COMPARING LINEAR TD TO SGTD ON HUS GAME

Semi-gradient TD(0)

Semi-gradient TD(0) for estimating $\hat{v} \approx v_\pi$

Input: the policy π to be evaluated
Input: a differentiable function $\hat{v} : \mathcal{S}^+ \times \mathbb{R}^n \to \mathbb{R}$ such that $\hat{v}(\text{terminal},\cdot) = 0$

Initialize value-function weights $\boldsymbol{\theta}$ arbitrarily (e.g., $\boldsymbol{\theta} = \mathbf{0}$)
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        Choose $A \sim \pi(\cdot|S)$
        Take action A, observe R, S'
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\big[R + \gamma\hat{v}(S',\boldsymbol{\theta}) - \hat{v}(S,\boldsymbol{\theta})\big]\nabla\hat{v}(S,\boldsymbol{\theta})$
        $S \leftarrow S'$
    until S' is terminal

Linear TD

- Update Rule Differs:

$$\theta_{k+1} = \theta_k + \alpha_k \delta_k \phi_k - \alpha\gamma\phi'_k(\phi_k^\top w_k)$$

$$w_{k+1} = w_k + \beta_k(\delta_k - \phi_k^\top w_k)\phi_k$$

$$\delta_k = r_k + \gamma\theta_k^\top \phi'_k - \theta_k^\top \phi_k$$

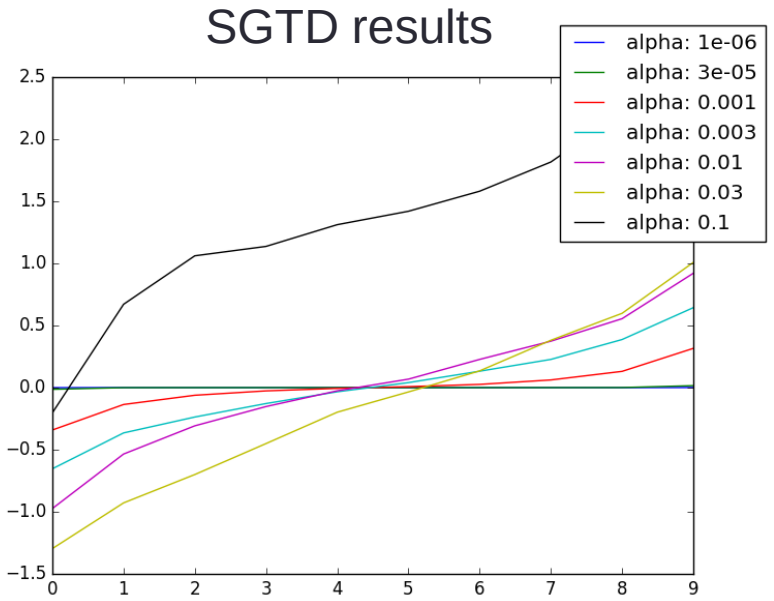
The difference between LTD and SGTD is the addition of an adjustment term during the weight updates.

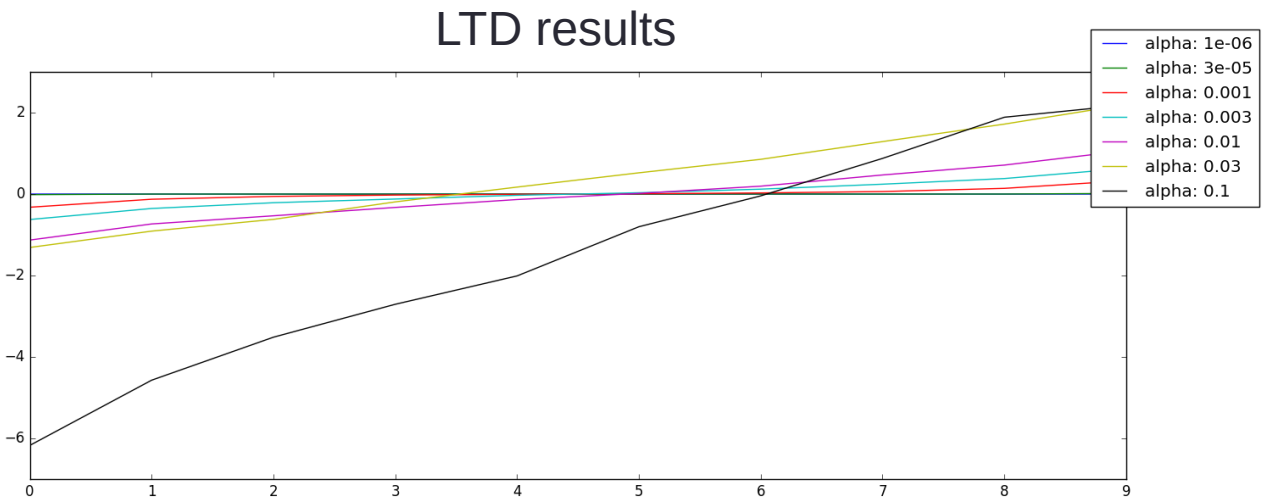
Methodology

- 1000 states random walk:
  - Used one hot encoding and grouped the states into 10 different groups
  - Used a linear value function V(s,w) = w^T*f(s) where f(s) is a vector of 10 entries, 1 for each group
  - As a consequence of using this parameterization, the weight associated with each feature represents the expected return for this group of states.
  - Ran both algorithms over 1000 episodes using various learning rates.

1000 states random walk results

Both methods perform well on the 1000 states random walk problem when a good learning rate is chosen.





Hus

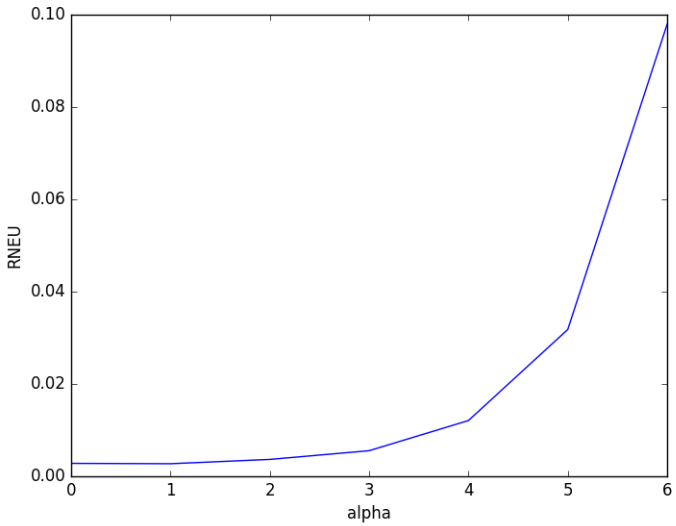
- Rules : http://mancala.wikia.com/wiki/Hus
- Each state is represented by a feature vector of 17 entries. One entry represents the sum of the seeds on the player's side of the board. The other 16 are binary variables representing whether or not the opponent can capture the seeds found in pits 1-16.
- The differentiable function is a linear combination of those features
- Ran both algorithms over 1000 episodes to learn valid weights and then again over 1000 episodes to estimate the RNEU. Did this 5 times for each learning rate value and averaged the results.
- I used a policy that picks a valid move randomly.

Hus Results

I used a reward of 1 when a player won the game, 0 otherwise

The estimate for the residual error is much lower when we use LTD than when we use SGTD, not matter the value of the learning rate. This seems to suggest that in the case of HUS, LTD is more effective at estimating the true value of every state than SGTD is.

SGTD results



LTD results