RL 16: Theoretical aspects

Michael Herrmann

University of Edinburgh, School of Informatics

14/03/2014

Overview

1. Convergence
2. Complexity

What means convergence of RL?

What you (may) get:

- Value function or policy does not change anymore
- Policy cannot be improved locally
- Policy is globally optimal
- Value function is optimal
- Value function or policy is close to the optimum

What you pay for:

- Simplified algorithms
- Space (memory) complexity
- Infinite learning times
- Results with a certain probability

Convergence of RL: General remarks

- It is not possible to a priori assess if TD(λ) will perform better than TD(0). See Sutton and Barto (1998).
- $\mathcal{Q}$-learning is an off-policy algorithm, which makes convergence control easier. SARSA and Actor-Critics (see below) are less easy to handle. It can be shown that under certain boundary conditions SARSA and $\mathcal{Q}$-learning will converge to the optimal policy if all state-action pairs are visited infinitely often.
- Actor-Critic algorithms: The Actor uses in general a set of predefined actions. Actions are not easily generated de novo. The Critic cannot generate actions on its own but must work together with the Actor. Convergence is slow if these methods are not augmented by additional mechanisms (Touzet and Santos 2001).

F. Woergoetter and B. Porr (2008) Reinforcement learning. *Scholarpedia.*

Szepesvari on contractions

- Convergence results based on norm contractions
- Basic results of the theory of Markovian decision processes
- Results for discounted expected total cost
- Based on contraction mappings and Banach's fixed-point theorem
- Applied to proof a number of basic results about value functions and optimal policies
- see Szepesvári (2009) Algorithms for RL, Appendix A

Convergence for $\mathcal{Q}$-learning

Convergence guaranteed for look-up table case.

Extremes: greedy vs. random acting (n-armed bandit models)

$\mathcal{Q}$-learning converges to optimal $\mathcal{Q}$-values if

- Every action is performed in every state infinitely often.
- The action selection is asymptotically greedy.
- The learning rate decreases according to the RM conditions

Convergence can be proven only with probability 1, as usually for stochastic gradient algorithms.

Convergence theorem for $\mathcal{Q}$-learning

Theorem: If every action is performed in every state infinitely often, $0 \leq \gamma < 1$, the initial values and the rewards are bounded, i.e. $\forall a, s : \left|\mathcal{Q}_0 (s, a)\right| < C_0$, $|r| < C_1$ then

$$\forall s, a : \lim_{t \to \infty} \mathcal{Q}_t (s, a) = \mathcal{Q}^* (s, a)$$

i.e. globally optimal $\mathcal{Q}$-values are asymptotically reached.

Convergence proof I

Proof:

Let

$$\Delta_t = \max_{s,a} \left| \mathcal{Q}_t(s,a) - \mathcal{Q}^*(s,a) \right|$$

denote the maximal error in the $\mathcal{Q}$-table.

Because $|r| < C = \max\{C_0, C_1\}$ we have
$\mathcal{Q}^* \leq \sum_{t=t_0}^{\infty} \gamma^{t-t_0} C = \frac{C}{1-\gamma}$

Because $\mathcal{Q}_0$ is bounded, also Δ_0 is bounded.

How is Δ_t affected of the agent move from state s to state s' using action a?

Convergence proof II

Immediate reward is identical for state s, but the max-$\mathcal{Q}$ action might not be the same.

$$
\begin{aligned}
\left|\mathcal{Q}_t\left(s,a\right)-\mathcal{Q}^*\left(s,a\right)\right| &= \left|\left(R+\gamma\max_{a'}Q_t\left(s',a'\right)\right)-\left(R+\gamma\max_{a''}Q^*\left(s',a''\right)\right)\right| \\
&= \gamma\left|\max_{a'}Q_t\left(s',a'\right)-\max_{a''}Q^*\left(s',a''\right)\right| \\
&\leq \gamma\max_{a'''}\left|Q_t\left(s',a'''\right)-Q^*\left(s',a'''\right)\right| \\
&\leq \gamma\max_{s'',a'''}\left|Q_t\left(s'',a'''\right)-Q^*\left(s'',a'''\right)\right| \\
&= \gamma\Delta_t
\end{aligned}
$$

i.e. after visiting the state s and performing a, $\mathcal{Q}$ differs from the optimal value by $\gamma\Delta_t$

Convergence proof III

We denote by τ_0 the start before the experiment, and by τ_N the first time when since τ_{N-1} every state-action pair has been encountered.

From the previous slide we can conclude that

$$\Delta_{\tau_N} \leq \gamma \Delta_{\tau_{N-1}}$$

The assumption the every state-action pair is visited infinitely often, gives us already

$$\lim_{t \to \infty} \Delta_t = 0$$

This completes the proof, but ...

Comments on the convergence proof

- We have assumed that $R - R = 0$ and should note that R is a random variable, Which requires more averaging than indicated by τ_N. The proof given here applies only to deterministic worlds.
- Also we did not enforce consistency within the $\mathcal{Q}$-table which is an asymptotic process (for $\eta < 1$), while here this is understood as instantaneous. Formally this is not a problem since $\mathcal{Q}^*$ is consistent by definition
- Exploration is not a problem for off-policy learning, but the exploration rate needs to decay asymptotically, which was not considered here either.
- See the proofs (with probability 1) by Jaakola, Jordan & Singh and Tsitsiklis

How many iterations ? (TD, discounted-reward)

Theorem $\forall t \;\; \|V_t - V^\pi\|_\infty \leq \gamma^t \|V_0 - V^\pi\|_\infty$

Proof: Let $\Delta_t = \|V_t - V^\pi\|_\infty$

$$\begin{aligned} V_{t+1} &= R^\pi + \gamma T^\pi V_t \\ &\leq R^\pi + \gamma T^\pi (V_k + \Delta_t) \\ &= (R^\pi + \gamma T^\pi V_k) + \gamma \Delta_t \\ &= V^\pi + \gamma \Delta_t \end{aligned}$$

Thus, if $t > \log_\gamma \frac{\varepsilon(1-\gamma)}{R\text{max}}$, then $\forall t' > t \;\; \|V_{t'} - V^\pi\|_\infty \leq \varepsilon$

Matrix inversion method (TD)

Infinite-horizon (discounted reward)

$$\begin{aligned} V^\pi &= R^\pi + \gamma T^\pi V^\pi \\ V^\pi - \gamma T^\pi V^\pi &= R^\pi \\ \left(I_{|S|} - \gamma T^\pi\right) V^\pi &= R^\pi \\ V^\pi &= \left(I_{|S|} - \gamma T^\pi\right) R^\pi \end{aligned}$$

Worst-case complexity if $|S|^3$

see: Satinder Singh

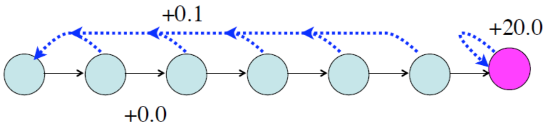
PAC-MDP Reinforcement Learning

- PAC: Probably approximately correct (Valiant 84)
- Extended to RL (Fiechter 95, Kakade 03, etc.).
- Given $\varepsilon > 0$, $\delta > 0$, A actions, S states, $\gamma < 1$.
- We say a strategy makes a mistake each time step t s.t. $\mathcal{Q}(s_t, a_t) < \max_a \mathcal{Q}(s_t, a) - \varepsilon$
- Let m be a bound on the number of mistakes that holds with probability $1 - \delta$.
- Want m to be polynomial in A, S, $1/\varepsilon$, $1/\delta$, $1/(1-\gamma)$.
- Must balance exploration and exploitation!

adapted from Michael L. Littman's talk on Model-based RL

Q-learning not PAC-MDP

- Family: initialisation, exploration, α_t decay
- Combination lock



- Initialise low, random exploration (&-greedy)
  - 2^n to find near-optimal reward. Keeps resetting.
  - Needs more external direction.

Optimism under uncertainty

- Exploration bonuses help to integrate exploration
- Shown to provide PAC-MDP guarantee (Kearns & Singh 02, Brafman & Tennenholtz 02).
- Key ideas:
  - Simulation lemma: Optimal for approximate model is near-optimal.
  - Explore or exploit lemma: If can't reach unknown states quickly, can achieve near-optimal reward.

Model-free PAC-MDP

- Solved by Strehl, Li, Wiewiora, Langford & Littman 2006
- Modifies $\mathcal{Q}$-learning to build rough model from recent experience.
- Total mistakes in learning $\sim SA/\left((1-\gamma)^8\varepsilon^4\right)$
- Compare to model-based methods: Mistakes in learning $\sim S^2A/\left(1-\gamma\right)^6\varepsilon^3$
- Lower bound, see Li 2009.

Learning: Approximation of the value function

Approximation ("probably almost correct")

$$Pr\left(\sup_{\theta\in\Theta}\left|\hat{V}(\theta)-V(\theta)\right|<\varepsilon\right)>1-\delta$$

Worst case in Θ has less than ε deviation with 1-δ confidence. How many samples H do we need for given Θ, ε, δ.
Furthermore, we set a bound η for the likelihood ratio. Then:

$$H\geq\eta\left(\frac{V\max}{\varepsilon}\right)^2\left(K(\Theta)+\log\left(\frac{8}{\delta}\right)\right)$$

So we need also $V_{\max}$ (maximum of the value) and $K(\Theta)$ the complexity of the policy space (e.g. similar to a k-means clustering): Assume there are experiences under k other policies $\theta_1, \ldots, \theta_k$. If they are sufficient to provide a good representative for any $\theta\in\Theta$ and for any $k-1$ of them this is not the case then $K(\Theta)=k$.

Comparison of results for trajectories of length T

Peshkin & Mukherjee (2001)

$$O\left(\left(\frac{V\max}{\varepsilon}\right)^2 2^T \left(K(\Theta) + \log\left(\frac{8}{\delta}\right)\right)\right)$$

Kearns, Mansour, Ng (2000)

$$O\left(\left(\frac{V\max}{\varepsilon}\right)^2 2^{2T} VC(\Theta)\log(T)\left(T + \log\left(\frac{V_{\max}}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right)\right)$$

use Vapnik-Chervonenkis (VC) dimension instead of covering number K for describing the complexity of the policy space and assume:

- Partial reuse of policies
- Fixed sampling policy (uniformly random)

VC is usually larger than K; can be related to depth of tree

Exponential time dependency required for generality

Using a model

- Large state spaces
  - factorisable transition probabilities
- POMDP with a restricted class of strategies Π
  - chose $\pi \in \Pi$ with maximal return
- what is *sample complexity*? From supervised learning
  - How many samples are needed to learn a function $f \in \mathcal{F}$ of a certain complexity?
  - e.g. neural network realises $h(x)$ with $h \in \mathcal{H}$ in order to approximate $f(x)$. Assume $|\mathcal{H}| = n$ then typically only $O(\log(n))$ samples are needed to find a good $h(n)$.
  - Since we are choosing from $\mathcal{H}$ the complexity of f does not play a role (if $|\mathcal{H}|$ is small and $|\mathcal{F}|$ is large)
- Assume a simulator (a generative model) of the POMDP
- Find bounds on the required amount of simulated experience

Sample complexity in a POMDP

- Using the policy $\pi \in \Pi$ and starting state s_0, generate many trials (MC-style) and find $V^{\pi}(s_0)$
- Now for a different $\pi' \in \Pi$ what use can we make of these trials?
- If we cannot re-use these trials we are left with a complexity $O(n)$ if $|\Pi| = n$ (instead of e.g. $O(\log(n))$)
  [Π does not have to be finite here]
- Several methods for generating reusable trajectories:
  - trajectory trees (easier, but specific generative model)
  - random trajectories (harder, but simple generative model)
  - likelihood ratios
- Number of required trajectories indep. of state space size
- Linear in complexity of policy space

Generative model

Given a POMDP M, then a model of M is

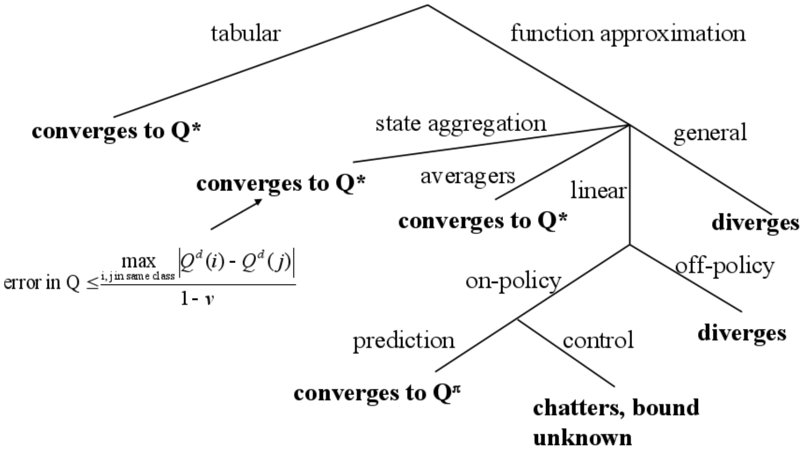
- a randomised algorithm that for a given state-action pair (s, a) outputs
  - a state s_0 that is distributed according to the next-state distribution $P(\cdot|s, a)$,
  - an observation o that is distributed according to the distribution $\mathcal{Q}(\cdot|s)$, and
  - the reward $R(s, a)$.

Task: Let M be a POMDP with start state s_0, and let Π be a class of strategies. Find

$$opt(M, \Pi) = \sup_{\pi \in \Pi} V^{\pi}(s_0)$$

where $V^{\pi}(s_0)$ is the expected return of π from s_0.

Convergence results of Q-learning



From: Tuomas Sandholm, Carnegie Mellon University.

Convergence with function approximation

- Somewhat incomplete for the analysis when function approximation is used (Chapter 6 of Bertsekas and Tsitsiklis, 1996).
- Bounding the behaviour of greedy policies obtained via function approximation (Williams and Baird, 1993; and Singh and Yee, 1994).
- TD methods using function approximation are known to converge (Sutton, 1984, 1992).
- Function approximation with state aggregation has also been been analysed (Tsitsiklis and van Roy, 1996).

Gosavi, Abhijit. "Reinforcement learning: A tutorial survey and recent advances." INFORMS J. on Computing 21.2 (2009): 178-192.

Conclusion and more

Conclusions

- Plain algorithms theoretically accessible, but usually prohibitively complex
- Convergence difficult for function approximation
- Model-based algorithms also theoretically preferable

More

- Algorithms derived from PAC bounds
- Martingales & reinforcement learning: Seldin et al. (2011, 2012)
- Efficient sampling (Kearns, M. NIPS 12, 1999).

Acknowledgements & References

Some material was adapted from web resources associated with Sutton and Barto's Reinforcement Learning book.

and on the slides by Dr. Subramanian Ramamoorthy from the previous years.

Today mainly based on C. Szepesvári: *Algorithms for RL*, Ch. 4.

See also:

Satinder Singh: Reinforcement Learning: A Tutorial + Rethinking State, Action, and Reward
http://learning.stat.purdue.edu/mlss/_media/mlss/singh.pdf

M. Littman: Model-based RL.