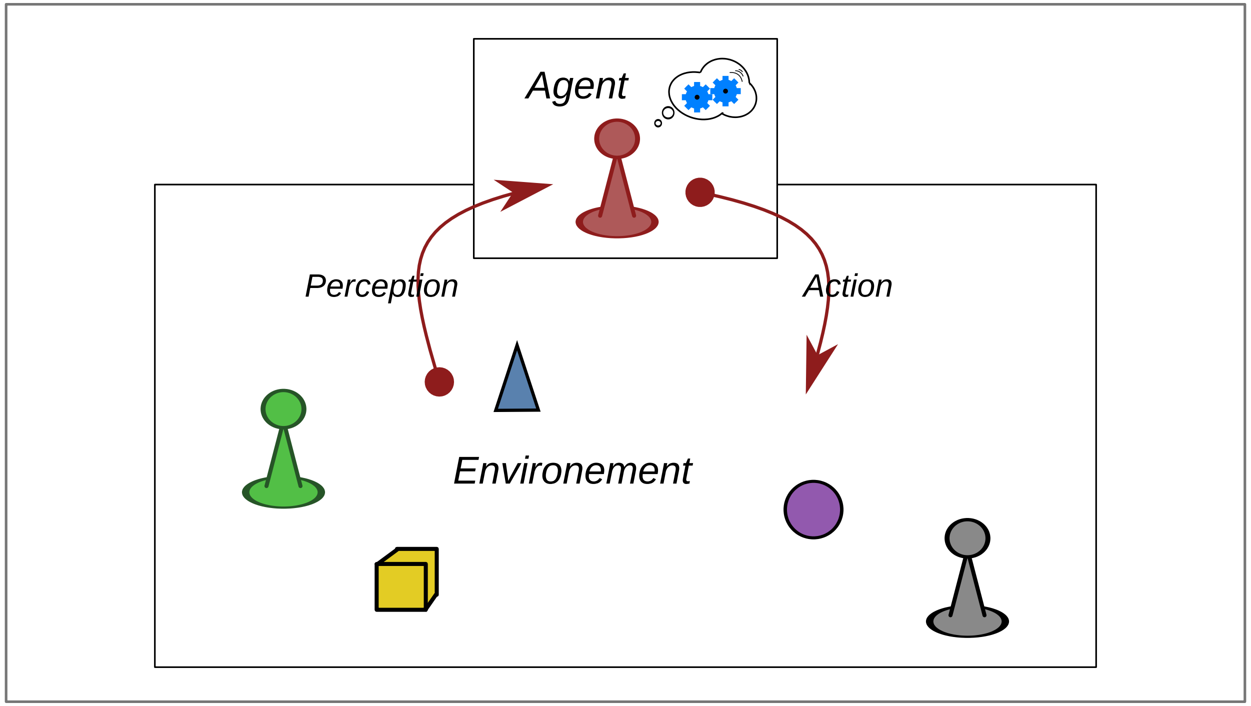
Decision Under Uncertainty

States, Actions and Policies

Guillaume.Lozenguez
@imt-nord-europe.fr

IMT Nord Europe
École Mines-Télécom
IMT-Université de Lille

Acting over a dynamic system: the agent



Rarely deterministic, Mostly uncertain

Rational Agent

"I act, therefore I am."

- My actions have an effect over the world **AND** I have the choice.
- *AI*: Model the effect **AND** explore the choices

Deliberativ Architecture - BDI:

- *Believe*: refers to the knowledge of the agent
- *Desire*: agent's goals (classically states to reach)
- *Intention*: succession of actions to perform oriented toward the goals

Acting over a system : formally

Markov Chain (Andreï Markov 1856-1922)

A tuple: $\langle States\ (S),\ Transitions\ (T)\rangle$

- **States**: set of configurations defining the studied system
- **Transitions**: Describe the possible evolution of the system state

$$T : S \times S \rightarrow [0, 1]$$
$$T(s_t,\ s_{t+1}) = P(s_{t+1}|s_t)$$

Vocabulary Parrenthesis: Hidden Markov Chain

> The system state is not directly observable.

Acting over a system : formally

Impact of the actions

- **Actions**: finite set of possible actions to perform

Updated Transition function:

The probabilistic evolution depends on the performed action.

$$T : S \times A \times S \to [0, 1]$$

$T(s^t,\ a,\ s^{t+1})$ return the probability to reach s^{t+1} by doing a from s^t:

$$T(s^t,\ a,\ s^{t+1}) = P(s^{t+1} | s^t, a)$$

Multi-Variable Systems

State and Action space:

> Cartesian product over State and Action variables

Multi-variable Transition function:

The probabilistic evolution depends on the performed action.

$$T : S \times A \times S \rightarrow [0, 1] \qquad T\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{bmatrix} \right) \in [0, 1]$$

Model of 421: States and actions

- **States**:
  - The value of each die's face ($d_n \in [1,6]$) and the re-roll number ($h \in [2,0]$)
  - So: **168** states (56 combinations over a horizon of 3).
- **Actions**:
  - The choice of roll again each die: $[roll,\ keep]$
  - so **8** actions (2^3)

Action Example :

By choosing to "roll-*keep*-roll" in state: "6-*4*-3 (2)" to expect a "4-2-1 (1)"

Model of 421: Transition function with 421-game

- **Transitions**:
  - All reachable states by rolling some dice with the probability to reach them.

$$T\left(\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ h \end{bmatrix}, \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \begin{bmatrix} d'_1 \\ d'_2 \\ d'_3 \\ h' \end{bmatrix}\right) \in [0,1] \quad \text{example: } T\left(\begin{bmatrix} 6 \\ 5 \\ 1 \\ 2 \end{bmatrix}, \begin{bmatrix} r \\ r \\ k \end{bmatrix}, \begin{bmatrix} 4 \\ 4 \\ 1 \\ 1 \end{bmatrix}\right) = 1/36$$

Model of 421: Transition function with 421-game

Transitions Example :

Choosing to "roll-*keep*-roll" from "6-*4*-3 (2)" implies *21* reachable states:

P(...)	**=**	**[0, 1]**		**P(...)**	**=**	**[0, 1]**
4-1-1 (1)	=	$1/36$		...		
4-2-1 (1)	=	$1/18$		6-*4*-4	=	$1/18$
4-2-2 (1)	=	$1/36$		6-5-*4*	=	$1/18$
...				6-6-*4*	=	$1/36$

Choosing : building a policy of actions

- *a policy* (π) : a function returning the action to perform
  Considering the current state of the system:

$$\pi : S \to A$$

$$\pi(s) : \text{ the action to perform in } s$$

Choosing : building a policy of actions

Example of policy :

"Always target a 4-2-1": keeping only one **4**, one **2** and one **1**

s	$\pi^{421}(s)$		s	$\pi^{421}(s)$
1-1-1	*keep*-roll-roll		…	
2-*1*-1	*keep*-*keep*-roll		*4*-*2*-*1*	*keep*-*keep*-*keep*
3-*1*-1	roll-*keep*-roll		…	
4-*1*-1	*keep*-*keep*-roll		6-6-5	roll-roll-roll
…			6-6-6	roll-roll-roll

(Invariant over the horizon h)

Automatize Decision Making

Choose an action in a given context (state):

- Evaluate tuples $\langle s,\ a \rangle$
- Selects the best action:

$$\pi^*(a) = \arg\max_{a \in A} \left(\ Eval(s, a)\ \right)$$

Evaluation in Game Theory

Can be done a posteriori, with a lot of data :

- $Eval(s, a) =$ the probability of winning if doing *a* in *s*.

$$Eval(s, a) = P(win \mid \pi(s) = a)$$

- But also depends on all the future actions...

Can be done a posteriori, with a lot of *good* data...

(AlphaGo mars 2016 wins 4-1 the best professional player *Lee Sedol*)

Evaluation in *4-2-1*

What is the optimal choices ?

- For example, from *6-1-1 (2)*: **roll-keep-keep** or **roll-roll-keep** ?

Scores in 421-game

Over the final combination (horizon $= 0$):

$$
\begin{aligned}
score(\text{4-2-1}) &= 800 \\
score(\text{1-1-1}) &= 700 \\
score(\text{x-1-1}) &= 400 + x \\
score(\text{x-x-x}) &= 300 + x \\
score(\text{(x+2)-(x+1)-x}) &= 202 + x \\
score(\text{2-2-1}) &= 0 \\
score(\text{x-x-y}) &= 100 + x \\
score(\text{y-x-x}) &= 100 + y
\end{aligned}
$$

Evaluation in *4-2-1*

What is the optimal choices ?

- For example, from *6-1-1 (2)*: **roll-keep-keep** or **roll-roll-keep** ?

Requires to evaluate the possible final combinations after 2 roll-agains...

- **It clearly depends on the horizon:**

  With an infinite costless roll-agains

  it is always preferable to target the **4-2-1**

Exercices, compute a policy

Basic learning process :

- Record a player experience (trace)
- Compute average values for tuples $\langle s,\ a \rangle$
- Use those values to select the actions.

But depends on the initial player behavior...

Make it iterative :

- Use your new player to create new data *and* learn again...

(AlphaGo Zero 2017 wins 100-0 AlphaGo Lee)