

Artificial Intelligence

Stefano Di Lena

2025

Indice

1	Intelligent Agents	1
1.1	Rationality	2
1.2	Specificare un ambiente	2
1.2.1	Proprieties of thask environments	3
1.3	The Structure of Agents	3
1.3.1	Simple reflex agents	4
1.3.2	Model-based reflex agents	4
1.3.3	Goal-based agents	5
1.3.4	Utility-based agents	5
1.3.5	Learning agents	6
2	Solving Problems By Searching	7

Capitolo 1

Intelligent Agents

Un **agent** è tutto ciò che può essere visto come un sistema che percepisce il suo *ambiente* tramite i *sensori* ed agisce su di esso tramite *attuatori*.

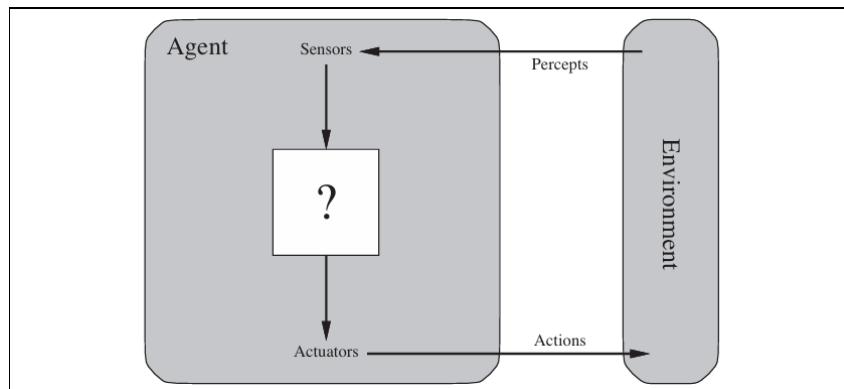


Figura 1.1: Gli agenti interagiscono con l'ambiente attraverso sensori ed attuatori.

Il comportamento di un agente è descritto, matematicamente, dal **agent function** che descrive la corrispondenza di una qualsiasi sequenza percettiva ed una specifica azione. Questa funzione è implementata, *internamente* (in maniera concreta, in esecuzione in un sistema fisico), da un **agent program**.

Consideriamo l'agente aspirapolvere che si muove nel seguente ambiente.

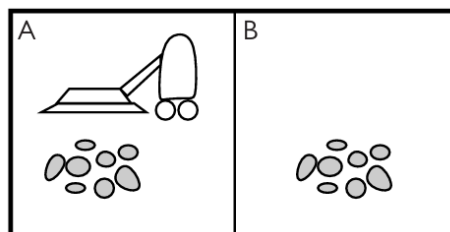


Figura 1.2: Un mondo per l'aspirapolvere con due sole posizioni.

Ogni posizione può essere pulita o sporca e l'agente può decidere di aspirare, andare a sinistra, a destra oppure non fare niente.

Un semplice esempio è: "se il riquadro corrente è sporco allora aspira, altrimenti muoviti sull'altro blocco". Altri esempi sono descritti nella tabella successiva.

Percept sequence	Action
$[A, Clean]$	<i>Right</i>
$[A, Dirty]$	<i>Suck</i>
$[B, Clean]$	<i>Left</i>
$[B, Dirty]$	<i>Suck</i>
$[A, Clean], [A, Clean]$	<i>Right</i>
$[A, Clean], [A, Dirty]$	<i>Suck</i>
\vdots	\vdots

Figura 1.3: Tabulazione parziale di una semplice funzione agente per il mondo dell'aspirapolvere mostrato sopra.

1.1 Rationality

Un **rational agent** sceglie un'azione che massimizza il valore atteso della sua misura di prestazione, per ogni possibile sequenza di percezioni.

Una **performance measure** valuta una sequenza di stati dell'ambiente.

Razionale non vuol dire onnisciente, né chiaroveggente (non è detto che i risultati delle azioni siano quelli attesi). Per razionale intendiamo che esplorando l'ambiente riesce ad apprendere in autonomia.

1.2 Specificare un ambiente

Nell'esempio dell'aspirapolvere abbiamo specificato la performance measure, l'ambiente esterno, gli attuatori ed i sensori dell'agente. Tutto questo può essere raggruppato nel termine **task environment**. Viene anche nominato considerando gli acronimi, come descrizione **PEAS** (Performance, Environment, Actuators, Sensors). Quanto meglio si riescono a descrivere questi elementi, tanto migliori saranno le performance dell'agente.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Figura 1.4: Descrizione PEAS dell'ambiente operativo di un taxi automatizzato.

1.2.1 Proprieties of thask environments

Un ambiente operativo si dice **completamente osservabile** quando i sensori dell'agente misurano tutti gli aspetti *rilevanti* per la scelta dell'azione. A causa della presenza di rumore o della scarsa precisione di alcuni sensori un ambiente potrebbe essere **parzialmente osservabile**. Se l'agente non dispone di sensori è **inosservabile**.

Un ambiente si dice **deterministico** se lo stato successivo dell'ambiente è completamente determinato dallo stato corrente e dall'azione eseguita dall'agente. In caso contrario si dice **stocastico**, in questo caso il modello è esplicitamente associato a probabilità.

In un ambiente operativo **episodico** l'esperienza dell'agente è divisa in episodi atomici. In ogni episodio l'agente riceve una percezione e poi esegue una singola azione. Un episodio non dipende dalle azioni intraprese in quelle precedenti. Negli ambienti **sequenziali**, invece, ogni decisione può influenzare le successive.

Un ambiente si dice **dinamico** per un agente se può cambiare mentre questo sta decidendo come agire. In caso contrario diciamo che è **statico**.

La distinzione tra **discreto** e **continuo** si applica allo *stato* dell'ambiente, al modo in cui è gestito il *tempo*, alle *percezioni* ed *azioni* dell'agente. [Esempi: la scacchiera ha un numero finito di stati distinti (discreto), la guida di un taxi è un problema continuo nel tempo (la sua velocità e posizione cambia continuamente nel tempo)].

Un agente che risolve un puzzle da solo è chiaramente in un ambiente **single-agent**. Se abbiamo più agenti che comunicano tra di loro, siamo in un ambiente **multi-agent**, il quale può essere:

- *competitivo* (ad esempio una partita a scacchi tra due agenti);
- *cooperativo* (ad esempio nella guida autonoma dove ogni agente deve collaborare per evitare gli incidenti).

Conoscere il tipo di ambiente è fondamentale per la corretta progettazione di un agente. Il mondo reale è parzialmente osservabile, multi-agent, non deterministico, sequenziale, dinamico, continuo e ignoto.

1.3 The Structure of Agents

Definiamo quattro tipi base di programma agente che contengono i principi alla base di quasi tutti i sistemi intelligenti. Vedremo anche come queste classi possono essere migliorate inserendo una forma di apprendimento.

1.3.1 Simple reflex agents

L'agente reattivo semplice sceglie le azioni sulla base della percezione corrente, ignorando la storia percettiva precedente.

Utilizza le regole *condizione-azione* (anche nota come *if-then*). Nell'esempio del taxi a guida autonoma, se l'agente nota che una macchina davanti sta frenando allora deve iniziare anche lui a frenare.

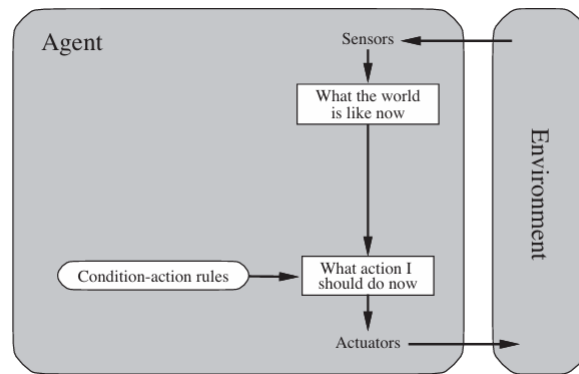


Figura 1.5: Diagramma schematico di un agente reattivo semplice.

Sono la tipologia più semplice di agente, ma la loro intelligenza è molto limitata. Funzionerà solo nel caso in cui l'ambiente è completamente osservabile (anche una minima parte di non-osservabilità può causare grandi problemi).

```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left
```

Figura 1.6: Programma per l'agente reattivo semplice nell'ambiente in Figura 1.2 che implementa la funzione agente rappresentata dalla Figura 2.3.

Nello schema di Figura 1.5 usiamo i *rettangoli* per indicare lo stato interno attuale del processo di decisione dell'agente e gli *ovali* per rappresentare le azioni di fondo usate nel processo.

1.3.2 Model-based reflex agents

L'agente che si basa di un *modello* del mondo, ovvero uno che mantiene uno stato interno che dipende dalla storia delle percezioni e che riflette alcuni aspetti dello stato corrente non-osservabili, viene detto model-based.

Notiamo dalla Figura 1.7 come lo stato percepito è combinato ad uno stato interno per generare una versione aggiornata dello stato corrente, basandosi sul modello dell'agente di "come il funziona il mondo".

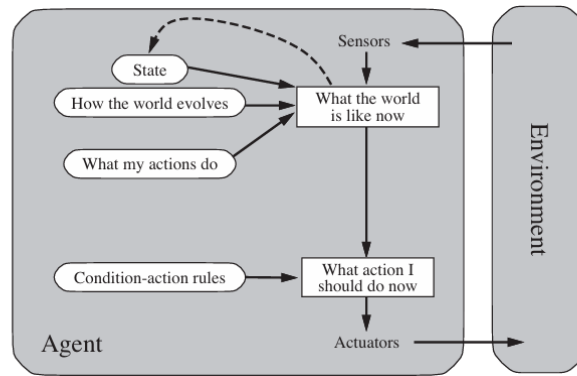


Figura 1.7: Un agente reattivo basato su modello.

1.3.3 Goal-based agents

L'agente che si muove per raggiungere un obiettivo. Considerando sempre l'esempio del taxi, questo ad un incrocio potrebbe girare a sinistra, a destra o proseguire dritto. La decisione giusta dipende dalla destinazione finale impostata. Quindi l'agente oltre allo stato corrente necessita anche di informazioni sull'obiettivo, per descrivere le situazioni *desiderabili*.

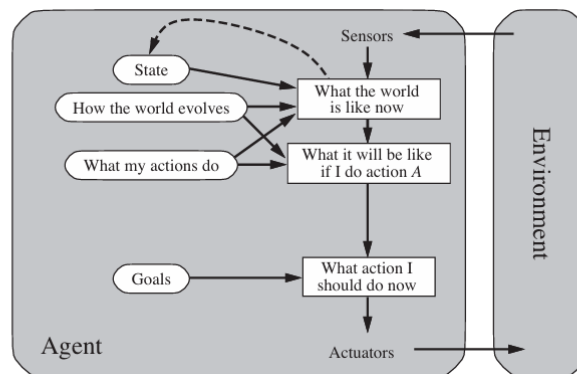


Figura 1.8: Un agente basato su modello che raggiunge degli obiettivi.

Questo tipo di agente sembra meno efficiente, ma risulta molto più flessibile (la conoscenza che guida le sue decisioni è rappresentata esplicitamente e può essere modificata).

1.3.4 Utility-based agents

Gli obiettivi forniscono una distinzione binaria tra stati "contenti" e "scontenti". Una misura delle performance potrebbe confrontare più stati e capire quanto renderebbero felice l'agente una volta raggiunti. Per usare un termine più scientifico diremo *utilità* al posto di felicità o contentezza.

L'agente basato sull'utilità sceglierà, quindi, un'azione considerando quanto uno stato è desiderabile o meno, in aggiunta all'importanza dell'obiettivo.

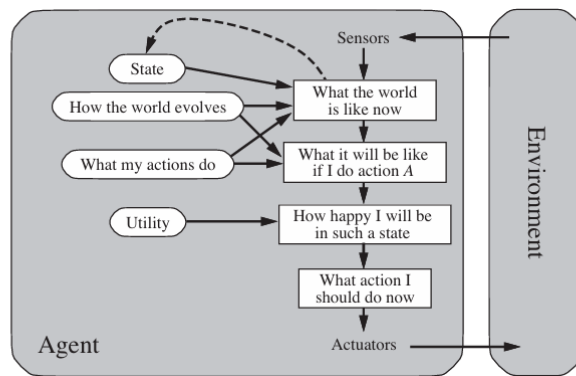


Figura 1.9: Un agente basato su modello che massimizza l'utilità.

1.3.5 Learning agents

Tramite l'apprendimento un agente riesce ad operare in ambienti inizialmente sconosciuti, diventando col tempo più competenti di quanto già fossero all'inizio (basandosi sulla conoscenza iniziale).

Un agente di apprendimento può essere scomposto in quattro componenti dal punto di vista concettuale, come mostrato in Figura 1.10:

- *learning element*
- *performance element*
- *critic*
- *problem generator*

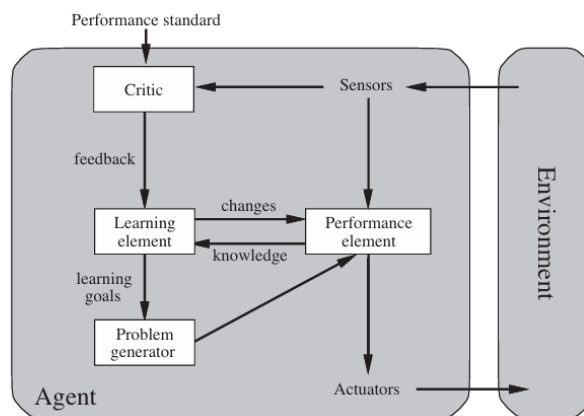


Figura 1.10: Un agente capace di apprendere.

Capitolo 2

Solving Problems By Searching