Linear Least-Squares algorithms for temporal difference learning

by Mohammad Pezeshki

Least-Squares algorithm

Batch learning algorithm

Learning from training data

Least-Squares algorithm

Batch learning algorithm

Learning from training data

We approximate the value function

using the experience set

by minimizing sum-squared error.

Least-Squares algorithm

Batch learning algorithm

Learning from training data

We approximate the value function $v_\pi(s) \approx \hat{v}(s, w)$

using the experience set

$$D = \{(s_1, v_\pi(s_1)), \\ (s_2, v_\pi(s_2)), \\ \ldots \\ (s_T, v_\pi(s_T))\}$$

by minimizing sum-squared error.

Linear Least-Squares algorithms for temporal difference learning

by Mohammad Pezeshki

Least-Squares algorithm

Batch learning algorithm

Learning from training data

We approximate the value function $v_\pi(s) \approx \hat{v}(s, w)$

using the experience set $D = \{(s_1, v_\pi(s_1)), (s_2, v_\pi(s_2)), \dots (s_T, v_\pi(s_T))\}$

by minimizing sum-squared error.

$$\mathrm{LS}(w) = \sum_{t=1}^{T} \big(v_\pi(s_t) - \hat{v}(s_t, w)\big)^2$$

Linear Least-Squares algorithms for temporal difference learning

Least-Squares

$$\mathrm{LS}(w) = \sum_{t=1}^{T} \left(v_\pi(s_t) - \hat{v}(s_t, w)\right)^2$$

How to solve?

Linear Least-Squares algorithms for temporal difference learning

Least-Squares

$$\mathrm{LS}(w) = \sum_{t=1}^{T} \left(v_\pi(s_t) - \hat{v}(s_t, w)\right)^2$$

How to solve?

Stochastic Gradient Descent

$$\Delta w = \alpha\left(v_\pi - \hat{v}(s, w)\right)\nabla_w \hat{v}(s, w)$$

Linear Least-Squares algorithms for temporal difference learning

Least-Squares

$$\mathrm{LS}(w) = \sum_{t=1}^{T} \big(v_\pi(s_t) - \hat{v}(s_t, w)\big)^2$$

How to solve?

Stochastic Gradient Descent

$$\Delta w = \alpha\big(v_\pi - \hat{v}(s, w)\big)\nabla_w \hat{v}(s, w)$$

Converges to

$$w_\pi = \mathrm{argmin}_w LS(w)$$

Linear Least-Squares algorithms for temporal difference learning

$$\Delta w = \alpha\Big(v_\pi - \hat{v}(s, w)\Big)\nabla_w \hat{v}(s, w)$$

Linear Least-Squares algorithms for temporal difference learning

$$\Delta w = \alpha\big(v_\pi - \hat{v}(s,w)\big)\nabla_w \hat{v}(s,w)$$

$$\hat{v}(s,w) = \phi(s)^T w$$

in the case of **linear** function approximation

Linear Least-Squares algorithms for temporal difference learning

$$\Delta w = \alpha\big(v_\pi - \hat{v}(s,w)\big)\nabla_w \hat{v}(s,w)$$

$$\hat{v}(s,w) = \phi(s)^T w$$

in the case of **linear** function approximation

$$\Delta w = \alpha \sum_{t=1}^{T} \phi(s_t)\big(v_\pi(s_t) - \phi(s_t)^T w\big)$$

Linear Least-Squares algorithms for temporal difference learning

$$\Delta w = \alpha\big(v_\pi - \hat{v}(s,w)\big)\nabla_w \hat{v}(s,w)$$

$$\hat{v}(s,w) = \phi(s)^T w$$ in the case of **linear** function approximation

$$\Delta w = \alpha \sum_{t=1}^{T} \phi(s_t)\big(v_\pi(s_t) - \phi(s_t)^T w\big)$$

When converges, then $E_D[\Delta w] = 0$

Linear Least-Squares algorithms for temporal difference learning

$$\Delta w = \alpha \big(v_\pi - \hat{v}(s,w)\big) \nabla_w \hat{v}(s,w)$$

$$\hat{v}(s,w) = \phi(s)^T w$$ in the case of **linear** function approximation

$$\Delta w = \alpha \sum_{t=1}^{T} \phi(s_t) \big(v_\pi(s_t) - \phi(s_t)^T w\big)$$

When converges, then $E_D[\Delta w] = 0$

$$\alpha \sum_{t=1}^{T} \phi(s_t) \big(v_\pi(s_t) - \phi(s_t)^T w\big) = 0$$

Linear Least-Squares algorithms for temporal difference learning

$$\Delta w = \alpha\big(v_\pi - \hat{v}(s,w)\big)\nabla_w \hat{v}(s,w)$$

$$\hat{v}(s,w) = \phi(s)^T w$$ in the case of **linear** function approximation

$$\Delta w = \alpha \sum_{t=1}^{T} \phi(s_t)\big(v_\pi(s_t) - \phi(s_t)^T w\big)$$

When converges, then $E_D[\Delta w] = 0$

$$\alpha \sum_{t=1}^{T} \phi(s_t)\big(v_\pi(s_t) - \phi(s_t)^T w\big) = 0$$

$$\sum_{t=1}^{T} \phi(s_t) v_\pi(s_t) = \sum_{t=1}^{T} \phi(s_t)\phi(s_t)^T w$$

Linear Least-Squares algorithms for temporal difference learning

$$\Delta w = \alpha\big(v_\pi - \hat{v}(s,w)\big)\nabla_w \hat{v}(s,w)$$

$$\hat{v}(s,w) = \phi(s)^T w$$

in the case of **linear** function approximation

$$\Delta w = \alpha \sum_{t=1}^{T} \phi(s_t)\big(v_\pi(s_t) - \phi(s_t)^T w\big)$$

When converges, then $E_D[\Delta w] = 0$

$$\alpha \sum_{t=1}^{T} \phi(s_t)\big(v_\pi(s_t) - \phi(s_t)^T w\big) = 0$$

$$\sum_{t=1}^{T} \phi(s_t) v_\pi(s_t) = \sum_{t=1}^{T} \phi(s_t)\phi(s_t)^T w$$

$$w = \left(\sum_{t=1}^{T} \phi(s_t)\phi(s_t)^T w\right)^{-1} \sum_{t=1}^{T} \phi(s_t) v_\pi(s_t)$$

Linear Least-Squares algorithms for temporal difference learning

The solution for $\mathrm{LS}(w)$

$$w = \left(\sum_{t=1}^{T} \phi(s_t)\phi(s_t)^T w\right)^{-1} \sum_{t=1}^{T} \phi(s_t) v_\pi(s_t)$$

Linear Least-Squares algorithms for temporal difference learning

The solution for $\mathrm{LS}(w)$

$$w = \left(\sum_{t=1}^{T} \phi(s_t)\phi(s_t)^T w \right)^{-1} \sum_{t=1}^{T} \phi(s_t) \underbrace{v_\pi(s_t)}_{\text{Unknown}}$$

$v_\pi(s_t)$

Linear Least-Squares algorithms for temporal difference learning

The solution for $\mathrm{LS}(w)$

$$w = \left(\sum_{t=1}^{T} \phi(s_t)\phi(s_t)^T w \right)^{-1} \sum_{t=1}^{T} \phi(s_t) \underbrace{v_\pi(s_t)}_{\text{Unknown}}$$

$v_\pi(s_t) \rightarrow G_t \rightarrow$ **LSMC**

$v_\pi(s_t) \rightarrow R_{t+1} + \gamma \hat{v}(S_{t+1}, w) \rightarrow$ **LSTD**

Linear Least-Squares algorithms for temporal difference learning

The solution for $\mathrm{LS}(w)$

$$w = \left(\sum_{t=1}^{T} \phi(s_t)\phi(s_t)^T w \right)^{-1} \sum_{t=1}^{T} \phi(s_t) \underbrace{v_\pi(s_t)}_{\text{Unknown}}$$

$$v_\pi(s_t) \dashrightarrow G_t \dashrightarrow \textbf{LSMC}$$

$$v_\pi(s_t) \dashrightarrow R_{t+1} + \gamma \hat{v}(S_{t+1}, w) \dashrightarrow \textbf{LSTD}$$

$$w = \left(\sum_{t=1}^{T} \phi(s_t)\big(\phi(s_t) - \gamma\phi(s_{t+1})\big)^T \right)^{-1} \sum_{t=1}^{T} \phi(s_t) R_{t+1}$$