Reinforcement learning and function approximation

Alexandre Bouchard

June 16, 2004
Contents

1 Problem formulation 0-2
2 Convergence of approximate $TD(\lambda)$ 0-7
3 Sparse Distributed Memories (SDM) 0-19
1 Problem formulation

- Discrete time
- \( \forall \) stage \( t \in \{0, 1, 2, \ldots \} \) the agent is in some state \( i_t \in S \)
- It chooses an action \( u \in U_{i_t} \)
- Assume that \( A := \bigcup_{s \in S} U_s \) and \( S \) are finite
- Represent the transition probability from state \( i \) to \( j \) given that \( u \) is selected by \( p_{ij}(u) \)
- Cost for transition from state \( i_t \) to \( i_{t+1} \) given that \( u \) is selected is denoted by \( g(i_t, u, i_{t+1}) \)
The goal of the agent is to find a policy

$$\pi := (\mu_k : S \rightarrow A)_{k=0}^{\infty}$$

that minimizes the expected long-term cost:

$$E\left[ \sum_{t=0}^{\infty} \alpha^t g(i_t, \mu_t(i_t), i_{t+1}) \right]$$

the expectation is taken with respect to the distribution of the Markov chain \((i_0, i_1, \ldots)\) characterized by the transition probabilities \(p_{i_t,i_{t+1}}(\mu_t(i_t))\) of the problem and \(0 < \alpha \leq 1\) is a fixed discount factor.

For today we will consider stationary policies, that is policies such that:

$$\mu := (\mu, \mu, \ldots)$$
Cost-to-go functions:

\[ J^\mu(i) := \liminf_{N \to \infty} E \left[ \sum_{t=0}^{N} \alpha^t g(i_t, \mu(i_t), i_{t+1}) | i_0 = i \right] \]

Optimal cost-to-go function:

\[ J^*(i) := \min_{\pi} J^\mu(i) \]
Linear approximators

\[ \tilde{J}(i, \vec{r}) := \sum_{k=1}^{K} \vec{r}[k] \phi_k(i) \]

where

\[ \vec{r} \in \mathbb{R}^K \]

is a vector of tunable parameters and

\[ \phi_k : S \rightarrow \mathbb{R} \]

are the basis functions.
Suppose we are given a RL problem with \( S, A, \alpha \) known, \( p(u) \), \( g \) unknown initially and that for a fixed policy \( \mu \) the agent is able to generate trajectories \( (i_t: i_t \in S)_{t=0}^{\infty} \) according to the transition probabilities \( p_{i_t,i_{t+1}}(\mu(i_t)) \).

A possible strategy to solve this problem:

1. Start with some policy \( \mu \)
2. Evaluate for this fixed policy its cost-to-go vector \( J^\mu \)
3. Apply some improvement to the current policy
4. If the policy can still be improved, go to step 2 (exist only a finite number of policies)

Today, we focus on step 2, namely the evaluation of the cost-to-go vector for a fixed policy.
2 Convergence of approximate $TD(\lambda)$

Proof based on:
Theorem 1 Assumptions:

• RL problem as stated before and a fixed stationary policy \( \mu \)

• A linear approximation architecture with bounded basis functions. For convenience, define:

\[
\vec{\phi}(i) := (\phi_1(i), \ldots, \phi_K(i))
\]

and also the \( n \) by \( K \) matrix:

\[
\Phi := 
\begin{pmatrix}
| & | \\
\phi_1 & \cdots & \phi_K \\
| & |
\end{pmatrix}
= 
\begin{pmatrix}
\begin{bmatrix}
- \vec{\phi}(1)^T \\
\vdots \\
- \vec{\phi}(n)^T
\end{bmatrix}
\end{pmatrix}
\]

• The update rule is:

\[
\vec{r}_{t+1} := \vec{r}_t + \gamma_t d_t \vec{z}_t
\]
In the update rule:
\[ \tilde{r}_{t+1} := \tilde{r}_t + \gamma_t d_t \hat{z}_t \]

d\(_t\) is called the temporal difference
\[ d_t := g(i_t, i_{t+1}) + \alpha \tilde{J}(i_{t+1}, \tilde{r}_t) - \tilde{J}(i_t, \tilde{r}_t) \]
In the update rule:
\[
\vec{r}_{t+1} := \vec{r}_t + \gamma_t d_t \vec{z}_t
\]

\(\vec{z}_t\) is called the eligibility vector

\[
\vec{z}_t := \sum_{k=0}^t (\alpha \lambda)^{t-k} \nabla \tilde{J}(i_t, \vec{r}_t)
\]

\[
= \sum_{k=0}^t (\alpha \lambda)^{t-k} \vec{\phi}(i_k)
\]

Note that \(\vec{z}_{t+1}\) can be computed easily by:

\[
\vec{z}_{t+1} := \alpha \lambda \vec{z}_t + \vec{\phi}(i_{t+1})
\]
In the update rule:

\[ \vec{r}_{t+1} := \vec{r}_t + \gamma_t d_t \vec{z}_t \]

\( \gamma_t \) is called the step size sequence. It is assumed to satisfy:

\[ \sum_{t=0}^{\infty} \gamma_t = \infty \]

\[ \sum_{t=0}^{\infty} \gamma_t^2 < \infty \]
Moreover, we assume that:

1. There exists positive numbers $\bar{\pi}[j]$ such that:

   $$\lim_{t \to 0} P(i_t = j | i_0 = i) = \bar{\pi}[j] \quad \forall i, j$$

2. We have $K \leq n$ and $\Phi$ has full rank.

Then, under these assumptions, the sequence $\vec{r}_t$ converges, with probability 1.
Lemma 1  Let $X_t$ be a Markov process taking values in a set $S$. Define the maps:

$$\begin{cases} 
A : S \rightarrow M_n(\mathbb{R}) \\
\vec{b} : S \rightarrow \mathbb{R}^n
\end{cases}$$

Then the sequence $\vec{r}_t$ defined by:

$$\vec{r}_{t+1} := \vec{r}_t + \gamma_t (A(X_t)\vec{r}_t + \vec{b}(X_t))$$

converges with probability one, provided that:
1. The sequence of stepsize $\gamma_t$ is deterministic and satisfies $\sum_{t=0}^{\infty} \gamma_t = \infty$ and $\sum_{t=0}^{\infty} \gamma_t^2 < \infty$.
2. The Markov process $X_t$ has an invariant distribution. Let $E_0[\cdot]$ denotes the expectation with respect to this distribution.
3. We have $A := E_0\left[A(X_t)\right]$ is negative definite.
4. There is a $K$ such that for all $X_t$, $\|A(X_t)\| \leq K$, $\|\vec{b}(X_t)\| \leq K$.
5. There are $\rho$, $C$ such that:

$$\|E\left[A(X_t)\|X_0\right] - A\| \leq C\rho^n \quad \forall n \geq 0, X_0$$

and

$$\|E\left[\vec{b}(X_t)\|X_0\right] - \vec{b}\| \leq C\rho^n \quad \forall n \geq 0, X_0$$

where $\vec{b} := E_0\left[\vec{b}(X_t)\right]$. 
We now reduce the problem of convergence of $TD(\lambda)$ to the case of a stochastic iterative algorithm with Markov noise. Let:

$$X_t := (i_t, i_{t+1}, \vec{z}_t)$$

This is indeed a Markov process. Recall that:

$$\vec{z}_{t+1} := \alpha \lambda \vec{z}_t + \vec{\phi}(i_{t+1})$$
Define the maps:

\[
\begin{align*}
\vec{b}(X_t) & := \vec{z}_t g(i_t, i_{t+1}) \\
A(X_t) & := \vec{z}_t (\alpha \vec{\phi}(i_{t+1})^T - \vec{\phi}(i_t)^T)
\end{align*}
\]

Check it works:

\[
\vec{r}_{t+1} := \vec{r}_t + \gamma_t (A(X_t)\vec{r}_t + \vec{b}(X_t))
\]
\[
= \vec{r}_t + \gamma_t (\vec{z}_t (\alpha \vec{\phi}(i_{t+1})^T - \vec{\phi}(i_t)^T) \vec{r}_t + [\vec{z}_t g(i_t, i_{t+1})])
\]
\[
= \vec{r}_t + \gamma_t \vec{z}_t (\alpha \vec{\phi}(i_{t+1})^T \vec{r}_t - \vec{\phi}(i_t)^T \vec{r}_t + g(i_t, i_{t+1}))
\]
\[
= \vec{r}_t + \gamma_t \vec{d}_t \vec{z}_t
\]
Lemma 2 Let:
\[ A := E_0[A(X_t)] \]
where \( E_0[\cdot] \) is the expectation with respect to the stationary distribution.
We claim that:
\[ A = \Phi^T D (M - I) \Phi \]
where:
\[ M := (1 - \lambda) \sum_{m=0}^{\infty} \lambda^m (\alpha P)^{m+1} \]
and
\[ D := \begin{pmatrix} \pi[1] \\ \vdots \\ \pi[n] \end{pmatrix} \]
We define the \textit{weighted quadratic norm} as follow:

\[
\|\vec{J}\|_D^2 := \vec{J}^T D \vec{J}
\]

where

\[
D := \begin{pmatrix}
\bar{\pi}[1] \\
\vdots \\
\bar{\pi}[n]
\end{pmatrix}
\]
3  Sparse Distributed Memories (SDM)


Elements of an SDM architecture:

1. Start with some *active locations* \( H := \{ \vec{h}_k \} \) uniformly distributed in the space. Active locations are points in the state space with an associated *weight* \( w_k \).

2. To predict the value of \( \vec{x} \in S \), we use the formula:

\[
\tilde{J}(\vec{x}) = \frac{\sum_{h_k \in H} \mu(\vec{h}_k, \vec{x}) w_k}{\sum_{h_k \in H} \mu(\vec{h}_k, \vec{x})}
\]

Where \( \mu \) is a similarity measure. An example of similarity measure: *symmetric triangular functions*

\[
\begin{align*}
\mu(\vec{h}, \vec{x}) &:= \min_{i=1,\ldots,n} \mu_i(\vec{h}, \vec{x}) \\
\mu_i(\vec{h}, \vec{x}) &:= \begin{cases} 
1 - \frac{|x_i - h_i|}{\beta_i} & \text{if } |x_i - h_i| \leq \beta_i \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

3. We also periodically rearrange the distribution of the active locations during the execution of the algorithm.