Algorithmic Game Theory and Applications

Lecture 15: a brief taster of Markov Decision Processes and Stochastic Games

Kousha Etessami

warning

▷ The subjects we will touch on today are vast: one can easily spend an entire course on them alone.

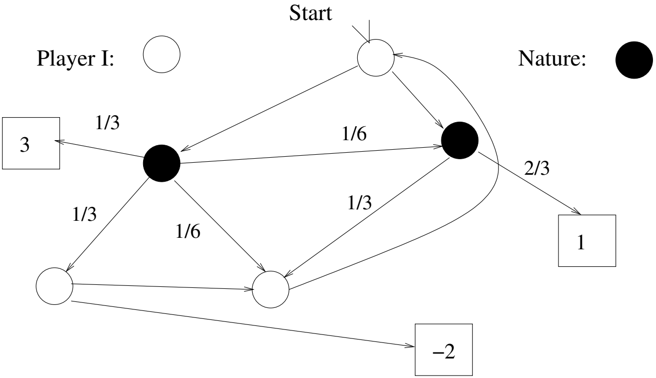
▷ So, what we discuss today is only a brief "taster". Please do explore further if the subject interests you.

▷ Here are two standard textbooks that you can look up if you are interested in learning more:

- M. Puterman, *Markov Decision Processes*, Wiley, 1994.
- J. Filar and K. Vrieze, *Competitive Markov Decision Processes*, Springer, 1997. (For 2-player zero-sum stochastic games.)

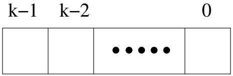
Games against Nature

Consider a game graph, where some nodes belong to player 1 but others are chance nodes of "Nature":



Question: What is Player 1's "optimal strategy" and "optimal expected payoff" in this game?

a simple finite game: “make a big number”



- Your goal: create as large a k-digit number as possible, using digits from $D = \{0, 1, 2, \ldots, 9\}$, which “nature/chance” will give you, one by one.
- The game proceeds in k rounds.
- In each round, “nature” chooses $d \in D$ “uniformly at random”, i.e., each digit has probability $1/10$.
- You then choose which “unfilled” position in the k-digit number should be “filled” with digit d. (Initially, all k positions are “unfilled”.)
- The game ends when all k positions are “filled”.

Your goal: maximize final number’s expected value.

Question: What should your strategy be?

▷ This is a “finite horizon” “Markov Decision Process”.

▷ Note that this is a finite PI-game and, *in principle*, we can solve it using the “bottom-up” algorithm.

▷ But we wouldn’t want to look at the entire tree if we can avoid it!

vast applications

Beginning in the 1950's with the work of Bellman, Wald, and others, these kinds of "Games Against Nature", a.k.a., "Markov Decision Processes", a.k.a. "Stochastic Dynamic Programs", have been applied to a huge range of subjects.

Examples where MDPs have actually been applied:

- highway repair scheduling.
- bus engine replacement scheduling.
- waste management.
- call center scheduling.
- . . .

(See [Puterman'94,Ross'83,Derman'72,Howard'70,Bellman'57,..])

▷ "Reinforcement Learning" (RL), see e.g., [Sutton-Barto'98], is founded on the underlying model of MDPs.

▷ However, in RL, the MDP itself is often not fully visible, and one has to "discover" it by a process of exploration/exploitation.

- The richness of applications shouldn't surprise you.
- We live in an uncertain world, where we constantly have to make decisions in the face of uncertainty about future events.
- But we may have some information, or "belief", about the "likelihood" of future events.
- "I know I may get hit by a car if I walk out of my apartment in the morning." But somehow I still muster the courage to get out.
- I don't however walk into a random pub in Glasgow and yell "I LOVE CELTIC FOOTBALL CLUB", because I know my chances of survival are far lower.

Markov Decision Processes

Definition A **Markov Decision Process** is given by a game graph $G_{v_0} = (V, E, pl, q, v_0, u)$, where:

- V is a (finite) set of vertices.
- $pl : V \mapsto \{0, 1\}$, maps each vertex either to player 0 ("Nature") or to player 1.
- Let $V_0 = pl^{-1}(0)$, and $V_1 = pl^{-1}(1)$.
- $E : V \mapsto 2^V$ maps each vertex v to a set E(v) of "successors" (or "actions" at v).
- For each "nature" vertex, $v \in V_0$, a probability distribution $q_v : E(v) \mapsto [0, 1]$, over the set of "actions" at v, such that $\sum_{v' \in E(v)} q_v(v') = 1$.
- A start vertex $v_0 \in V$.
- A **payoff function**: $u : \Psi_{T_{v_0}} \mapsto \mathbb{R}$, from plays to payoffs for player 1.

Player 1 want to *maximize its expected payoff*.

Many different payoff functions

Many different payoff functions have been studied in the MDP and stochastic game literature. Examples:

1. **Mean payoff**: for every state $v \in V$, associate a payoff $u(v) \in \mathbb{R}$ to that state. For a play $\pi = v_0 v_1 v_2 v_3 \ldots$, the goal is to maximize the expected *mean payoff*:

$$\mathbb{E}(\liminf_{n\to\infty} \frac{\sum_{i=0}^{n-1} u(v_i)}{n})$$

Many different payoff functions

Many different payoff functions have been studied in the MDP and stochastic game literature. Examples:

1. **Mean payoff**: for every state $v \in V$, associate a payoff $u(v) \in \mathbb{R}$ to that state. For a play $\pi = v_0 v_1 v_2 v_3 \ldots$, the goal is to maximize the expected *mean payoff*:

$$\mathbb{E}(\liminf_{n\to\infty} \frac{\sum_{i=0}^{n-1} u(v_i)}{n})$$

2. **Discounted total payoff:** For a given **discount** factor $0 < \beta < 1$, the goal is to maximize:

$$\mathbb{E}(\lim_{n\to\infty} \sum_{i=0}^{n} \beta^i u(v_i))$$

Many different payoff functions

Many different payoff functions have been studied in the MDP and stochastic game literature. Examples:

1. **Mean payoff**: for every state $v \in V$, associate a payoff $u(v) \in \mathbb{R}$ to that state. For a play $\pi = v_0 v_1 v_2 v_3 \ldots$, the goal is to maximize the expected *mean payoff*:
$$\mathbb{E}(\liminf_{n \to \infty} \frac{\sum_{i=0}^{n-1} u(v_i)}{n})$$
2. **Discounted total payoff:** For a given **discount** factor $0 < \beta < 1$, the goal is to maximize:
$$\mathbb{E}(\lim_{n \to \infty} \sum_{i=0}^{n} \beta^i u(v_i))$$
3. **Probability of reaching target:** Given target $v_T \in V$, goal: maximize (or minimize) probability of reaching v_T. Can be rephrased: for a play $\pi = v_0 v_1 v_2 \ldots$, let $\chi(\pi) := 1$ if $v_i = v_T$ for some $i \geq 0$. Otherwise, $\chi(\pi) := 0$. Goal: maximize/minimize $\mathbb{E}(\chi(\pi))$.

Expected payoffs

▷ Intuitively, we want to define expected payoffs as the sum of payoffs of each play times its probability.

▷ However, of course, this is not possible in general because after fixing a strategy it may be the case that all plays are infinite, and every infinite play has probability 0!

▷ In general, to define the expected payoff for a fixed strategy requires a proper *measure theoretic* treatment of the probability space of infinite plays involved, etc.

▷ This is the same thing we have to do in the theory of *Markov chains* (where there is no player).

▷ We will avoid all the heavy probability theory.
(You have to take it on faith that the intuitive notions can be formally defined appropriately, or consult the cited textbooks.)

memoryless optimal strategies

A **strategy** is again any function that maps each history of the game (ending in a node controlled by player 1), to an *action* (or a probability distribution over actions) at that node.

Theorem *(memoryless optimal strategies)* For every finite-state MDP, with any of the the following objectives:

- *Mean payoff*,
- *Discounted total payoff*, or
- *Probability of reaching target*,

player 1 has a pure memoryless optimal strategy.

In other words, player 1 has an optimal strategy where it just picks one edge from $E(v)$ for each vertex $v \in V_1$.

(For a proof see, e.g., [Puterman'94].)

Bellman Optimality Equations

For the objective of **maximizing probability to reach target vertex** v_T, consider the following system of equations. Let $V = \{v_1, \ldots, v_n\}$ be the set of vertices of the MDP, G. Consider the following system of equations, with one variable x_i for every vertex v_i.

$$\begin{array}{rcl} x_T & = & 1 \\ x_i & = & \max\{x_j \mid v_j \in E(v_i)\} \quad \text{for } v_i \in V_1 \\ x_i & = & \displaystyle\sum_{v_j \in E(v_i)} q_{v_i}(v_j) \cdot x_j \quad \text{for } v_i \in V_0 \end{array}$$

Theorem *These max-linear* **Bellman equations** *for the MDP, have a (unique) least non-negative solution vector* $x^* = (x_1^*, \ldots, x_n^*) \in [0,1]^n$*, in which* x_i^* *is the optimal probability for player 1 to reach the target* v_T *in the MDP* G_{v_i} *starting at* v_i*.* We won't prove this (but it is not difficult).

computing optimal values

One way to compute the solution x^* for the Bellman equations $\mathrm{x} = L(\mathrm{x})$ is **value iteration**: consider the sequence $L(0)$, $L(L(0))$, ..., $L^m(0)$. **Fact**: $\lim_{m \to \infty} L^m(0) = x^*$. Unfortunately, value iteration can be very slow in the worst case (requiring exponentially many iterations). Instead, we can use LP. Let $V = \{v_1, \ldots, v_n\}$ be the vertices of MDP, G. We have one LP variable x_i for each vertex $v_i \in V$.

Minimize $\sum_{i=1}^{n} x_i$

Subject to:

$x_T = 1$;

$x_i \geq x_j$, *for each* $v_i \in V_1$, *and* $v_j \in E(v_i)$;

$x_i = \sum_{v_j \in E(v_i)} q_{v_i}(v_j) \cdot x_j$, *for each* $v_i \in V_0$;

$x_i \geq 0$ *for* $i = 1, \ldots, n$.

Theorem *For* $(x_1^*, \ldots, x_n^*) \in \mathbb{R}^n$ *an optimal solution to this LP (which must exist), each* x_i^* *is the optimal value for player 1 in the game* G_{v_i}. (This follows from Bellman equations.)

extracting the optimal strategy

Suppose you computed the optimal values x^* for each vertex. How do you find an optimal (memoryless) strategy?
One way to find an optimal strategy for player 1 in this MDPs is to solve the **dual LP**.
First, remove all vertices v_i such that the maximum probability of reaching the target from v_T is 0. This is easy to do, by just doing reachability analysis on the underlying graph of the MDP, and ignoring probabilities.
Once this is done, it turns out that an optimal solution to the dual LP encodes an optimal strategy of player 1 in the MDP associated with the primal LP. And, furthermore, if you use Simplex, the optimal basic feasible solution to the dual will yield a pure strategy. (Too bad we don't have time to prove this.)

Stochastic Games

What if we introduce a second player in the game against nature?

In 1953 L. Shapley, one of the major figures in game theory, introduced "stochastic games", a general class of zero-sum, not-necessarily perfect info, two-player games which generalize MDPs. This was about the same time that Bellman and others were studying MDPs.

In Shapley's stochastic games, at each state, both players simultaneously and independently choose an action. Their joint actions yield both a 1-step reward, and a probability distribution on the next state.

We will confine ourselves to a restricted perfect information stochastic games where the objective is the probability of reaching a target.

These are callled "simple stochastic games" by [Condon'92].

simple stochastic games

Definition A zero-sum **simple stochastic game** is given by a game graph $G_{v_0} = (V, E, pl, q, v_0, u)$, where:

▷ V is a (finite) set of vertices.

▷ $pl : V \mapsto \{0, 1, 2\}$, maps each vertex to one of player 0 ("Nature"), player 1, or player 2.

▷ Let $V_0 = pl^{-1}(0)$, $V_1 = pl^{-1}(1)$, & $V_2 = pl^{-1}(2)$.

▷ $E : V \mapsto 2^V$ maps each vertex v to a set E(v) of "successors" (or "actions" at v).

▷ Let $V_{dead} = \{v \in V \mid E(v) = \emptyset\}$.

▷ For each "nature" vertex, $v \in V_0$, a probability distribution $q_v : E(v) \mapsto [0, 1]$, over the set of "actions" at v, such that $\sum_{v' \in E(v)} q_v(v') = 1$.

▷ A start vertex $v_0 \in V$.

▷ A target vertex $v_T \in V$.

memoryless determinacy

▷ The goal of player 1 is to *maximize* the probability of hitting the target state v_T.
▷ The goal of of player 2 is to *minimize* this probability. (So, the game is a zero-sum 2-player game.)
▷ We call the game *memorylessly determined* if both players have (deterministic) memoryless optimal strategies.

Theorem([Condon'92]) Every simple stochastic game is memorylessly determined.

computing optimal strategies

▷ Memoryless determinacy immediately gives us one algorithm for computing optimal strategies:

- "Guess" the strategy for one of the two players.
- The "residual game" is a MDP; solve corresponding LP.

▷ This gives a **NP** ∩ **co-NP** procedure for solving simple stochastic games.

▷ [Hoffman-Karp'66] studied a "strategy improvement algorithm" for stochastic games based on LP, which can be adapted to simple stochastic games ([Condon'92]). Strategy improvement works well in practice, but recent results show that this algorithm requires exponential time for both MDPs and stochastic games, with SOME objective functions.

▷ Is there a P-time algorithm for solving simple stochastic games? This is an open problem.

▷ Solving parity games and mean payoff games (Lecture 13) can be reduced to solving SSGs ([Zwick-Paterson'96]).

food for thought

> *What is the relationship between computing Nash Equilibria in finite (two-person, n-person) strategic games and computing solutions to (simple) stochastic games?*

In other words:

> *What does Nash have to do with Shapley?*

To put it more concretely: is either computational problem efficiently reducible to the other?
ANSWER: it turns out that both computing the value of Simple Stochatic Games, and approximating the (irrational) value of Shapley's Stochastic Games are reducible to computing a NE in 2-player strategic games. In other words, both problems are in **PPAD**.
(See [Etessami-Yannakakis,'07,SICOMP'10]).