Q-learning with linear function approximation

Francisco S. Melo and M. Isabel Ribeiro
Institute for Systems and Robotics
[fmelo,mir]@isr.ist.utl.pt

Conference on Learning Theory,
COLT'2007

Outline of the presentation

- **Motivation and problem formulation**
- Related work
- Q-learning with LFA
- Addressing partial observability
- Concluding remarks

Motivation

- Markov decision processes (MDPs) provide useful models to address sequential decision problems;
- Many powerful methods are available (*e.g.*, TD(λ), Q-learning, SARSA).

However...

- Many such methods rely on *explicit representations of the state-space*;
- Many interesting problems have a state-space unsuited for explicit representation (*e.g.*, infinite or partially observable);

Problem formulation

- In this paper we address Markov decision problems with *infinite state-space* or *partial observability*;
- We propose a modified version of Q-learning that accomodates MDPs with infinite state-space;
- To this end, we make use of *linear function approximation* to achieve compact representation;
- We identify conditions under which this same algorithm can be applied to partially observable scenarios.

Some notation

We represent a MDP as a tuple $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ where

- $\mathcal{X}$ and $\mathcal{A}$ are the state and action-spaces, respectively;
- P is the transition probability kernel

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

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

Some notation (cont.)

- The agent should choose the sequence of actions $\{A_t\}$ maximizing

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

- For the optimal action sequence, the corresponding values verify

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

- The optimal Q-function is defined as

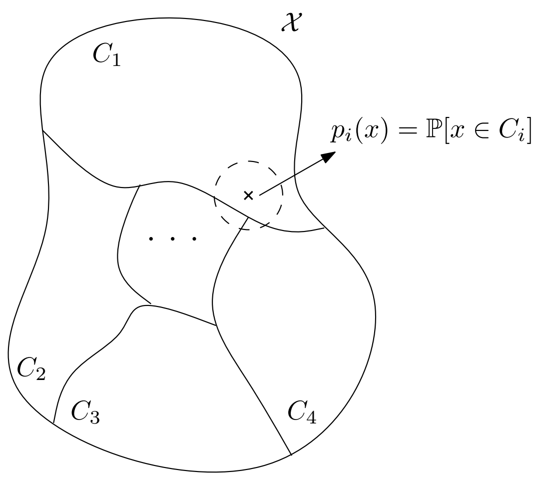
$$Q^*(x,a) = \int_{\mathcal{X}} \big[r(x,a,y) + \gamma V^*(y)\big] \mathsf{P}_a(x, dy).$$

Outline of the presentation

- Motivation and problem formulation
- **Related work**
- Q-learning with LFA
- Addressing partial observability
- Concluding remarks

Related work

- *Soft-state aggregation methods* [4, 9, 11] partition the state-space into M regions $C_1, \ldots, C_M$;
- Each $x \in \mathcal{X}$ "belongs" to region C_i with probability $p_i(x)$;
- The algorithms consider each C_i as a "hyper-state" and compute the corresponding values, $\theta(i, a)$;



- The optimal Q-function is then approximated as

$$\hat{Q}(x, a) = \sum_i \theta(i, a) p_i(x).$$

Related work (cont.)

- Tsitsiklis and Van Roy [12] consider a finite-dimensional function space $\mathcal{V}$ obtained as the linear span of a set of M linearly independent functions $\xi_1, \ldots, \xi_M$;
- The authors implement a stochastic approximation algorithm to determine the fixed point

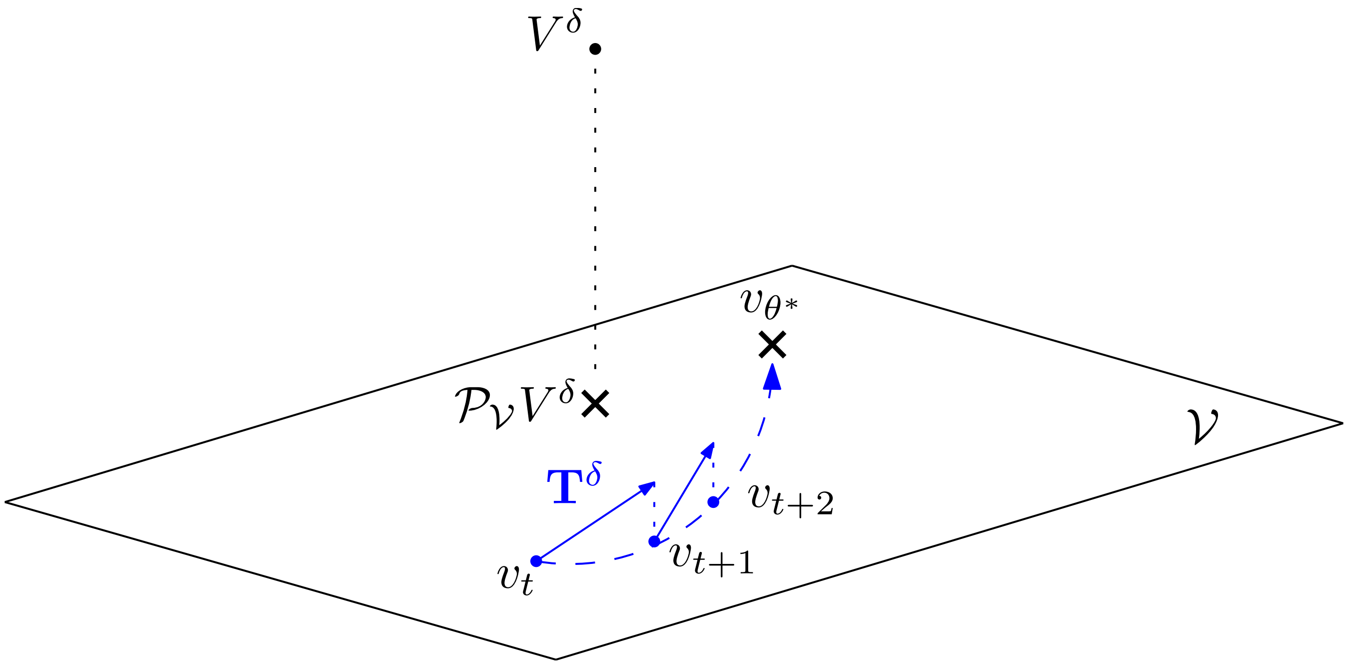
$$v_{\theta^*} = \mathcal{P}_{\mathcal{V}} \mathbf{T}^{\delta} v_{\theta^*},$$

  where $\mathcal{P}_{\mathcal{V}}$ is the orthogonal projection on $\mathcal{V}$ and $\mathbf{T}^{\delta}$ is the TD-operator;

Related work (cont.)

- Convergence is established by showing the algorithm to follow the trajectories of a globally asymptotically stable ODE

$$\dot{v}_t = \mathcal{P}_{\mathcal{V}} \mathbf{T}^{\delta} v_t - v_t.$$



Related work (cont.)

- Szepesvári and Smart [10] consider Q-learning with interpolative function approximation;
- The authors consider a sample-based operator $\mathcal{P}$ that projects a generic function q to a finite-dimensional parameter space by considering the value of q at a pre-specified set of sample points;
- Combined with an interpolation operator F, this yields a non-expansive, equipotent operator $\mathcal{G} = F\mathcal{P}$;
- The algorithm proceeds by determining the fixed point

$$q_{\theta^*} = \mathcal{G}\mathbf{H}q_{\theta^*},$$

where $\mathbf{H}$ is the Bellman operator.

Outline of the presentation

- Motivation and problem formulation
- Related work
- **Q-learning with LFA**
- Addressing partial observability
- Concluding remarks

Q-learning with LFA

- Our approach is a combination of that in [12] with the one in [10];
- As in [12], we consider a finite-dimensional function space $\mathcal{Q}$ obtained as the linear span of a set of M linearly independent functions $\xi_1, \ldots, \xi_M$ and implement a stochastic approximation algorithm to determine the fixed point

$$q_{\theta^*} = \mathcal{G}\mathbf{H}q_{\theta^*},$$

where now $\mathcal{G}$ is the sample-based projection on $\mathcal{Q}$ defined on [10];

Q-learning with LFA (cont.)

- To ensure that, for a generic function q, $\mathcal{G}(q)$ lies in $\mathcal{Q}$, we define the interpolation operator F from the functions ξ_i;
- Each sample point is chosen so that *one* of the functions ξ_i attains its maximum value of 1 at that point and require that $\sum_i |\xi_i(x,a)| \leq 1$;
- Then, given a parameter vector $\theta \in \mathbb{R}^M$,

$$F_\theta(x,a) = \xi^\top(x,a)\theta.$$

- Convergence is established by showing the algorithm to follow the trajectories of a globally asymptotically stable ODE

$$\dot{q}_t = \mathcal{G}\mathbf{H}q_t - q_t.$$

Two important remarks

1. In order to establish the convergence of the method by means of an associated ODE requires the underlying Markov process to be *geometrically ergodic*;

2. Since $\mathbf{H}$ is contractive in the sup-norm and $\mathcal{G}$ is non-expansive in that same norm, the combined operator $\mathcal{G}\mathbf{H}$ is contractive in the sup-norm. This, in particular, implies that

   - The fixed-point q_{θ^*} of the combined operator $\mathcal{G}\mathbf{H}$ is a globally asymptotically stable equilibrium point of the associated ODE;
   - The obtained approximation verifies

$$\|q_{\theta^*} - Q^*\|_\infty \leq \frac{1}{1-\gamma} \|\mathcal{G}(Q^*) - Q^*\|_\infty .$$

Outline of the presentation

- Motivation and problem formulation
- Related work
- Q-learning with LFA
- **Addressing partial observability**
- Concluding remarks

Addressing partial observability

A partially observable MDP (POMDP) is a tuple $(\mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathsf{P}, \mathsf{O}, r, \gamma)$ where

- $\mathcal{X}$, $\mathcal{A}$, P, r and γ are as before;
- $\mathcal{Z}$ is the set of possible observations;
- O is the observation probability function

$$\mathsf{O}_a(x, z) = \mathbb{P}\left[Z_{t+1} = z \mid X_{t+1} = x, A_t = a\right].$$

We assume $\mathcal{X}$ and $\mathcal{Z}$ to be finite sets.

Internal state

- Due to partial observability, the agent no longer accesses the state X_t of the process;
- The action choice must now depend on the history of past observations;
- Defining the vector b_t to be

$$b_t(x) = \mathbb{P}\left[X_t = x \mid \mathcal{F}_t\right]$$

  it can be updated using a simple bayesian update [2]

$$b_{t+1}(y) = \frac{\sum_{x \in \mathcal{X}} b_t(x) \mathsf{P}_{A_t}(x, y) \mathsf{O}_{A_t}(y, Z_{t+1})}{\sum_{x, w \in \mathcal{X}} b_t(x) \mathsf{P}_{A_t}(x, w) \mathsf{O}_{A_t}(w, Z_{t+1})}$$

Internal state (cont.)

- The vector b_t translates the *belief* of the agent on the current state;
- In terms of beliefs, the agent should now choose the sequence of actions $\{A_t\}$ maximizing

$$V(\{A_t\}, b) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \mid B_0 = b\right];$$

- For the optimal action sequence, the corresponding values now verify

$$V^*(b) = \max_{a \in \mathcal{A}} \sum_{x,y \in \mathcal{X}} b(x) \mathsf{P}_a(x, y) \left[r(x, a, y) + \gamma \sum_{z \in \mathcal{Z}} \mathsf{O}_a(y, z) V^*(b'_{a,z})\right],$$

where $b'_{a,z}$ is the updated belief given action a and observation z.

Internal state (cont.)

However...

- The belief vector b_t is Markovian in its dependence of the past;
- We can thus define a fully observable MDP $(\mathbb{S}^n, \mathcal{A}, \bar{\mathsf{P}}, \bar{r}, \gamma)$ from the $(\mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathsf{P}, \mathsf{O}, r, \gamma)$ where [1]
  - $\mathbb{S}^n$ is the $n-1$-dimensional probability simplex, where $n = |\mathcal{X}|$;
  - $\bar{\mathsf{P}}$ is the transition probability kernel

$$\bar{\mathsf{P}}_a(b, U) = \sum_{z \in \mathcal{Z}} \sum_{x,y \in \mathcal{X}} b(x)\mathsf{P}_a(x, y)\mathsf{O}_a(y, z)\mathbb{I}_U(b'_{a,z});$$

  - $\bar{r}$ is the reward function

$$\bar{r}(b, a, b') = \sum_{x,y \in \mathcal{X}} b(x)\mathsf{P}_a(x, y)r(x, a, y).$$

QL with LFA in POMDPs

- Exact methods for POMDPs are of little use in all but the smallest problems [6, 8];
- Since solving a POMDP $(\mathcal{X}, \mathcal{A}, \mathcal{Z}, \mathsf{P}, \mathsf{O}, r, \gamma)$ is equivalent to solving the MDP $(\mathbb{S}^n, \mathcal{A}, \bar{\mathsf{P}}, \bar{r}, \gamma)$, we can apply our Q-learning algorithm with LFA to the MDP $(\mathbb{S}^n, \mathcal{A}, \bar{\mathsf{P}}, \bar{r}, \gamma)$;
- As seen, we need only guarantee that the underlying process is *geometrically ergodic*;
- We thus conclude with a very simple result: if the MDP $(\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ is ergodic and there is *one distinguishable state*, then the MDP $(\mathbb{S}^n, \mathcal{A}, \bar{\mathsf{P}}, \bar{r}, \gamma)$ is geometrically ergodic.

Outline of the presentation

- Motivation and problem formulation
- Related work
- Q-learning with LFA
- Addressing partial observability
- **Concluding remarks**

Concluding remarks

- Error bounds depend on $\|\mathcal{G}(Q^*) - Q^*\|_\infty$ – bad approximations may yield bad policies;
- The choice of a "good approximation" is a topic of current research [3, 5, 7];
- The algorithm uses a *fixed learning policy*; extension to a θ-dependent policy should be possible, by requiring the dependence on θ to be smooth;
- The use of a θ-dependent policy suggests that an on-policy version of the algorithm could probably be derived from our algorithm;
- Although we do not consider them, we belief that the algorith can easily be modified to accomodate eligibility traces, eventually improving the obtained error bounds;

Concluding remarks (cont.)

- In the partially observable setup, belief tracking requires knowledge of the dynamic model (transition and observation probabilities); this is a common assumption in several situations (*e.g.*, robotic tasks);
- The use of learning algorithms and function approximation, even if relying on belief tracking, may constitute an appealing alternative, given the complexity of exact methods;
- Finally, requiring one state to be distinguishable is often acceptable (the goal state is often observable); furthermore, this condition is sufficient (not necessary) and often simple to check in practice.

*

References

[1] A. R. Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes*. PhD thesis, Brown University, May 1998.

[2] A. R. Cassandra. Optimal policies for partially observable Markov decision processes. Technical Report CS-94-14, Department of Computer Sciences, Brown University, August 1994.

[3] R. Glaubius and W. D. Smart. Manifold representations for value-function approximation in reinforcement learning. Technical Report 05-19, Department of Computer Science and Engineering, Washington University in St. Louis, 2005.

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

[5] P. W. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In

Proceedings of the 23rd International Conference on Machine Learning (ICML'06), pages 449–456, New York, NY, 2006. ACM Press.

[6] C. Lusena, J. Goldsmith, and M. Mundhenk. Nonapproximability results for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 14:83–103, 2001.

[7] I. Menache, S. Mannor, and N. Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research*, 134(1): 215–238, February 2005.

[8] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov chain decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[9] 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.

[10] 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.

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

[12] 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.