Explicit Explore or Exploit

Michael Kearns, Satinder Singh

Structure

- Basics
- Definition
- Algorithm
- Future Work

Markov Decision Process

- set of states 1,…,N
- set of actions $a_1, \ldots, a_k$
- Transition probabilities: $P_M^a(ij) \geq 0$ probability of reaching state j after executing action a from state i in M
- Payoff distributions: $R_M(i)$, determine the random payoff received when state i is visited
- Policy $\pi : \{1, \ldots, N\} \rightarrow \{a_1, \ldots, a_k\}$

Definitions

- **T-path** is a sequence p of T+1 states: $p = i_1, i_2, \ldots, i_T, i_{T+1}$
  - probability that p is traversed in M, starting in i_1 and executing policy π

$$\Pr_M^{\pi}[p] = \prod_{k=1}^{T} P_M^{\pi(i_k)}(i_k i_{k+1})$$

- expected undiscounted return along p in M $U_M(p) = \frac{1}{T}\left(R_{i_1} + \cdots + R_{i_T}\right)$

- expected discounted return along p in M

$$V_M(p) = R_{i_1} + \gamma R_{i_2} + \gamma^2 R_{i_3} + \cdots + \gamma^{T-1} R_{i_T}$$

Definitions

- **T-step** undiscounted return from state i $$U_M^\pi(i,T) = \sum_p \Pr_M^\pi[p] U_M(p)$$

- **T-step** discounted return from state i $$V_M^\pi(i,T) = \sum_p \Pr_M^\pi[p] V_M(p)$$

  - In both cases the sum is over all T-paths p that start in state i

Definitions: Mixing Time

Every MDP has a stationary distribution

- the time t distribution converges to the stationary distribution π as t tends to infinity
- We need a finite number of steps T to get close to the stationary distribution

We look for the smallest number T of steps required to ensure that the distribution on states after T steps of π is within ε of the stationary distribution $$T' \geq T, \left| U_M^\pi(i, T') - U_M^\pi \right| < \varepsilon$$

The distance is measured by the Kullback-Leibler divergence

Definitions: Horizon Time

the expected discounted return of any policy after $T \approx 1/(1-\gamma)$ steps approaches the expected asymptotic discounted return.

so if $T \geq (1/(1-\gamma))\log(R_{\max} /(\varepsilon(1-\gamma)))$

Then for any state i $V_M^\pi(i,T) \leq V_M^\pi(i) \leq V_M^\pi(i,T) + \varepsilon$

Algorithm

- Maintain a partial model for the transition probabilities and the expected payoffs for some subset of states in M
- States of the algotrithm divided into three categories
  - Known states
  - States that have been visited before
  - Unknown states
- M_s is naturally induced on S by the full MDP M
  - All transistions in M between states in S are preserved in M_s
  - All other transistions in M are redirected in M_s to a single additional absorbing state that represents all of the unknown and unvisited states

Algorithm

TECHNISCHE UNIVERSITÄT DARMSTADT

the algorithm has an approximation $\hat{M}_S$

performing two off-line, polynomial-time computations on this approximation

$\hat{M}_S$ is an α-approximation of M_S if:

- for any state i

$$R_{M_S}(i) - \alpha \leq R_{\hat{M}_S}(i) \leq R_{M_S}(i) + \alpha$$

- for any states i and j, and any action a

$$P^a_{M_S}(ij) - \alpha \leq P^a_{\hat{M}_S}(ij) \leq P^a_{M_S}(ij) + \alpha$$

Algorithm

Uniscounted Case:

- Let $\hat{M}$ be an $O\left(\left(\varepsilon/\left(NTG_{\max}^{T}\right)\right)^2\right)$ - approximation of M
- For any policy π in $\prod_{M}^{T,\varepsilon/2}$ and any state i

$$U_M^{\pi}(i,T) - \varepsilon \le U_{\hat{M}}^{\pi}(i,T) \le U_M^{\pi}(i,T) + \varepsilon$$

Discounted Case:

- Let $T \ge (1/(1-\gamma))\log\left(R_{\max}/(\varepsilon(1-\gamma))\right)$ and $\hat{M}$ be an $O\left(\left(\varepsilon/\left(NTG_{\max}^{T}\right)\right)^2\right)$ - approximation of M
- For any policy π and any state i

$$\boldsymbol{V_M^{\pi}(i) - \varepsilon \le V_{\hat{M}}^{\pi}(i) \le V_M^{\pi}(i) + \varepsilon}$$

Algorithm

We need a definition of a known state

- The state has been visited enough times to ensure that the estimated transition probabilities and the estimated payoff are all within

$$O\left(\left(\varepsilon / \left(NTG_{\max}^{T}\right)\right)^{2}\right)$$ of their true values

After at least m_{known} steps a state is known

$$m_{known} = O\left(\left(\left(NTG_{\max}^{T}\right)/\varepsilon\right)^{4} Var_{\max} \log(1/\delta)\right)$$

Algorithm

T-step Undiscounted value Iteration

$$Initialize: \; for\ all\ i \in \hat{M}_S, U_{T+1}(i) = 0.0$$

$$For\ t = T, T-1, T-2, \ldots, 1:$$

$$for\ all\ i, U_t(i) = R_{\hat{M}_S}(i) + \max_a \sum_j P^a_{\hat{M}_S}(ij) U_{t+1}(j)$$

$$for\ all\ i, \pi^*_t(i) = \arg\max_a \left[R_{\hat{M}_S}(i) + \sum_j P^a_{\hat{M}_S}(ij) U_{t+1}(j) \right]$$

T-step Discounted value iteration

$$Initialize: \; for\ all\ i \in \hat{M}_S, V_{T+1}(i) = 0.0$$

$$For\ t = T, T-1, T-2, \ldots, 1:$$

$$for\ all\ i, V_t(i) = R_{\hat{M}_S}(i) + \gamma \max_a \sum_j P^a_{\hat{M}_S}(ij) V_{t+1}(j)$$

$$for\ all\ i, \pi^*_t(i) = \arg\max_a \left[R_{\hat{M}_S}(i) + \gamma \sum_j P^a_{\hat{M}_S}(ij) V_{t+1}(j) \right]$$

Algorithm

- M is a MDP and S any subset of the states of M
- The induced MDP on S, denoted M_S has states, transitions and payoffs as follows:
  - For any state $i \in S, R_{M_S}(i) = R_M(i)$
  - $R_{M_S}(s_0) = 0$
  - $P^a_{M_S}(ij) = P^a_M(ij)$ transitions in M between states in S are preserved in M_S
  - All transitions in M that are not between states in S are redirected to s_0 in M_S

Algorithm

- At certain points we will perform the value iteration twice
  - Once at $\hat{M}_S$
  - Second time on $\hat{M}_S^{'}$

- Balanced Wandering
  - Is the algorithm arriving in a state it has never visited before it takes a arbitary action
  - But reaching a state it visited before, it takes the action it has tried the fewest times

Algorithm

- ►(Initialization) the set S of known states is empty
- ►Balanced Wandering (the current state is not in S)
  - ►After $N(m_{known}-1)+1$ steps of balanced wandering some states become known (worst case)
- ►(Off-line Optimizations) reaching a known state the algorithm performs the two off-line Optimizations
  - ►(Attempted Exploitation) if the resulting exploitation policy $\hat{\pi}$ achieves return that is at least $U^{*}-\varepsilon/2$, the algorithm executes $\hat{\pi}$ for the next T steps
  - ►(attempted Exploration) otherwise the algorithm executes the resulting exploration policy $\hat{\pi}'$ for the next T steps

Algorithm

- Any time an attempted exploitation or attempted exploration visits a state not in S, the algorithm goes to step 1

Future Work

- There is no implemented Algorithm yet

- Model-free version of the algorithm

Any Questions

Thank you for your Attention