# Presentation_Mar30_MTanneau

April 1, 2017

## 1 Expected behaviour of TD(0) in the off-policy prediction

### 1.1 Aka: what makes the deadly triad so deadly

Hint: it's all about linear algebra :)

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        from scipy.optimize import fsolve
        import time as time
```

## 2 1. Off-policy TD(0) with Linear VFA

### 2.1 1.1 Using State Value functions

We consider the off-policy case, extended to linear function approximation.

Recall the semi-gradient TD update :

$$\begin{aligned}
\delta_t &= R_{t+1} + \gamma \hat{v}(S_{t+1}, \theta_t) - \hat{v}(S_t, \theta_t) \\
\theta_{t+1} &= \theta_t + \alpha \rho_t \delta_t \nabla \hat{v}(S_t, \theta_t)
\end{aligned}$$

The expected behaviour then writes:

$$\begin{aligned}
E[\theta_{t+1} | \theta_t] &= \theta_t + \alpha E[\rho_t \delta_t \nabla \hat{v}(S_t, \theta_t) | \theta_t] \\
&= \theta_t + \alpha E\left[\rho_t \big(R_{t+1}\phi_t - \phi_t(\phi_t - \gamma\phi_{t+1})^T \theta_t\big) | \theta_t\right] \\
&= \theta_t + \alpha E\left[\rho_t R_{t+1} \phi_t\right] - \alpha E\left[\rho_t \phi_t (\phi_t - \gamma \phi_{t+1})^T\right] \theta_t
\end{aligned}$$

Now, defining $A, b$ as follow:

$$\begin{aligned}
A &= E\left[\rho_t \phi_t (\phi_t - \gamma \phi_{t+1})^T\right] \\
b &= E\left[\rho_t R_{t+1} \phi_t\right]
\end{aligned}$$

If we expand these expectations, we can write them under the following matrix form. Actually, these equations are at the beginning of the 2015 paper on emphatic TD (off-policy section).

$$
\begin{aligned}
A &= \Phi^T D_\mu (I - \gamma P_\pi)\Phi \\
b &= \Phi^T D_\mu r_\pi
\end{aligned}
$$

Let's sum up what we know so far :

- On-policy. $TD(0)$ converges to the TD fixed point, all is fine

$$
\begin{aligned}
A &= \Phi^T D_\pi (I - \gamma P_\pi)\Phi \\
b &= \Phi^T D_\pi r_\pi
\end{aligned}
$$

- Off policy, no IS. Not having any IS means we're actually learning the value function of behaviour policy, not that of the target.

$$
\begin{aligned}
A &= \Phi^T D_\mu (I - \gamma P_\mu)\Phi \\
b &= \Phi^T D_\mu r_\mu
\end{aligned}
$$

- Off-policy, IS. Introducing IS actually corrects part of the equation, but not all of it. In expectation, everything goes as if the transitions occured according to the target policy, but states were sampled according to another distribution (that of the behaviour policy). This corresponds to the case considered in section IX of [Tsitsiklis, Van Roy 1997], where they show that TD can diverge if states are not sampled according to the right distribution.

$$
\begin{aligned}
A &= \Phi^T D_\mu (I - \gamma P_\pi)\Phi \\
b &= \Phi^T D_\mu r_\pi
\end{aligned}
$$

In the on-policy case, the convergence prrof relied on the fact that states are sampled from the stationnary distribution $d_\pi$ (see 9.4 in Sutton's book). It is no longer the case in an off-policy. As a consequence, $A$ is no longer guaranteed to be positive definite, and TD may diverge.

Indeeed, assuming $A\theta_{TD} = b$, recall the error behaves as $\theta_{t+1} - \theta_{TD} = \epsilon_{t+1} = (I - \alpha A)\epsilon_t$. Thus, if $A$ has at least one negative eigenvalue, then TD is bound to diverge.

#### 2.1.1 1.1.1 An example

Let's illustrate this in Baird's famous counterexample (picture taken from Sutton's book):

```
In [3]: #number of states
        n=7

        #discount factor
        gamma=0.99
        r_pi=np.zeros(n)

        #define stationnary distribution under pi
        pi=np.zeros(n)
        pi[n-1]=1.0
        D_pi=np.diag(pi)

        #define stationnary distribution under mu
        mu=1.0/n * np.ones(n)
        D_mu =np.diag(mu)

        #Compute transition probability matrices for target and behaviour policy
        P_pi=np.zeros((7,7))
        P_pi[:,6]=1

        P_mu=np.zeros((7,7))
        P_mu[:,:]=1.0/7.0


        #Define features matrix
        Phi = np.zeros((7,8))
        for i in range(6):
            Phi[i,i]=2
            Phi[i,7]=1
        Phi[6,6]=1
        Phi[6,7]=2


        I=np.eye(7)

        #Compute matrix A
        A=Phi.T.dot(D_mu).dot(I-gamma*P_pi).dot(Phi)

        #by computing the column sums of the key matrix, we see whether it is posit
        print 'Column sums of key matrix'
        print np.ones(n).T.dot(D_mu).dot(I-gamma*P_pi)

        #We TD is bound to diverge (unless the initial theta is really-well chosen)
        #if A has at least one negative eigenvalue
        [eA,vA]=np.linalg.eig(A)
        print '\nEigenvalues of A'
        print eA

Column sums of key matrix
[ 0.14285714  0.14285714  0.14285714  0.14285714  0.14285714  0.14285714
 -0.84714286]

Eigenvalues of A
[  5.71428571e-01  -2.39250464e-01  -2.21781072e-02  -7.94718827e-17
   5.71428571e-01   5.71428571e-01   5.71428571e-01   5.71428571e-01]
```

### 2.1.2 1.1.2 Are we doomed, though ?

Not necessarily, as shows the last line of the proof (still 9.4 in Sutton's book). We need the column sum of the key matrix to be positive. This ought to be the case, assuming $\mu$ is close enough to $\pi$.

The column sums of the key matrix write as :

$$\begin{aligned} 1^T D_\mu (I - \gamma P_\pi) &= d_\mu^T (I - \gamma P_\pi) \\ &= d_\mu^T - \gamma d_\mu^T P_\pi \end{aligned}$$

We now assume that, whether we consider the behaviour policy or the target policy, the induced markov chain (with transition matrix $P_\pi$ or $P_\mu$) is irreductible (otherwise, we may restrict to a smaller set of states) and ergodic (otherwise, ie all states are visited infinitely many times when time goes to infinity). Note this assumption is always met when we have exploring starts.

Therefore, the associated stationary distribution is unique.

As $\mu \to \pi$, we have $P_\pi \to P_\mu$. We also know that $d_\pi$ (resp $d_\mu$) is a left-eigenvector of $P_\pi$ (resp $P_\mu$) associated to left-eigenvalue 1. We just have to show that $d_\mu$ is close to $d_\pi$ when $\mu$ is close to $\pi$

Let $\mu_k$ be a sequence of policies that converge to $\pi$. For simplicity, denote $d_k = d_{\mu_k}$, $P_k = P_{\mu_k}$ and $\epsilon_k = d_k - d_\pi$. We have :

$$\begin{aligned} \epsilon_k^T &= d_k^T - d_\pi^T \\ &= d_k^T P_k - d_\pi^T P_\pi \\ &= (d_k - d_\pi)^T P_k + d_\pi^T (P_k - P_\pi) \end{aligned}$$

The sequence $\epsilon_k$ is bounded, so we can extract a convergent subsequence. For simplicity, we assume this is already the case, ie :

$$\epsilon := \lim_{k \to +\infty} \epsilon_k$$

which we re-write with respect to $d_k$ :

$$d := \lim_{k \to +\infty} d_k$$

As $k$ goes to infinity, we have :

$$(d - d_\pi)^T (I - P_\pi) = 0$$

And since $d_\pi^T = d_\pi^T P_\pi$, we have $d^T = d^T P_\pi$. Therefore :

$$\lim_{\mu \to \pi} d_\mu = d_\pi$$

Finally, when $\mu$ is close to $\pi$, $d_\mu^T - \gamma d_\mu^T P_\pi$ is close to $d_\pi^T - \gamma d_\pi^T P_\pi$ which coefficients are all positive. It immediately follows that the key matrix is positive definite if the behaviour policy is close enough to the target policy.

**Conclusion** Just like there always exist a behaviour policy that will make off-policy TD with linear approximation diverge, there always exists at least one behaviour policy for which TD(0) is guaranteed to converge.

### 2.1.3 1.2.3 Why is TD guaranted to converge in the tabular case ?

I believe the answer is in the projector operator $\Pi$. Recall that :

$$\Pi = \Phi(\Phi^T D \Phi)^{-1} \Phi^T D$$

The convergence of TD(0) is based on $\Pi \cdot TD$ being a contraction mapping, which is the case when $D$ is the stationnary distribution of the Markov Chain.

The problem with off-policy is that, since $D$ may not be the stationnary distribution of the Markov Chain, then the TD operator may not be a contraction with respect to $D$ (see [Tsitsiklis 97]). However, in the tabular case, $\Phi$ is the identity matrix, which yields immediately $\Pi = I$, no matter which distribution we consider. As a consequence, $\Pi \cdot TD$ is a contraction (because TD is), regardless of the behaviour policy (we still require coverage).

## 2.2 1.2 Using State-Action functions (aka, Q-learning)

We now consider state-action values. The corresponding algorithm is Expected Sarsa, aka Q-learning when the policy is greedy. Recall the semi-gradient algorithm :

$$\begin{aligned}\delta_t &= R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) \hat{q}(S_{t+1}, a, \theta_t) - \hat{q}(S_t, a, \theta_t) \\ \theta_{t+1} &= \theta_t + \alpha \delta_t \nabla \hat{q}(S_t, A_t, \theta_t)\end{aligned}$$

We use linear function approximation, ie : $\hat{q}(s,a) = \theta^T \phi(s,a)$. We then have :

$$\theta_{t+1} = \theta_t + \alpha \left[ R_{t+1}\Phi(S_t, A_t) - \Phi(S_t, A_t)\big(\Phi(S_t, A_t) - \gamma \sum_a \pi(a|S_{t+1})\Phi(S_{t+1}, a)\big)^T \theta_t \right]$$

In expectation, updates thus write as (to ease the reading, I voluntarily skip a few lines of calculus here):

$$\begin{aligned}E[\theta_{t+1}|\theta_t] &= \theta_t + \alpha E\left[R_{t+1}\Phi(S_t, A_t)\right] - E\left[\Phi(S_t, A_t)\big(\Phi(S_t, A_t) - \gamma \sum_a \pi(a|S_{t+1})\Phi(S_{t+1}, a)\big)^T\right] \theta_t \\ &= \theta_t + \alpha(b - A\theta_t)\end{aligned}$$

where (all matrices and vectors are now indexed by state-action pairs) :

$$\begin{aligned}A &= \Phi^T D_\mu (I - \gamma P_\pi) \Phi \\ b &= \Phi^T D_\mu r\end{aligned}$$

The main result here, is that these expression are obtained without using importance sampling. This is due to the fact that Q-learning samples directly from state-action together. Therefore, the transition to the next state $S_{t+1}$ does not depend on the behaviour policy. Besides, the expectation in the target is taken explicitely with respect to $\pi$. Finally, notice the reward vector $r$ also does not depend on the behaviour policy (nor the target).

We have the exact same structure as we had before. Ergo, the conclusions are the same.

## 3 2. What's next ?

We've seen that introducing importance sampling partially solves the problem of off-policy. However, it does not correct for the distribution of states. Therefore, we would like to introduce a second set of importance ratios, $\sigma(s) = \dfrac{d_\pi(s)}{d_\mu(s)}$, which depend only on the states (not the actions).

The update rule would then be, if we use state-value function :

$$\theta_{t+1} = \theta_t + \alpha \rho_t \sigma_t \delta_t \nabla \hat{v}(S_t, \theta_t)$$

And for state action values :

$$\theta_{t+1} = \theta_t + \alpha \sigma_t \delta_t \nabla \hat{q}(S_t, A_t, \theta_t)$$

where $\sigma_t = \dfrac{d_\pi(S_t)}{d_\mu(S_t)}$. Using this trick would yield (in expectation) the correct expressions :

$$\begin{aligned} A &= \Phi^T D_\pi (I - \gamma P_\pi) \Phi \\ b &= \Phi^T D_\pi r \end{aligned}$$

Of course, we have no way of computing neither $d_\pi(s)$ nor $d_\mu(s)$ beforehand…

One way to go is developped in [Precup, Sutton, and Dasgupta 2001] (http://www.cs.mcgill.ca/~dprecup/publications/PSD-01.pdf). However, this approach is based on multiplying the importance ratios over an episode, which leads to very high variance.

State weighting as is done in Emphatic TD might be another way to tackle the issue.