Temporal Difference Methods are Not Gradient Descent

COMP 767

Matthew Smith

Summary of: *Temporal Difference Methods and Markov Models* by Etienne Barnard

Overview

Problem Setting: Sequential Prediction

- Tabular Markov Process, in which the state s_t of the system is observed at each time step t.
- After m_σ such steps, a terminal state z_σ is reached. (σ indexes the trajectory)
- The goal is to predict the (binary) value of the terminal state from the previous states: $P(z_\sigma = Z | s_t = s)$.

Temporal Difference Methods and Markov Models

- We can express the probability of terminating in state Z, given a current state, s as the empirical ratio:

$$w_s = n_Z s / n_s$$

where n_s is the number of trajectories in which state s is reached, and $n_Z s$ is the number of such trajectories that end in Z

Temporal Difference Methods and Markov Models

- We can express the probability of terminating in state Z, given a current state, s as the empirical ratio:

  $$w_s = n_Z s / n_s$$

  where n_s is the number of trajectories in which state s is reached, and $n_Z s$ is the number of such trajectories that end in Z
- However, this is not data efficient - we don't leverage the Markov structure.

TD Methods and Markov Models

- Instead, represent w_s as the sum:

$$w_s = h_s + \sum_{s'} P(s'|s) w_{s'}$$

where h_s is the probability of terminating in Z directly from state s, and P denotes the transition probability.

TD Methods and Markov Models

- Instead, represent w_s as the sum:

$$w_s = h_s + \sum_{s'} P(s'|s) w_{s'}$$

  where h_s is the probability of terminating in Z directly from state s, and P denotes the transition probability.
- Since h and P are also probabilities, we can estimate them using empirical frequencies:

$$h_s = m_{sZ}/n_s \qquad P(s'|s) = m_{ss'}/n_s$$

TD Methods and Markov Models

- multiply through by $n_{s'}$:

$$\sum_{s'} m_{ss'} w_s - n_{s'} w_s' + m_{sZ} = 0$$

TD Methods and Markov Models

- multiply through by $n_{s'}$:

$$\sum_{s'} m_{ss'} w_s - n_{s'} w_s' + m_{sZ} = 0$$

- Or in matrix form:

$$(\mathbf{M} - \mathbf{N})\mathbf{w} + \mathbf{m} = \mathbf{0}$$

TD Methods and Markov Models

- multiply through by $n_{s'}$:

$$\sum_{s'} m_{ss'} w_s - n_{s'} w'_s + m_{sZ} = 0$$

- Or in matrix form:

$$(\mathbf{M} - \mathbf{N})\mathbf{w} + \mathbf{m} = \mathbf{0}$$

- Which can be solved iteratively by:

$$\mathbf{w} \rightarrow \mathbf{w} + \alpha \left[(\mathbf{M} - \mathbf{N})\mathbf{w} + \mathbf{m}\right]$$

TD Methods and Markov Models

- Here we apply the one-step contributions to M and N at every step.

TD Methods and Markov Models

- Here we apply the one-step contributions to M and N at every step.
- Assuming $\mathbf{x}_t$ represents the one-hot state encoding vector, we can express this as:

$$\mathbf{w} \to \mathbf{w} + \alpha \left[(\mathbf{x}_t \mathbf{x}_{t+1}^\top - \mathbf{x}_t \mathbf{x}_t^\top)\mathbf{w} + \mathbf{x}_t \delta_{s_{t+1} Z} \right] \tag{1}$$

$$= \mathbf{w} + \alpha \left[(\mathbf{x}_{t+1}^\top \mathbf{w} - \mathbf{x}_t^\top \mathbf{w} + \delta_{s_{t+1} Z})\mathbf{x}_t \right] \tag{2}$$

which looks like TD.

Value Accumulation

- Similarly, we can express the value as the sum:

$$v_s = r_s + \sum_{s'} P(s'|s) v_{s'}$$

.

Value Accumulation

- Similarly, we can express the value as the sum:

$$v_s = r_s + \sum_{s'} P(s'|s) v_{s'}$$

.

- Now r_s is no longer a probability, so this gives us:

$$(\mathbf{M} - \mathbf{N})\mathbf{v} + \mathbf{N}\mathbf{r} = \mathbf{0}$$

Value Accumulation

- Similarly, we can express the value as the sum:

$$v_s = r_s + \sum_{s'} P(s'|s) v_{s'}$$

.

- Now r_s is no longer a probability, so this gives us:

$$(\mathbf{M} - \mathbf{N})\mathbf{v} + \mathbf{N}\mathbf{r} = \mathbf{0}$$

- And using the same approximations as before gives the TD update that we all know:

$$\begin{aligned} \mathbf{v} \to \mathbf{v} + \alpha \left[(\mathbf{x}_t \mathbf{x}_{t+1}^\top - \mathbf{x}_t \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t \mathbf{x}_t^\top \mathbf{r} \right] && (3) \\ = \mathbf{v} + \alpha \left[(\mathbf{x}_{t+1}^\top \mathbf{v} - \mathbf{x}_t^\top \mathbf{v} + \mathbf{x}_t^\top \mathbf{r})\mathbf{x}_t \right] && (4) \end{aligned}$$

TD is Not Gradient Descent

- If TD were gradient descent, we would have:

$$\nabla J(v) = (\mathbf{x}_t \mathbf{x}_{t+1}^\top - \mathbf{x}_t \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t \mathbf{x}_t^\top \mathbf{r}$$

TD is Not Gradient Descent

- If TD were gradient descent, we would have:

$$\nabla J(v) = (\mathbf{x}_t \mathbf{x}_{t+1}^\top - \mathbf{x}_t \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t \mathbf{x}_t^\top \mathbf{r}$$

- However, we then have:

$$\frac{\partial J}{\partial v_i} = x_{ti}\left[(\mathbf{x}_{t+1}^\top - \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t^\top \mathbf{r}\right] \tag{5}$$

and

$$\frac{\partial J}{\partial v_j} = x_{tj}\left[(\mathbf{x}_{t+1}^\top - \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t^\top \mathbf{r}\right] \tag{6}$$

TD is Not Gradient Descent

- We have:

$$\frac{\partial J}{\partial v_i} = x_{ti}\left[(\mathbf{x}_{t+1}^\top - \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t^\top \mathbf{r}\right] \text{ and}$$
$$\frac{\partial J}{\partial v_j} = x_{tj}\left[(\mathbf{x}_{t+1}^\top - \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t^\top \mathbf{r}\right]$$

TD is Not Gradient Descent

- We have:

$$\frac{\partial J}{\partial v_i} = x_{ti}\left[(\mathbf{x}_{t+1}^\top - \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t^\top \mathbf{r}\right] \text{ and}$$
$$\frac{\partial J}{\partial v_j} = x_{tj}\left[(\mathbf{x}_{t+1}^\top - \mathbf{x}_t^\top)\mathbf{v} + \mathbf{x}_t^\top \mathbf{r}\right]$$

- But this means that:

$$\frac{\partial^2 J}{\partial v_i \partial v_j} = x_{ti}(x_{(t+1)j} - x_{tj})$$

and

$$\frac{\partial^2 J}{\partial v_j \partial v_i} = x_{tj}(x_{(t+1)i} - x_{ti})$$

TD is Not Gradient Descent

- Since:

$$\frac{\partial^2 J}{\partial v_i \partial v_j} \neq \frac{\partial^2 J}{\partial v_j \partial v_i}$$

TD updates do not come from the derivative of a differentiable function.

TD is Not Gradient Descent

Note that this slide does not actually prove anything, but in a two parameter, nonabsorbing, 4 state environment, with 0 reward everywhere, we see that TD does not follow the gradient of the MSVE function.

(Yellow is SGD, blue is TD, red is the gradient, green is MSVE)