Slide 1

Reinforcement learning in general state spaces

Francisco S. Melo
fmelo@isr.ist.utl.pt

Reading group on Sequential Decision Making

Slide 2

Outline of the presentation

- **Background**
- Some history on RL with FA
- TD(0) with linear function approximation
- Interpolated Q-learning
- References

Slide 3

Background

A *Markov chain* is a pair $(\mathcal{X}, \mathsf{P})$ where

- $\mathcal{X}$ is the (general) state-space;
- P is a *transition probability kernel*:

$$\mathsf{P}(x, U) = \mathbb{P}\left[X_{t+1} \in U \mid X_t = x\right];$$

- *Positive chains* admit an invariant probability measure μ;
- *Geometrically ergodic chains* converge exponentially fast to μ:

$$\sum_{t=0}^{\infty} \rho^t \left\|\mathsf{P}^t(x, \cdot) - \mu(\cdot)\right\| \leq \infty,$$

for all $x \in \mathcal{X}$, with $\rho > 1$.

Slide 4

Background (2)

An *MDP* is a tuple $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ where

- $\mathcal{X}$ is the (general) state-space;
- $\mathcal{A}$ is the finite action-space;
- P is a controlled *transition probability kernel*:

$$\mathsf{P}_a(x, U) = \mathbb{P}\left[X_{t+1} \in U \mid X_t = x, A_t = a\right];$$

- $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$ is a reward function;
- γ is a discount factor.

Slide 5

Background (3)

- The optimal value function is

$$V^*(x) = \max_{\{A_t\}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(X_t, A_t) \mid X_0 = x\right];$$

- V^* verifies the Bellman optimality equation

$$V^*(x) = \max_{a \in \mathcal{A}} \int_{\mathcal{X}} \big[r(x,a,y) + \gamma V^*(y)\big] \mathsf{P}_a(x, dy);$$

- The optimal Q-function is simply

$$Q^*(x,a) = \int_{\mathcal{X}} \big[r(x,a,y) + \gamma V^*(y)\big] \mathsf{P}_a(x, dy).$$

Slide 6

Background (4)

- TD(0) evaluates a policy δ using the update

$$V_{t+1}(X_t) = V_t(X_t) + \alpha_t\big[R_t + \gamma V_t(X_{t+1}) - V_t(X_t)\big];$$

- Q-learning determines the optimal Q^* using the update

$$Q_{t+1}(X_t, A_t) = Q_t(X_t, A_t) + \alpha_t\big[R_t + \gamma \max_{b \in \mathcal{A}} Q_t(X_{t+1}, b) - Q_t(X_t, A_t)\big].$$

Slide 7

Outline of the presentation

- Background
- **Some history on RL with FA**
- TD(0) with linear function approximation
- Interpolated Q-learning
- References

Slide 8

Some history

- Samuel's pioneer works in machine learning (late 50's/early 60's) describe an artificial checker's player and a "feature"-based approximation [19, 20];
- In 1992, Tesauro combined $TD(\lambda)$ and non-linear function approximation (using a neural network). Its results boosted the interest int the problem of generalization [27, 28, 29];
- In 1993, Thrun and Schwartz discuss the use of reinforcement learning with function approximation [30];
- Singh et al. propose the use of *soft-state aggregation* with reinforcement learning, *proving convergence w.p.1* of the obtained method;

Slide 9

Some history (2)

- Soft-state aggregation was further addressed by Gordon [12] and Tsitsiklis and Van Roy [31];
- Baird [3] and Gordon [11] provide divergent counter-examples for Q-learning and SARSA. Baird proposes the use of gradient ascent algorithms, to overcome the convergence limitations of standard RL methods with function approximation [2, 4];
- Boyan and Moore [7] and Sutton [23] experimentally evaluate several approximation architectures;
- Tsitsiklis and Van Roy provide a fundamental analysis of TD(λ) with function approximation, establishing convergence w.p.1 for linear function approximation [32];
- The fundamental insight was further explored in other works [5, 8, 9]

Slide 10

Some history (3)

Recent days have witnessed several new and exciting works, namely:

- Functional Q-learning [6];
- Off-policy TD(λ) with established convergence [17];
- Kernel-based reinforcement learning [16];
- Interpolation based Q-learning [26];
- Other [1, 13, 15, 25].

Slide 11

Outline of the presentation

- Background
- Some history on RL with FA
- **TD(0) with linear function approximation**
- Interpolated Q-learning
- References

Slide 12

Main idea

- Require the sampling policy to yield a geometrically ergodic chain;
- Aproximate the algorithm

$$\theta_{t+1} = \theta_t + \alpha_t H(\theta_t, X_{t+1})$$

by considering the "average" algorithm

$$\tilde{\theta}_{t+1} = \tilde{\theta}_t + \alpha_t \mathbb{E}_\mu \left[H(\theta_t, X_{t+1}) \right].$$

- Writing $h(\theta) = \mathbb{E}_\mu \left[H(\theta, X_{t+1}) \right]$, analyze the ODE

$$\dot{\theta}_t = h(\theta_t).$$

Slide 13

TD(0) with linear function approximation

- Consider a fixed policy δ;
- Aproximate the function V^δ as a linear combination of basis functions ξ_i, $i = 1, \ldots, M$:

$$V^\delta(x) \approx \tilde{V}(x, \theta) = \sum_{i=1}^{M} \xi_i(x)\theta_i = \xi^\top(x)\theta.$$

- Iterate in θ using the update

$$\theta_{t+1} = \theta_t + \alpha_t \xi(X_t)\big[R_t + \gamma V(X_{t+1}, \theta_t) - V(X_t, \theta_t)\big].$$

Slide 14

TD(0) with linear function approximation

- Using the previous approach, the ODE becomes

$$\dot{\theta}_{t+1} = \mathbb{E}_\mu\left[\mathbf{A}(X_t)\right]\theta_t + \mathbb{E}_\mu\left[\mathbf{b}(X_t)\right],$$

with a negative definite matrix A. Then,

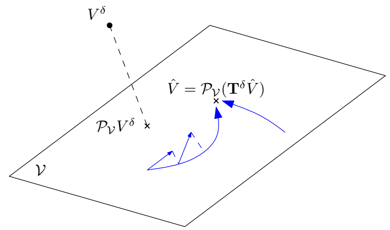
$$\theta^* = -\mathbb{E}_\mu\left[\mathbf{A}(X_t)\right]^{-1}\mathbb{E}_\mu\left[\mathbf{b}(X_t)\right]$$

is a *globally asymptotically stable equilibrium point* of the ODE;

Slide 15

TD(0) with linear function approximation

The limit point is the fixed point $\hat{V} = \mathcal{P}_\mathcal{V}\mathbf{T}^\delta\hat{V}$.



Slide 16

Outline of the presentation

- Background
- Some history on RL with FA
- TD(0) with linear function approximation
- **Interpolated Q-learning**
- References

Slide 17

Problems with Q-learning:

The fixed point $\hat{V} = \mathcal{P}_{\mathcal{V}} \mathbf{T}^{\delta} \hat{V}$ exists because:

- $\mathbf{T}^{\delta}$ is a *contraction* in the 2-norm;
- $\mathcal{P}_{\mathcal{V}}$ is a *non-expansion* in the 2-norm;
- The combined operator $\mathcal{P}_{\mathcal{V}} \mathbf{T}^{\delta}$ is a contraction in the 2-norm.

But the operator $\mathbf{H}$ associated with Q^* is a contraction *in the sup-norm*...

Slide 18

Interpolation-based Q-learning

- An idea is to use a "projection-like" operator $\mathcal{P}$ that is a non-expansion in the sup-norm;
- Interpolation-based Q-learning defines a set of points $I = \{(x_1, a_1), \ldots, (x_M, a_M)\}$ and uses the projection-like operator

$$(\mathcal{P}q)(x, a) = \mathcal{F}_{\theta}(x, a),$$

where $\mathcal{F}_{\theta}$ is a *convex interpolator* and θ is a vector such that $\theta_i = q(x_i, a_i)$;

Slide 19

Interpolation-based Q-learning

- The parameters θ are updated using the rule

$$\theta_{t+1} = (1 - \alpha)\theta_t + \alpha_t g_{\varepsilon}(X_t, A_t)\big[R_t + \gamma \max_{b \in \mathcal{A}} F_{\theta_t}(X_{t+1}, b), \theta_t)\big];$$

- The limit point is now the fixed point $\hat{Q} = \mathcal{P}\hat{\mathbf{H}}\hat{Q}$.

Slide 20

*****

References

[1] A. Antos, C. Szepesvári, and R. Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT'06)*, pages 574–588, 2006.

[2] L. C. Baird. *Reinforcement learning through gradient descent.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, May 1999.

[3] L. C. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*, pages 30–37, San Francisco, CA, 1995. Morgan Kaufman Publishers.

[4] L. C. Baird and A. More. Gradient descent for general reinforcement learning. In D. A. Cohn, editor, *Advances in Neural Information Processing Systems*, volume 11, pages 968–974, Cambridge, Massachussets, 1999. MIT Press.

Slide 21

[5] D. P. Bertsekas, V. S. Borkar, and A. Nedić. *Improved temporal difference methods with linear function approximation*, chapter 9, pages 235–260. Wiley Publishers, 2004.

[6] V. S. Borkar. A learning algorithm for discrete-time stochastic control. *Probability in the Engineering and Informational Sciences*, 14:243–258, 2000.

[7] J. Boyan and A. Moore. Generalization in reinforcement learning: Safely approximating the value function. In G. Tesauro, D. Touretzky, and T. Lee, editors, *Neural Information Processing Systems 7*, pages 369–376, Cambridge, MA, 1995. The MIT Press.

[8] J. A. Boyan. Least-squares temporal difference learning. In *Proceedings of the 16th International Conference on Machine Learning (ICML'99)*, pages 49–56, San Francisco, CA, 1999. Morgan Kaufmann.

[9] J. A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49:233–246, 2002.

[10] G. J. Gordon. Reinforcement learning with function approximation converges to a region. In *Proceedings of the Neural Information Processing Systems*, pages 1040–1046, 2000.

Slide 22

[11] G. J. Gordon. Chattering in SARSA(λ). Cmu learning lab internal report, CMU Learning Lab, Carnegie Mellon University, 1996.

[12] G. J. Gordon. Stable function approximation in dynamic programming. Technical Report CMU-CS-95-103, School of Computer Science, Carnegie Mellon University, 1995.

[13] F. S. Melo and M. I. Ribeiro. Q-learning and linear function approximation. In *Proceedings of the 20th Annual Conference on Learning Theory (COLT'07)*, page (to appear), 2007.

[14] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Communications and Control Engineering Series. Springer-Verlag, New York, 1993.

[15] R. Munos. Performance bounds in L_p-norm for approximate value iteration. *SIAM Journal on Control and Optimization*, (to appear), 2007.

[16] D. Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. *Machine Learning*, 49:161–178, 2002.

Slide 23

[17] D. Precup, R. S. Sutton, and S. Dasgupta. Off-policy temporal-difference learning with function approximation. In *Proceedings of the 18th International Conference on Machine Learning (ICML'01)*, pages 417–424, San Francisco, CA, 2001. Morgan Kaufmann.

[18] J. Rust. Using randomization to break the curse of dimensionality. *Econometrica*, 65(3):487–516, 1997.

[19] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959. Reprinted in IBM J. Res. Devel. 44:1/2, pp. 206-226, 2000.

[20] A. L. Samuel. Some studies in machine learning using the game of checkers II: Recent progress. *IBM Journal of Research and Development*, 11:601–617, 1967.

[21] S. P. Singh, T. Jaakkola, and M. I. Jordan. Reinforcement learning with soft state aggregation. In *Advances in Neural Information Processing Systems*, volume 7, pages 361–368. 1994.

[22] W. D. Smart and L. P. Kaelbling. Practical reinforcement learning in continuous spaces. In P. Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML'00)*, pages 903–910, 2000.

Slide 24

[23] R. S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems*, 8:1038–1044, 1996.

[24] R. S. Sutton. Open theoretical questions in reinforcement learning. *Lecture Notes in Computer Science*, 1572:11–17, 1999. URL `citeseer.ist.psu.edu/sutton99open.html`.

[25] C. Szepesvári and R. Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, volume 119 of *ACM International Conference Proceeding Series*, pages 880–887. ACM Press, 2005.

[26] C. Szepesvári and W. D. Smart. Interpolation-based Q-learning. In *Proceedings of the 21st International Conference on Machine learning (ICML'04)*, pages 100–107, New York, USA, July 2004. ACM Press.

[27] G. Tesauro. Practical issues in temporal difference learning. *Machine Learning*, 8 (3-4):257–277, 1992.

[28] G. Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.

Slide 25

[29] G. Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[30] S. Thrun and A. Schwartz. Issues in using function approximation for reinforcement learning. In M. Mozer, P. Smolensky, D. Touretzky, J. Elman, and A. Weigend, editors, *Proceedings of the 1993 Connectionist Models Summer School*, Hillsdale, NJ, 1993. Erlbaum Associates.

[31] J. N. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22:59–94, 1996.

[32] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5): 674–690, May 1996.

[33] J. N. Tsitsiklis and B. Van Roy. On average versus discounted reward temporal-difference learning. *Machine Learning*, 49(2):179–191, 2002.

[34] B. Van Roy. *Learning and value function approximation in complex decision processes*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1998.