Reinforcement Learning from an optimization's persective

How I tried to bring optimization into RL

Mathieu Tanneau

Reinforcement Learning - Class project presentation

April 13, 2017

POLYTECHNIQUE MONTRÉAL
GERAD GROUP FOR RESEARCH IN DECISION ANALYSIS
CANADA EXCELLENCE RESEARCH CHAIR DATA SCIENCE FOR REAL-TIME DECISION-MAKING

One problem, two approaches

Policy evaluation, linear approximation $\longrightarrow$ solve $A\theta = b$

One problem, two approaches

Policy evaluation, linear approximation $\longrightarrow$ solve $A\theta = b$

Fixed point

$$\theta_{k+1} = \theta_k + \alpha(b - A\theta_k)$$

One problem, two approaches

Policy evaluation, linear approximation $\longrightarrow$ solve $A\theta = b$

Fixed point

$$\theta_{k+1} = \theta_k + \alpha(b - A\theta_k)$$

Least square

$$\min_\theta \frac{1}{2}\theta^T A^T A\theta - (A^T b)^T \theta$$

One problem, two approaches

Policy evaluation, linear approximation $\longrightarrow$ solve $A\theta = b$

Fixed point	Least square
$\theta_{k+1} = \theta_k + \alpha(b - A\theta_k)$	$\min_\theta \frac{1}{2}\theta^T A^T A\theta - (A^T b)^T \theta$

- Dynamic programming
- Convergence: $Sp(A)$
- Speed: linear

One problem, two approaches

Policy evaluation, linear approximation $\longrightarrow$ solve $A\theta = b$

Fixed point

$$\theta_{k+1} = \theta_k + \alpha(b - A\theta_k)$$

- Dynamic programming
- Convergence: $Sp(A)$
- Speed: linear

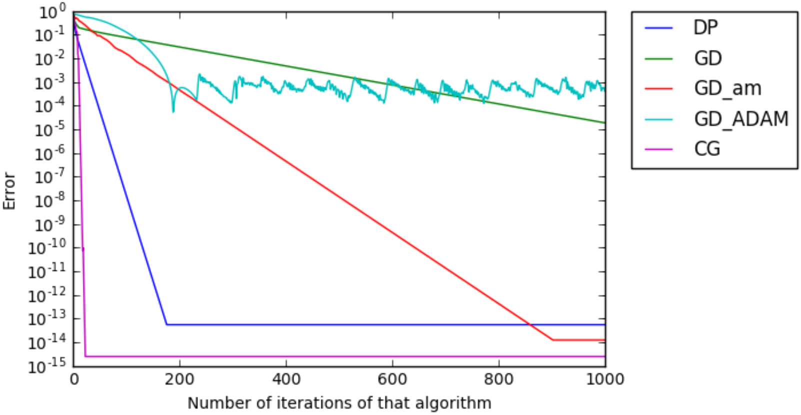
Least square

$$\min_\theta \frac{1}{2}\theta^T A^T A\theta - (A^T b)^T \theta$$

- Quadratic programming
- Convex
- Speed: superlinear

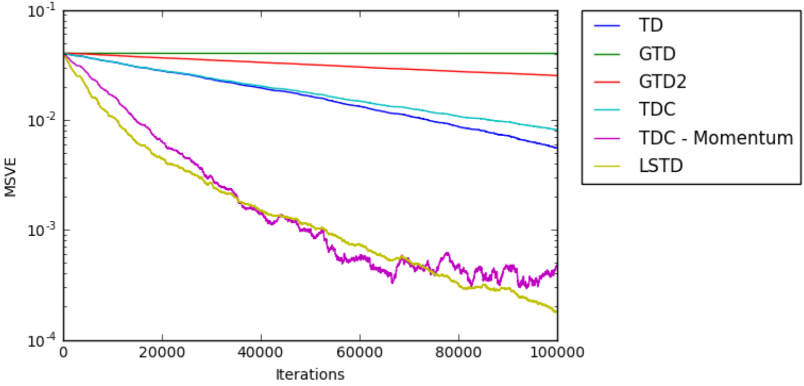
DP vs QP: tabular case

Random walk, 1000 states, $\gamma = 0.9$ (one run)



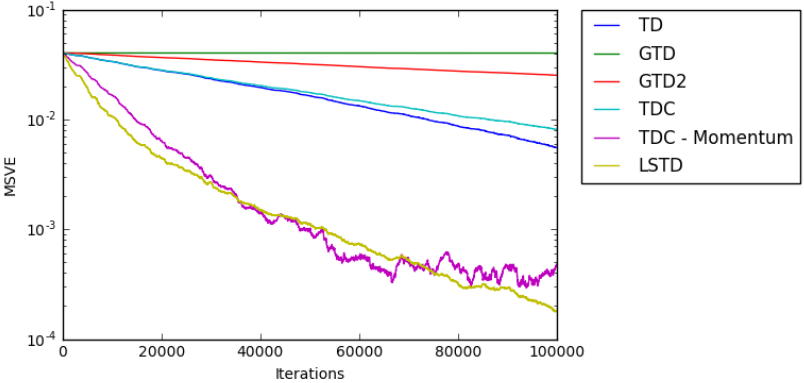
Extension: Gradient-based TD

Random walk, 1000 states, 100 features, $\gamma = 0.9$ (one run)



Extension: Gradient-based TD

Random walk, 1000 states, 100 features, $\gamma = 0.9$ (one run)



To do: adaptive learning rate (AdaGrad, RMSprop, ADAM...)

Limited-Memory LSTD

Sherman-Morrison formula, with $\hat{B}_t = \hat{A}_t^{-1}$:

$$\hat{B}_t \cdot \hat{b} = \hat{B}_{t-1}\hat{b} - \frac{\hat{B}_{t-1}\phi_t(\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\hat{b}}{1 + (\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\phi_t}$$

Limited-Memory LSTD

Sherman-Morrison formula, with $\hat{B}_t = \hat{A}_t^{-1}$:

$$\hat{B}_t \cdot \hat{b} = \hat{B}_{t-1}\hat{b} - \frac{\hat{B}_{t-1}\phi_t(\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\hat{b}}{1 + (\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\phi_t} = v - \frac{u^T v}{1 + u^T w} w$$

Limited-Memory LSTD

Sherman-Morrison formula, with $\hat{B}_t = \hat{A}_t^{-1}$:

$$\hat{B}_t \cdot \hat{b} = \hat{B}_{t-1}\hat{b} - \frac{\hat{B}_{t-1}\phi_t(\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\hat{b}}{1 + (\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\phi_t} = v - \frac{u^T v}{1 + u^T w} w$$

L-LSTD:

- Only remember last $m \ll n$ transitions, memory cost $O(mn)$

Limited-Memory LSTD

Sherman-Morrison formula, with $\hat{B}_t = \hat{A}_t^{-1}$:

$$\hat{B}_t \cdot \hat{b} = \hat{B}_{t-1}\hat{b} - \frac{\hat{B}_{t-1}\phi_t(\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\hat{b}}{1 + (\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\phi_t} = v - \frac{u^T v}{1 + u^T w} w$$

L-LSTD:

- Only remember last $m \ll n$ transitions, memory cost $O(mn)$
- $\hat{B}_t\phi_t$, $\hat{B}_t\hat{b}$ computed iteratively

Limited-Memory LSTD

Sherman-Morrison formula, with $\hat{B}_t = \hat{A}_t^{-1}$:

$$\hat{B}_t \cdot \hat{b} = \hat{B}_{t-1}\hat{b} - \frac{\hat{B}_{t-1}\phi_t(\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\hat{b}}{1 + (\phi_t - \gamma\phi_{t+1})^T \hat{B}_{t-1}\phi_t} = v - \frac{u^T v}{1 + u^T w} w$$

L-LSTD:

- Only remember last $m \ll n$ transitions, memory cost $O(mn)$
- $\hat{B}_t\phi_t$, $\hat{B}_t\hat{b}$ computed iteratively
- Cost $O(m^2 n)$ per update

Limited-Memory LSTD

In practice:

:) Indeed faster than LSTD for small m

:(Very unstable

Limited-Memory LSTD

In practice:

- :) Indeed faster than LSTD for small m
- :(Very unstable

Questions :

- *Convergence??*
- How often should we update θ?
- Which ϵ is best?
- Should we forget all information about b?