Machine Learning Theory 2021
Lecture 14

Wouter M. Koolen

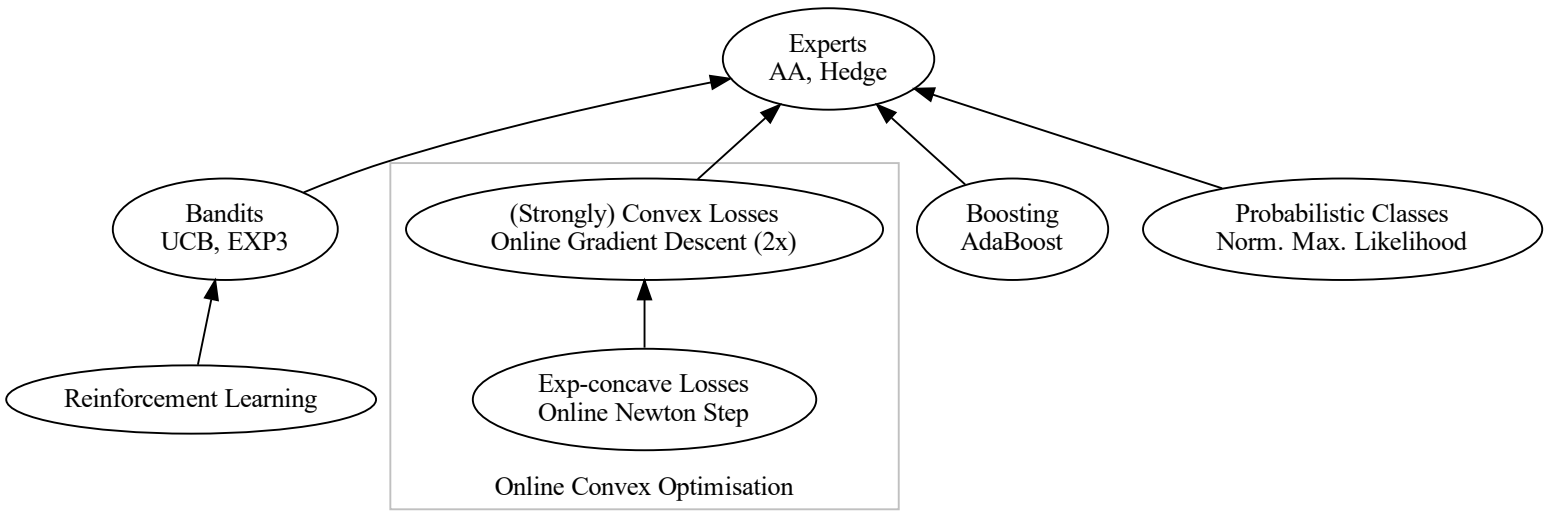
Download these slides now from elo.mastermath.nl!

- Reinforcement Learning:
  - Markov Decision Processes
  - REPS approach for solving known MDP
  - UCRL approach for learning in unknown MDP

homework roulette
in the break

Recap

Overview of Second Half of Course



Material: course notes and selection of sources on MLT website.

Revisiting our Abstractions

We can deal with adversarial data.

This allows us to learn in **any** environment. Even in settings where our actions change the state of the world.

Revisiting our Abstractions

We can deal with adversarial data.

This allows us to learn in **any** environment. Even in settings where our actions change the state of the world.

Our baseline so far, best expert/arm/action/point, does not exploit **what would have happened to future state if we had taken a different action**

But what if we **can model how** the world changes?

Reinforcement Learning:

- Explicit about state
- Stochastic assumptions
- More demanding baseline
- Learning possible!

Known MDP

Basic Setup

An MDP is a stochastic model for the environment with **state**

- State space S
- Action set A
- Transition probability $P(s'|s,a)$
- Reward $R(r|s,a)$
- Initial state $s_1 \in S$.

Stochastic Bandit corresponds to $S = 1$.

Definition (Tabular Case)

- S, A are finite.
- $P(\cdot|s,a)$ is presented as an $S \times A \times S$ table of probabilities.
- $R(\cdot|s,a)$ is either parametric (Gaussian/Bernoulli/...) or bounded-support, and is represented by an $S \times A$ table of means.

Definition (Tabular Case)

- S, A are finite.
- $P(\cdot|s,a)$ is presented as an $S \times A \times S$ table of probabilities.
- $R(\cdot|s,a)$ is either parametric (Gaussian/Bernoulli/...) or bounded-support, and is represented by an $S \times A$ table of means.

For the impatient: One can consider MDPs with infinite state and action spaces, with prior knowledge/structural assumptions, feature maps...

Interaction with an MDP

Protocol

For $t = 1, 2, \ldots$

- Learner sees current state s_t
- Learner picks action a_t
- Learner receives reward $r_t \sim R(\cdot|s_t, a_t)$
- The state evolves according to $s_{t+1} \sim P(\cdot|s_t, a_t)$

Objective (high level)

Large cumulative reward

Examples

- Queueing systems (k-server)
  inventory management, cooling, routing, ...
- Control (local aspects of) drone flight, self-driging, ...
- Many one-player games
- ...

For contrast: in POMDP we do not see the (full) state.

Basic Setup

Definition

A (randomised) policy π assigns a next action a_t to each history $(s_1, a_1, r_1), \ldots, (s_{t-1}, a_{t-1}, r_{t-1}), s_t$.

In general policies can be complicated. Two simple cases

- A **memoryless policy** $\pi : S \to \triangle_A$ maps current state s_t to randomised action.
- A **deterministic memoryless policy** $\pi : S \to A$ maps current state s_t to deterministic action.

Quality

Definition

Long-term average reward of a policy π is

$$V_\pi = \lim_{T\to\infty} \frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$$

where

s_1 is the initial state,
$a_t \sim \pi(\cdot|s^t, r^{t-1}, a^{t-1})$,
$r_t \sim R(\cdot|s_t, a_t)$, and
$s_{t+1} \sim P(\cdot|s_t, a_t)$.

Quality

Definition

Long-term average reward of a policy π is

$$V_\pi = \lim_{T\to\infty} \frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$$

where

s_1 is the initial state,
$a_t \sim \pi(\cdot|s^t, r^{t-1}, a^{t-1})$,
$r_t \sim R(\cdot|s_t, a_t)$, and
$s_{t+1} \sim P(\cdot|s_t, a_t)$.

Definition

The **optimal policy** is $\pi^* \in \operatorname{argmax}_\pi V_\pi$, with optimal value $V^* = V^{\pi^*}$.

Quality

Definition

Long-term average reward of a policy π is

$$V_\pi = \lim_{T\to\infty} \frac{1}{T}\,\mathbb{E}\left[\sum_{t=1}^{T} r_t\right] \qquad \text{where}$$

s_1 is the initial state,
$a_t \sim \pi(\cdot|s^t, r^{t-1}, a^{t-1})$,
$r_t \sim R(\cdot|s_t, a_t)$, and
$s_{t+1} \sim P(\cdot|s_t, a_t)$.

Definition

The **optimal policy** is $\pi^* \in \operatorname{argmax}_\pi V_\pi$, with optimal value $V^* = V^{\pi^*}$.

Popular alternatives:

- Discounted reward $\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r_t\right]$ for $\gamma \in (0, 1)$.
- Finite horizon reward $\mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$ for fixed T

MDP Facts

- The optimal policy π^* is memoryless. $\Leftarrow$ simple!
- Each memoryless policy π has a **stationary distribution**

$$\mu_\pi(s, a) = \lim_{t \to \infty} \mathbb{P}\{s_t = s, a_t = a\}$$

I.e. if you run π for long, it will be at $s, a \sim \mu_\pi$.

Solving MDPs

Problem

Given MDP (S, A, P, R, s_1), how to obtain (approximate) π^*?

Solving MDPs

Problem

Given MDP (S, A, P, R, s_1), how to obtain (approximate) π^*?

Many options:

- Policy Iteration (Policy Gradient)
- Value Iteration
- Linear Programming
- Online learning
- ...

(Possibly alternated with **learning/estimating** P, R, μ_π, V^π, ...)

Online Linear Optimisation

Taken from (Neu, Gómez, and Jonsson, 2017)

Stationarity is an **linear constraint**. Can search over stationary distributions (the optimal policy is among those).

Problem

Optimise expected reward

$$V^* = \max_{\mu \text{ stationary}} \sum_{s,a} \mu(s,a) r(s,a)$$

under the constraint that $\mu \in \triangle_{S \times A}$ *is stationary for the MDP, i.e.*

$$\forall s : \sum_{a} \mu(s,a) = \sum_{a',s'} P(s|a',s') \mu(s',a').$$

NB: the policy giving rise to μ is $\pi(a|s) = \mu(a|s) = \frac{\mu(s,a)}{\sum_{a'} \mu(s,a')}$.

In fact solving a more general problem

Suppose MDP is fixed but reward functions $r_1, r_2 \in \mathbb{R}^{S \times A}$ are **adversarial**.

The goal is to optimise the regret

$$R_T \coloneqq \sum_{t=1}^{T} \langle \mu^* - \mu_t, r_t \rangle$$

where the learner's iterates μ_t and the optimal policy μ are stationary distributions.

Practical? If policy is revised infrequency, learner is at μ_π and observes entire average reward r_t.

Mirror Descent Learning Algorithm

We saw (homework) a particular presentation of Online Gradient Descent

$$\boldsymbol{w}_{t+1} \;=\; \operatorname*{argmin}_{\boldsymbol{w}\in\mathcal{U}} \; \langle \nabla f_t(\boldsymbol{w}_t), \boldsymbol{w}\rangle + \frac{1}{\eta} \underbrace{\|\boldsymbol{w}-\boldsymbol{w}_t\|^2}_{\text{squared Eucl. norm}}$$

and of Hedge

$$\boldsymbol{w}_{t+1} \;=\; \operatorname*{argmin}_{\boldsymbol{w}\in\triangle} \; \langle \boldsymbol{\ell}_t, \boldsymbol{w}\rangle + \frac{1}{\eta} \underbrace{\sum_k w^k \ln \frac{w^k}{w_t^k}}_{\text{KL divergence}}$$

Mirror Descent Learning Algorithm

We saw (homework) a particular presentation of Online Gradient Descent

$$\boldsymbol{w}_{t+1} = \operatorname*{argmin}_{\boldsymbol{w}\in\mathcal{U}} \langle \nabla f_t(\boldsymbol{w}_t), \boldsymbol{w}\rangle + \frac{1}{\eta} \underbrace{\|\boldsymbol{w}-\boldsymbol{w}_t\|^2}_{\text{squared Eucl. norm}}$$

and of Hedge

$$\boldsymbol{w}_{t+1} = \operatorname*{argmin}_{\boldsymbol{w}\in\triangle} \langle \boldsymbol{\ell}_t, \boldsymbol{w}\rangle + \frac{1}{\eta} \underbrace{\sum_k w^k \ln \frac{w^k}{w_t^k}}_{\text{KL divergence}}$$

This suggests the upgrade (Online Relative Entropy Policy Search)

$$\mu_{t+1} = \operatorname*{argmax}_{\mu \text{ stationary}} \langle r_t, \mu\rangle - \frac{1}{\eta} \underbrace{\sum_{s,a} \mu(a,s) \ln \frac{\mu(s,a)}{\mu_t(s,a)}}_{\text{KL divergence}}$$

Result

O-REPS has near-closed-form iterates

$$\pi_{t+1}(a|s) \;=\; \pi_t(a|s)e^{\eta\left(r_t(s,a)+\sum_{s'} P(s'|s,a)V_t(s')-V_t(s)\right)}$$

See (Zimin and Neu, 2013)

Theorem

The regret of O-REPS is $\sqrt{T \ln(SA)}$ (full info) or $\sqrt{TSA \ln(SA)}$ (bandit).

Conclusion

How to put O-REPS in a larger system?

- Leverage regret bounds in motivation
- There are **bandit** versions too (more practical)
- The learner plays μ_t
- Estimate P, R on the way?

Unknown MDPs

Basic Setup

Now suppose the learner is in an MDP, but does not know P (and/or R).

Learner needs to *learn* P by *interacting*

There is a reward **overhead** for learning P vs knowing P up front.

Basic Setup

Now suppose the learner is in an MDP, but does not know P (and/or R).

Learner needs to *learn* P by *interacting*

There is a reward **overhead** for learning P vs knowing P up front.

Definition

The expected regret of policy π in a given MDP (S, A, P, R, s_1) is

$$\text{Regret}_T(\pi) \;=\; TV^* - \mathbb{E}_{\pi,\text{MDP}}\left[\sum_{t=1}^{T} r(A_t, S_t)\right]$$

where $r(a, s)$ is the mean reward of action a in state s.

Result

Theorem

There is an algorithm with regret at most

$$R_T \leq C_{diam} S \sqrt{2AT \ln T}$$

where C_{diam} is the **diameter** *of the MDP*

Result

Theorem

There is an algorithm with regret at most

$$R_T \leq C_{diam} S \sqrt{2AT \ln T}$$

where C_{diam} is the **diameter** *of the MDP*

Lower bound $\sqrt{C_{\text{diam}} SAT}$. Very active research area!

UCRL

Main message: Not significantly more involved than UCB analysis

UCRL

Main message: Not significantly more involved than UCB analysis

High-level overview of UCRL2 (Jaksch, Ortner, and Auer, 2010)

- Work in **phases**. In each phase k
  - Construct a confidence interval on MDPs. (Hoeffding)
    Empirical transition probability $\pm\sqrt{\frac{\text{exp. bonus}}{\#\text{ visits}}}$
  - Compute optimal policy π_k in plausible MDP of largest gain.
    $\Rightarrow$ Use methods for known MDP from first part.
  - Follow π_k for the duration of the phase.

Need about $SA\ln(T)$ phases.

See (Lattimore and Szepesvári, 2020) for context/analysis.

Further References

Good starting point:

Simons Fall 2020 Theory of Reinforcement Learning program
Boot Camp video/slides archive

Conclusion

Conclusion

- Explicitly model state of the world.
- Provides structure for learner, and also raises the benchmark.
- Beyond tabular
  - Theory is developing under other assumptions:
    - Linear MDPs
    - Low-rank MDPs
    - ...
  - Far-ranging practical success based on approximation (DNN) of
    - policy
    - value function
    - stationary distribution
    - ...

Conclusion

This concludes the lectures.

- It has been a pleasure
- Good luck for the exam
- If you have an idea that you want to work on ...

 Jaksch, T., R. Ortner, and P. Auer (2010). "Near-optimal Regret Bounds for Reinforcement Learning". In: *Journal of Machine Learning Research* 11, pp. 1563–1600.

 Lattimore, T. and C. Szepesvári (2020). *Bandit Algorithms*. Cambridge University Press.

 Neu, G., V. Gómez, and A. Jonsson (2017). "A unified view of entropy-regularized Markov decision processes". In: *Deep Reinforcement Learning Symposium, NIPS*.

 Zimin, A. and G. Neu (2013). "Online learning in episodic Markovian decision processes by relative entropy policy search". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 1583–1591.