Online Markov Decision Processes

Iheb Belgacem

February 2020

1 Introduction

Markov Decision Processes provide a very useful framework for Decision Making under uncertainty. They enable the modeling of agents which given the state of the environment they perceive, follow certain action policies and receive rewards. MDPs are widely used in many fields : Robotics, Finance, Logistics ...
MDPs can be combined with Online Convex Optimization to model situations where the agent is aware of the transition probabilities between states given the actions chosen, but where the rewards are selected dynamically by an adversary. This work is based on the study of Large Scale Markov Decision Processes with Changing Rewards made by Cardoso A.R., Wang H. and Xu H..[2]
In this article, we propose to start by deriving a Linear Programming (2) formulation of the Average Reward MDP (3) , in dual form. Then we present the Follow The Regularized Leader (4) method and the online learning framework in general. The next step will be to use the results from the previous sections to explain the online learning algorithm used in [2] : which is based on the FTRL method applied to the dual form of the linear optimization formulation of the MDP problem. And finally we explain how the authors managed to make the algorithm more computationally efficient and adapted to large scale MDPs.

2 Linear Programming

A convex optimization problem is an optimization problem in which the objective function is convex and the feasible set is also convex.
A set S is convex if and only if for any x_1 and x_2 in S and $\forall \lambda \in [0, 1]$ we have $\lambda \times x_1 + (1 - \lambda) \times x_2 \in S$.
Examples for convex sets are the half-spaces H_+ and H_- defined by $H_+ = \{x \in R^n | p^T x \geq \alpha\}$ and $H_- = \{x \in R^n | p^T x \leq \alpha\}$ with $p \in R^n$ a normal vector for the hyper-plane H defined by $H = \{x \in R^n | p^T x = \alpha\}$
It can be easily proven using the definition of convexity for sets that the intersection of convex sets is also a convex set.
A function f defined over a convex set S is convex if and only if for any x_1 and x_2 in S and $\forall \lambda \in [0, 1]$ we have $f(\lambda \times x_1 + (1-\lambda) \times x_2) \leq \lambda \times f(x_1) + (1-\lambda) \times f(x_2))$.
Linear functions are convex functions.

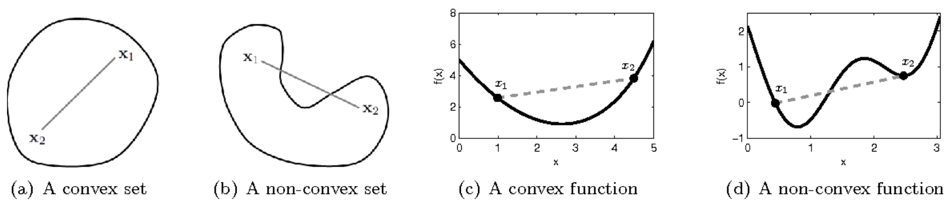

Figure 1: Convexity for sets and functions [5]

Using the previous definitions we can say that linear optimization problems defined by :

$$\min_{x \in R^n} c^T x \tag{1}$$

$$s.t \;\; Ax \geq b$$

with $x \in R^n$, $c \in R^n$, $b \in R^m$ and $A \in R^{m \times n}$, are a special form of convex optimization problems.
The feasible set of a linear optimization problem, which is the intersection of a finite number of half-spaces, is closed and convex. we call it a polyhedral set (or polyhedron).
An extreme point of a convex set is a point of the set that cannot be written as the convex combination of two distinct points in the set.
Similarly, an extreme direction is a direction that cannot be written as a positive linear combination of two distinct directions. It can be proven that every point inside a polyhedron can be written as a combination of its extreme points and extreme directions.(This fact is known as the representation theorem of polyhedral sets.) It can be shown using the representation theorem that if the LP

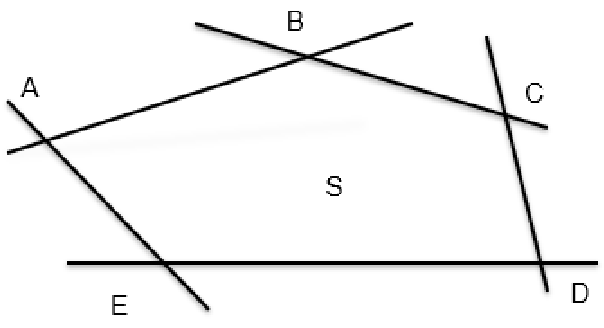

Figure 2: Polyhedral set in R^2 with 5 extreme points and no extreme directions

program has a finite solution, than one of the extreme points is an optimizer of

the LP problem.
Linear optimization problems can be solved efficiently using the simplex algorithm. The simplex algorithm uses the previous results to find the optimizer for the LP problem. In every iteration, the algorithm moves from one extreme point to another extreme point with a lower objective value.[6]
As for any convex optimization problem, we can introduce the Lagrangian dual problem of any LP Problem. For a general convex optimization problem :

$$\min_{s\in S} f(x) \tag{2}$$

$$s.t \;\; g_i(x) \leq 0 \;\; \forall i \in \{1..l\}$$

$$s.t \;\; h_i(x) = 0 \;\; \forall i \in \{1..m\}$$

The dual Lagrangian problem is defined as :

$$\max_{u\in R^l, v\in R^m} \Theta(u,v) \tag{3}$$

$$s.th \;\; u \geq 0$$

where $\Theta(u,v) = \min_{x\in S}\phi(x,u,v)$ and $\phi(x,u,v) = f(x) + u^T g(x) + v^T h(x)$
For the LP problem we have :

$$\phi(x,u) = c^T x + u^T(-Ax+b)$$

$$u \geq 0, u \in R^m$$

$$\Theta(u) = \min_{x\in S}\phi(x,u)$$

$$\Theta(u) = \min_{x\in S} c^T x + u^T(-Ax+b)$$

$$\Theta(u) = \min_{x\in S}(c^T - u^T A)x + u^T b$$

$$\Theta(u,v) = u^T b \;\; if \;\; A^T u = c$$

$$\Theta(u,v) = -\infty \;\; otherwise$$

We obtain the dual problem for the initial linear optimization problem :

$$\max_{u\in R^m} b^T u \tag{4}$$

$$s.th \;\; A^T u = c$$

$$u \geq 0$$

The strong duality theorem holds in the case of linear optimization problems. It means that if the primal has an optimal value p^*, than the dual has also an optimal value d^* and $p^* = d^*$.
If the primal problem has way more constraints that it has decision variables, it may be very efficient to solve the dual problem instead.

3 Markov Decision Process

A Markov Decision Process is a framework for modeling decision making under uncertainty. In the context of MDPs, an agent observes the state of a stochastic environment and takes an action from a set A of possible actions and receives a reward R(s). The system than transitions to a new state s'. The transition probabilities to a new state s' only depend on the previous state s and the last action a taken by the agent : We say that the state transitions of an MDP satisfy the Markov property.

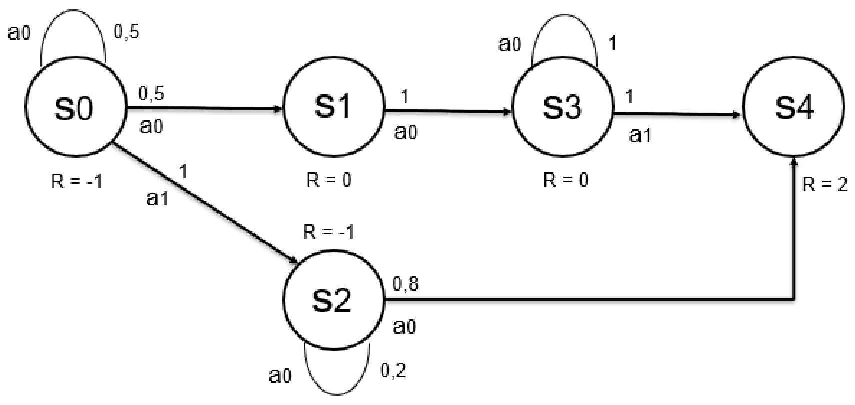

Figure 3: An example of a Markov Decision Process with $|S| = 5$

The goal of the agent is to adopt a policy that maximizes the total expected rewards.
A policy π is a mapping from the state space S into the action space A.

$$\pi : S \to A : \pi(s) = a \quad \forall s \in S$$

The total expected reward for an agent starting from a state s is known as the value function of the state. The values of the different states in the state space for a specific policy π can be determined recursively using the famous Bellman's Equations :

$$V(s) = R(s) + \gamma \times \sum_{s' \in S} (P(s'|s, \pi(s)) * V(s')) \quad \forall s \in S \tag{5}$$

γ is a real number in $[0, 1]$ known as the discount factor. It represents for the agent the importance it gives to future rewards compared to present rewards.
We say that a policy π_1 dominates a policy π_2 if :

$$V^{\pi_1}(s) \geq V^{\pi_2}(s) \quad \forall s \in S \text{ and } \exists s \in S \text{ such that } V^{\pi_1}(s) > V^{\pi_2}(s)$$

It can been easily proven that for every MDP, there exists an optimal policy π^* And in the case of the optimal policy we can write the Bellman equation in the following way :

$$V^*(s) = R(s) + \gamma \times \max_{a \in A(s)} \sum_{s' \in S} (P(s'|s,a) * V^*(s')) \quad \forall s \in S \tag{6}$$

If we have the values of the states for the optimal policy, we can very easily derive the actions associated with it, using the following formula :

$$\forall s \in S, \pi^*(s) = \arg\max_{a \in A(s)} \sum_{s' \in S} P(s'|s,a) \times V^*(s') \tag{7}$$

There are many possible methods to find the optimal policy for a MDP (Policy iteration methods, value iteration methods ...). In this article, we are going to concentration on LP approaches. We can modelize the MDP problem as a LP Problem by transforming Eq. 6:

$$V^*(s) = R(s) + \gamma \times \max_{a \in A(s)} \textstyle\sum_{s' \in S} (P(s'|s,a) * V^*(s')) \quad s \in S$$

into $|A|$ inequalities :

$$V^*(s) \geq R(s) + \gamma \times \textstyle\sum_{s' \in S} (P(s'|s,a) * V^*(s')) \quad s \in S \quad \forall a \in A(s)$$

Thus, we will have in total $|S| \times |S|$ linear inequality constraints. (Provided we have for each state s, s possible transitions, i.e to all states).
A good objective function would be : $\min_{V(s) \in R} \sum_{s \in S} V(s) \times W(s)$
With W(s) a probability distribution over S, such as $W(s) > 0 \quad \forall s \in S$
We can choose for example $W(s) = 1/|S| \quad \forall s \in S$ (of course we can substitute $W(s) = 1/|S|$ by 1 without changing the optimization problem.)

To sum up, a LP formulation for the MDP problem is :

$$\min_{V(s) \in R} \sum_{s \in S} V(s) \times 1/|S| \tag{8}$$

$$s.t \quad V(s) - \gamma \times \textstyle\sum_{s' \in S} (P(s'|s,a) * V(s')) \geq R(s) \quad \forall s \in S \quad \forall a \in A(s)$$

If we take the dual of this problem we obtain :

$$\max_{\mu(s,a) \in R} \sum_{s \in S} (R(s) \times \sum_{a \in A} \mu(s,a)) \tag{9}$$

$$s.t \quad \textstyle\sum_{a \in A} \mu(s',a) - \nu \times \sum_{s \in S} \sum_{a \in A} (P(s'|s,a) * \mu(s,a)) = 1/|S| \quad \forall s' \in S$$

$$\mu(s,a) \geq 0 \quad \forall s \in S \quad \forall a \in A$$

If we interpret the dual variables $\mu(s,a)$ as the number of times we are in state s and make the action a [4]: the term $1/|S| + \nu \times \sum_{s\in S}\sum_{a\in A}(P(s'|s,a) * \mu(s,a))$ would correspond to the expected number of times that we arrive in a given state s'.
In fact, we have a $1/|S|$ probability of the first state being s', and, every time that we are in a state s and take action a, we have a probability of $P(s'|s,a)$ that the MDP will end up in s' in the next round. This must be equal to $\sum_{a\in A}\mu(s',a)$ which is equal to the number of times we leave the state s'. The value of the objective function correspond to the expected reward.
Once we obtain the dual variables $\mu(s,a)\ \ \forall s \in S$ and $\forall a \in A$, we can find the optimal policy π^* in the following way :

$$\pi^*(s) = \arg\max_{a\in A} \mu(s,a) \quad \forall s \in S \tag{10}$$

If we go back to the primal problem in Eq. 8 :

$$\min_{V(s)\in R} \textstyle\sum_{s\in S} V(s) \times 1/|S|$$

$$s.t \quad V(s) - \gamma \times \textstyle\sum_{s'\in S}(P(s'|s,a) * V(s')) \geq R(s,a) \quad \forall s \in S \quad \forall a \in A(s)$$

We know that an important issue we may encounter is the problem known as the "Curse of Dimensionalty" : For complex MDPs, the cardinality of the state space may become very large. To tackle this problem, we may want to approximate the value function $V(s)\ \ \forall s \in S$. To achieve that, we should start by designing suitible feature $\theta(s)\ \ \forall s \in S$ with $\theta(s) \in R^d$ and $d << |S|$. [1] These features are of course problem-dependant. The goal is to find a good linear regression model that will map our features $\theta(s) \forall s \in S$ to $V(s)$.
The best linear regression model is defined as :

$$\Phi^* = \arg\min_{\Phi\in R^d} \sum_{s\in S}(\Phi^T\theta(s) - V(s))^2 \tag{11}$$

In this optic, we can modify the primal problem by substituting the value function $V(s)$ with $\Phi^T\theta(s)$ and minimizing over Φ. And this way we will reduce the number of our decision variables from $|S|$ to $d << |S|$. We can make the same reasoning for the dual problem .
To sum up, the Approximate Primal Linear Optimization model of MDP is :

$$\min_{\Phi\in R^d} \sum_{s\in S} \Phi^T\theta(s) = \min_{\Phi\in R^d} \sum_{s\in S}\sum_{i=1}^{d} \Phi_i\theta_i(s) \tag{12}$$

$$s.t \quad \Phi^T\theta(s) - \gamma \times \textstyle\sum_{s'\in S}(P(s'|s,a) * \Phi^T\theta(s')) \geq R(s,a) \quad \forall s \in S \quad \forall a \in A(s)$$

The approximate variation of the dual problem is :

$$\max_{\Phi\in R^d} \sum_{s\in S}\sum_{a\in A} \Phi^T\theta(s,a) \times R(s,a) \tag{13}$$

$$s.t \quad \sum_{a\in A} \Phi^T\theta(s',a) - \nu \times \sum_{s\in S} \sum_{a\in A} (P(s'|s,a) * \Phi^T\theta(s,a)) = 1/|S| \quad \forall s' \in S$$

$$\Phi^T\theta(s,a) \geq 0 \quad \forall s \in S \quad \forall a \in A$$

We should note that the design of the features is more intuitive in the case of the primal problem.

4 Online Optimization

Online learning is a well-established learning paradigm. Its goal is to make a sequence of accurate predictions given knowledge of the correct answer to previous prediction tasks and possibly additional available information.
An important problem in online learning is the online optimization game. This game is a sequential game played against an intelligent adversary. The player can choose at each time an action from a set of possible actions A. As a result of this action, the player receives a loss $l_i(a)$ that the adversary chooses each time, just before the player shares his intention. The goal of this game is to minimize the players regret : which is defined as the difference between the actual loss suffered by the player and the losses of the best fixed action in hindsight.

$$Regret = \sum_{i=1}^{i=N} l_i(a_i) - \min_{a\in A} \sum_{i=1}^{i=N} l_i(a) \tag{14}$$

A natural approach to this problem is to choose at time i the action that until time i-1 would have resulted in the smallest loss. This approach is know as Follow the Leader :

$$a_i = \arg\min_{a\in A} \sum_{j=1}^{j=i-1} l_j(a) \tag{15}$$

This approach may seem very appealing at first, but it can be improved in a significant way. To see that let's consider a game where the action space is $1, -1$ and the loss function chosen by the adversary l such that : $l_1 = a/2$ and $\forall i \in [2..N]$ $l_i = a$ if i is odd and $l_i = -a$ if i is even. [3]

i	$l_j(1)$	$l_j(-1)$	$\sum_{j=1}^{j=i-1} l_j(1)$	$\sum_{j=1}^{j=i-1} l_j(-1)$	choice	loss
1	1/2	-1/2	X	X	1	1/2
2	-1	1	1/2	-1/2	-1	1
3	1	-1	-1/2	1/2	1	1
4	-1	1	1/2	-1/2	-1	1
5	1	-1	-1/2	1/2	1	1

We saw in the previous example that we had linear regret, caused by the lack of stability. We can improve this by adding a regularization term ϕ. We obtain the Follow the regularized Leader Method :

$$a_i = \arg\min_{a\in A} \sum_{j=1}^{j=i-1} (\nu \times l_j(a) + \phi(a)) \tag{16}$$

With ν is a tuning parameter.
We can instead of choosing one action in a deterministic way at each round choose a sample from the probability distribution of actions we think would minimize the loss. Our system becomes :

$$p_i = \arg\min_p \sum_{j=1}^{j=i-1} \nu \times p_j^T l_j + \phi(p_j) \tag{17}$$

A possible choice for the regularization function is the negative of the entropy $\phi(p) = \sum_i pi \times log(pi)$. The entropy is maximized for a uniform distribution, that's why this regularization term will stabilize our predictor.

5 A sublinear Regret Algorithm for Online MDP (using FTRL)

Now we can use all the preliminary results from the previous sections to explain the Online Algorithm developed in [2].
Let's go back to the Eq. 9, which represent the classical LP Formulation of the MDP optimal policy problem in dual form :

$$\max_{\mu(s,a)\in R} \sum_{s\in S}(R(s) \times \sum_{a\in A} \mu(s,a)) \tag{18}$$

$$s.t \quad \textstyle\sum_{a\in A} \mu(s',a) - \nu \times \sum_{s\in S} \sum_{a\in A}(P(s'|s,a) * \mu(s,a)) = 1/|S| \quad \forall s' \in S$$

$$\mu(s,a) \geq 0 \quad \forall s \in S \quad \forall a \in A$$

With the dual variables $\mu(s,a)$ representing the number of times we are in state s and make the action a.
We can normalize the $\mu(s,a)$ (for example to 1) by adding the following constraint :

$$\sum_{s\in S} \sum_{a\in A} \mu(s,a) = Cte = 1 \tag{19}$$

Considering the situation described in the introduction : MDP with varying reward and known transition probabilities, we can't use the dual LP because we don't have access at time t before we take the action to the reward function r_t.
The Regularized Follow The Leader approach is more useful in this context.
For the MDP-RFTL Algorithm developed in [2], we will calculate at each timestep $\mu_t(s,a)$ $\forall s \in S$ and $\forall a \in A$ using the RFTL method.

1. First, we initialize $\mu_1(s,a)$ with any feasible values. ($\mu_1(s,a)$ $\forall s \in S$ and $\forall a \in A$ should respect the $|S|$ equality constraints and $|A| \times |S|$ inequality constraints of the dual linear optimization problem).

2. We iterate for i = 1, ...N (length of the "game"):

- We observe the current state s_i
- If $\sum_{a\in A} \mu_i(s_i, a) > 0$
  - We choose the action to be made exactly as we did in Section 3 : $a_i = \arg\max_{a\in A} \mu(s_i, a)$
  - Otherwise we pick randomly an action $a \in A$
- We observe the reward $r_i(s') = r_i(s, a)$ for the next state s'
- The $\mu_{i+1}(s, a)\ \forall s \in S$ and $\forall a \in A$ are calculated with FTRL as the values that maximize the cumulative reward observed so far plus a regularization term

$$\mu_{i+1} = \underset{\mu feasible}{\arg\max} \sum_{j=1}^{i} [< r_j, \mu > -R(\mu)/\nu] \tag{20}$$

With (provided the normalization constraint in Eq. 19 is respected)

$$< r_j, \mu >= \sum_{s\in S} \sum_{a\in A} r_j(s) \times \mu(s, a) \tag{21}$$

The Regularization term can be the negative of the entropy. (to force the stabilization = to avoid $\mu(s, a)$ to change drastically)

$$R(\mu) = \sum_{s\in S} \sum_{a\in A} \mu(s, a) \times ln(\mu(s, a)) \tag{22}$$

The authors of the article [2] have proven that the previous algorithm has sub-linear regret. With regret defined as :

$$MDP - Regret(N) = \sup_{\pi} \mathrm{E}[\sum_{i=1}^{N} r_i({s_i}^{\pi}, {a_i}^{\pi})] - \mathrm{E}[\sum_{i=1}^{N} r_i(s_i, a_i)] \tag{23}$$

More precisely, it has been proven that :

$$MDP - Regret(N) \leq O(\tau + 4\sqrt{\tau N(ln(|S| + ln(|A|)ln(N)}) \tag{24}$$

for $|R(s)| \leq 1$ and for big enough N. τ being the mixing time for MDP .

When the state space considered for our problem is large, it is always useful to consider approximations of the $\mu(s, a)\ \forall s \in S$ and $\forall a \in A$.
We want to approximate $\mu(s, a)$ like in Eq. 13 by $\Phi^T \theta(s, a)$ with $\Phi \in R^d$,d : number of features, and maximize instead over Φ .
In this case the FTRL Algorithm is modified. The action is selected this time according to the rule :

$$a_i = \arg\max_{a\in A} {\Phi_i}^T \theta(s_i, a)$$

The update rule becomes :

$\Phi_{i+1} = \arg\max_{\Phi \in R^d} \sum_{j=1}^{i-1} [\sum_{s \in S} \sum_{a \in A} r_j(s) \times {\Phi_j}^T \theta(s,a) - R^\alpha({\Phi_j}^T \theta(s,a)]$

With R^α is a modified version of the negative entropy function, that has a finite gradient at 0.
The previous maximization problem can be solved efficiently with the Stochastic Gradient Ascent Algorithm. And for the whole procedure, we should make sure that the Φ_i we calculate for every step i are included in the feasible set of the approximate dual problem for MDPs. We should of course make sure to initialize correctly Φ. We also note that the authors of [2] chose to take out some of the constraints in the dual formulation of the approximation of the problem and transformed them as penalty terms in the objective function to simplify the feasible set. They proved in their paper they achieved a good bound on the adapted version of the Regret. Namely :

$$\Phi - MDP - Regret(N) \leq O(cte \times ln(|S||A|)\sqrt{\tau T l} \times ln(T)) \tag{25}$$

6 Conclusion

This work is a summary of the study of Large Scale Markov Decision Processes with Changing Rewards made by Cardoso A.R., Wang H. and Xu. We analyzed the case when we have known transition properties but changing rewards set by an adversary. We explained the approach used by the authors: who applied the Regularized Follow The Leader method to the dual form of the linear programming formulation of the MDP. And we explained an efficient way to reduce the computational complexity of the problem using function approximation and stochastic gradient ascent.

References

[1] Pieter Abbeel. Advanced robotics lecture.

[2] Adrian Rivera Cardoso, He Wang, and Huan Xu. Large scale markov decision processes with changing rewards, 2019.

[3] Nicolo Cesa-Bianchi. Online learning and online convex optimization.

[4] Vincent Conitzer. *Linear and Integer Programming Course (CPS 196/296.2).* Spring 2008.

[5] H. Narasimhan. *Machine Learning Course, Optimization Tutorial 2.*

[6] Prof. Dr.-Ing. Wolfgang Utschick. Lecture notes in convex optimization.