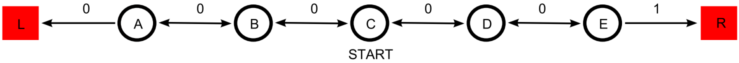
Least Squares Temporal Difference Learning

Yaroslav Rosokha

April 26, 2011

Random Walk

Consider a random walk example from chapter 6:



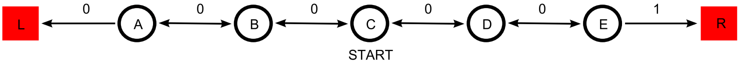
Our objective is to learn value function $V(s)$ for $s \in \{L, A, B, C, D, E, R\}$

- Matrix/vector representation of the initial state C, one step transition probabilities, and rewards:

$$\mathbf{s}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ .5 & 0 & .5 & 0 & 0 & 0 & 0 \\ 0 & .5 & 0 & .5 & 0 & 0 & 0 \\ 0 & 0 & .5 & 0 & .5 & 0 & 0 \\ 0 & 0 & 0 & .5 & 0 & .5 & 0 \\ 0 & 0 & 0 & 0 & .5 & 0 & .5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- What about two-step transition probabilities and rewards?

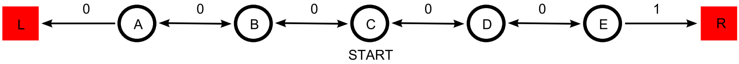
A Solution (when we know the model...)



Two-step transition probabilities and rewards:

$$\mathbf{P} \times \mathbf{P} = \mathbf{P}^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ .5 & 0.25 & 0 & .25 & 0 & 0 & 0 \\ .25 & 0 & .5 & 0 & .25 & 0 & 0 \\ 0 & .25 & 0 & .5 & 0 & .25 & 0 \\ 0 & 0 & .25 & 0 & .5 & 0 & .25 \\ 0 & 0 & 0 & .25 & 0 & .25 & .5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{R_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A Solution (when we know the model...)



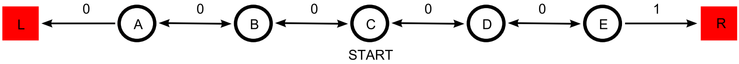
Two-step transition probabilities and rewards:

$$\mathbf{P} \times \mathbf{P} = \mathbf{P}^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ .5 & 0.25 & 0 & .25 & 0 & 0 & 0 \\ .25 & 0 & .5 & 0 & .25 & 0 & 0 \\ 0 & .25 & 0 & .5 & 0 & .25 & 0 \\ 0 & 0 & .25 & 0 & .5 & 0 & .25 \\ 0 & 0 & 0 & .25 & 0 & .25 & .5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{R_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- And so on...

$$\mathbf{P}^{\infty} = \begin{bmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .833 & 0 & 0 & 0 & 0 & 0 & .167 \\ .667 & 0 & 0 & 0 & 0 & 0 & .333 \\ .500 & 0 & 0 & 0 & 0 & 0 & .500 \\ .333 & 0 & 0 & 0 & 0 & 0 & .667 \\ .167 & 0 & 0 & 0 & 0 & 0 & .833 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \end{bmatrix} \mathbf{R}_{\infty} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A Solution (when we know the model...)



Two-step transition probabilities and rewards:

$$\mathbf{P} \times \mathbf{P} = \mathbf{P}^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ .5 & 0.25 & 0 & .25 & 0 & 0 & 0 \\ .25 & 0 & .5 & 0 & .25 & 0 & 0 \\ 0 & .25 & 0 & .5 & 0 & .25 & 0 \\ 0 & 0 & .25 & 0 & .5 & 0 & .25 \\ 0 & 0 & 0 & .25 & 0 & .25 & .5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{R_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

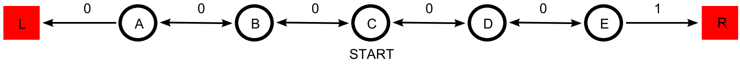
- And so on...

$$\mathbf{P}^{\infty} = \begin{bmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .833 & 0 & 0 & 0 & 0 & 0 & .167 \\ .667 & 0 & 0 & 0 & 0 & 0 & .333 \\ .500 & 0 & 0 & 0 & 0 & 0 & .500 \\ .333 & 0 & 0 & 0 & 0 & 0 & .667 \\ .167 & 0 & 0 & 0 & 0 & 0 & .833 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \end{bmatrix} \mathbf{R}_{\infty} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Which implies the following value function

$$V = \begin{bmatrix} 0 & .167 & .333 & .500 & .667 & .833 & 0 \end{bmatrix}$$

Notation



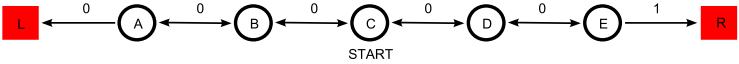
a *featurizer* $\phi(x)$ maps states to feature vectors

- For a single-state-per-feature representation, $\phi(x)$, C is represented by $\phi(C) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Example

Suppose L, C, R represeted by [1,0,0] [0,1,0], and [0,0,1] repspectively. What will be representation of states A and B if we interpolate linearly?

Notation



a *featurizer* $\phi(x)$ maps states to feature vectors

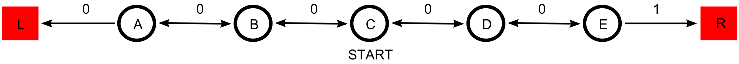
- For a single-state-per-feature representation, $\phi(x)$, C is represented by $\phi(C) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

Example

Suppose L, C, R represeted by [1,0,0] [0,1,0], and [0,0,1] repspectively. What will be representation of states A and B if we interpolate linearly?

- $\begin{bmatrix} \frac{2}{3} & \frac{1}{3} & 0 \end{bmatrix}$ and $\begin{bmatrix} \frac{1}{3} & \frac{2}{3} & 0 \end{bmatrix}$

Notation

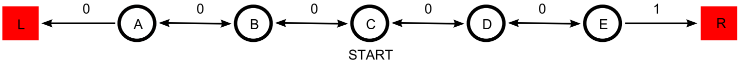


How do we construct feature's eligibility vector Z?

Example

Suppose we start in state C at time $t = 1$ and transition to B at time $t = 2$. What are Z_1 and Z_2 for some general λ and single-state-per-feature $\phi(x)$? $[Z_{t+1} = \lambda Z_t + \phi(x)]$

Notation



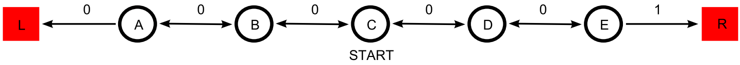
How do we construct feature's eligibility vector Z?

Example

Suppose we start in state C at time $t = 1$ and transition to B at time $t = 2$. What are Z_1 and Z_2 for some general λ and single-state-per-feature $\phi(x)$? $[Z_{t+1} = \lambda Z_t + \phi(x)]$

- $Z_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$ and $Z_2 \begin{bmatrix} 0 & 0 & 1 & 1 \times \lambda & 0 & 0 & 0 \end{bmatrix}$

More Notation



- β a coeficient vector for which $V(x) = \beta \cdot \phi(x)$. For the example above: $\beta = \begin{bmatrix} 0 & .167 & .333 & .500 & .667 & .833 & 0 \end{bmatrix}^T$ and $\phi(C) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T \implies V(C) = .5$
- One step TD error

$$R + (\phi(C) - \phi(B))^T \beta = 0 + \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right)^T \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \end{bmatrix} = -\beta_3 + \beta_4$$

TD

TD(λ) for approximate policy evaluation:

Given: • a *simulation model* for a proper policy π in MDP X;
• a *featurizer* $\phi : X \to \Re^K$ mapping states to feature vectors, $\phi(\text{END}) \stackrel{\text{def}}{=} \mathbf{0}$;
• a parameter $\lambda \in [0,1]$; and
• a sequence of *stepsizes* $\alpha_1, \alpha_2, \ldots$ for incremental coefficient updating.

Output: a coefficient vector $\boldsymbol{\beta}$ for which $V^\pi(x) \approx \boldsymbol{\beta} \cdot \phi(x)$.

Set $\boldsymbol{\beta} := \mathbf{0}$ *(or an arbitrary initial estimate)*, $t := 0$.
for $n := 1, 2, \ldots$ **do:** {
  Set $\boldsymbol{\delta} := 0$.
  Choose a start state $x_t \in X$.
  Set $\mathbf{z}_t := \phi(x_t)$.
  while $x_t \neq \text{END}$, **do:** {
    Simulate one step of the process, producing a reward R_t and next state x_{t+1}.
    Set $\boldsymbol{\delta} := \boldsymbol{\delta} + \mathbf{z}_t\big(R_t + (\phi(x_{t+1}) - \phi(x_t))^\mathsf{T}\boldsymbol{\beta}\big)$.
    Set $\mathbf{z}_{t+1} := \lambda \mathbf{z}_t + \phi(x_{t+1})$.
    Set $t := t + 1$.
  }
  Set $\boldsymbol{\beta} := \boldsymbol{\beta} + \alpha_n \boldsymbol{\delta}$.
}

Figure 1. Ordinary TD(λ) for linearly approximating the undiscounted value function of a fixed proper policy.

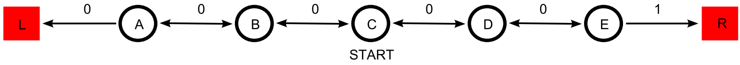
TD

Example

Suppose $\lambda = .5$, $\alpha = .1$, and suppose during first two episodes you only move to the right.

- What will be the value of δ and Z after:
  - one step? two steps? three steps?
- What will be the value of β after:
  - first episode? second episode?

while $x_t \neq$ END, **do**: {
  Simulate one step of the process, producing a reward
  Set $\boldsymbol{\delta} := \boldsymbol{\delta} + \mathbf{z}_t\big(R_t + (\boldsymbol{\phi}(x_{t+1}) - \boldsymbol{\phi}(x_t))^\mathsf{T}\boldsymbol{\beta}\big)$.
  Set $\mathbf{z}_{t+1} := \lambda\mathbf{z}_t + \boldsymbol{\phi}(x_{t+1})$.
  Set $t := t + 1$.
}
Set $\boldsymbol{\beta} := \boldsymbol{\beta} + \alpha_n\boldsymbol{\delta}$.



LSTD

LSTD(λ) for approximate policy evaluation:

Given: a *simulation model*, *featurizer*, and λ as in ordinary TD(λ).
(No stepsize schedules or initial estimates of $\boldsymbol{\beta}$ are necessary.)

Output: a coefficient vector $\boldsymbol{\beta}$ for which $V^{\pi}(x) \approx \boldsymbol{\beta} \cdot \boldsymbol{\phi}(x)$.

Set $\mathbf{A} := \mathbf{0}$, $\mathbf{b} := \mathbf{0}$, $t := 0$.
for $n := 1, 2, \ldots$ **do**: {
  Choose a start state $x_t \in X$.
  Set $\mathbf{z}_t := \boldsymbol{\phi}(x_t)$.
  while $x_t \neq$ END, **do**: {
    Simulate one step of the chain, producing a reward R_t and next state x_{t+1}.
    Set $\mathbf{A} := \mathbf{A} + \mathbf{z}_t(\boldsymbol{\phi}(x_t) - \boldsymbol{\phi}(x_{t+1}))^{\mathsf{T}}$. /* *outer product* */
    Set $\mathbf{b} := \mathbf{b} + \mathbf{z}_t R_t$.
    Set $\mathbf{z}_{t+1} := \lambda \mathbf{z}_t + \boldsymbol{\phi}(x_{t+1})$.
    Set $t := t + 1$.
  }
  Whenever updated coefficients are desired: Set $\boldsymbol{\beta} := \mathbf{A}^{-1}\mathbf{b}$. /* *Use SVD.* */
}

Figure 2. A least-squares version of TD(λ) (compare figure 1). Note that $\mathbf{A}$ has dimension $K \times K$, and $\mathbf{b}$, β, $\mathbf{z}$, and $\phi(x)$ all have dimension $K \times 1$.

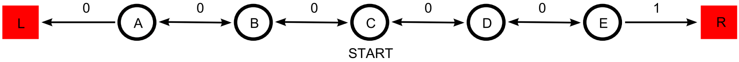
LSTD

Example

Suppose $\lambda = 0$, $\alpha = .1$, and suppose during first two episodes you only move to the right.

- What will be the value of Z, A, and b after:
  - one step? two steps?
  - first episode? second episode?

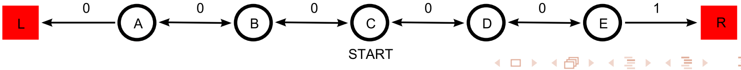
while $x_t \neq$ END, **do**: {
  Simulate one step of the chain, producing a reward
  Set $\mathbf{A} := \mathbf{A} + \mathbf{z}_t(\phi(x_t) - \phi(x_{t+1}))^\mathsf{T}$. /* o
  Set $\mathbf{b} := \mathbf{b} + \mathbf{z}_t R_t$.
  Set $\mathbf{z}_{t+1} := \lambda \mathbf{z}_t + \phi(x_{t+1})$.
  Set $t := t + 1$.
}



TD vs LSTD: RMS error

$\lambda = .5$

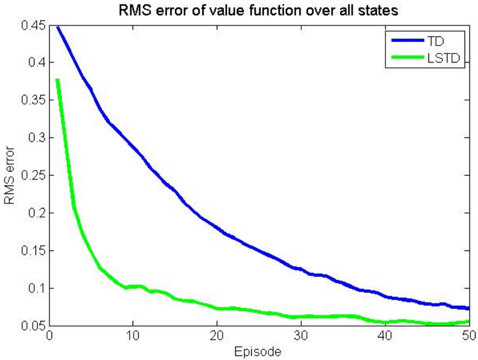
Which one does better?

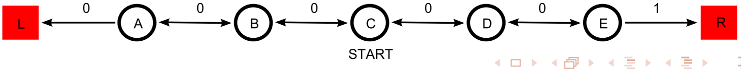


TD vs LSTD: RMS error

$\lambda = .5$

Which one does better?

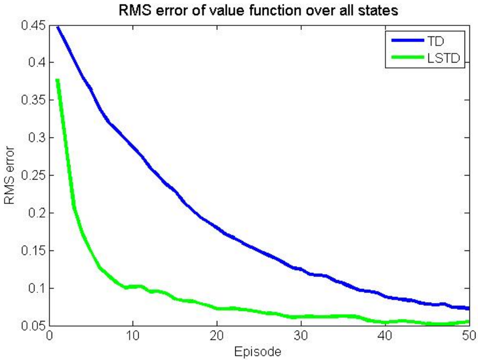




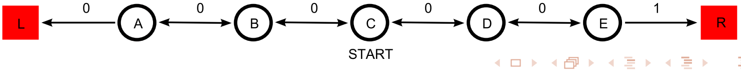
TD vs LSTD: RMS error

$\lambda = .5$

Which one does better?

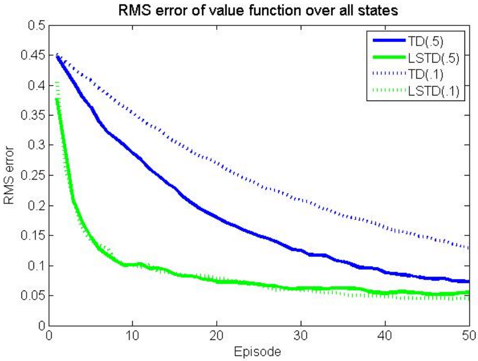


-
- What if we change λ to .1?



TD vs LSTD: RMS error

$\lambda = .5$ vs $\lambda = .1$

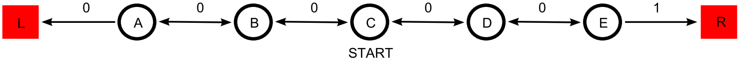


TD vs LSTD: time performance

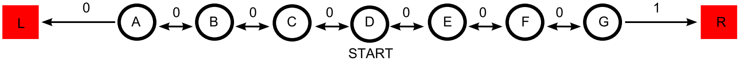
Example

Suppose we vary number of intermediate states:

from



to



Which one is faster? More accurate?

TD vs LSTD: time performance