MS&E 338: Reinforcement Learning
Lecture 11 & 2 Outline

Ian Osband

February 4, 2015

Warning

These notes are work in progress and principally meant to guide my own lecture preparation. Please take any questions/clarifications to Piazza. There will be typos

If you are looking for more complete/precise notes then I would recommend you look at the papers:

- Near-optimal Regret Bounds for Reinforcement Learning `http://www.jmlr.org/papers/volume11/jaksch10a/jaksch10a.pdf`
- (More) Efficient Reinforcement Learning via Posterior Sampling `http://arxiv.org/abs/1306.0940`

In particular we will basically follow the analysis and setting from "(More) Efficient Reinforcement Learning via Posterior Sampling". You should read that, because it's almost exactly the notation and setting we have been studying in class.

There are just a few small differences in the notation/presentation:

- episode length $H \leftrightarrow \tau$
- episode number $\ell \leftrightarrow k$
- Time-homogeneous R, P within each episode
- Some small changes to confidence sets/scaling/constants in analysis

Overall these are very minor differences.

1 Problem formulation (copy/paste)

We consider the problem of learning to optimize a random finite horizon MDP $M = (\mathcal{S}, \mathcal{A}, R^M, P^M, \tau, \rho)$ in repeated finite episodes of interaction. $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $R^M(s,a)$ is the reward distibution over $\mathbb{R}$ in state s with action a, $P^M(\cdot|s,a)$ is the transition probability over $\mathcal{S}$ from state s with action a, τ is the time horizon, and ρ the initial state distribution. We define the MDP and all other random variables we will consider with respect to a probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

A deterministic policy μ is a function mapping each state $s \in \mathcal{S}$ and $i = 1, \ldots, \tau$ to an action $a \in \mathcal{A}$. For each MDP $M = (\mathcal{S}, \mathcal{A}, R^M, P^M, \tau, \rho)$ and policy μ, we define a value function

$$V^M_{\mu,i}(s) := \mathbb{E}_{M,\mu}\left[\sum_{j=i}^{\tau} \overline{R}^M(s_j, a_j) \Big| s_i = s\right],$$

where $\overline{R}^M(s,a)$ denotes the expected reward realized when action a is selected while in state s, and the subscripts of the expectation operator indicate that $a_j = \mu(s_j, j)$, and $s_{j+1} \sim P^M(\cdot|s_j, a_j)$ for $j = i, \ldots, \tau$. A policy μ is optimal for the MDP M if $V^M_{\mu,i}(s) = \max_{\mu'} V^M_{\mu',i}(s)$ for all $s \in \mathcal{S}$ and $i = 1, \ldots, \tau$. We will associate with each MDP M a policy μ^M that is optimal for M.

The reinforcement learning agent interacts with the MDP over episodes that begin at times $t_k = (k-1)\tau + 1$, $k = 1, 2, \ldots$. At each time t, the agent selects an action a_t, observes a scalar reward r_t, and then transitions to s_{t+1}. Let $H_t = (s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1})$ denote the history of observations made *prior* to time t. A reinforcement learning algorithm is a deterministic sequence $\{\pi_k | k = 1, 2, \ldots\}$ of functions, each mapping H_{t_k} to a probability distribution $\pi_k(H_{t_k})$ over policies which the agent will employ during the kth episode. We define the regret incurred by a reinforcement learning algorithm π up to time T to be:

$$\text{Regret}(T, \pi, M^*) := \sum_{k=1}^{\lceil T/\tau \rceil} \Delta_k,$$

where Δ_k denotes regret over the kth episode, defined with respect to the MDP M^* by

$$\Delta_k := \sum_{\mathcal{S}} \rho(s)(V^{M^*}_{\mu^*,1}(s) - V^{M^*}_{\mu_k,1}(s))$$

with $\mu^* = \mu^{M^*}$ and $\mu_k \sim \pi_k(H_{t_k})$. Note that regret is not deterministic since it can depend on the random MDP M^*, the algorithm's internal random sampling and, through the history H_{t_k}, on previous random transitions and random rewards. We will assess and compare algorithm performance in terms of regret and its expectation.

2 Confidence sets

We will follow the same style of analysis as in the bandit setting. Specifically, we will build up "confidence sets" that contain the true MDP with high probability. From here we will either use optimism or posterior sampling to equate the "imagined" optimal rewards, and from there bound the mis-estimation error within the confidence sets.

So to specify a family of MDPs we will use two essential lemmas which bound the deviation of the reward and transition functions from their empirical mean.

Lemma 1 (Azuma-Hoeffding)**.**
For X_t martingale with $|X_t - X_{t+1}| \leq C$ a.s. for all t then for any $\beta > 0$:

$$\mathbb{P}(|X_N - X_0| \geq \beta) \leq 2\exp\left(\frac{-\beta^2}{2NC^2}\right)$$

We can equivalently write this equation that for any $\delta > 0$:

$$\mathbb{P}\left(\frac{|X_N - X_0|}{N} \geq C\sqrt{\frac{2\log(2/\delta)}{N}}\right) \leq \delta$$

Lemma 2 (L1 concentration of empirical measure)**.**
For $\hat{P}$ the empirical measure from N samples of a distribution P^* over S distinct values, the L1 deviation of the empirical measure is bounded for any $\beta > 0$:

$$\mathbb{P}\left(\|\hat{P} - P^*\|_1 \geq \beta\right) \leq \exp\left(S\log(2) - \frac{N\beta^2}{2}\right)$$

We can equivalently write this equation that for any $\delta > 0$:

$$\mathbb{P}\left(\|\hat{P} - P^*\|_1 \geq \sqrt{\frac{2S\log(2/\delta)}{N}}\right) \leq \delta$$

So now, if we want to make all the rewards and transitions lie within the confidence sets with probability greater than $1-\delta$ we should rescale $\tilde{delta} = \delta/\tau SA$. As you can see though, the δ only appears within the log... For the style of analysis that we're doing we really don't care much about logarithmic factors.

Theorem 1 (Osband's inequality)**.** *For all $X > 0$:*

$$\sqrt{\log(X)} \leq 10$$

Note that there are some results that suggest this inequality does not hold for $X > \exp(100)$.

Now actually Theorem 1 is meant to be a joke... just so you know $\exp(100) \simeq 10^{50}$ so if you're dealing with problems that big then these sorts of algorithms aren't really relevant in practice.

Next, let's rescale again $\tilde{\tilde{\delta}} = \tilde{\delta}/k^2$ so that when we sum over all episodes $k = 1,..,\infty$ we get a bounded contribution from missed confidence sets. This is because $\sum_{k=1}^{\infty} k^{-2} = \pi^2/6 < 2 < \infty$.

Using this together we define

$$\mathcal{M}_k = \left\{ R, P \;\middle|\; |R - \hat{R}| \leq \sqrt{\frac{4\log(2SA\tau k/\delta)}{n(s,a)}} \;,\; \|P - \hat{P}\|_1 \leq \sqrt{\frac{4S\log(2SA\tau k/\delta)}{n(s,a)}} \;\forall\; s,a \right\}$$

So now condition on the event that we lie within this set and we'll only lose $\mathbb{P} = \delta$. If we want a high probability guarantee we are done, otherwise we can scale δ with T to make sure that the resultant term is subdominant.

3 Analysis

For a real coverage of the analysis you're better off going straight to the papers listed above. Specifically, you should look at sections 4 and 5 of (More) Efficient Reinforcement Learning via Posterior Sampling. However, I will give a quick rundown here:

3.1 Optimism

For each episode k:

1. Form $\mathcal{M}_k$ subset of MDPs M that are statistically plausible given the data.

2. Use policy $\mu_k \in \arg\max_{\mu} \left\{ \max_{M \in \mathcal{M}_k} V_\mu^M(s) \right\}$.

Proof sketch:

$$\begin{aligned}
\Delta_k &= V_{*,1}^*(s) - V_{k,1}^*(s) \\
&= \underbrace{\left(V_{k,1}^k(s) - V_{k,1}^*(s)\right)}_{\text{Imagined - Actual}} + \underbrace{\left(V_{*,1}^*(s) - V_{k,1}^k(s)\right)}_{\leq 0 \text{ by optimism}}
\end{aligned}$$

We can decompose this into Bellman error:

$$V_{k,1}^k - V_{k,1}^* = \underbrace{\sum_{i=1}^{\tau} \left(\mathcal{T}_{k,i}^k - \mathcal{T}_{k,i}^*\right) V_{k,i+1}^k}_{B:=\text{Bellman error}} + \underbrace{\sum_{i=1}^{\tau} d_{t_k+1}}_{\mathbb{E}=0 \text{ martingale}} .$$

We can now use the Hölder inequality to bound:

$$B \leq \sum_{i=1}^{\tau} \left\{ \underbrace{|\overline{R}^k - \overline{R}^*|}_{\text{reward error}} + \frac{1}{2} \underbrace{\Psi_k}_{\text{MDP span}} \underbrace{\|P^k - P^*\|_1}_{\text{transition error}} \right\}$$

We conclude the proof by upper bounding these deviations by maximum possible within $\mathcal{M}_k$. Concentration inequalities allows us to build tight $\mathcal{M}_k$ that contain M^* with high probability.

3.2 Posterior sampling

For each episode k:

1. Sample an MDP from the posterior distribution for the true MDP: $M_k \sim \phi(\cdot|H_t)$.
2. Use policy $\mu_k \in \arg\max_{\mu} V_\mu^{M_k}$.

Proof sketch:

$$\begin{aligned}
\Delta_k &= V_{*,1}^*(s) - V_{k,1}^*(s) \\
&= \underbrace{\left(V_{k,1}^k(s) - V_{k,1}^*(s)\right)}_{\text{Imagined - Actual}} + \underbrace{\left(V_{*,1}^*(s) - V_{k,1}^k(s)\right)}_{\mathbb{E}[\cdot]=0}
\end{aligned}$$

Then follow the analysis as per optimism.

3.3 Final result

The dominant contribution comes from learning the transition functions P. This gives us a final scaling:

$$\text{Regret}(T) = \tilde{O}\left(\tau S \sqrt{A \tau L}\right)$$

These results hold in expectation for PSRL and high probability for UCRL. This should look pretty similar to the bandit setting. You might naturally ask, is this the same thing you'd just get from an application of the bandit theory?

3.4 Naive application of bandits

We could naively think of each episode as a single round of a bandit problem. In each episode you need to pick a *policy* to apply. For now, we will restrict outselves to deterministic policies $\mu : \mathcal{S} \to \mathcal{A}$. Well, if we treated each policy as an indepenedent arm of the bandit we'd have A^S possible pollicies. Standard bandit results would end up with regret after L episodes

$$\tilde{O}\left(\tau\sqrt{A^S L}\right)$$

4 Beyond episodic RL

If you look a the actual UCRL paper, you will notice that the problem setting is slightly different. In particular, they do not assume that every H steps the MDP will reset its episode. In fact, this is a common theme to a lot of the literature... how do you deal with learning when there is no episodic reset? This can get pretty complicated and/or technical, but it's something that (so far) we don't really know how to deal with for PSRL.

They manage to produce an algorithm, with the same $\sqrt{T}$ scaling of regret in a setting without episodic regret. To do this, the full version of UCRL2 actually solves for the optimal long-term average reward problem at the start of each pseudo-episode. UCRL2 then recomputes this policy every time a particular (s, a) pair has been visited double the amount of times it had at the start of the policy. This means that we get contributions from $\sum_{t=1}^{T}\sqrt{\frac{1}{n_t(s,a)}}$ that still grow slowly like $O(\sqrt{T})$ but also that the total number of episode switching (which can incur a large regret) only grows logarithmically.

In practice, you can start off with a small "episode" and grow it as you gather more and more data. In some sense the timescale of the optimal policy acts as a form of regularization, bandit algorithms with $H = 1$ will come up with less complicated schemes to increase future rewards than a policy to optimize long term average cost.

4.1 Measure of MDP connectedness

In our previous analysis we knew that the value in any episode was bounded $\in [0, \tau]$ so that when we decomposed the $(P^k - P^*)^T V^k \leq \|P^k - P^*\|_1 \tau$. However we could have actually decomposed it more finely by using the $\max_{s,s'} V^k(s) - V^k(s')$. This is the maximum difference in future value between any two states under the MDP M_k. This quantity is called the *span* on an MDP.

Definition 1 (MDP Span)**.** *For an MDP M with optimal value function V^M, the span of M $\Psi(M)$ is defined:*

$$\Psi(M) := \max_{s,s'} V^M(s) - V^M(s')$$

We define another measure of MDP connectedness

Definition 2 (MDP Diameter)**.** *For an MDP M we define the diameter $D(M)$:*

$$D(M) := \max_{s,s'} \min_{\pi} T^{\pi}_{s \to s'}$$

Where $T^{\pi}_{s \to s'}$ is the expected number of timesteps to transition between s and s' under the policy π.

We make a few simple observations:

- $\Psi(M) \leq D(M)$ and the difference can be arbitrarily large.
- For posterior sampling $\Psi(M_k) = \Psi(M^*)$ in distribution.
- For UCRL $D(M_k) \leq D(M^*)$ when confidence sets hold.

5 Beyond Tabula Rasa RL

In fact, there is a fundamental lower bound that any general RL algorithm must have expected regret $\Omega(\sqrt{SAT})$ on some MDP. We only have bounds $O(S\sqrt{AT})$ but we will now consider two examples that show you that even if we managed to get these optimal scalings it wouldn't be enough.

5.1 Production line

Production line with 100 machines, each with 3 states and 3 actions. Each machine generates some revenue we want to maximize jointly.

Figure 1: automated production line

This MDP has state $s = (s_1, .., s_{100})$ and action $a = (a_1, .., a_{100})$. Here $S = A = 3^{100} \simeq 10^{50}$, so even a maximally efficient general-purpose learner would have regret $\Omega(\sqrt{SAT}) \simeq 10^{50}\sqrt{T}$.

5.1.1 Factored MDPs

If over a single timestep, each machine depends directly only upon its neighbours then this becomes a factored MDP. Now $|\mathcal{X}[Z_j^P]| \leq 3^3$ and $|S_j| \leq 3$ for each machine j. We can exploit this graphical structure for exponentially smaller regret $\simeq 100\sqrt{3^3 \times 3 \times T} \simeq 10^3\sqrt{T}$.

Check out `http://papers.nips.cc/paper/5445-near-optimal-reinforcement-learning-in-factored-mdps` for the scoop. You can exploit this structure via PSRL in the natural way (or with a UCRL modification).

5.2 Helicopter guidance

Imagine that we want to pilot a helicopter through a specific trajectory, with a quadratic cost for deviation from the desired flight path/velocity. The state here can be described by a 12-dimensional vector of (position, velocity, acceleration, rotation). Actions are similarly high dimensional, perhaps 6-dimensions. We only assume that the helicopter can observe the state and reward. Discretization to any reasonable precision will mean that $S = \Omega(\epsilon^{-12})$, $A = \Omega(\epsilon^{-6})$ and the whole problem dimensionality explodes!

5.2.1 Eluder dimension

In this enlarged state space the state dynamics are linear. Now that we know this, it seems like we should be able to learn in time that scales polynomially in the dimension of the problem.

Check out `http://papers.nips.cc/paper/5245-model-based-reinforcement-learning-and-the-eluder-dimension` for the scoop. You can exploit this structure via PSRL in the natural way (or with a UCRL modification).

5.3 Beyond model-based RL

So far, the majority of reinforcement learning algorithms we have studied have been model-based. That is to say they directly build up approximations to R and P and then attempt to solve the resulting MDP. Solving the MDP in itself can be an intractable problem, even if you know the reward and transition functions!

One nice thing we should note is that, for UCRL and PSRL if you use an approximate MDP solver in the algorithm then it will (at worst) give an additive term in the regret bound. If your problem is really that intractable that you cannot solve it anyways to begin with then effectively we have reduced the RL problem to an MDP planning problem (at least for PSRL where you only need to solve over one MDP) and it must be that this is more tractable.

However, if all we really care about is getting good performance (or rather, learning a good policy) then why don't we simply try to approximate the policy or value function directly? So far there is only really one algorithm (Delayed Q-learnind) that gives polynomial learning guarantees for general MDPs... and it's not a very practical algorithm. But, this sort of direct value/policy generalization is going to be key for most real applications of reinforcement learning. We'll start on that next week.