Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation

Rich Sutton, University of Alberta
Hamid Maei, University of Alberta
Doina Precup, McGill University
Shalabh Bhatnagar, Indian Institute of Science, Bangalore
David Silver, University of Alberta
Csaba Szepesvari, University of Alberta
Eric Wiewiora, University of Alberta

a breakthrough in RL

- function approximation in TD learning is now straightforward
  - as straightforward as it is in supervised learning
- TD learning can now be done as gradient-descent in a novel Bellman error

limitations (for this paper)

- linear function approximation
- one-step TD methods ($\lambda = 0$)
- prediction (policy evaluation), not control

limitations (for this paper)

- linear function approximation
- one-step TD methods ($\lambda = 0$)
- prediction (policy evaluation), not control

all of these are being removed in current work

keys to the breakthrough

- a new Bellman error objective function
- an algorithmic trick—a second set of weights
  - to estimate one of the sub-expectations
  - and avoid the need for double sampling
  - introduced in prior work (Sutton, Szepesvari & Maei, 2008)

outline

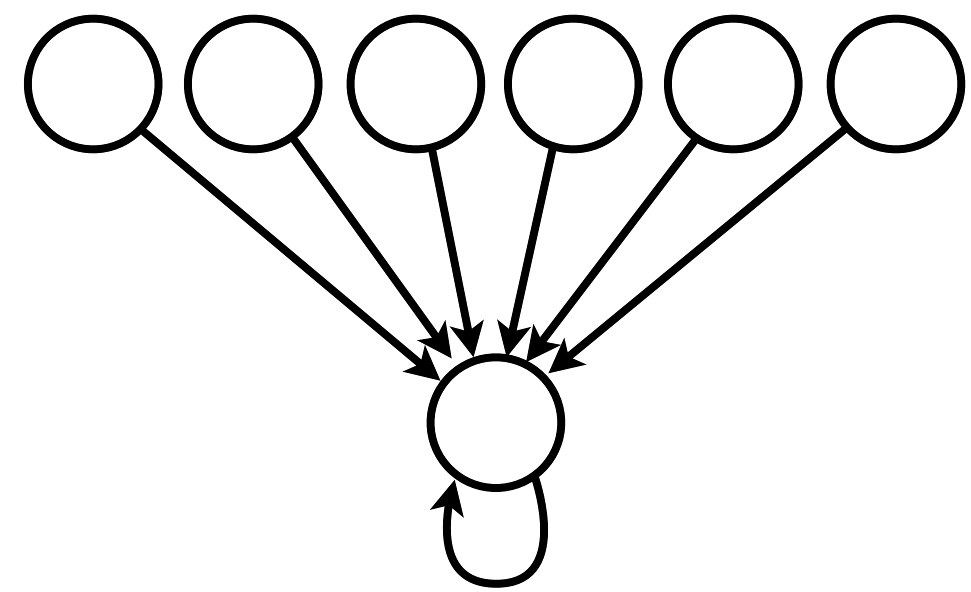
- ways in which TD with FA has not been straightforward
- the new Bellman error objective function
- derivation of new algorithms (the trick)
- results (theory and experiments)

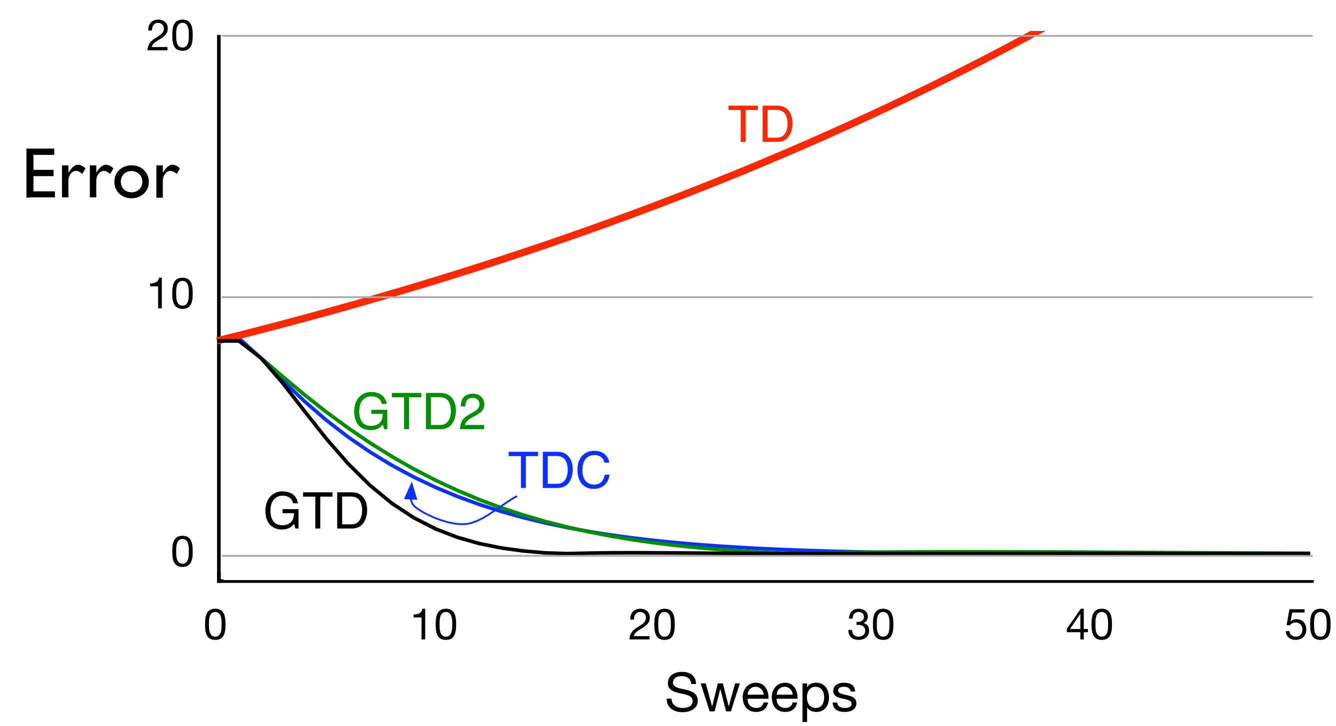
TD+FA was not straightforward

- with linear FA, off-policy methods such as Q-learning diverge on some problems (Baird, 1995)
- with nonlinear FA, even on-policy methods can diverge (Tsitsiklis & Van Roy, 1997)
- convergence guaranteed only for one very important special case—linear FA, learning about the policy being followed
- second-order or importance-sampling methods are complex, slow or messy
- no true gradient-descent methods

Baird's counterexample

- a simple Markov chain
- linear FA, all rewards zero
- deterministic, expectation-based full backups (as in DP)
- each state updated once per sweep (as in DP)
- weights can diverge to ±∞



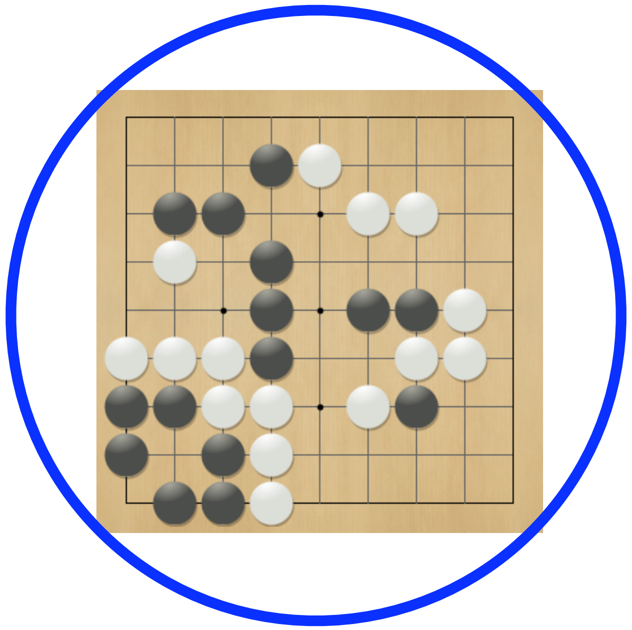


outline

- ways in which TD with FA has not been straightforward
- the new Bellman error objective function
- derivation of new algorithms (the trick)
- results (theory and experiments)

e.g. linear value-function approximation in Computer Go

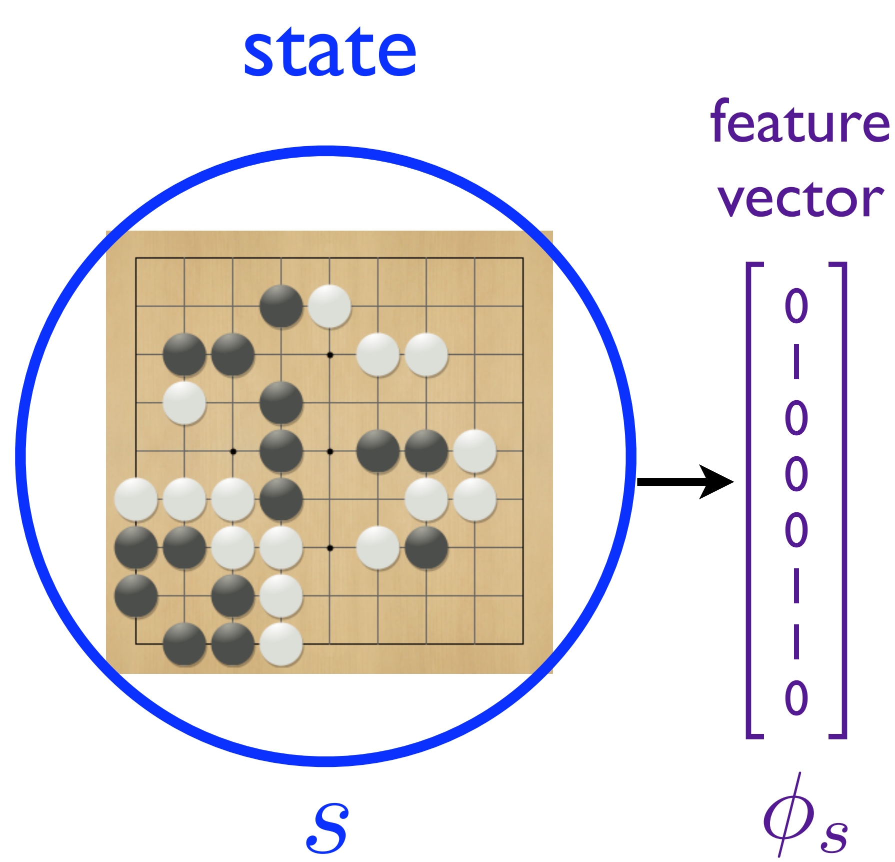
state



s

10^{35} states

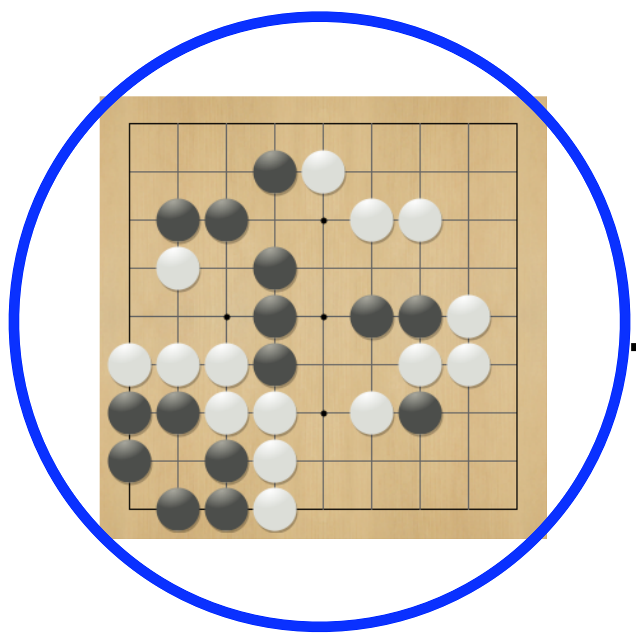
e.g. linear value-function approximation in Computer Go



10^{35} states $\quad$ 10^{5} binary features

e.g. linear value-function approximation in Computer Go

state



s

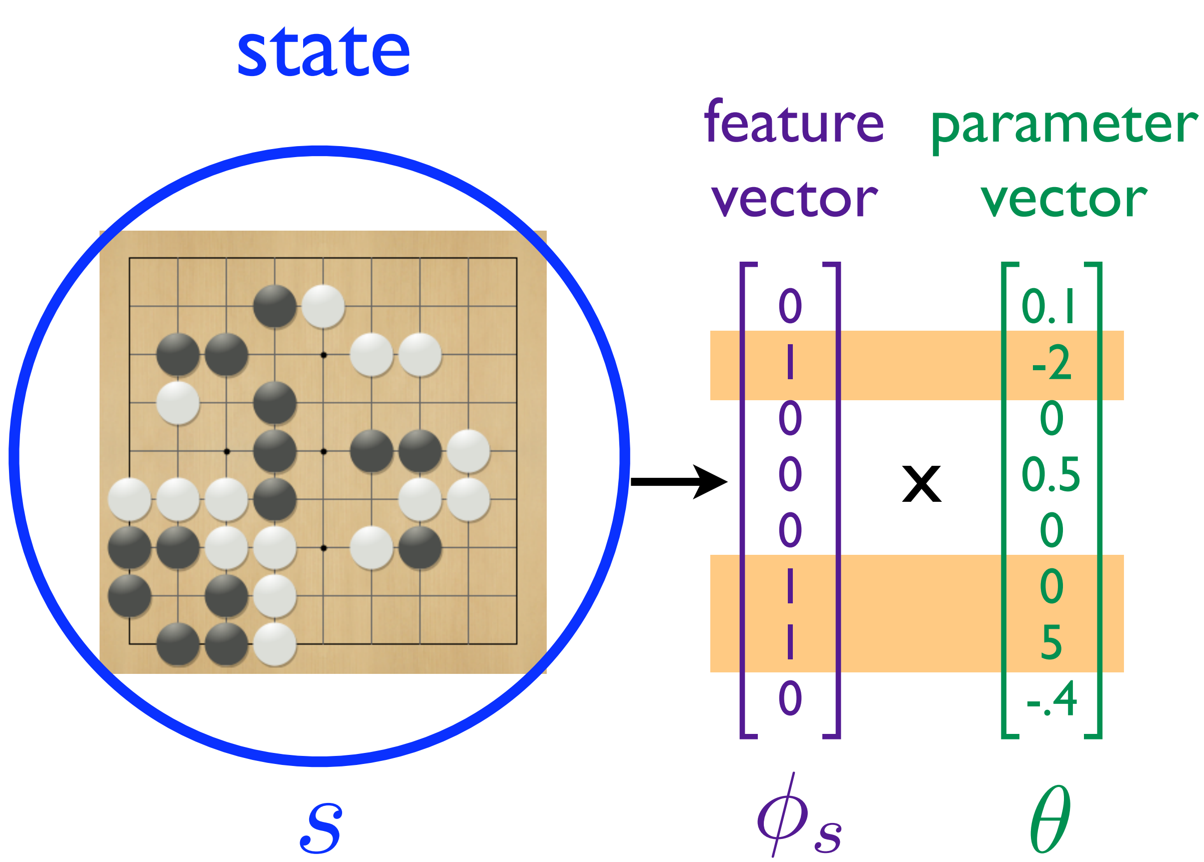
feature vector | parameter vector

$$\phi_s = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \times \theta = \begin{bmatrix} 0.1 \\ -2 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 5 \\ -.4 \end{bmatrix}$$

10^{35} states

10^{5} binary features and parameters

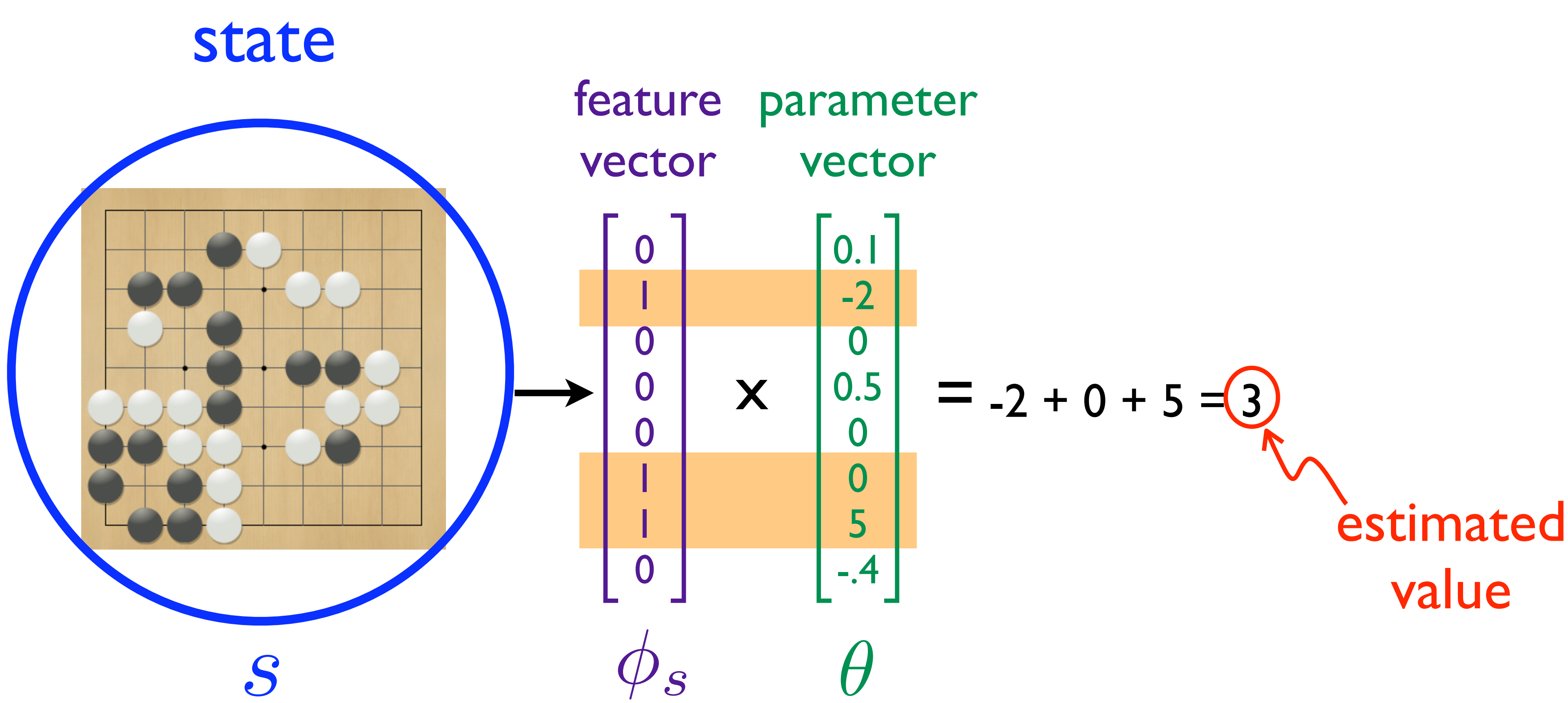
e.g. linear value-function approximation in Computer Go



10^{35} states

10^{5} binary features and parameters

e.g. linear value-function approximation in Computer Go



10^{35} states

10^{5} binary features and parameters

Notation

- state transitions: $s \xrightarrow{r} s'$
- feature vectors: $\phi \quad \phi' \in \Re^n \qquad n \ll \#\text{states}$
- approximate values: $V_\theta(s) = \theta^\top \phi \qquad \theta \in \Re^n$ parameter vector

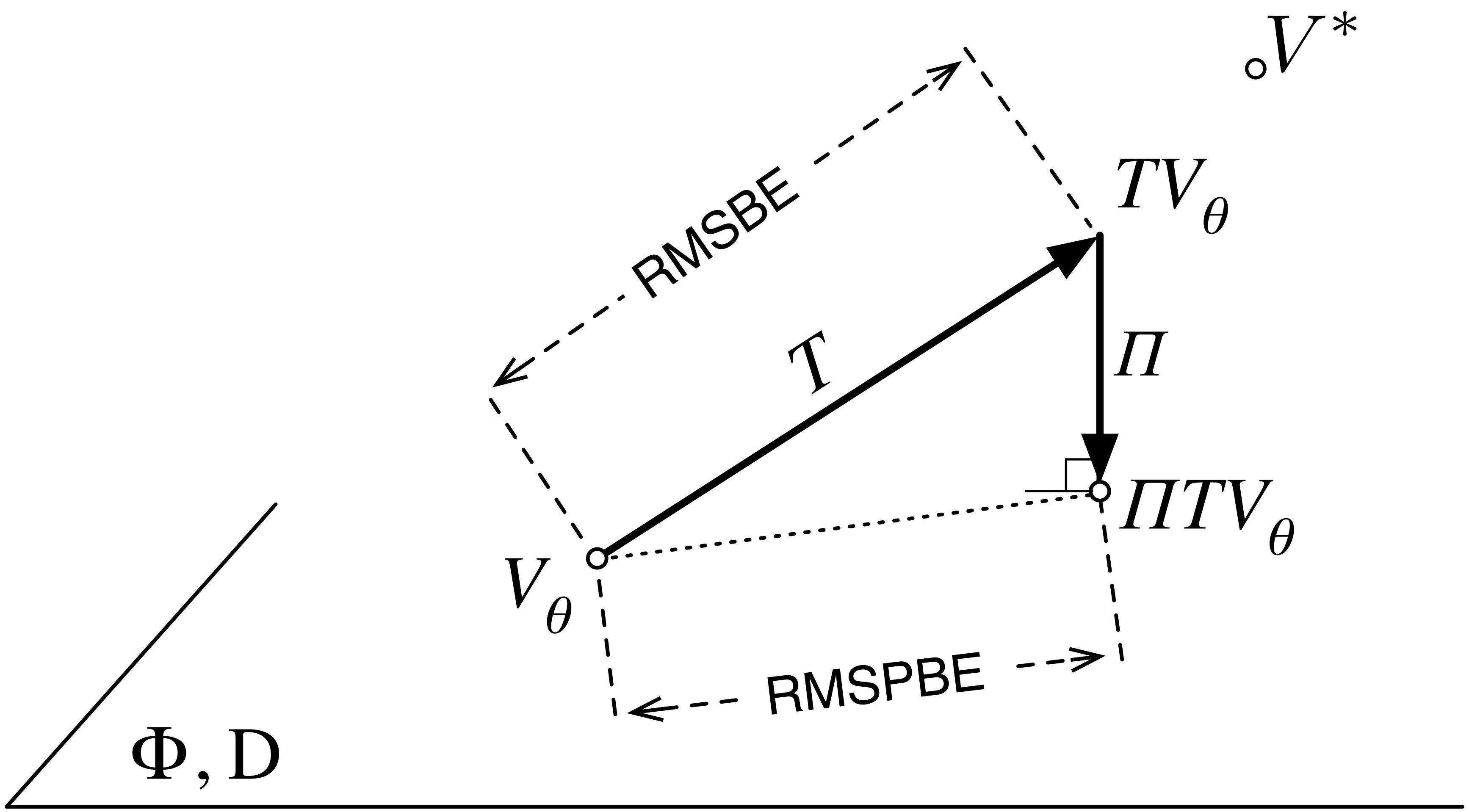
Notation

- state transitions: $s \xrightarrow{r} s'$
- feature vectors: $\phi \quad \phi' \in \Re^n$ $\qquad n \ll \#\text{states}$
- approximate values: $V_\theta(s) = \theta^\top \phi$ $\qquad \theta \in \Re^n$ parameter vector
- TD error: $\delta = r + \gamma\theta^\top\phi' - \theta^\top\phi$ $\qquad \gamma \in [0, 1)$
- TD(0) algorithm: $\Delta\theta = \alpha\delta\phi$ $\qquad \alpha > 0$

Notation

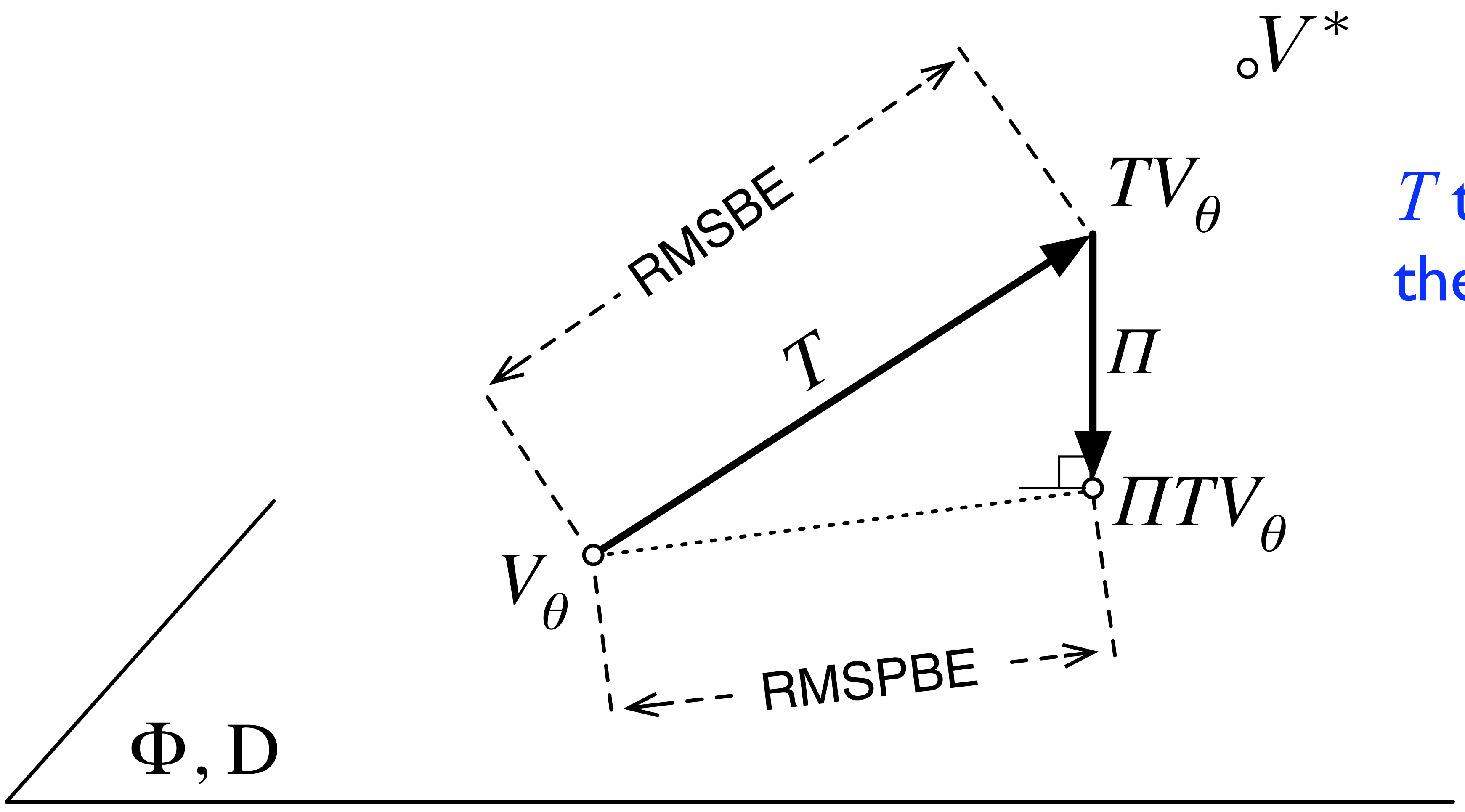
- state transitions: $s \xrightarrow{r} s'$
- feature vectors: $\phi \quad \phi' \quad \in \Re^n$ $\qquad n \ll \#\text{states}$
- approximate values: $V_\theta(s) = \theta^\top \phi$ $\qquad \theta \in \Re^n$ parameter vector
- TD error: $\delta = r + \gamma\theta^\top\phi' - \theta^\top\phi$ $\qquad \gamma \in [0, 1)$
- TD(0) algorithm: $\Delta\theta = \alpha\delta\phi$ $\qquad \alpha > 0$
- true values: $V^*(s) = \mathbb{E}[r|s] + \gamma \sum_{s'} P_{ss'} V^*(s')$
- Bellman operator: $TV = R + \gamma PV$ $\qquad V^* = TV^*$
  over per-state vectors

Value function geometry



The space spanned by the feature vectors,
weighted by the state visitation distribution

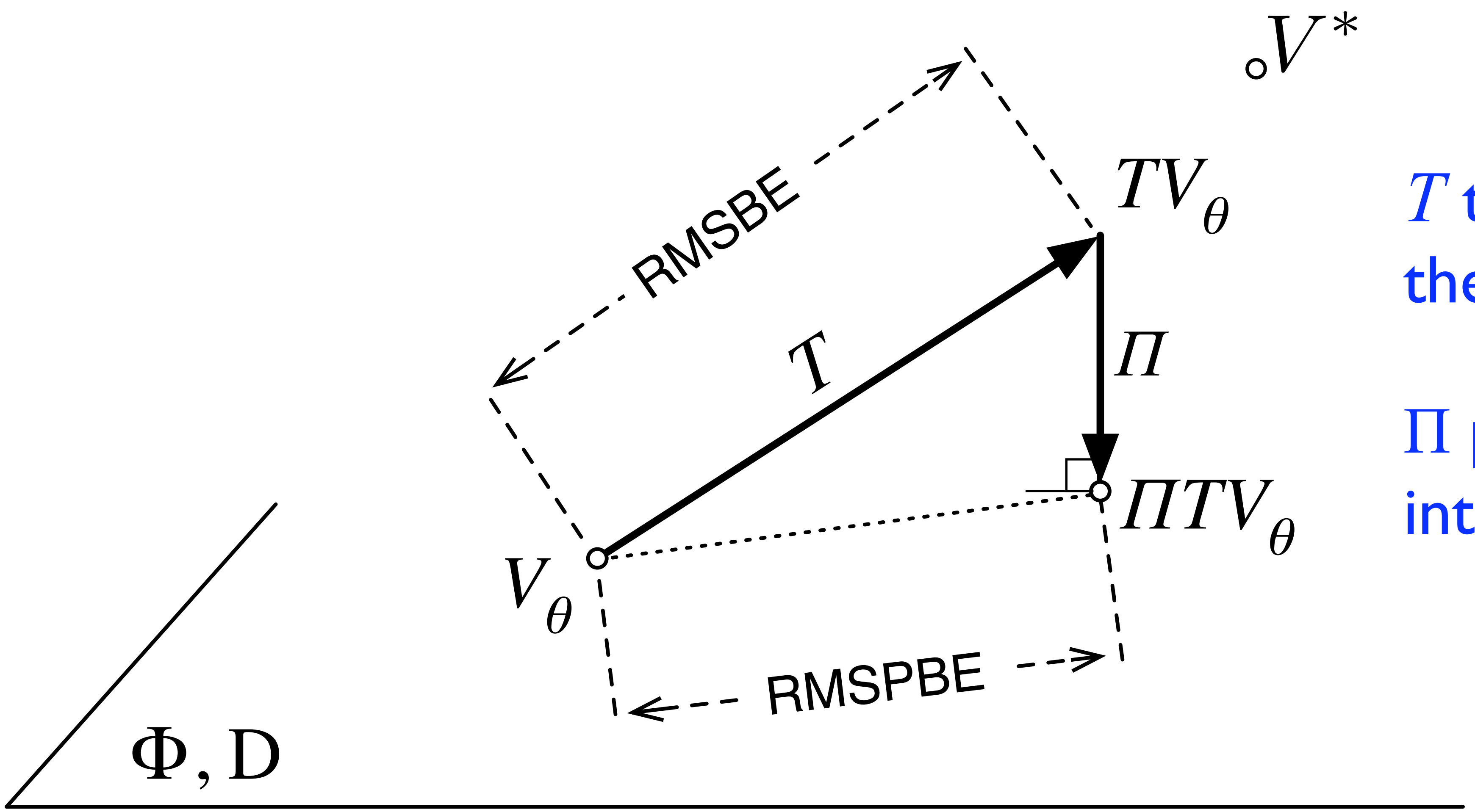
Value function geometry



The space spanned by the feature vectors,
weighted by the state visitation distribution

T takes you outside the space

Value function geometry



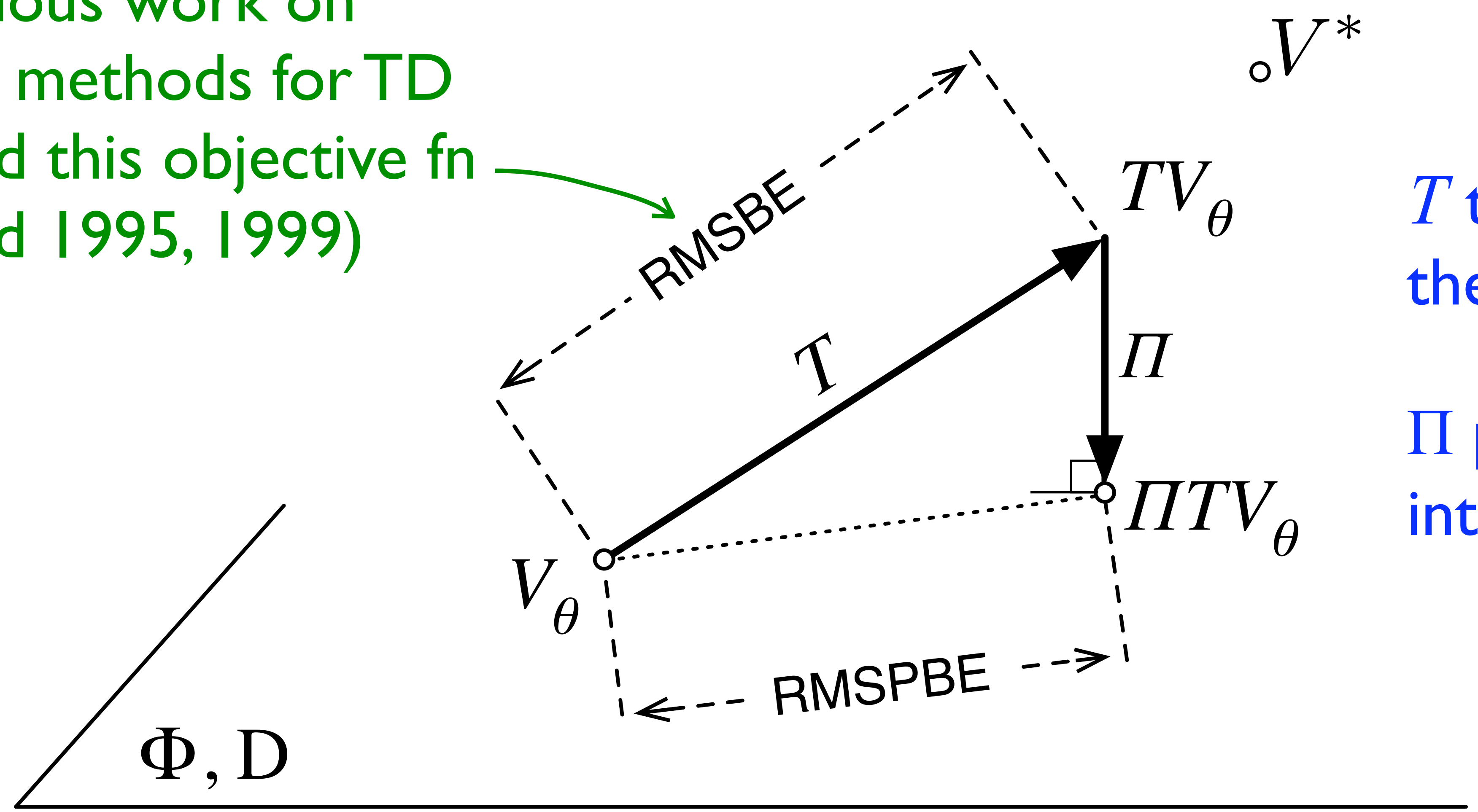
The space spanned by the feature vectors,
weighted by the state visitation distribution

T takes you outside the space

Π projects you back into it

Value function geometry

Previous work on gradient methods for TD minimized this objective fn (Baird 1995, 1999)



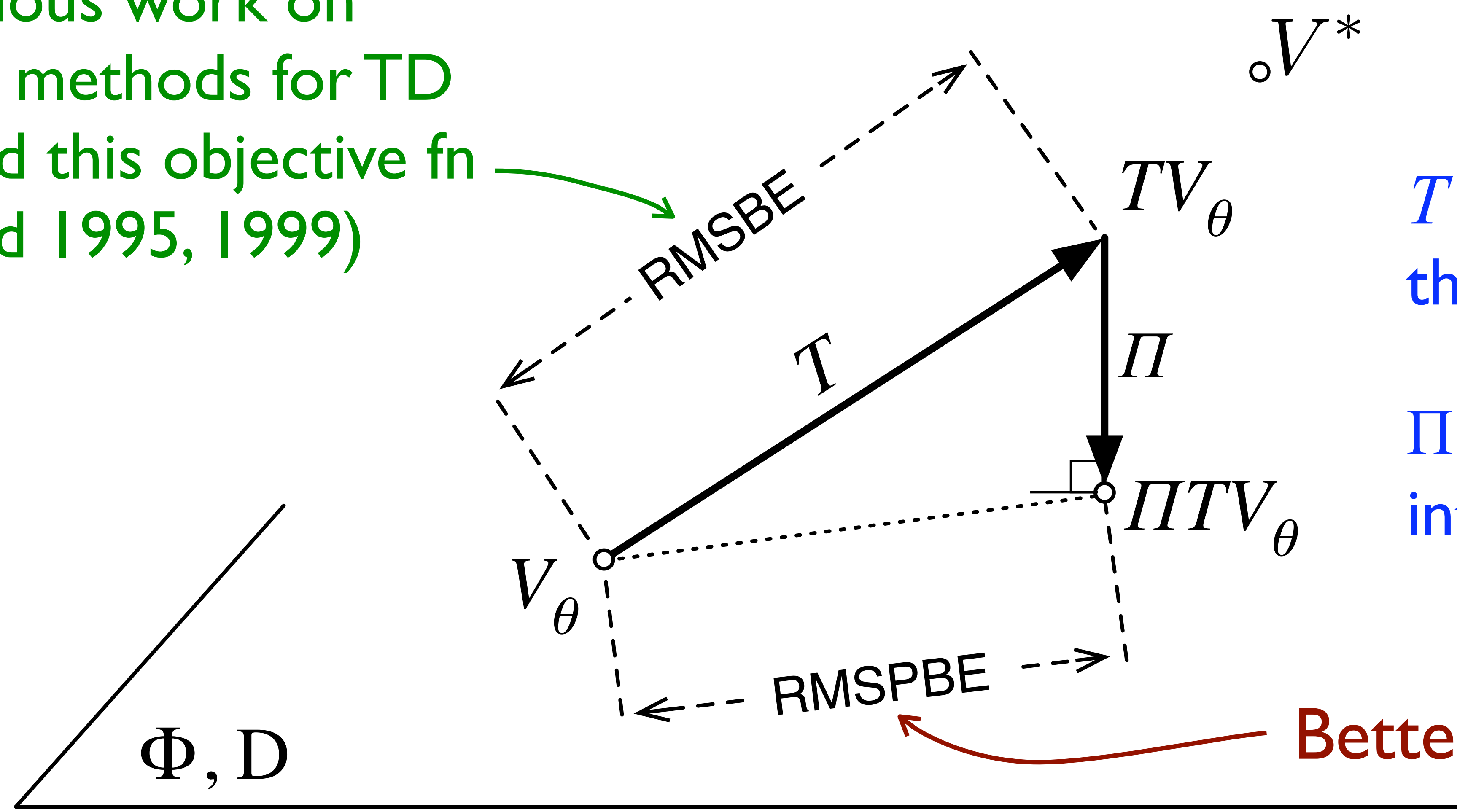
T takes you outside the space

Π projects you back into it

The space spanned by the feature vectors, weighted by the state visitation distribution

Value function geometry

Previous work on gradient methods for TD minimized this objective fn (Baird 1995, 1999)



The space spanned by the feature vectors, weighted by the state visitation distribution

T takes you outside the space

Π projects you back into it

Better objective fn?

Mean Square ***Projected*** Bellman Error (MSPBE)

TD objective functions
(to be minimized)

- Error from the true values $\| V_\theta - V^* \|_D^2$
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$

TD objective functions
(to be minimized)

- Error from the true values $\| V_\theta - V^* \|_D^2$ Not TD
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$

TD objective functions
(to be minimized)

- Error from the true values $\| V_\theta - V^* \|_D^2$ Not TD
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$ Not right
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$

TD objective functions
(to be minimized)

- Error from the true values $\| V_\theta - V^* \|_D^2$ Not TD
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$ Not right
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$ Right!

TD objective functions
(to be minimized)

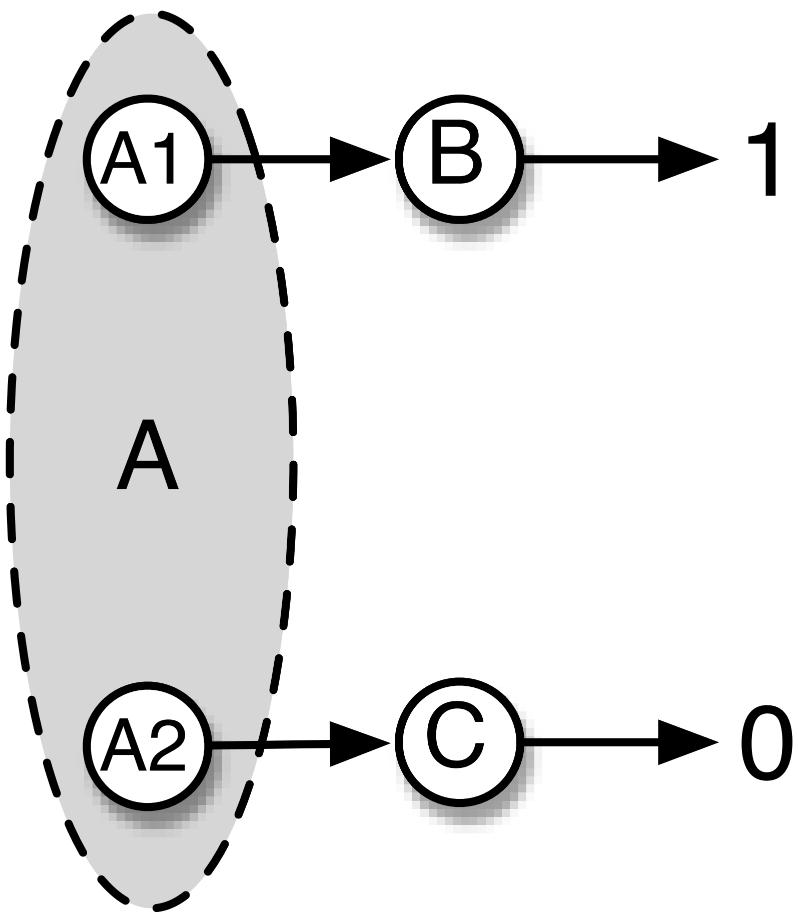
- Error from the true values $\| V_\theta - V^* \|_D^2$ Not TD
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$ Not right
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$ Right!
- Zero expected TD update $V_\theta = \Pi TV_\theta$

TD objective functions

(to be minimized)

- Error from the true values $\| V_\theta - V^* \|_D^2$ Not TD
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$ Not right
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$ Right!
- Zero expected TD update $V_\theta = \Pi TV_\theta$ Not an objective

backwards-bootstrapping example



- The two 'A' states look the same; they share a single feature and must be given the same approximate value

$$V(A1) = V(A2) = \frac{1}{2}$$

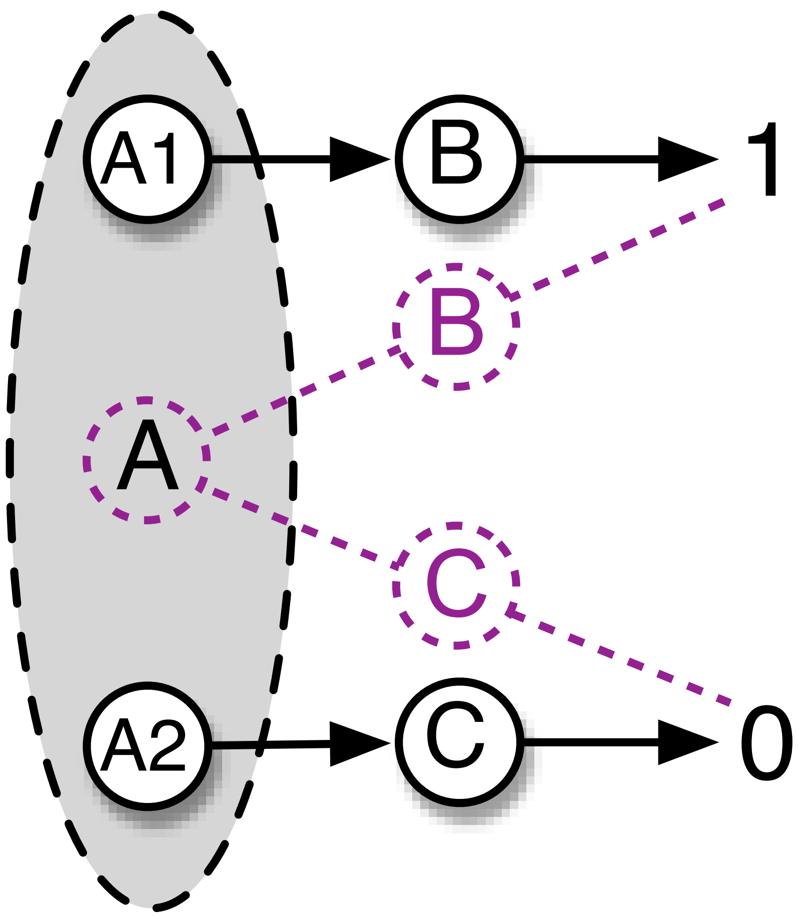
- All transitions are deterministic; Bellman error = TD error
- Clearly, the right solution is

$$V(B) = 1, \; V(C) = 0$$

- But the solution the minimizes the Bellman error is

$$V(B) = \frac{3}{4}, \; V(C) = \frac{1}{4}$$

backwards-bootstrapping example



- The two 'A' states look the same; they share a single feature and must be given the same approximate value

$$V(A1) = V(A2) = \frac{1}{2}$$

- All transitions are deterministic; Bellman error = TD error
- Clearly, the right solution is

$$V(B) = 1, \; V(C) = 0$$

- But the solution the minimizes the Bellman error is

$$V(B) = \frac{3}{4}, \; V(C) = \frac{1}{4}$$

TD objective functions
(to be minimized)

- Error from the true values $\| V_\theta - V^* \|_D^2$ Not TD
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$ Not right
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$ Right!
- Zero expected TD update $V_\theta = \Pi TV_\theta, \; \mathbb{E}[\Delta\theta_{TD}] = \vec{0}$ Not an objective
- Norm Expected TD update $\| \mathbb{E}[\Delta\theta_{TD}] \|$
- Expected squared TD error $\mathbb{E}[\delta^2]$

TD objective functions

(to be minimized)

- Error from the true values $\| V_\theta - V^* \|_D^2$ Not TD
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$ Not right
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$ Right!
- Zero expected TD update $V_\theta = \Pi TV_\theta, \; \mathbb{E}[\Delta\theta_{TD}] = \vec{0}$ Not an objective
- Norm Expected TD update $\| \mathbb{E}[\Delta\theta_{TD}] \|$ previous work
- Expected squared TD error $\mathbb{E}[\delta^2]$

TD objective functions
(to be minimized)

- Error from the true values $\| V_\theta - V^* \|_D^2$ Not TD
- Error in the Bellman equation (Bellman residual) $\| V_\theta - TV_\theta \|_D^2$ Not right
- Error in the Bellman equation *after projection* (MSPBE) $\| V_\theta - \Pi TV_\theta \|_D^2$ Right!
- Zero expected TD update $V_\theta = \Pi TV_\theta, \ \mathbb{E}[\Delta\theta_{TD}] = \vec{0}$ Not an objective
- Norm Expected TD update $\| \mathbb{E}[\Delta\theta_{TD}] \|$ previous work
- Expected squared TD error $\mathbb{E}[\delta^2]$ Not right; residual gradient

outline

- ways in which TD with FA has not been straightforward
- the new Bellman error objective function
- derivation of new algorithms (the trick)
- results (theory and experiments)

Gradient-descent learning

1. Pick an objective function $J(\theta)$, a parameterized function to be minimized
2. Use calculus to analytically compute the gradient $\nabla_\theta J(\theta)$
3. Find a "sample gradient" that you can sample on every time step and whose expected value equals the gradient
4. Take small steps in θ proportional to the sample gradient:

$$\Delta\theta = -\alpha \nabla_\theta J_t(\theta)$$

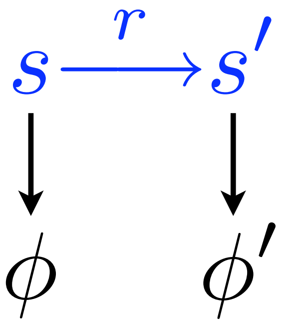
Derivation of the TDC algorithm

$$\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta)$$

Derivation of the TDC algorithm

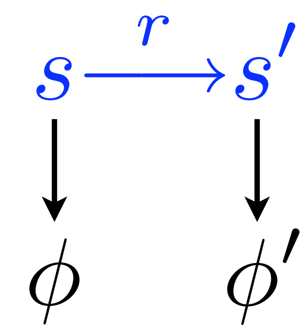
$$\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) \quad = \quad -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2$$

Derivation of the TDC algorithm



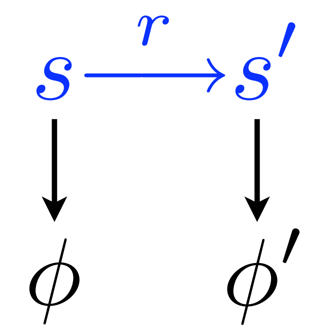
$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) &= -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]\right)
\end{aligned}
$$

Derivation of the TDC algorithm



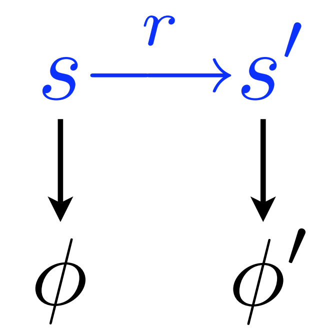
$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) &= -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]\right) \\
&= -\alpha\left(\nabla_\theta\mathbb{E}\left[\delta\phi\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]
\end{aligned}
$$

Derivation of the TDC algorithm



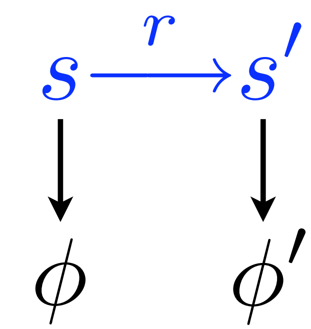
$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) \quad &= \quad -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= \quad -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]\right) \\
&= \quad -\alpha\left(\nabla_\theta\mathbb{E}\left[\delta\phi\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\nabla_\theta\left(r + \gamma\theta^\top\phi' - \theta^\top\phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]
\end{aligned}
$$

Derivation of the TDC algorithm



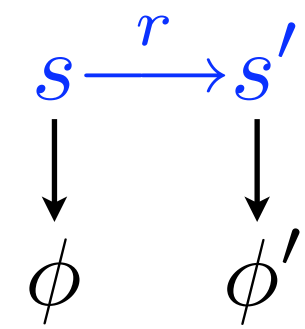
$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) \quad &= \quad -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= \quad -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right] \right) \\
&= \quad -\alpha \left(\nabla_\theta \mathbb{E}\left[\delta\phi\right]\right) \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha \mathbb{E}\left[\nabla_\theta \left(r + \gamma\theta^\top\phi' - \theta^\top\phi\right)\phi\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha \mathbb{E}\left[\left(\gamma\phi' - \phi\right)\phi\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right]
\end{aligned}
$$

Derivation of the TDC algorithm



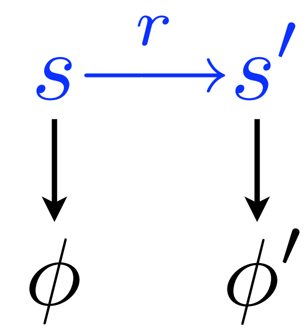
$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) \quad &= \quad -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= \quad -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]\right) \\
&= \quad -\alpha\left(\nabla_\theta\mathbb{E}\left[\delta\phi\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\nabla_\theta\left(r + \gamma\theta^\top\phi' - \theta^\top\phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\left(\gamma\phi' - \phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad \alpha\left(\mathbb{E}\left[\phi\phi^\top\right] - \gamma\mathbb{E}\left[\phi'\phi^\top\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]
\end{aligned}
$$

Derivation of the TDC algorithm



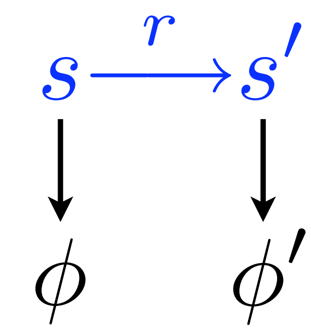
$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) \quad &= \quad -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= \quad -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]\right) \\
&= \quad -\alpha\left(\nabla_\theta\mathbb{E}\left[\delta\phi\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\nabla_\theta\left(r + \gamma\theta^\top\phi' - \theta^\top\phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\left(\gamma\phi' - \phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad \alpha\left(\mathbb{E}\left[\phi\phi^\top\right] - \gamma\mathbb{E}\left[\phi'\phi^\top\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad \alpha\mathbb{E}\left[\delta\phi\right] - \alpha\gamma\mathbb{E}\left[\phi'\phi^\top\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]
\end{aligned}
$$

Derivation of the TDC algorithm



$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) \quad &= \quad -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= \quad -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right]\right) \\
&= \quad -\alpha\left(\nabla_\theta \mathbb{E}\left[\delta\phi\right]\right) \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\nabla_\theta\left(r + \gamma\theta^\top\phi' - \theta^\top\phi\right)\phi\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\left(\gamma\phi' - \phi\right)\phi\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right] \\
&= \quad \alpha\left(\mathbb{E}\left[\phi\phi^\top\right] - \gamma\mathbb{E}\left[\phi'\phi^\top\right]\right) \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right] \\
&= \quad \alpha\mathbb{E}\left[\delta\phi\right] - \alpha\gamma\mathbb{E}\left[\phi'\phi^\top\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}\left[\delta\phi\right] \\
&\approx \quad \alpha\mathbb{E}\left[\delta\phi\right] - \alpha\gamma\mathbb{E}\left[\phi'\phi^\top\right] w
\end{aligned}
$$

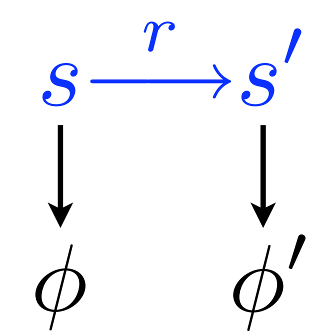
Derivation of the TDC algorithm



$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) &= -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]\right) \\
&= -\alpha\left(\nabla_\theta\mathbb{E}\left[\delta\phi\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= -\alpha\mathbb{E}\left[\nabla_\theta\left(r + \gamma\theta^\top\phi' - \theta^\top\phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= -\alpha\mathbb{E}\left[\left(\gamma\phi' - \phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \alpha\left(\mathbb{E}\left[\phi\phi^\top\right] - \gamma\mathbb{E}\left[\phi'\phi^\top\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \alpha\mathbb{E}\left[\delta\phi\right] - \alpha\gamma\mathbb{E}\left[\phi'\phi^\top\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&\approx \alpha\mathbb{E}\left[\delta\phi\right] - \alpha\gamma\mathbb{E}\left[\phi'\phi^\top\right] w
\end{aligned}
$$

This is the trick!
$w \in \Re^n$ is a second set of weights

Derivation of the TDC algorithm

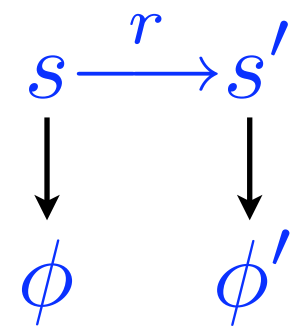


$$
\begin{aligned}
\Delta\theta = -\frac{1}{2}\alpha\nabla_\theta J(\theta) \quad &= \quad -\frac{1}{2}\alpha\nabla_\theta \parallel V_\theta - \Pi T V_\theta \parallel_D^2 \\
&= \quad -\frac{1}{2}\alpha\nabla_\theta \left(\mathbb{E}\left[\delta\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right]\right) \\
&= \quad -\alpha\left(\nabla_\theta\mathbb{E}\left[\delta\phi\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\nabla_\theta\left(r + \gamma\theta^\top\phi' - \theta^\top\phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad -\alpha\mathbb{E}\left[\left(\gamma\phi' - \phi\right)\phi\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad \alpha\left(\mathbb{E}\left[\phi\phi^\top\right] - \gamma\mathbb{E}\left[\phi'\phi^\top\right]\right)\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&= \quad \alpha\mathbb{E}\left[\delta\phi\right] - \alpha\gamma\mathbb{E}\left[\phi'\phi^\top\right]\mathbb{E}\left[\phi\phi^\top\right]^{-1}\mathbb{E}\left[\delta\phi\right] \\
&\approx \quad \alpha\mathbb{E}\left[\delta\phi\right] - \alpha\gamma\mathbb{E}\left[\phi'\phi^\top\right]w \\
\text{(sampling)} \quad &\approx \quad \alpha\delta\phi - \alpha\gamma\phi'\phi^\top w
\end{aligned}
$$

This is the trick!
$w \in \Re^n$ is a second set of weights

The complete *TD with gradient correction* (TDC) algorithm

- on each transition



- update two parameters

$$\theta \leftarrow \theta + \alpha\delta\phi - \alpha\gamma\phi' \left(\phi^\top w\right)$$

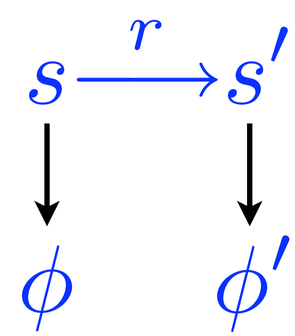
$$w \leftarrow w + \beta(\delta - \phi^\top w)\phi$$

- where

$$\delta = r + \gamma\theta^\top\phi' - \theta^\top\phi$$

The complete *TD with gradient correction* (TDC) algorithm

- on each transition



- update two parameters TD(0)

$$\theta \leftarrow \theta + \alpha\delta\phi - \alpha\gamma\phi' \left(\phi^\top w\right)$$

$$w \leftarrow w + \beta(\delta - \phi^\top w)\phi$$

- where

$$\delta = r + \gamma\theta^\top\phi' - \theta^\top\phi$$

The complete *TD with gradient correction* (TDC) algorithm

- on each transition

$$\begin{array}{ccc} s & \xrightarrow{r} & s' \\ \downarrow & & \downarrow \\ \phi & & \phi' \end{array}$$

- update two parameters

TD(0) — with gradient correction

$$\theta \leftarrow \theta + \alpha\delta\phi - \alpha\gamma\phi'\left(\phi^\top w\right)$$

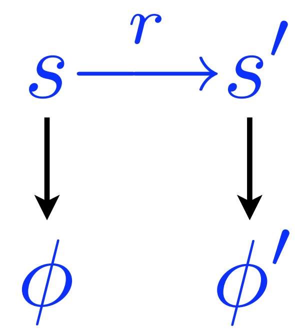
$$w \leftarrow w + \beta(\delta - \phi^\top w)\phi$$

- where

$$\delta = r + \gamma\theta^\top\phi' - \theta^\top\phi$$

The complete *TD with gradient correction* (TDC) algorithm

- on each transition



- update two parameters

$$\theta \leftarrow \theta + \alpha\delta\phi - \alpha\gamma\phi' \left(\phi^\top w\right)$$

$$w \leftarrow w + \beta(\delta - \phi^\top w)\phi$$

estimate of the TD error (δ) for the current state ϕ

- where

$$\delta = r + \gamma\theta^\top\phi' - \theta^\top\phi$$

outline

- ways in which TD with FA has not been straightforward
- the new Bellman error objective function
- derivation of new algorithms (the trick)
- results (theory and experiments)

Three new algorithms

- GTD, the original *gradient TD algorithm* (Sutton, Szepevari & Maei, 2008)
- GTD2, a second-generation GTD
- TDC

Convergence theorems

- For arbitrary on- or off-policy training
- All algorithms converge w.p.1 to the TD fix-point:

$$\mathbb{E}\left[\delta\phi\right] \longrightarrow 0$$

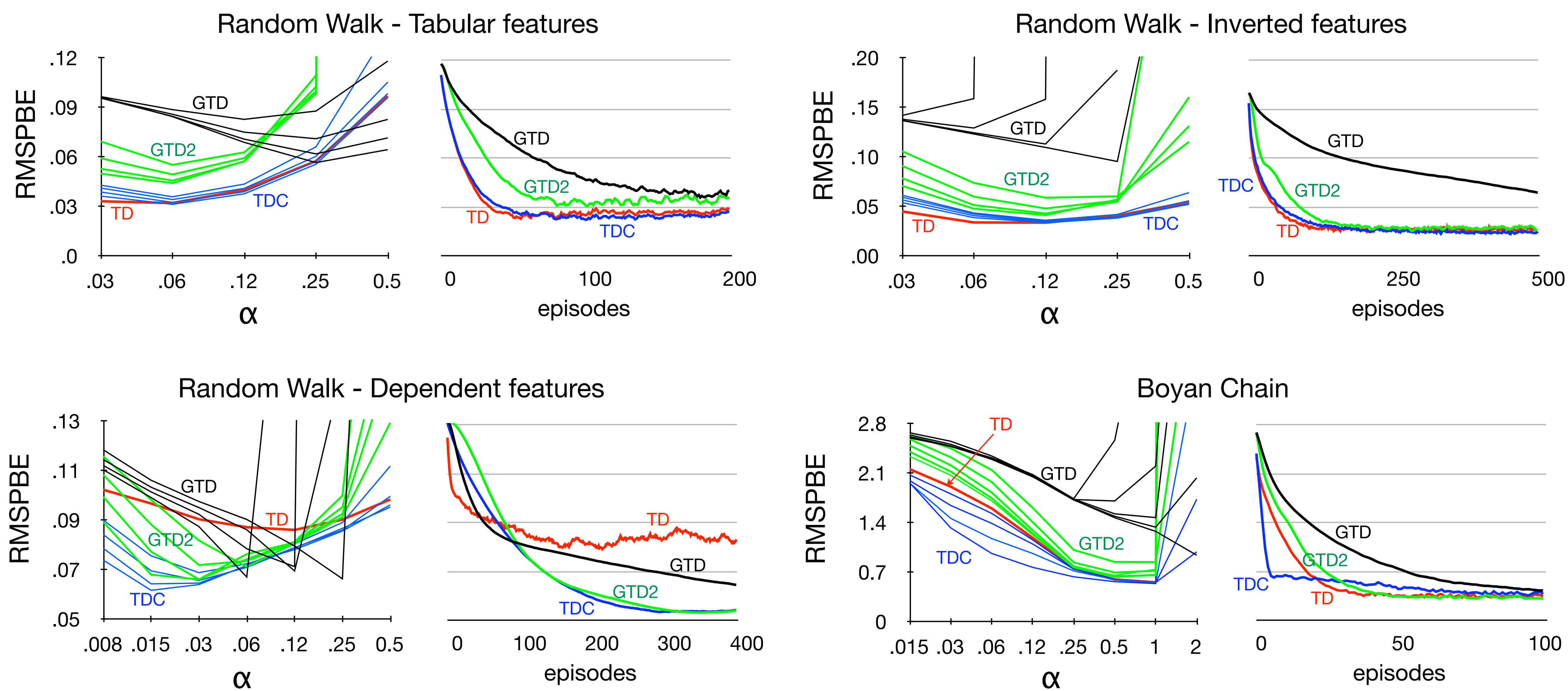
- GTD, GTD2 converge at one time scale

$$\alpha = \beta \longrightarrow 0$$

- TDC converges in a two-time-scale sense

$$\alpha, \beta \longrightarrow 0 \qquad \frac{\alpha}{\beta} \longrightarrow 0$$

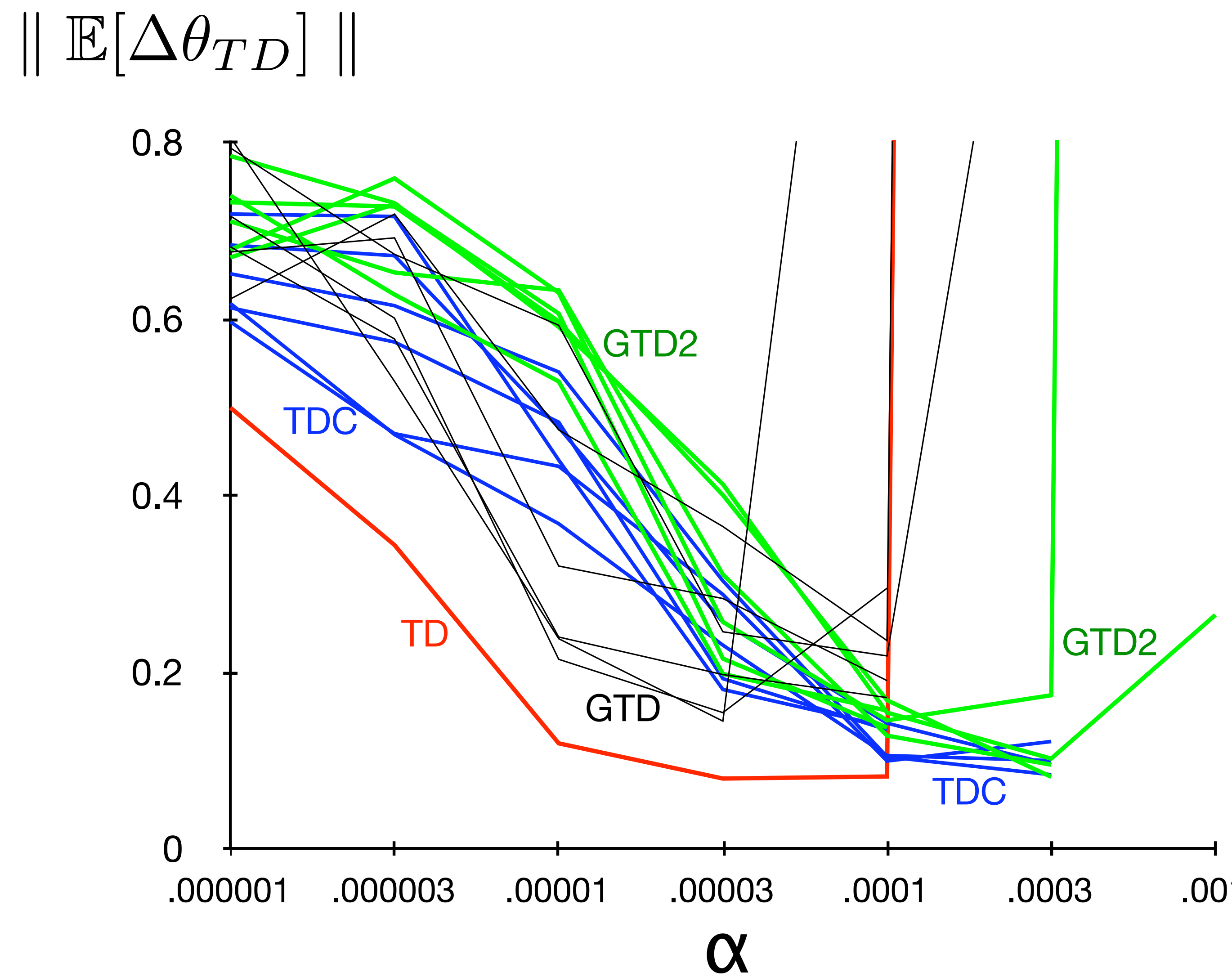
Summary of empirical results on small problems



Usually GTD << GTD2 < TD, TDC

Computer Go experiment

- Learn a linear value function (probability of winning) for 9x9 Go from self play
- One million features, each corresponding to a template on a part of the Go board
- An established experimental testbed



conclusions

- the new algorithms are roughly the same efficiency as conventional TD on on-policy problems
- but are guaranteed convergent under general off-policy training as well
- their key ideas appear to extend quite broadly, to control, general λ, non-linear settings, DP, intra-option learning, TD nets...
- *TD with FA is now straightforward*
- *the curse of dimensionality is removed*