

Sistemi Intelligenti Learning: l'apprendimento degli agenti

Alberto Borghese

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Scienze dell'Informazione
borghese@dsi.unimi.it



A.A. 2006-2007

1/36

<http://homes.dsi.unimi.it/~borghese/>



Riassunto



- **Gli agenti**
- Il Reinforcement Learning
- Gli elementi del RL
- Un esempio: tris

A.A. 2006-2007

2/36

<http://homes.dsi.unimi.it/~borghese/>



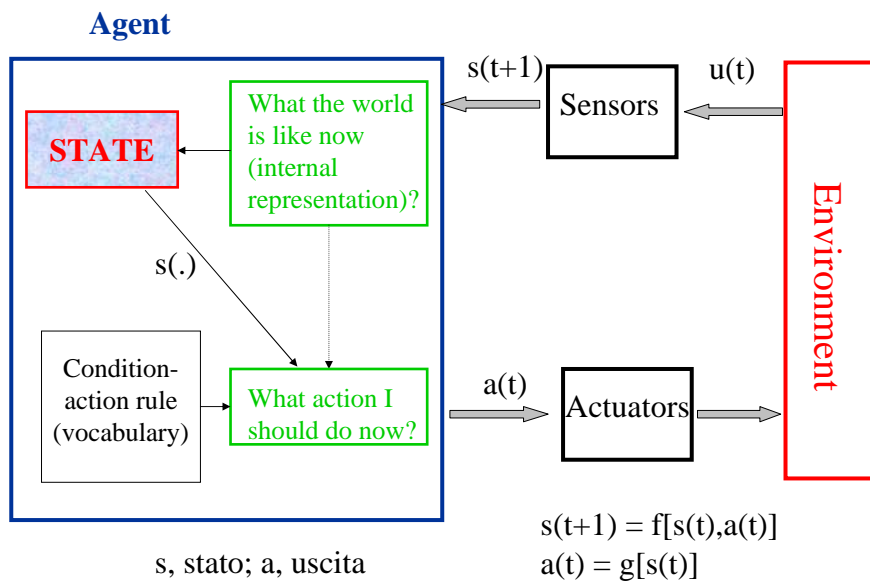
Agente



- Può scegliere un'azione sull'ambiente tra un insieme continuo o discreto.
- L'azione dipende dalla situazione. La situazione è riassunta nello **stato** del sistema.
- L'agente monitora continuamente l'ambiente (**input**) e modifica continuamente lo stato.
- La scelta dell'azione è non banale e richiede un certo grado di "intelligenza".



Schematic diagram of an agent





L'agente



- Inizialmente l'attenzione era concentrata sulla progettazione dei sistemi di "controllo". Valutazione, sintesi...
- L'intelligenza artificiale e la "computational intelligence" hanno consentito di spostare l'attenzione sull'apprendimento delle strategie di controllo e più in generale di comportamento.
- **Macchine dotate di meccanismi (algoritmi, SW), per apprendere.**



I vari tipi di apprendimento



$$x(t+1) = f[x(t), u(t)]$$
$$y(t) = g[x(t), u(t)]$$

Supervisionato (learning with a teacher). Viene specificato per ogni pattern di input, il pattern desiderato in output.

Non-supervisionato (learning without a teacher). Estrazione di similitudine statistiche tra pattern di input. Clustering. Mappe neurali.

Apprendimento con rinforzo (reinforcement learning, learning with a distal teacher). L'ambiente fornisce un'informazione puntuale, del tipo success or fail.



I - Apprendimento supervisionato



$$y(t) = g[w; u(t)]$$

$$\min_{\{w\}} J(\cdot) \quad J = \|Y^D - g(W; U)\|$$

$$u \in \mathbb{R}^n \quad y \in \mathbb{R}^m$$

Y^D è l'uscita desiderata nota.

- Si tratta di un problema di minimizzazione di una cifra di merito, $J(\cdot)$, nello spazio dei parametri W , che caratterizzano l'agente.
- Apprendimento supervisionato si applica a problemi di *classificazione* o di *regressione*.

Soluzione iterativa: Obiettivo: se esiste una soluzione, trovare ΔW in modo iterativo tale che l'insieme dei pesi W^{nuovo} ottenuto come:

$$W^{\text{nuovo}} = W^{\text{vecchio}} + \Delta W$$

dia luogo a un errore sulle uscite di norma minore che con W^{vecchio}

- Capacità di generalizzazione
- Modello globale vs insieme di modelli locali.

A.A. 2006-2007

7/36

<http://homes.dsi.unimi.it/~borghese/>



Esempio Apprendimento Supervisionato

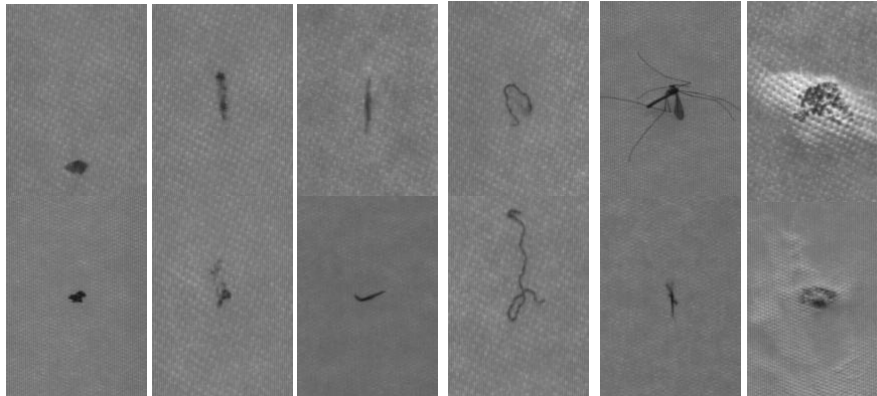
Task di classificazione
Uscita booleana

A.A. 2006-20

<http://homes.dsi.unimi.it/~borghese/>



Riconoscimento difetti in linee di produzione (finanziato da Electronic Systems: 2006-)



regolari

irregolari

allungati

fili

insetti

macchie su
denso

Anche: pieghe, arruffati

Difetti – Classificazione real-time e apprendimento.
Feature extraction + SVM

A.A. 2006-2007

9/36

<http://homes.dsi.unimi.it/~borghese/>



Riconoscimento della scrittura (progetto su un problema ridotto)



1. Determinare il tipo di “training example” adatti al problema.
2. Definire un training set rappresentativo. Training set = coppie di input / output (ruolo del supervisore umano).
3. Preprocessing. Determinare quali feature è opportuno estrarre dai pattern in ingresso perchè costituiscano l’input per l’agente.
4. Determinare i dati utilizzati per l’apprendimento (quali -> rappresentatività; e quanti -> tanti ma non troppi....).
5. Determinare le modalità di apprendimento: empirical risk minimization, structural risk minimization....
6. Determinare come valutare l’apprendimento (e.g. through cross-validation).

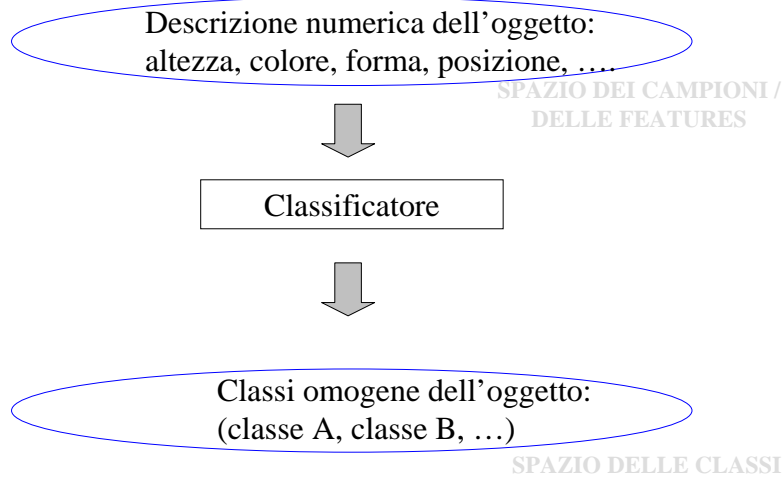
A.A. 2006-2007

10/36

<http://homes.dsi.unimi.it/~borghese/>



Apprendimento non-supervisionato: Clustering



A.A. 2006-2007

11/36

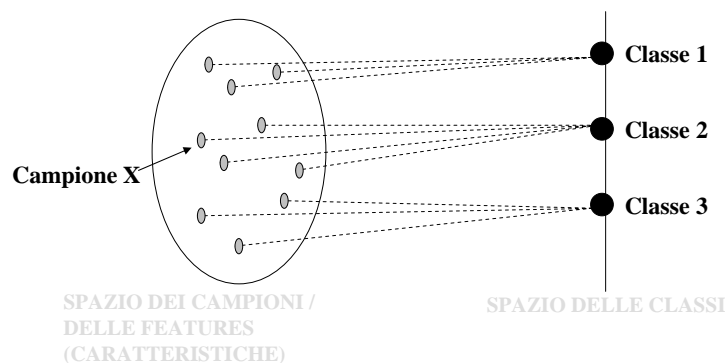
<http://homes.dsi.unimi.it/~borghese/>



Classificazione



Un'interpretazione geometrica:
Mappatura dello spazio dei campioni nello spazio delle classi.



***Che differenza c'è rispetto al clustering?
Cos'è un concetto?***

A.A. 2006-2007

12/36

<http://homes.dsi.unimi.it/~borghese/>



Esempio di clustering



Tesi: ricerca immagini su WEB.



Clustering -> Indicizzazione



A cosa serve il clustering?



- Compressione dati (telecomunicazioni, immagini, ...);
- Segmentazione (bio)immagini;
- Controllo robot;
- Indicizzazione

A cosa serve la classificazione?

- Riconoscimento automatico;
- Pattern recognition;
- Ricostruzione superfici;
- ...



Riassunto



- Gli agenti
- **Il Reinforcement Learning**
- Gli elementi del RL
- Un esempio: tris



Reinforcement learning



Nell'apprendimento supervisionato, esiste un "teacher" che dice al sistema quale è l'uscita corretta (learning with a teacher). Non sempre è possibile.

Spesso si ha a disposizione solamente un'informazione qualitativa (a volte binaria, giusto/sbagliato successo/fallimento), puntuale.

Questa è un'informazione qualitativa.

*L'informazione disponibile si chiama **segnale di rinforzo**. Non dà alcuna informazione su come aggiornare il comportamento dell'agente (e.g. i pesi). Non è possibile definire una funzione costo o un gradiente.*

Obiettivo: creare degli agenti "intelligenti" che abbiano una "machinery" per apprendere dalla loro esperienza.



Reinforcement Learning: caratteristiche



- Apprendimento mediante interazione con l'ambiente. Un agente isolato non apprende.
- L'apprendimento è funzione del raggiungimento di uno o più obiettivi.
- Non è necessariamente prevista una ricompensa ad ogni istante di tempo.
- Le azioni vengono valutate mediante la ricompensa a lungo termine ad esse associata (**delayed reward**). Il meccanismo di ricerca delle azioni migliori è imparentato con la ricerca euristica: **trial-and-error**.
- L'agente sente l'input, modifica lo stato e genera un'azione che massimizza la ricompensa a lungo termine.



Exploration vs Exploitation



Esplorazione (**exploration**) dello spazio delle azioni per scoprire le azioni migliori. Un agente che esplora solamente raramente troverà una buona soluzione.

Le azioni migliori vengono scelte ripetutamente (**exploitation**) perchè garantiscono ricompensa (**reward**). Se un agente non esplora nuove soluzioni potrebbe venire surclassato da nuovi agenti più dinamici.

Occorre non interrompere l'esplorazione.

Occorre un approccio statistico per valutare le bontà delle azioni.

Exploration ed exploitation vanno bilanciate. Come?



Dove agisce un agente?



- L'agente ha un comportamento goal-directed ma agisce in un **ambiente incerto** non noto a priori o parzialmente noto.
- Esempio: planning del movimento di un robot.
- Un agente impara interagendo con l'ambiente. Planning può essere sviluppato mentre si impara a conoscere l'ambiente (mediante le misure operate dall'agente stesso). La strategia è vicina al trial-and-error.



L'ambiente



Model of the environment. E' uno sviluppo relativamente recente. Da valutazione implicita dello svolgersi delle azioni future (trial-and-error) a valutazione esplicita mediante modello dell'ambiente della sequenza di azioni e stati futuri (planning) = dal sub-simblico al simbolico; emerging intelligence.

Incorporazione di AI:

- Planning (pianificazione delle azioni).
- Viene rinforzato il modulo di pianificazione dell'agente.

Incorporazione della conoscenza dell'ambiente:

- Modellazione dell'ambiente (non noto o parzialmente noto).



Relazione con l'AI



- Gli agenti hanno dei goal da soddisfare. Approccio derivato dall'AI.
- Nell'apprendimento con rinforzo vengono utilizzati strumenti che derivano da aree diverse dall'AI:
 - ◆ Ricerca operativa.
 - ◆ Teoria del controllo.
 - ◆ Statistica.
- L'agente impara facendo. Deve selezionare i comportamenti che **ripetutamente** risultano favorevoli a lungo termine.



Esempi



Un giocatore di scacchi. Per ogni mossa ha informazione sulle configurazioni di pezzi che può creare e sulle possibili contro-mosse dell'avversario.

Una gazzella in 6 ore impara ad alzarsi e correre a 40km/h.

Come fa un robot veramente autonomo ad imparare a muoversi in una stanza per uscirne? (cf. competizione Robocare@home).

Come impostare i parametri di una raffineria (pressione petrolio, portata....) in tempo reale, in modo da ottenere il massimo rendimento o la massima qualità?



Caratteristiche degli esempi



Parole chiave:

- Interazione con l'ambiente. L'agente impara dalla **propria** esperienza.
- Obiettivo dell'agente.
- Incertezza o conoscenza parziale dell'ambiente.

Osservazioni:

- Le azioni modificano lo stato (la situazione), cambiano le possibilità di scelta in futuro (**delayed reward**).
- L'effetto di un'azione non si può prevedere completamente.
- L'agente ha a disposizione una valutazione globale del suo comportamento. Deve sfruttare questa informazione per migliorare le sue scelte. **Le scelte migliorano con l'esperienza.**
- I problemi possono avere orizzonte temporale finito od infinito.



Riassunto



- Gli agenti
- Il Reinforcement Learning
- **Gli elementi del RL**
- Un esempio: tris



I tue tipi di rinforzo



L'agente deve scoprire quale azione (**policy**) fornisca la ricompensa massima provando le varie azioni (trial-and-error) sull'**ambiente**.

“Learning is an adaptive change of behavior and that is indeed the reason of its existence in animals and man (K. Lorentz, 1977).”

Rinforzo puntuale istante per istante, azione per azione (**condizionamento classico**).

Rinforzo puntuale “una-tantum” (**condizionamento operante**).



Il Condizionamento classico



L'agente deve imparare una (o più) trasformazione tra input e output. Queste trasformazioni forniscono un comportamento che l'ambiente premia.

Il segnale di rinforzo è sempre lo stesso per ogni coppia input – output.

Esempio: risposte riflesse Pavloviane. Campanello (stimolo condizionante) prelude al cibo. Questo induce una risposta (salivazione). La risposta riflessa ad uno stimolo viene evocata da uno stimolo condizionante.

Stimolo-Risposta. Lo stimolo condizionante (campanello = input) induce la salivazione (uscita) in risposta al campanello.

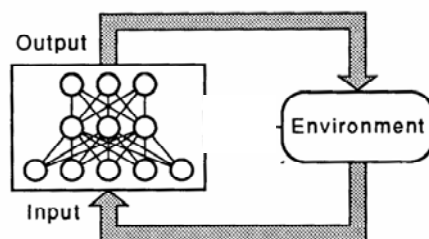


Condizionamento operante



Reinforcement learning (operante).

Interessa un **comportamento**. Una **sequenza di input / output** che può essere modificata agendo sui parametri che definiscono il comportamento dell'agente. Il condizionamento arriva in un certo istante di tempo (spesso una-tantum) e deve valutare tutta la sequenza temporale di azioni, anche quelle precedenti nel tempo.



A.A. 2006-2007

<http://homes.dsi.unimi.it/~borghese/>



Gli attori del RL



Policy. Descrive l'azione scelta dall'agente: mapping tra stato (input dall'ambiente) e azioni. Funzione di controllo. Le policy possono avere una componente stocastica. Viene utilizzato un modello adeguato del comportamento dell'agente (e.g. tabella, funzione continua parametrica...).

Reward function. Ricompensa **immediata**. Associata all'azione intrapresa in un certo stato. Può essere data al raggiungimento di un goal (esempio: successo / fallimento). E' uno scalare. Rinforzo primario.

Value function. "Cost-to-go". Ricompensa a **lungo termine**. Somma dei reward: costi associati alle azioni scelte istante per istante + costo associato allo stato finale. Orizzonte temporale ampio. Rinforzo secondario.

Ambiente. Può essere non noto o parzialmente noto. L'agente deve costruirsi una rappresentazione dell'ambiente.

- Quale delle due è più difficile da ottenere?
- L'agente agisce per massimizzare la funzione Value o Reward?

A.A. 2006-2007

28/36

<http://homes.dsi.unimi.it/~borghese/>



Proprietà del rinforzo

L'ambiente o l'interazione può essere complessa.

Il rinforzo può avvenire solo dopo una più o meno lunga sequenza di azioni (**delayed reward**).

E.g. agente = giocatore di scacchi.
 ambiente = avversario.

Problemi collegati:

temporal credit assignement.
structural credit assignement.

L'apprendimento non è più da esempi, ma dall'osservazione del proprio comportamento nell'ambiente.



Riassunto

- Reinforcement learning. L'agente viene modificato, rinforzando le azioni che sono risultate buone a lungo termine. E' quindi una classe di algoritmi iterativi.
- Self-discovery of a successful strategy (it does not need to be optimal!). La strategia (di movimento, di gioco) non è data a-priori ma viene appresa attraverso **trial-and-error**.
- Credit assignement (temporal and structural).
- Come possiamo procedere in modo efficiente nello scoprire una strategia di successo? Cosa vuol dire modificare l'agente?



Riassunto



- Gli agenti
- Il Reinforcement Learning
- Gli elementi del RL
- **Un esempio: tris (progetto)**



Gli attori del RL



Policy. Descrive l'azione scelta dall'agente: mapping tra stato e azioni. Funzione di controllo.

Ambiente. Descrive tutto quello su cui agisce la policy. Generalmente è non noto o parzialmente noto e deve essere "scoperto". Non viene modellato esplicitamente, ma viene stimata la sua reazione alla policy mediante la value function.

Reward function. Ricompensa **immediata**. Associata all'azione intrapresa in un certo stato. Può essere data al raggiungimento di un goal (esempio: successo / fallimento). E' uno scalare associato allo stato dell'agente. Rinforzo primario.

Value function. "Cost-to-go". Ricompensa a **lungo termine**. Somma dei reward + costi associati alle azioni scelte istante per istante. Orizzonte temporale ampio. Rinforzo secondario. Ricompensa attesa.

Ciclo dell'agente (le tre fasi sono sequenziali):

- 1) Implemento una policy
- 2) Aggiorno la Value function
- 3) Aggiorno la policy.



Commenti



Con molti stati è impossibile esplorarli tutti →
Generalizzazione. Dalle funzioni combinatorie alle funzioni continue.

Può essere inserita della conoscenza a-priori sia sulla policy che sulla Value function.



Esempio: gioco del tris



X	O	O
O	X	X
		X

Ciclo dell'agente (le tre fasi sono sequenziali):

- 1) Implemento una policy
- 2) Aggiorno la Value function
- 3) Aggiorno la policy.



Riflessioni su RL ed il gioco del tris



Supponete che l'agente dotato di RL giochi, invece che con un avversario, contro sé stesso. Cosa pensate che succeda? Secondo voi imparerebbe una diversa strategia di gioco?

Molte posizioni del tris sembrano diverse ma sono in realtà la stessa per effetto delle simmetrie. Come si può modificare l'algoritmo di RL (definizione dello stato) per sfruttare le simmetrie? Come si può migliorare il meccanismo di apprendimento?

Supponiamo che l'avversario non sfrutti le simmetrie. In questo caso noi possiamo sfruttarle? E' vero che configurazioni della scacchiera equivalenti per simmetria devono avere la stessa funzione valore.

Potete pensare a modi per migliorare il gioco dell'agente? Potete pensare a metodi migliori (più veloci, più robusti....) perché un agente impari a giocare a tris?



Riassunto



- Gli agenti
- Il Reinforcement Learning
- Gli elementi del RL
- Un esempio: tris (progetto)