

Linear Least-Squares algorithms for temporal difference learning

by Mohammad Pezeshki

Least-Squares algorithm



Batch learning algorithm



Learning from training data

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

by minimizing sum-squared error. $(s_2, v_{\pi}(s_2)),$

...
 $(s_T, v_{\pi}(s_T)) \}$

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 $D = \{ (s_1, v_{\pi}(s_1)),$
by minimizing sum-squared error. $(s_2, v_{\pi}(s_2)),$
 \dots
 $(s_T, v_{\pi}(s_T)) \}$

↪

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

Linear Least-Squares algorithms for temporal difference learning


Least-Squares

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

How to solve?

Linear Least-Squares algorithms for temporal difference learning

Least-Squares



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



How to solve?




Stochastic Gradient Descent


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

Linear Least-Squares algorithms for temporal difference learning

Least-Squares



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



How to solve?



Stochastic Gradient Descent


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



Converges to


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

Linear Least-Squares algorithms for temporal difference learning

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

Linear Least-Squares algorithms for temporal difference learning

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



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

Linear Least-Squares algorithms for temporal difference learning

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

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

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

Linear Least-Squares algorithms for temporal difference learning

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

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

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

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

Linear Least-Squares algorithms for temporal difference learning

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

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

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

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

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

Linear Least-Squares algorithms for temporal difference learning

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

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

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

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

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

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

Linear Least-Squares algorithms for temporal difference learning

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

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

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

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

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

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

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

Linear Least-Squares algorithms for temporal difference learning

The solution for $LS(w)$

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

Linear Least-Squares algorithms for temporal difference learning

The solution for $LS(w)$

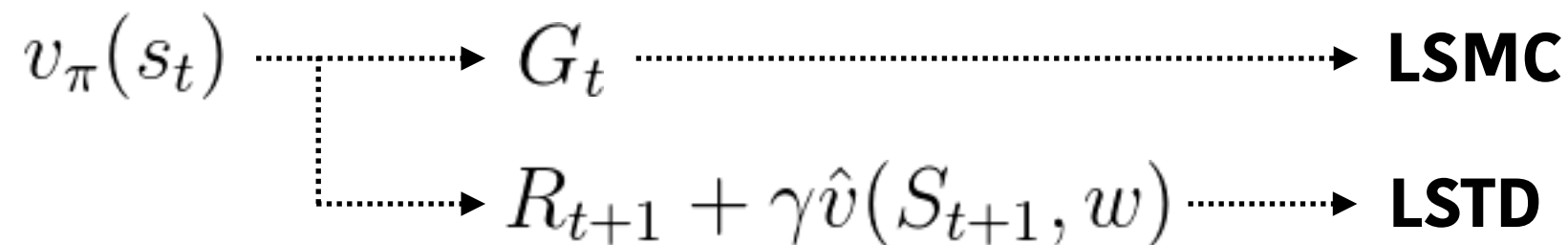
$$\hookrightarrow w = \left(\sum_{t=1}^T \phi(s_t) \phi(s_t)^T \right)^{-1} \sum_{t=1}^T \phi(s_t) \underbrace{v_{\pi}(s_t)}_{\text{Unknown}}$$

$v_{\pi}(s_t)$

Linear Least-Squares algorithms for temporal difference learning

The solution for $LS(w)$

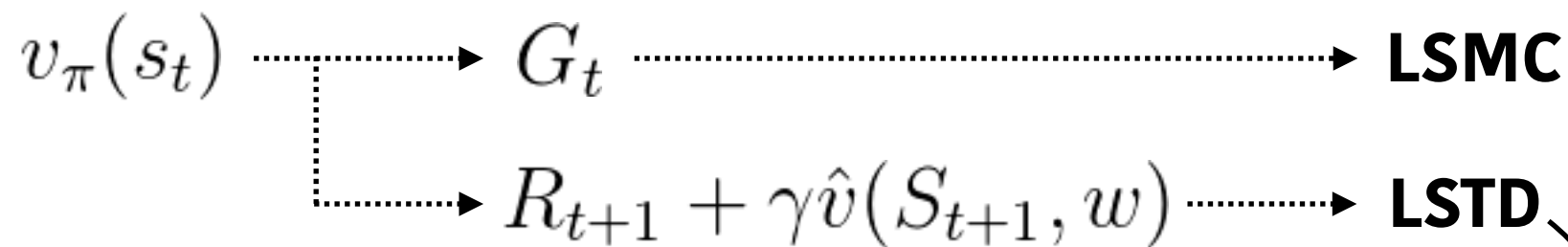
$$w = \left(\sum_{t=1}^T \phi(s_t) \phi(s_t)^T \right)^{-1} \sum_{t=1}^T \phi(s_t) \underbrace{v_{\pi}(s_t)}_{\text{Unknown}}$$



Linear Least-Squares algorithms for temporal difference learning

The solution for $LS(w)$

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



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