Selecting the state representation in reinforcement Learning

Odalric-Ambrym Maillard, Daniil Ryabko and Rémi Munos

NIPS Workshop

December 16, 2011
States in MDP
A practical challenge.

We wonder how to deal with many, possibly improper, state-representations in reinforcement learning.
Motivation

When we deal with MDP, we know algorithms that can solve it reasonably well (see LSTD, BRM, UCRL, etc.).

Yet in practice, we generally start from observations only, and it is not always easy to build a proper MDP.

In particular, one can generate a priori various notions of states, many being irrelevant, too complex or leading to improper MDPs.

What can we do?
State representation

Sets Observations $\mathcal{O}$, Actions $\mathcal{A}$ (finite), Rewards $0 \in \mathcal{R} \subset [0, 1]$. Set of histories up to time $t \in \mathbb{N}$: $\mathcal{H}_t \overset{\text{def}}{=} \mathcal{O} \times (\mathcal{A} \times \mathcal{R} \times \mathcal{O})^{t-1}$, of all possible histories: $\mathcal{H} \overset{\text{def}}{=} \bigcup_{t=1}^{\infty} \mathcal{H}_t$.

Definition A state representation function $\phi$ is a function from $\mathcal{H}$ to $\mathcal{S}$, where $\mathcal{S} \overset{\text{def}}{=} S_\phi \subset \mathbb{N}$ is some finite set. We write $s_{t, \phi} := \phi(h_{<t})$. 
**Challenge**

**Setting**
The learner is given a set $\Phi = \{\phi_1, ..., \phi_J\}$ of $J$ many state representation functions. One of which, $\phi_{j^*}$, is such that the process $(s_t, \phi_{j^*}, a_t, r_t)_{t \in \mathbb{N}}$ is a Markov Decision Process.

**Assumption**
We further assume this MDP to be weakly communicating, with finite diameter $D$ (the expected minimum time required to reach any state starting from any other state).

**Challenge**
If $j^*$ is known, we have a standard MDP that we can solve. Can we perform as well if we do not know $j^*$? i.e. can we get as much reward as the optimal policy for the (best of the) correct model(s) $\phi_{j^*}$?
Some Motivations

**High dimensional observations** $\Phi$ is a set of high-level feature extraction functions $\{\phi_j\}_{j \in J}$, and the observed dynamic is Markov w.r.t. this high-level representation. There are many of them most of which are useless.

**Example** Observations $=$ video stream (high dimensional).
Some Motivations (II)

Bounding the order of an MDP  The process is known to be $k$-order Markov, $k$ unknown but $K \gg k$ is given. Thus it is also 1-order MDP with $s^k$ states, where $s$ is the size of the state space.

But if $k \ll K$ we don’t want to use $2^K$-state representation to make it Markov. We’d prefer finding the right $k$ as we go. We want to perform as well as if we knew $k$. 
Example: high-level feature selection

- Observations = video stream (high dimensional).
- $\Phi$ is a set of feature extraction functions $\{\phi_j\}_{j \in J}$. 
States in MDP

A practical challenge.

We now present one possible solution.
Regret

Regret of any strategy at time $T$:

$$R_T \overset{\text{def}}{=} T \rho^* - \sum_{t=1}^{T} r_t,$$

where $r_t$ are the rewards received when following the proposed strategy and $\rho^*$ is the average optimal value in the best Markov model, i.e.,

$$\rho^* = \lim_{T} \frac{1}{T} \mathbb{E}\left( \sum_{t=1}^{T} r_t(\pi^*) \right)$$

where $r_t(\pi^*)$ are the rewards received when following the optimal policy for the best wc-Markov model.
UCRL2 as a subroutine algorithm

We base our algorithm on the UCRL2 algorithm.

- It is shown in [Jaksch et al. 2010] that with probability higher than $1 - \delta'$, the regret of UCRL2 when run for $\tau$ consecutive many steps in the true model $\phi^*$ is upper bounded as

  $$\frac{R_\tau}{\tau} \leq B_D(\tau, \phi, \delta'),$$

  where $B_D(\tau, \phi, \delta') \overset{\text{def}}{=} 34D|S_\phi|\sqrt{\frac{A\log(\frac{\tau}{\delta'})}{\tau}}$.

- The diameter $D$ does not need to be known by the algorithm.
The Best Lower Bound algorithm

In stage $i$ of length $\tau_i = 2^i$:

**Exploration**

$\tau_{i,1}$ steps
($\tau_{i,1} = \tau_i^{2/3}$)

**Exploitation**

$\tau_{i,2}$ steps
The Best Lower Bound algorithm

In stage $i$ of length $\tau_i = 2^i$:

**Exploration** $\tau_{i,1}$ steps 
($\tau_{i,1} = \tau_i^{2/3}$)

$\hat{\mu}_{i,1}(\phi_1)$ $\hat{\mu}_{i,1}(\phi_2)$ $\hat{\mu}_{i,1}(\phi_3)$ $\ldots$ $\hat{\mu}_{i,1}(\phi_J)$

**Exploitation** $\tau_{i,2}$ steps
The Best Lower Bound algorithm

In stage $i$ of length $\tau_i = 2^i$:

Exploration

$\tau_{i,1}$ steps ($\tau_{i,1} = \tau_i^{2/3}$)

$\hat{\mu}_{i,1}(\phi_1) \quad \hat{\mu}_{i,1}(\phi_2) \quad \hat{\mu}_{i,1}(\phi_3) \quad \ldots \quad \hat{\mu}_{i,1}(\phi_J)$

Exploitation

$\tau_{i,2}$ steps

Exploit the best model according to:

$$\hat{j} = \arg\max_{j \in J} \hat{\mu}_{i,1}(\phi_j) - 2B(i, \phi_j, \delta).$$

Run UCRL2 on it as long as

$$\hat{\mu}_{i,2,t}(\phi_{\hat{j}}) \geq \hat{\mu}_{i,1}(\phi_{\hat{j}}) - 2B(i, \phi_{\hat{j}}, \delta).$$
The Best Lower Bound algorithm

In stage $i$ of length $\tau_i = 2^i$:

Exploration

$\tau_{i,1}$ steps

($\tau_{i,1} = \tau_i^{2/3}$)

$\hat{\mu}_{i,1}(\phi_1)$ $\hat{\mu}_{i,1}(\phi_2)$ $\hat{\mu}_{i,1}(\phi_3)$ ... $\hat{\mu}_{i,1}(\phi_J)$

Exploitation

$\tau_{i,2}$ steps

5 2 3 4 J ... 1
The Best Lower Bound algorithm

In stage $i$ of length $\tau_i = 2^i$:

**Exploration**

$\tau_{i,1}$ steps

($\tau_{i,1} = \tau_i^{2/3}$)

$\hat{\mu}_{i,1}(\phi_1)$  $\hat{\mu}_{i,1}(\phi_2)$  $\hat{\mu}_{i,1}(\phi_3)$  \ldots  $\hat{\mu}_{i,1}(\phi_J)$

**Exploitation**

$\tau_{i,2}$ steps

5  2  3  4  J  \ldots  1
The Best Lower Bound algorithm

In stage $i$ of length $\tau_i = 2^i$:

**Exploration**

$\tau_{i,1}$ steps
($\tau_{i,1} = \tau_i^{2/3}$)

$\hat{\mu}_{i,1}(\phi_1)$, $\hat{\mu}_{i,1}(\phi_2)$, $\hat{\mu}_{i,1}(\phi_3)$, ..., $\hat{\mu}_{i,1}(\phi_J)$

**Exploitation**

$\tau_{i,2}$ steps

5 2 3 4 J ... 1

...
Intuition of the proposed algorithm (I)

- Each model $\phi_j \in \Phi$ yields to a Decision Process with states $S_{\phi_j}$ (and $\phi^*$ gives a MDP). We can run UCRL2 with each $S_j$ and measure the cumulative rewards for all models during an **exploration** phase.

- We know that for the true model, after enough **consecutive** steps, the cumulative **rewards** must be high with high probability.

- We test this property during an **exploitation** phase, which enables to discard wrong models.

The quantity used in the algorithm is:

$$B(i, \phi, \delta) \overset{\text{def}}{=} 34f(\tau_i - 1 + \tau_i, 1)|S_{\phi}|\sqrt{\frac{A \log \left( \frac{\tau_{i,1}}{J} \right)}{\tau_{i,1}/J}},$$

where $f$ is used to guess $D$ and where $\delta_i(\delta)$ is tuned using $\delta$. 
Intuition of the algorithm (II)

- **Exploration** of all models long enough to estimate the optimal reward level and policy in the true model.
- “Wrong” models used during *exploitation* stages only as long as they are giving rewards higher than the rewards that could be obtained in the “true” model.
- Generally in *hypothesis testing*, one has to make assumptions about alternative models so that the Type II error is small (the power of the test is large). Here, this role is played by the *rewards*, thus nothing has to be known about the “wrong” models.
- We only have to ensure that the true model passes the test.
Result

Single run, no reset, no assumption on “wrong” models.

Theorem (Main result)

The regret of the BLB algorithm w.r.t the optimal policy corresponding to $\phi^* \in \Phi$, run with parameter $\delta$, is bounded, for any horizon $T$, with probability higher than $1 - \delta$, as

$$\mathcal{R}_T \leq c f(T) S \left( A J \log((J\delta)^{-1}) \log_2(T) \right)^{1/2} T^{2/3} + c' DS \left( A \log(\delta^{-1}) \log_2(T) T \right)^{1/2} + c(f, D), \quad (1)$$

for some numerical constants $c, c', c(f, D)$ and $S = |S_{\phi^*}|$. The function $f(t)$ is any increasing, non negative function.

Choosing $f(t) := \log_2 t + 1$ gives a bound which is of order $T^{2/3}$ in $T$ but is exponential in $D$ (with $c(f, D) \leq 2^D$); taking $f(t) := t^{\epsilon}$ we get a bound of order $T^{2/3+\epsilon}$ in $T$ but of polynomial order $1/\epsilon$ in $D$. 
Assumptions. A crucial assumption is that the “true” model $\phi^*$ belongs to a known finite set (realizable-case). It seems difficult to remove it without additional assumptions on the models.

Another approach would be to try to build a good state representation function, as suggested e.g. in [Hutter (2009)].

Estimating the diameter? A possibly large additive constant $c(f, D)$ appears in the regret since we do not know a bound on the diameter of the MDP in the “true” model, and use $\log T$ instead. Finding a way to estimate the diameter of the MDP online is an interesting and challenging question.
Future work, open questions

- Use structure of \( \Phi \) (Hierarchy?, Aggregation?).
- Online estimation of \( D \).
- Smoother hypothesis with respect to \( \phi^* \in \Phi \): Need for approximate-MDPs