1. Levels and goals

I. Probability in Game Design

Giacomo Bergami

Newcaslte University

Objectives

- Characterizing and classifying video-games as AI's task environments.
- Representing AI/Game problems as graphs.
- Differentiating probability from actual execution time via strategies.

Credits

Dr. Graham Morgan

Randomness and Probability

Randomness is often associated to:

1. *disorder*
2. to *complex systems* for which we cannot grasp all the possible variables.
3. to *rationality* while avoiding the pitfalls of predictability.

A random number generator generates a given number with a given probability (PDF).

- A non-rigged cubic die generates the numbers in $I = \{ 1, 2, 3, 4, 5, 6 \}$ with an uniform probability, i.e. $\forall i \in I.\, \mathbb{P}(\text{die}{=}i) = 1/6$.
- A pair of non-rigged cubic dice generate the pairs in $I^2 = \{ (1,1), \ldots, (6,6) \}$ with an uniform probability, i.e. $\forall i \in I^2.\, \mathbb{P}(\text{2\_dice}{=}i) = 1/6^2$.

If a die returns a value $i = 5$, then the event "die=*5*" is **true** and "die=*2*" is false.

Events

Events in videogames can be:

1. Actions performed by the player (e.g., right kick).
2. Any NPC answer to the player's actions (e.g., left punch).
3. Any change of state of the environment (e.g., a new enemy is spawned).

The interplay of (1) and (2) implies that we have causation (e.g., any player action implies an environmental response). To make the game more challenging, we do not want predictable responses, and therefore we need *randomness*:

- NPC/environment events could also happen with a given probability.
- Similarly, human players' actions could be associated to a probability distribution (*stats*).

Event Ordering

- If the events are not perceived as ordered in the game play, the player would feel that changes happening in the environment (e.g., NPC reactions) are happening at random.
- Turns are a common way to structure action/reaction time frames among the players.

In this lecture, we narrow down our analysis to games (task environments) having the following characteristics:

1. **Discrete**: events are marked by **turns**.
2. **Single Agent**: a player is challenging their own faculties.
3. **Known**: all the actions and their consequences are known.
4. In particular, the player performs a set of finite actions.

Strategy

Sequential and deterministic/non-stochastic games enable strategies:

- If the game is **episodic**, learning from my previous experience will not (positively) affect my future decisions.
- If the game is **sequential**, the current decision could affect all future decisions.

The total order of a game's events provides a pattern of events:

- the recognition of specific sub-patterns might be used as *optimal heuristics* for determining the next moves $\Rightarrow$ we need either memory and/or a fully-observable environment.
- A game becomes challenging when the number of possible patterns are *either* hard to remember *or* can be enumerated in a computationally challenging amount of time.

Balancing Game Play

In medio stat virtus:

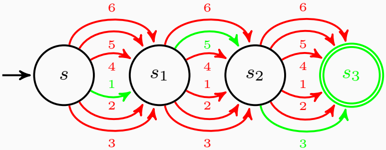
1. Too strict rules allow no free will to the player, and the game becomes boring.
2. Too much variety and/or contingencies make any action useless for state progression.
3. When both problems are combined, we lose the grasp over the correlations on these over the state progression, and we cannot assess whether the game is good or not.

Patterns of game play make the game more interesting, but they are extremely difficult to achieve.

In the creation of games, exhaustive testing is required to determine if a game is "fun":

- State progression analysis reduces the time required for such testing.
- The user should feel that his set of good choices make him progress towards the goal (e.g., life bar).

Use Case: A Simple Board Game (1/2)



- **State Space**: a set of rooms, $S = \{ s, s_1, s_2, s_3 \}$).
- **Initial State**: s, all doors are closed and not selected.
- **Actions**: pick a door available in a room, e.g. d_6.
- **Transition Model**: each door leads to a new room: $s \xrightarrow{d_6} s_1$
- **Path/Pattern**: states connected by actions.
- **Goal Test**: reaching s_3 by opening all the correct (green) doors.
- The game is **Partially Observable**: the player doesn't know a priori whether a door is correct or not (one per state).

For each state in S, there are $1/6$ chances of opening the correct door and $1 - 1/6 = 5/6$ changes of opening a wrong one.

Deterministic Automata

In our incoming lectures, we are **only** going to consider deterministic automata:

- for each pair of state s and action a there is one and only one transition to a next state s',
- if an automaton contains a state having multiple transactions "with the same label" connecting you to multiple states, you have a **non deterministic automata**, as one state might lead to the goal test, while the other might not.

Scenario #1: Locked Doors

For each state, we test the doors sequentially; all the incorrect doors are closed (but this is not known by the player until they open the door)

- Locked doors have no **applicable** actions: we can perform only one correct transition per non-accepting state.
- We **always** win: s_3 is always reached through correct doors.
- In the best case scenario, for each state we exploit an oracle to picking the correct door (1).
- In the worst case scenario, for each state (3) we need to test all the doors (6).

$\Rightarrow$ the player tests at most 18 doors and at least 3 doors.

Scenario #2: Open Doors, no Strategy

*For each state, we pick one door; all the incorrect doors are left open. If we reach s_3 through at least one incorrect door (i.e., **fail**), we backtrack to s and keep the same three correct doors.*

- As there is only one correct door per state, we have $(1/6)^3$ chances of winning and $1 - (1/6)^3 = 215/216$ chances of losing.
- If the player uses no strategy (i.e., by not remembering the previous combination), in the worst case scenario they will never get the right door triplet (*livelock*).
- The best case scenario is the same.

Scenario #3: Open Doors, with Strategy

*For each state, we pick one door; all the incorrect doors are left open. If we reach s_3 through at least one incorrect door (i.e., **fail**), we backtrack to s and keep the same three correct doors.*

For avoiding the livelock, we can keep track of all the failing door strategies:

- We might guess that each single door d_i at a given state h is not correct after visiting 36 failing strategies containing the same door.
- We might guess that each pair of doors d_i (and d_j) at a given state h (and k with $h \neq k$) is not correct after visiting 6 failing doors at a given state $l \neq h$ and $l \neq k$.

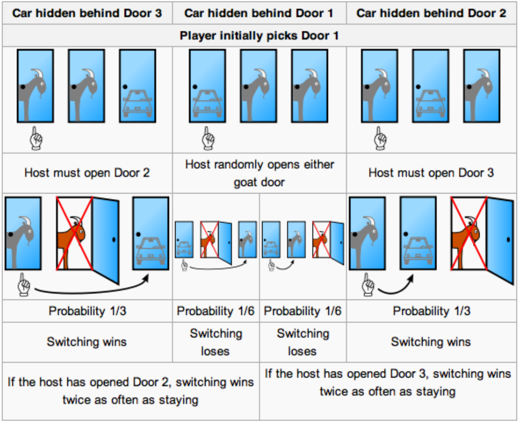
In the worst case scenario, we will backtrack $6^3 - 1$ times before guessing the right door triplet.

Scenario #4: Stochastic selection of correct doors at failure

*For each state, we pick one door. If we reach s_3 through at least one incorrect door (i.e., **fail**), we backtrack to s and randomly select another triplet of correct doors.*

- The games becomes **episodic**, as the actions to perform for reach the goal test in next turn after a failure might not be related to any other strategy or door configuration.
- In the worst case scenario, the player is stuck in a livelock as in the *no Strategy* scenario.

Monty Hall Problem (1/2)



In such a game, we have one room, and only one winning door over three. After picking one door, the environment shows you one loosing door among the remaining ones. Then, you are allowed to change your door before making your final decision.

Monty Hall Problem (2/2)

- If you stick with your original decision, you will always have a $1/3$ chance of winning
- but, if you switch to another door, you have a $2/3$ chance of winning!

In the most general formulation of the game by *D. L. Ferguson* (1975), we have one room, and only one winning door over N. After picking one door, the environment shows you $p < N - 1$ loosing doors among the remaining ones. Then, you are allowed to change your door before making your final decision.

- If you stick with your original decision, you will always have a $1/N$ chance of winning
- but, if you switch to another door, you have a $\frac{1}{N} \cdot \frac{N-1}{N-p-1}$ chance of winning!

Scenario #5: Stochastic selection + Monty Hall

*For each state, we pick one door after playing the Monty Hall Game. If we reach s_3 through at least one incorrect door (i.e., **fail**), we backtrack to s and randomly select another triplet of correct doors.*

- The games is still **episodic**.
- In the worst case scenario, the player is stuck in a livelock, but probability of this losing each time greatly reduces the more p tends to $N-2$.

Considering state progression

- Searching the game state exhaustively might not be applicable for even the most simplistic of games.
- Additional players might make such a search even more impractical.
- Completely removing the notion of turns makes such a search even more impractical.
- However, isolated scenarios of a game may be analyzed.
  - In an adventure style games, just considering quests might ease the search of the state space.
  - Ignoring any mathematical analysis is not to be advised.