaggia sono disposti degli ombrelloni. Gli ombrelloni sono disposti come i punti del piano cartesiano a coordinate intere. Immaginiamo di trovarci nell'origine degli assi e di guardarci intorno: non tutti gli ombrelloni saranno visibili da lì, perché alcuni copriranno quelli che si trovano dietro; ad esempio, (1,1) è visibile, ma (2,2), (3,3), ecc, non sono visibili perché coperti proprio da (1,1).

La domanda è questa: dove si trova il quadrato di lato 1 più vicino all'origine (nel senso della distanza Euclidea) tale che i suoi quattro vertici sono tutti invisibili? Si scriva un programma CLP che calcola le coordinate di tale quadrato.

Esercizio 3 (punti 8)

Si abbia un problema di planning per spostare oggetti, modellato alla STRIPS con stato iniziale, goal e le seguenti azioni:

Stato iniziale: `at(robot, table), at(a, table), at(b, table), robotfree`

Stato goal: `at(a, room)`

Caricamento di un oggetto

`load(X, Pos)`

`PREC:at(robot, Pos), at(X, Pos), robotfree`

`EFFECT:in(robot, X), ¬robotfree`

Spostamento di un oggetto

`move(X, Pos1, Pos2)`

`PREC:at(robot, Pos1), in(robot, X)`

`EFFECT:at(robot, Pos2), ¬at(robot, Pos1), ¬at(X, Pos1)`

Scaricamento di un oggetto

`deliver(X, Pos)`

`PREC:at(robot, Pos), in(robot, X)`

`EFFECT:at(X, Pos), ¬in(robot, X)`

Si modelli questo problema con la formulazione di Kowalski.

Esercizio 4 (punti 6)

Si applichi la formulazione di Kowalski delle azioni descritte al punto 3) precedente per pianificare e raggiungere il goal specificato attraverso deduzione.

SOLUZIONE

Esercizio 1:

a) $\text{info}(S) = -7/12 * \log_2 7/12 - 5/12 * \log_2 5/12 = 0,980$

b)

$$\text{info}_{\text{Att1}}(S) = 4/12 * (-3/4 * \log_2 3/4 - 1/4 * \log_2 1/4) + 4/12 * (-3/4 * \log_2 3/4 - 1/4 * \log_2 1/4) + 4/12 * (-3/4 * \log_2 3/4 - 1/4 * \log_2 1/4) =$$

$$= 0,333 * 0,811 + 0,333 * 0,811 + 0,333 * 0,811 = 0,810$$

$$\text{gain}(\text{Att1}) = 0,980 - 0,810 = 0,170$$

$$\text{splitinfo}(\text{Att1}) = -4/12 * \log_2(4/12) - 4/12 * \log_2(4/12) - 4/12 * \log_2(4/12) = 1,585$$

$$\text{gainratio}(\text{Att1}) = 0,170 / 1,585 = 0,107$$

Per calcolare il guadagno dell'attributo Att2 non si usa l'entropia calcolata su tutto il training set ma solo sugli esempi che hanno Att2 noto (insieme F):

$$\text{info}(F) = -6/11 * \log_2 6/11 - 5/11 * \log_2 5/11 = 0,994$$

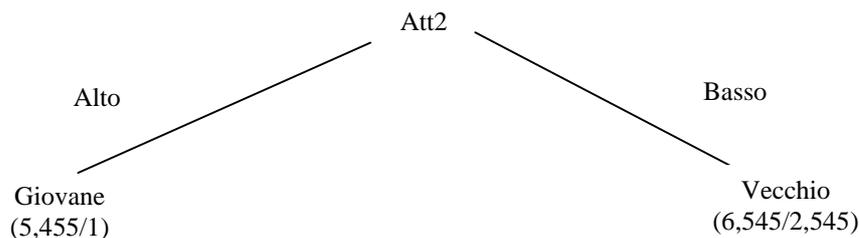
$$\text{info}_{\text{Att2}}(F) = 5/11 * (-4/5 * \log_2 4/5 - 1/5 * \log_2 1/5) + 6/11 * (-2/6 * \log_2 2/6 - 4/6 * \log_2 4/6) =$$
$$= 0,455 * 0,721 + 0,545 * 0,918 = 0,828$$

$$\text{gain}(\text{Att2}) = 11/12 * (0,994 - 0,828) = 0,152$$

$$\text{splitinfo}(\text{Att2}) = -5/12 * \log_2(5/12) - 6/12 * \log_2(6/12) - 1/12 * \log_2(1/12) = 1,325$$

$$\text{gainratio}(\text{Att2}) = 0,152 / 1,325 = 0,115$$

c)



d) l'istanza viene divisa in due parti, una di peso $5,455/12=0,455$ e l'altra di peso $6,545/12=0,545$. La prima parte viene mandata lungo il ramo Alto e viene classificata come Giovane con probabilità $4,455/5,455=81,7\%$ e come Vecchio con probabilità $1/5,455=18,3\%$. La seconda parte viene mandata lungo il ramo Basso e viene classificata come Vecchio con probabilità $4/6,545=61,1\%$ e come Giovane con probabilità $2,545/6,545=38,9\%$. Quindi in totale la classificazione dell'istanza è
Giovane: $0,455 * 81,7\% + 0,545 * 38,9\% = 58,4\%$
Vecchio: $0,455 * 18,3\% + 0,545 * 61,1\% = 41,6\%$

Esercizio 2

```
:- lib(fd).

ombrelloni(X,Y):-
  [X,Y,X1,Y1]::0..100,
  X1 #= X+1,
  Y1 #= Y+1,
  invisibile(X,Y,XC1,YC1,M1),
  invisibile(X1,Y,XC2,YC2,M2),
  invisibile(X,Y1,XC3,YC3,M3),
  invisibile(X1,Y1,XC4,YC4,M4),
  Cost #= X*X + Y*Y,
  minimize(
    labeling([X,Y,XC1,XC2,XC3,XC4,YC1,YC2,YC3,YC4,M1,M2,M3,M4])
    ,Cost).

% L'ombrellone (XC,YC) copre l'ombrellone (X,Y) se c'e`
% un coefficiente M tale che XC*M=X e YC*M=Y
invisibile(X,Y,XC,YC,M):-
  [XC,YC,M]::0..100,
  M#>1,
  XC*M #= X,
  YC*M #= Y.
```

Esercizio 3

Goal

```
:- poss(s), holds(at(a, room), s).
```

Stato iniziale:

```
poss(s0).
holds(at(robot,table), s0).
holds(at(a, table), s0).
holds(at(b, table), s0).
holds(robotfree, s0).
```

Azioni:

Caricamento di un oggetto

```
load(X,Pos)
PREC:at(robot,Pos), at(X,Pos), robotfree
EFFECT:in(robot,X), ¬robotfree
```

Effetti dell'azione `load(X,Pos)`:

```
holds(in(robot,X),do(load(X,Pos),S)).
```

Clausola che esprime le precondizioni dell'azione `load(X,Pos)`:

```
pact(load(X,Pos),S):-
holds(at(robot,Pos),S),holds(at(X,Pos),S), holds(robotfree,S).
```

Clausola per esprimere le condizioni di frame:

```
holds(V,do(load(X,Pos),S)):-
holds(V,S),
V\=robotfree.
```

Spostamento di un oggetto

```
move(X,Pos1,Pos2)
PREC:at(robot,Pos1), in(robot,X)
EFFECT:at(robot,Pos2), ¬at(robot,Pos1), ¬at(X,Pos1)
```

Effetti dell'azione `move(X,Pos1,Pos2)`:

```
holds(at(robot,Pos2),do(move(X,Pos1,Pos2),S)).
```

Clausola che esprime le precondizioni dell'azione `move(X,Pos1,Pos2)`:

```
pact(move(X,Pos1,Pos2),S):-
holds(at(robot,Pos1),S),holds(in(robot,X),S).
```

Clausola per esprimere le condizioni di frame:

```
holds(V,do(move(X,Pos1,Pos2),S)):-
holds(V,S),
V\= at(robot,Pos1),
V\= at(X,Pos1).
```

Scaricamento di un oggetto

```
deliver(X,Pos)
PREC:at(robot,Pos), in(robot,X)
EFFECT:at(X,Pos), ¬in(robot,X)
```

Effetti dell'azione `deliver(X,Pos)`:

```
holds(at(X,Pos),do(deliver(X,Pos),S)).
```

Clausola che esprime le precondizioni dell'azione `deliver(X,Pos)`:

```
pact(deliver(X,Pos),S):-
holds(at(robot,Pos),S),holds(in(robot,X),S).
```

Clausola per esprimere le condizioni di frame:

```
holds(V,do(deliver(X,Pos),S)):-
holds(V,S),
V\= in(robot,X).
```

Clausola per esprimere la raggiungibilità di uno stato:

```
poss(do(U,S)) :-  
    poss(S),  
    pact(U,S).
```

Esercizio 4

Tracing del programma Prolog al punto 3 (per il goal al punto 3).