

## Introduzione all'Intelligenza Artificiale – a.a. 2017/18

Seconda prova intermedia – 4 giugno 2018

### Domande [punti 7]

La risposta corretta alla maggior parte delle domande abilita la correzione della seconda parte. Fornire risposte brevi, entro una riga di testo.

1. Nel passaggio di eliminazione degli esistenziali della trasformazione in forma a clausole, quando si deve usare una costante di Skolem e quando una funzione di Skolem?
2. Che proprietà ha un *unbiased learner*?
3. Definire il *decision boundary* di un modello lineare (matematicamente).
4. Scrivere l'equazione dell'ipotesi nel caso generale di una *linear basis expansion*.
5. Definire la *loss* della *ridge regression* (equazione).
6. Che significa regolarizzazione nel Machine Learning?
7. Definire la *loss* dell'algoritmo *K-means* (equazione).

### Esercizio 1. Unificazione [5 punti]

1. Si definisca l'operazione di unificazione tra due espressioni.
2. Per ciascuna delle coppie di espressioni che seguono, si calcoli, se esiste, l'unificatore più generale.
  - a.  $p(x, x)$  e  $p(f(y), y)$
  - b.  $p(f(x, y), g(A, A))$  e  $p(f(f(w, z), v), w)$

### Esercizio 2. Le borsette di Maria [13 punti]

Maria deve scegliere una nuova borsetta. Sapete che gradisce una borsa chiara piccola ma non gradisce né una borsetta scura piccola, né una chiara grande. Pensate che comprerebbe una borsetta grande scura? Proporre una soluzione in base a diversi modelli, codificando il colore chiaro con 0 e quello scuro con 1 e la dimensione piccola con 0 e grande con 1. In particolare:

1. Definire il compito (task) di apprendimento (input, output, data set e tipo di task)
2. Proporre la soluzione che forniremmo costruendo l'ipotesi con Find-S (mostrando i passi di esecuzione dell'algoritmo per costruire l'ipotesi)
3. Proporre una soluzione utilizzando Candidate Elimination
4. Proporre una soluzione mostrando la costruzione con ID3 tramite il calcolo dei Gain
5. Proporre una soluzione (direttamente, senza necessità di eseguire algoritmi di learning) con un modello lineare (per via grafica ed esprimendo la  $h(\mathbf{x})$ )
6. Proporre una soluzione (direttamente, senza necessità di eseguire algoritmi di learning) con un modello SVM *hard-margin* (per via grafica ed esprimendo la  $h(\mathbf{x})$ , motivandola)
7. Formulare la risposta che si darebbe con un K-NN (motivandola)
8. Formulare la risposta che si darebbe con un modello di tipo "Lookup table" (motivandola)

### Esercizio 3. Quando fermarsi? [7 punti]

Nell'algoritmo a discesa di gradiente si parte con valori dei  $w$  nulli e si deve scegliere un criterio di stop delle iterazioni. L'algoritmo viene fatto iterare a lungo (con una iterazione per ogni *epoca*) conservando i modelli  $i$  prodotti in ogni iterazione  $i$ , fino ad un massimo di  $K$ . Potete quindi scegliere quale modello  $i$  utilizzare. Avete diviso il data set in tre parti: training (TR), validation (VL) e test (TS) set e misurate ad ogni iterazione l'errore  $TR_i$ ,  $VL_i$  e  $TS_i$ .

1. Commentate ciascuna di queste tre possibilità e scegliete quella che potrebbe essere la migliore:
  - a. Prendete l' $i$  per cui l'errore di training è minimo (di indice  $i(TR_{min})$ )
  - b. Prendete l' $i$  per cui l'errore di validation è minore (di indice  $i(VL_{min})$ )
  - c. Prendete l' $i$  per cui l'errore sul test set è minore (di indice  $i(TS_{min})$ )
2. Che modello scegliete e quale errore riportate come stima di accuratezza del modello?
3. Potreste infine suggerire un criterio di fermata che arresti l'iterazione prima di arrivare a  $K$  (anche assumendo un andamento dell'errore rispetto ad  $i$  che considerate tipico)?