New Temporal-Difference Methods Based on Gradient Descent

Rich Sutton
Hamid Maei
Doina Precup (McGill)
Shalabh Bhatnagar (IIS Bangalore)
Csaba Szepesvari
Eric Wiewiora
David Silver

Outline

- The promise and problems of TD learning
- Value-function approximation
- Gradient-descent methods - LMS example
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence
- Empirical results
- Conclusions

What is temporal-difference learning?

- The most important and distinctive idea in reinforcement learning
- A way of learning to predict, from changes in your predictions, without waiting for the final outcome
- A way of taking *advantage of state* in multi-step prediction problems
- Learning a guess from a guess

Examples of TD learning opportunities

- Learning to evaluate backgammon positions from changes in evaluation within a game
- Learning where your tennis opponent will hit the ball from his approach
- Learning what features of a market indicate that it will have a major decline
- Learning to recognize your friend's face

Function approximation

- TD learning is sometimes done in a table-lookup context - where every state is distinct and treated totally separately
- But really, to be powerful, we must generalize between states
  - The same state never occurs twice

For example, in Computer Go,
we use 10^6 parameters to learn about 10^{170} positions

Advantages of TD methods for prediction

1. Data efficient.
   Learn much faster on Markov problems
2. Cheap to implement.
   Require less memory, peak computation;
3. Able to learn from incomplete sequences.
   In particular, able to learn *off-policy*

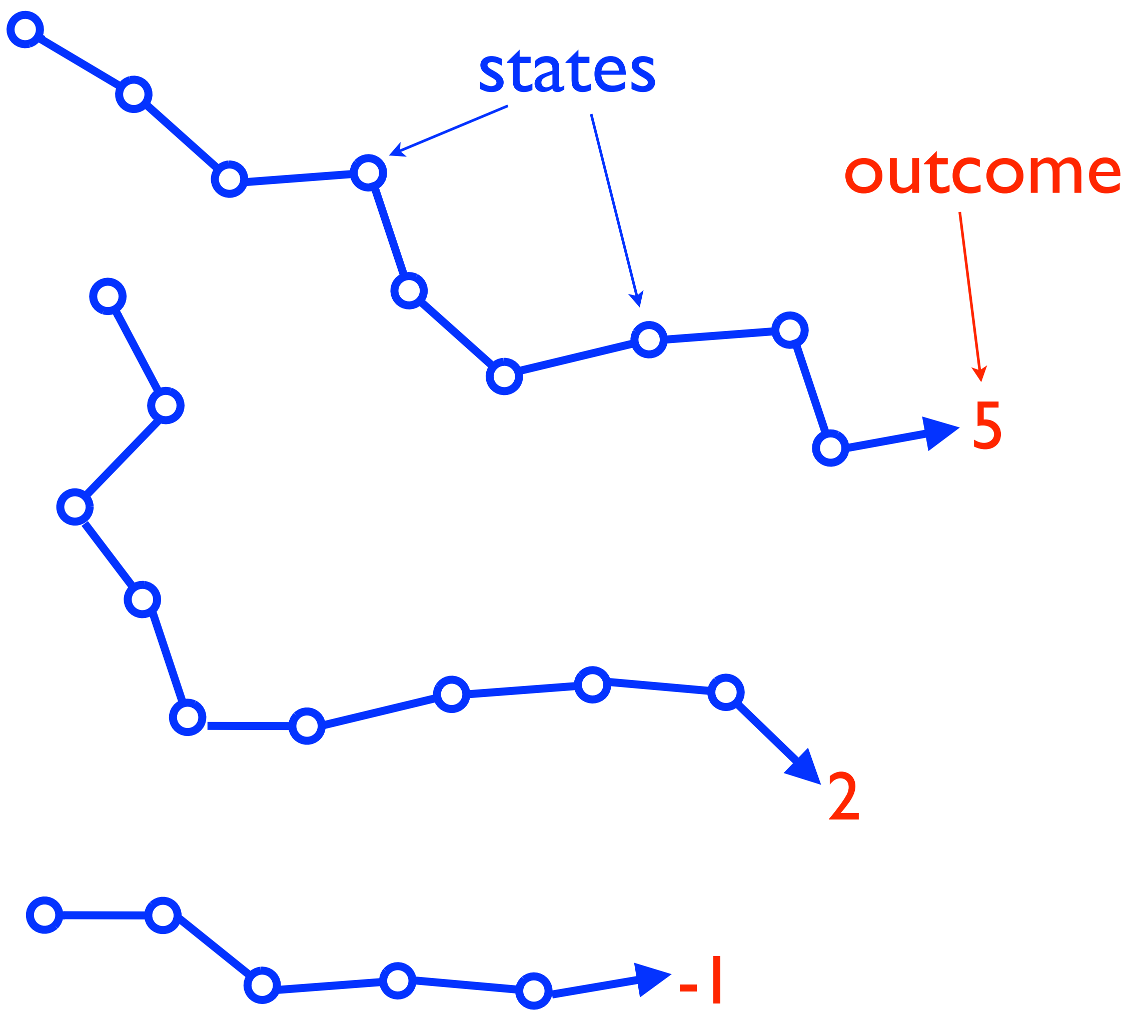
Off-policy learning

- Learning about a policy different than the one being used to generate actions
  - Most often used to learn optimal behavior from a given data set, or from more exploratory behavior
  - Key to ambitious theories of knowledge and perception as continual prediction about the outcomes of options

Outline

- The promise and problems of TD learning
- Value-function approximation
- Gradient-descent methods - LMS example
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence
- Empirical results
- Conclusions

Value-function approximation from sample trajectories



- True values:

$$V(s) = \mathbb{E}[\text{outcome}|s]$$

- Estimated values:

$$V_\theta(s) \approx V(s), \qquad \theta \in \Re^n$$

- Linear approximation:

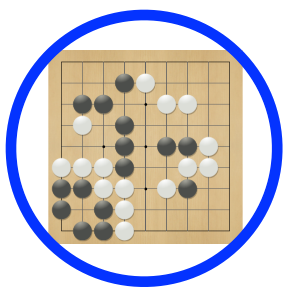
$$V_\theta(s) = \theta^\top \phi_s, \qquad \phi_s \in \Re^n$$

modifiable parameter vector

feature vector
for state s

Value-function approximation from sample trajectories

state



$$\begin{array}{ccccc} & \text{feature vector} & & \text{parameter vector} & \\ & \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} & \times & \begin{bmatrix} 0.1 \\ -2 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 5 \\ -.4 \end{bmatrix} & = -2 + 0 + 5 = 3 \\ s & \phi_s & & \theta & \end{array}$$

- True values:

$$V(s) = \mathbb{E}[\text{outcome}|s]$$

- Estimated values:

$$V_\theta(s) \approx V(s), \qquad \theta \in \Re^n$$

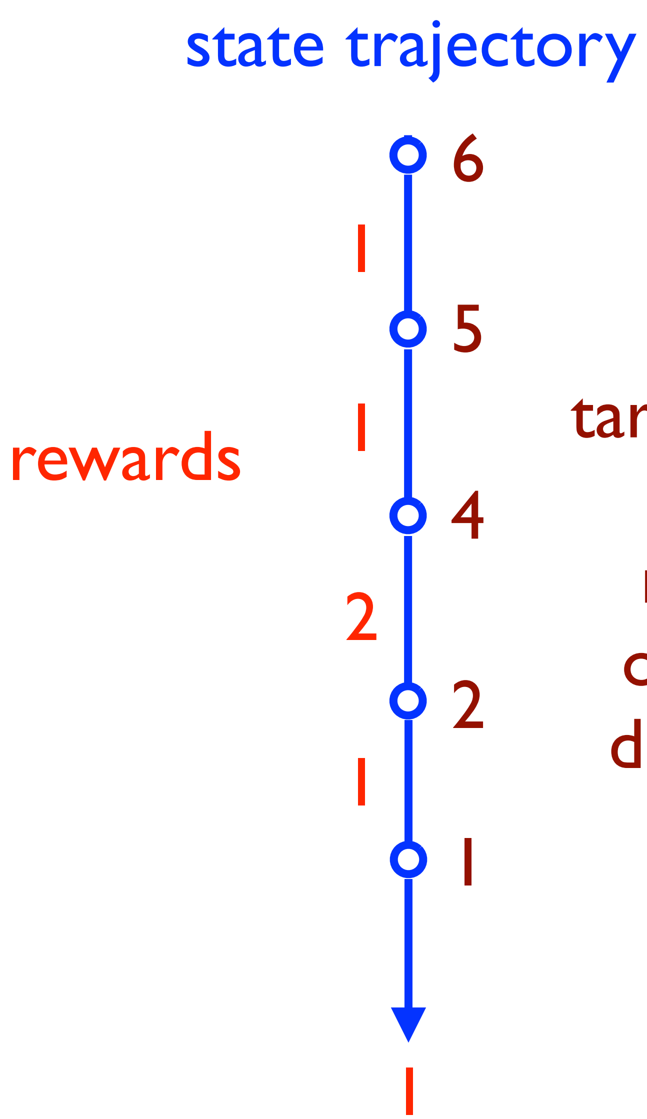
- Linear approximation:

$$V_\theta(s) = \theta^\top \phi_s, \qquad \phi_s \in \Re^n$$

modifiable parameter vector

feature vector for state s

From terminal outcomes to per-step rewards



target values (returns)
= sum of future
rewards until end
of episode, or until
discounting horizon

- True values:

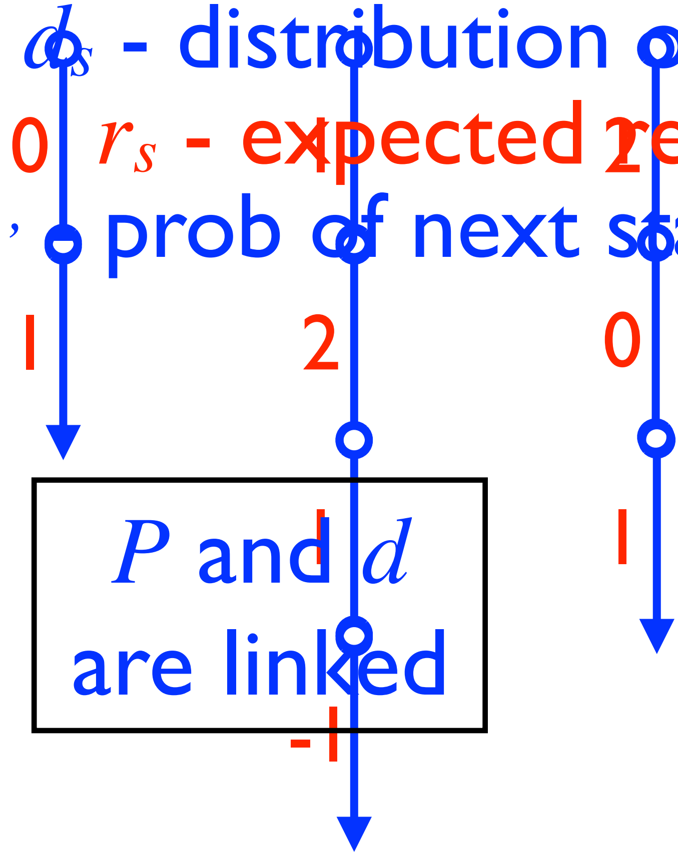
$$V(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\right]$$

discount rate,
$0 \leq \gamma \leq 1$

TD methods operate on individual transitions

trajectories

transitions



d_s - distribution of first state s

r_s - expected reward given s

$P_{ss'}$ - prob of next state s' given s

P and d are linked

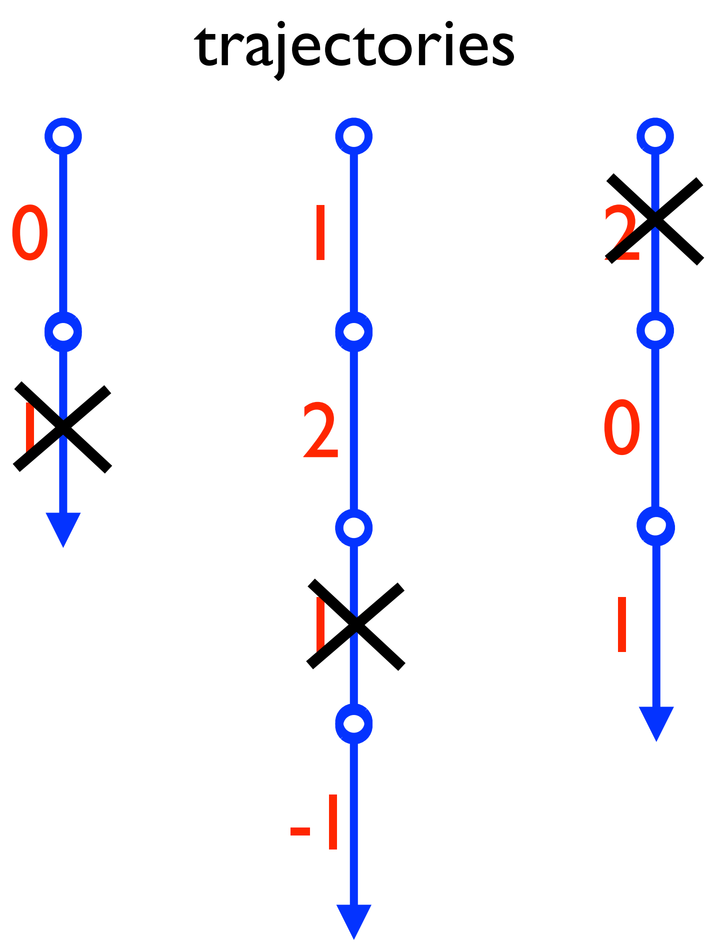
Training set is now a bag of transitions
Select from them i.i.d.
(independently, identically distributed)

Sample transition: (s, r, s') or (ϕ, r, ϕ')

TD(0) algorithm: $\theta \leftarrow \theta + \alpha\delta\phi$

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

Off-policy training



transitions

d_s

r_s

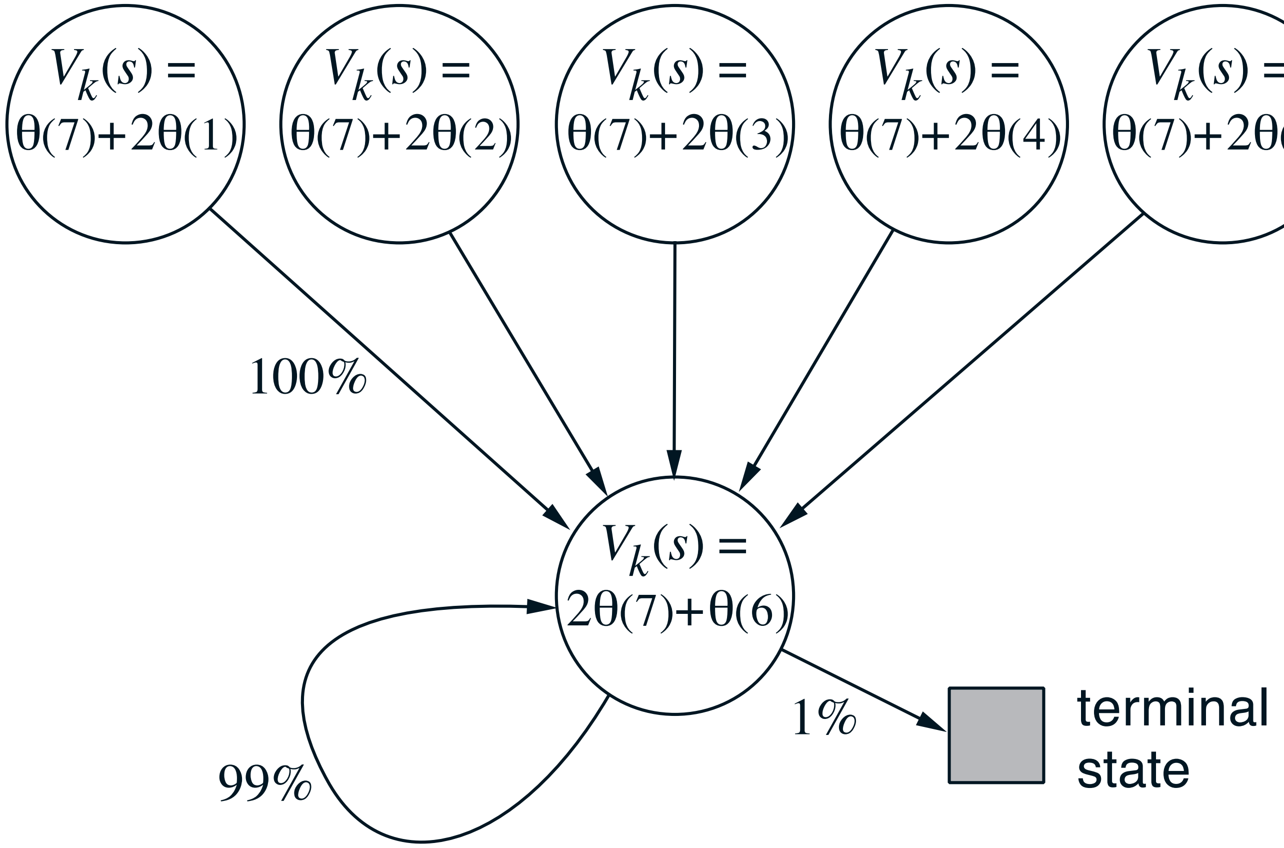
$P_{ss'}$

P and d are no longer linked

TD(0) may diverge!

Baird's counter-example

- P and d are not linked
  - d is all states with equal probability
  - P is according to this Markov chain:



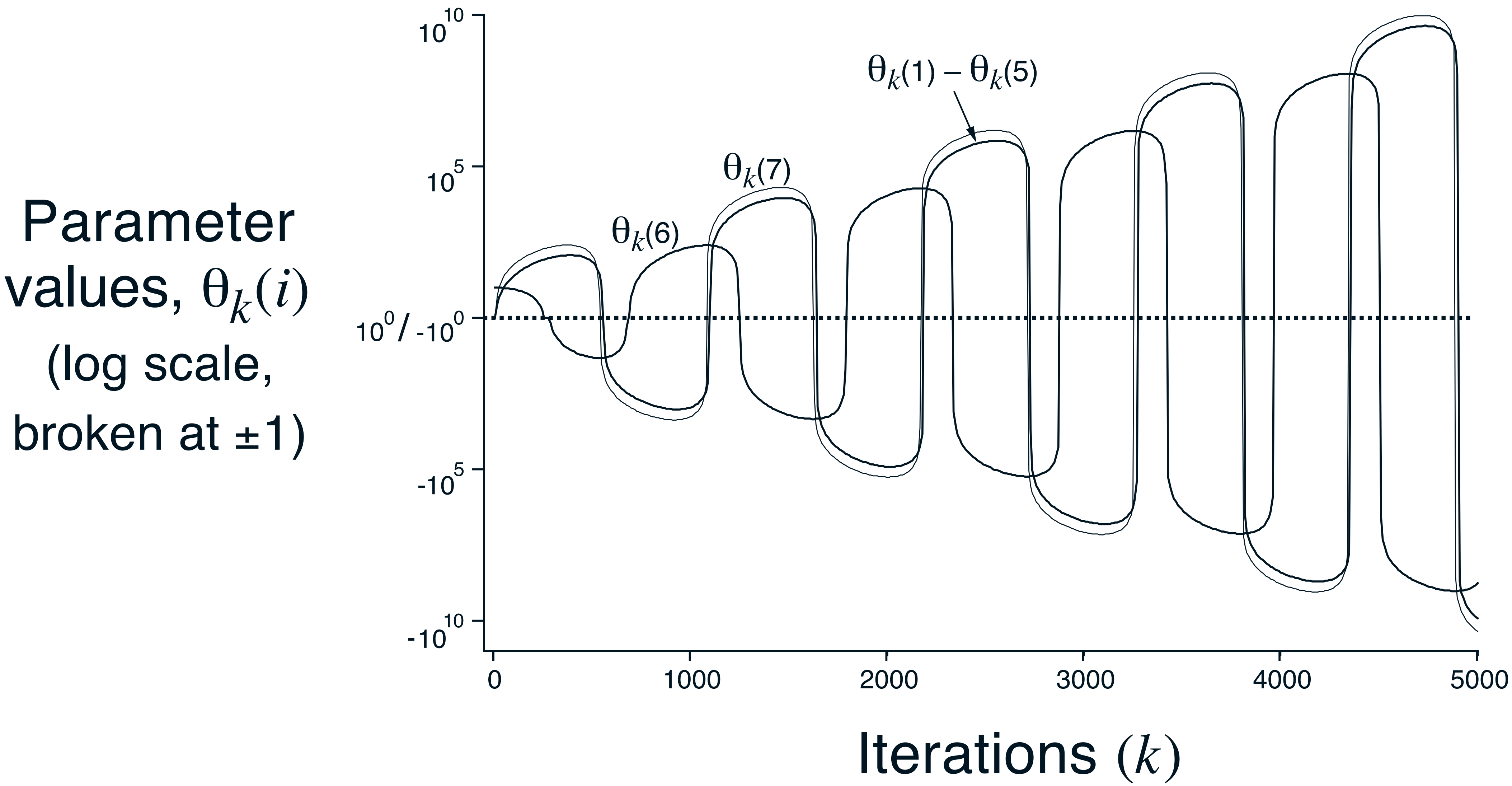
$$\alpha = 0.01$$

$$\gamma = 0.99$$

$$\theta_0 = (1, 1, 1, 1, 1, 10, 1)^\top$$

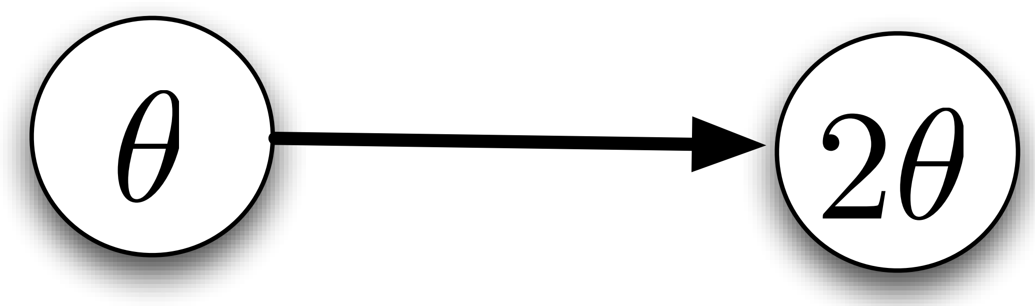
$$r = 0$$

TD can diverge: Baird's counter-example



$\alpha = 0.01$ $\gamma = 0.99$ $\theta_0 = (1, 1, 1, 1, 1, 10, 1)^\top$ **deterministic updates**

TD(0) can diverge: A simple example



$$\begin{aligned}\delta &= r + \gamma\theta^\top\phi' - \theta^\top\phi \\ &= 0 + 2\theta - \theta \\ &= \theta\end{aligned}$$

TD update: $$\begin{aligned}\Delta\theta &= \alpha\delta\phi \\ &= \alpha\theta\end{aligned}$$ Diverges!

TD fixpoint: $$\theta^* = 0$$

Previous attempts to solve the off-policy problem

- Importance sampling
  - With recognizers
- Least-squares methods, LSTD, LSPI, iLSTD
- Averagers
- Residual gradient methods

Desiderata:
We want a TD algorithm that

- Bootstraps (genuine TD)
- Works with linear function approximation (stable, reliably convergent)
- Is simple, like linear TD — O(n)
- Learns fast, like linear TD
- Can learn off-policy (arbitrary P and d)
- Learns from online causal trajectories (no repeat sampling from the same state)

Outline

- The promise and problems of TD learning
- Value-function approximation
- Gradient-descent methods - LMS example
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence
- Empirical results
- Conclusions

Gradient-descent learning methods - the recipe

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:

$$\theta \leftarrow \theta - \alpha \nabla_\theta J_t(\theta)$$

Conventional TD is not the gradient of anything

TD(0) algorithm:

$$\Delta\theta = \alpha\delta\phi$$

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

Assume there is a J such that: $\frac{\partial J}{\partial \theta_i} = \delta\phi_i$

Then look at the second derivative:

$$\left.\begin{aligned} \frac{\partial^2 J}{\partial\theta_j\partial\theta_i} &= \frac{\partial(\delta\phi_i)}{\partial\theta_j} = (\gamma\phi'_j - \phi_j)\phi_i \\ \frac{\partial^2 J}{\partial\theta_i\partial\theta_j} &= \frac{\partial(\delta\phi_j)}{\partial\theta_i} = (\gamma\phi'_i - \phi_i)\phi_j \end{aligned}\right\} \quad \frac{\partial^2 J}{\partial\theta_j\partial\theta_i} \neq \frac{\partial^2 J}{\partial\theta_i\partial\theta_j}$$

Contradiction!

Real 2nd derivatives must be symmetric

Outline

- The promise and problems of TD learning
- Value-function approximation
- Gradient-descent methods - LMS example
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence
- Empirical results
- Conclusions

Gradient descent for TD:
What should the objective function be?

- Close to the true values?

Mean-Square Value Error

$$\begin{aligned} \mathrm{MSE}(\theta) &= \sum_s d_s \left(V_\theta(s) - V(s)\right)^2 \\ &= \| V_\theta - V \|_D^2 \end{aligned}$$

True value function

- Or close to satisfying the Bellman equation?

Mean-Square *Bellman* Error

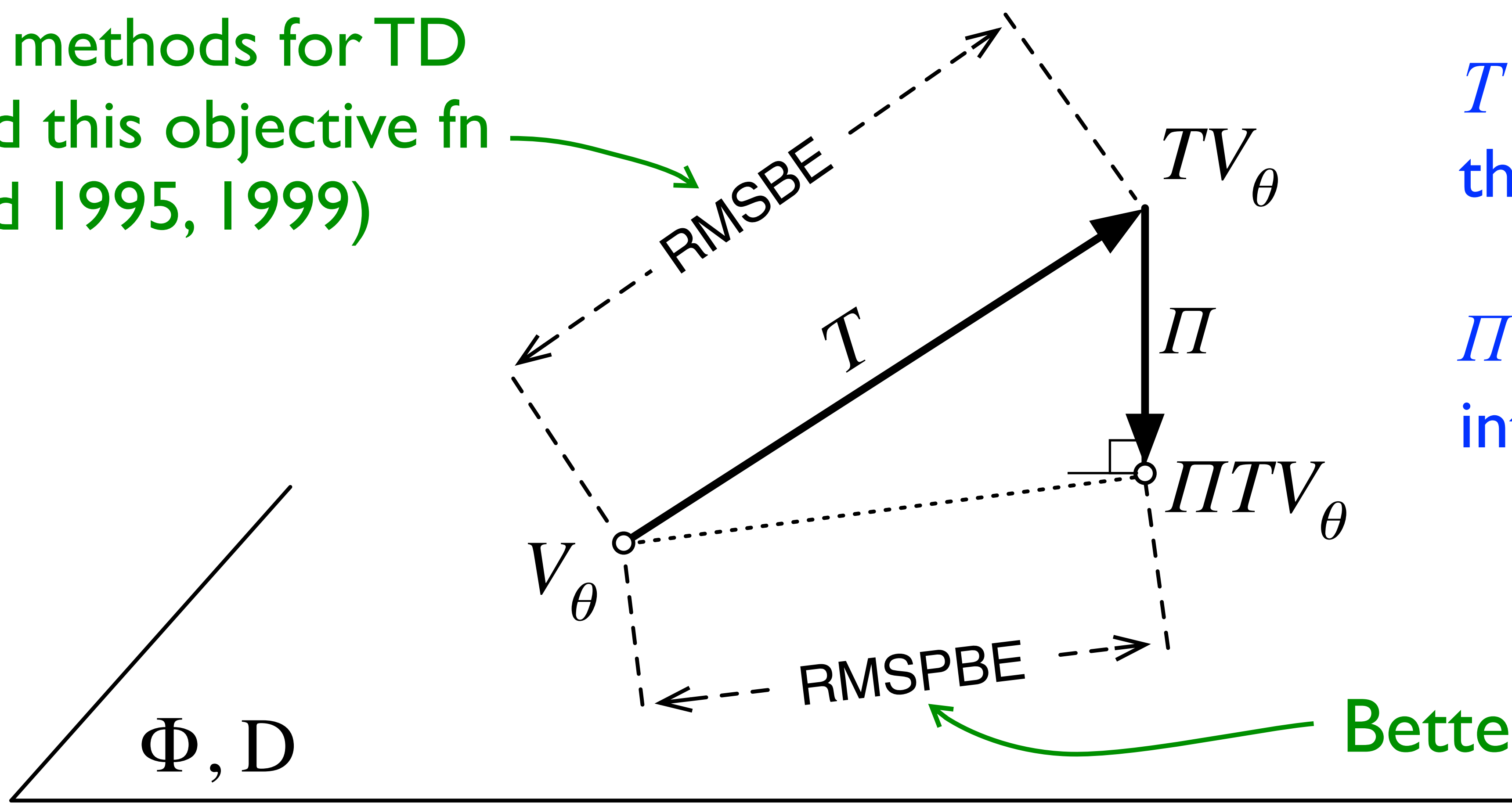
$$\mathrm{MSBE}(\theta) = \| V_\theta - TV_\theta \|_D^2$$

where T is the Bellman operator defined by

$$\begin{aligned} V &= r + \gamma P V \\ &= TV \end{aligned}$$

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

Better objective fn?

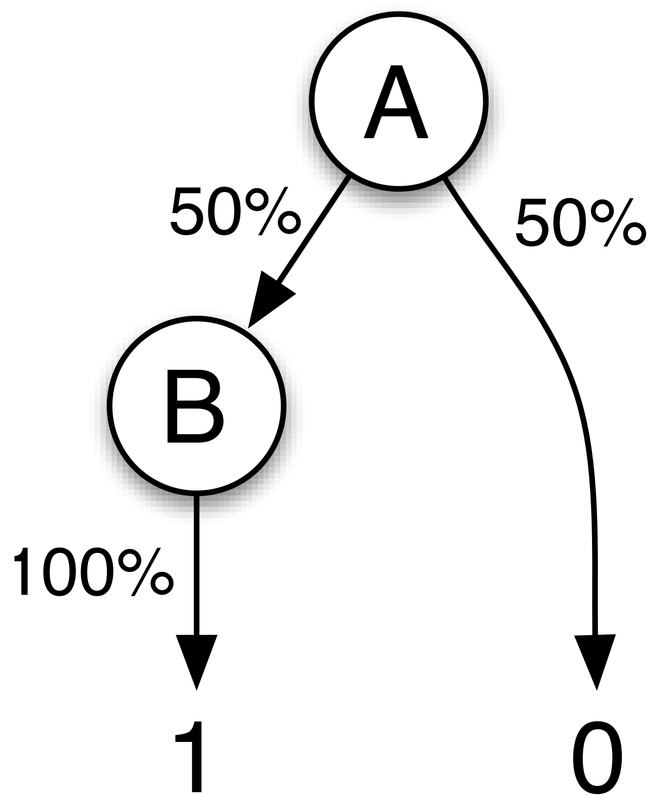
$$V_\theta = \Pi T V_\theta$$

Is the TD fix-point

The space spanned by the feature vectors,
weighted by the state visitation distribution
$D = \mathrm{diag}(d)$

Mean Square *Projected* Bellman Error (MSPBE)

A-split example (Dayan 1992)



Clearly, the true values are

$$V(A) = 0.5$$

$$V(B) = 1$$

But if you minimize the naive objective fn,

$$J(\theta) = \mathbb{E}[\delta^2],$$

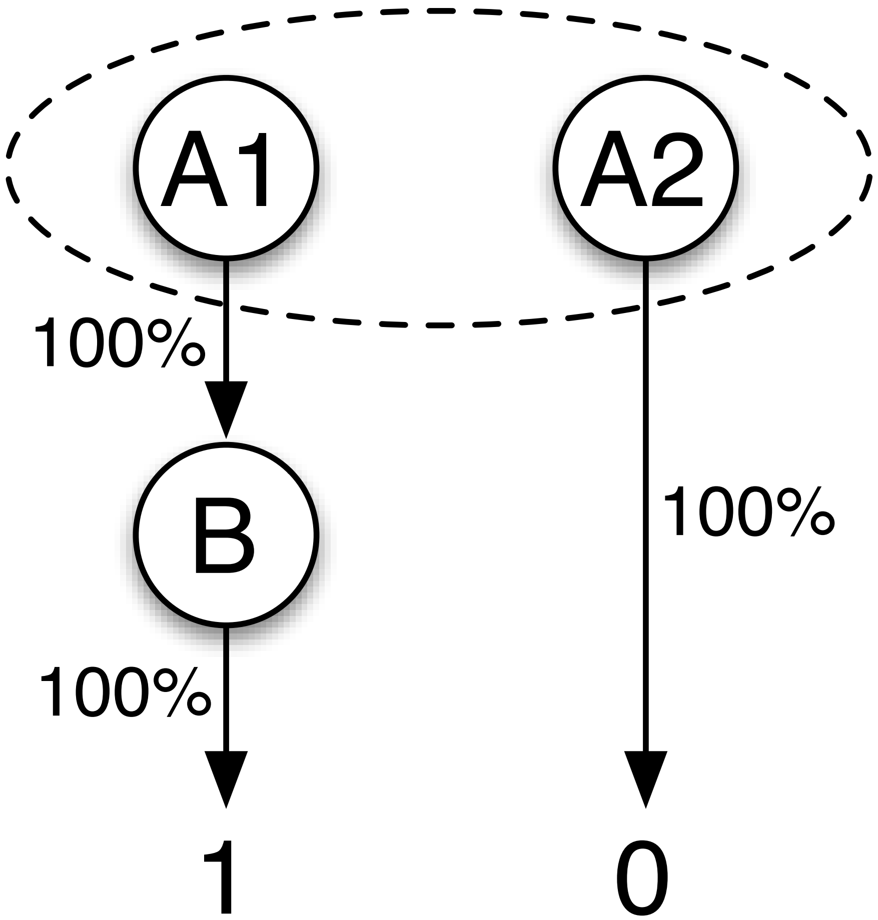
then you get the solution

$$V(A) = 1/3$$

$$V(B) = 2/3$$

Even in the tabular case (no FA)

Split-A example



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

The example appears just like the previous, and the minimum MSBE solution is

$$V(A) = 1/3$$

$$V(B) = 2/3$$

Outline

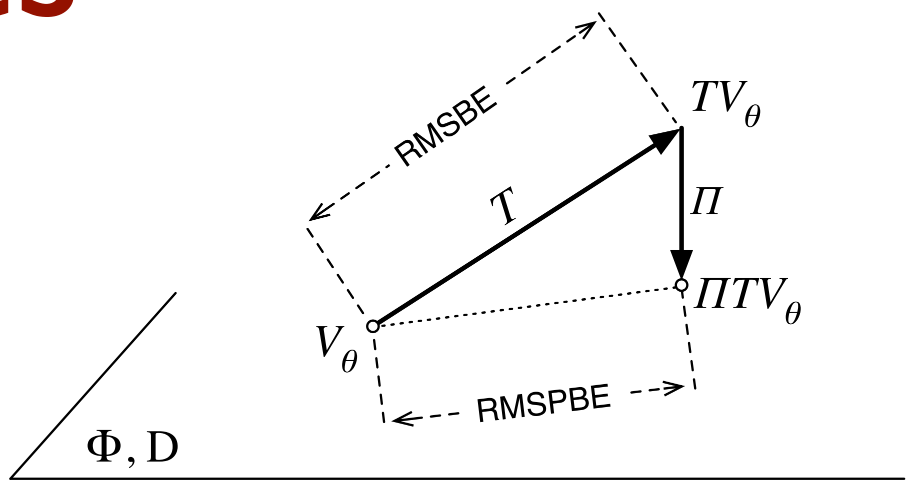
- The promise and problems of TD learning
- Value-function approximation
- Gradient-descent methods - LMS example
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence
- Empirical results
- Conclusions

Three new algorithms

- GTD, the original *gradient TD algorithm* (Sutton, Szepevari & Maei, 2008)
- GTD-2, a second-generation GTD
- TD-C, *TD with gradient correction*

- GTD(λ), GQ(λ)

First relate the geometry to the iid statistics



$$\Phi^T D(TV_\theta - V_\theta) = \mathbb{E}[\delta\phi]$$

$$\Phi^T D\Phi = \mathbb{E}[\phi\phi^T]$$

$$\begin{aligned}
&\mathrm{MSPBE}(\theta) \\
&\quad = \;\| V_\theta - \Pi T V_\theta \|_D^2 \\
&\quad = \;\| \Pi(V_\theta - TV_\theta) \|_D^2 \\
&\quad = \;(\Pi(V_\theta - TV_\theta))^\top D(\Pi(V_\theta - TV_\theta)) \\
&\quad = \;(V_\theta - TV_\theta)^\top \Pi^\top D \Pi (V_\theta - TV_\theta) \\
&\quad = \;(V_\theta - TV_\theta)^\top D^\top \Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D(V_\theta - TV_\theta) \\
&\quad = \;(\Phi^\top D(TV_\theta - V_\theta))^\top (\Phi^\top D\Phi)^{-1}\Phi^\top D(TV_\theta - V_\theta) \\
&\quad = \;\mathbb{E}[\delta\phi]^\top \, \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}[\delta\phi]\,.
\end{aligned}$$

Derivation of the GTD-2 algorithm as gradient descent in the MSPBE

$$\begin{aligned} \frac{1}{2}\nabla \mathrm{MSPBE}(\theta) &= \mathbb{E}\big[(\phi - \gamma\phi')\phi^\top\big]\,\mathbb{E}\big[\phi\phi^\top\big]^{-1}\mathbb{E}[\delta\phi] \\ &\approx \mathbb{E}\big[(\phi - \gamma\phi')\phi^\top\big]\,w, \end{aligned}$$

This is the main trick!

Assuming $w \approx \mathbb{E}\big[\phi\phi^\top\big]^{-1}\mathbb{E}[\delta\phi]$.

Sampling the expectation yields the O(n) update:

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

with

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

where

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

Gradient TD Algorithm #2

Derivation of the original GTD algorithm as gradient descent in $\mathrm{NEU}(\theta) = \mathbb{E}[\delta\phi]^\top \mathbb{E}[\delta\phi]$

$$\begin{aligned} \frac{1}{2}\nabla_\theta \mathrm{NEU}(\theta) &= \mathbb{E}[(\phi - \gamma\phi')\phi^\top]\mathbb{E}[\delta\phi] \\ &\approx \mathbb{E}[(\phi - \gamma\phi')\phi^\top] w \end{aligned}$$

Assuming $w \approx \mathbb{E}[\delta\phi]$

Sampling the expectation yields the same θ update as GTD-2, but with a different w update:

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

Derivation of the TD-C algorithm as gradient descent in the MSPBE

$$\begin{aligned}
&\frac{1}{2}\nabla \mathrm{MSPBE}(\theta) \\
&= \mathbb{E}\left[(\phi - \gamma\phi')\phi^\top\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}[\delta\phi] \\
&= \left(\mathbb{E}\left[\phi\phi^\top\right] - \gamma\mathbb{E}\left[\phi'\phi^\top\right]\right) \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}[\delta\phi] \\
&= \mathbb{E}[\delta\phi] - \gamma\mathbb{E}\left[\phi'\phi^\top\right] \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}[\delta\phi] \\
&\approx \mathbb{E}[\delta\phi] - \gamma\mathbb{E}\left[\phi'\phi^\top\right] w,
\end{aligned}$$

Assuming $w \approx \mathbb{E}\left[\phi\phi^\top\right]^{-1} \mathbb{E}[\delta\phi]$.

Sampling the expectation yields

$$\theta \leftarrow \theta + \underbrace{\alpha\delta\phi}_{\text{conventional TD(0)}} - \underbrace{\alpha\gamma\phi'(\phi^\top w)}_{\text{gradient correction term}}$$

With w updated as in GTD-2

Outline

- The promise and problems of TD learning
- Value-function approximation
- Gradient-descent methods - LMS example
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence (sketch and remarks)
- Empirical results

Convergence theorems

- For arbitrary P and d
- All algorithms converge w.p.1 to the TD fix-point:

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

- GTD, GTD-2 converges at one time scale

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

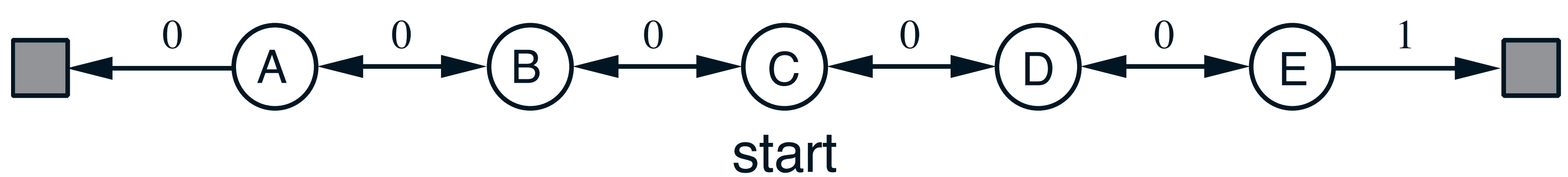
- TD-C converges in a two-time-scale sense

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

Outline

- The promise and problems of TD learning
- Value-function approximation
- Gradient-descent methods - LMS example
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence
- Empirical results
- Conclusions

Random walk problem (on-policy)

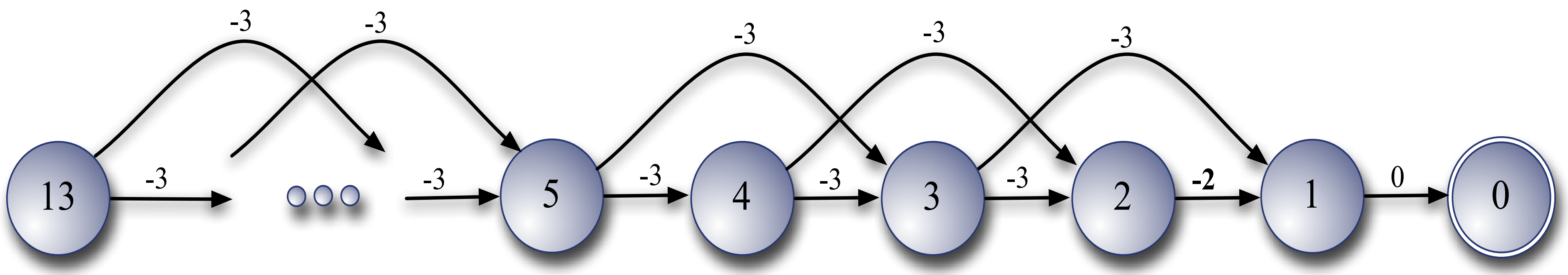


3 different feature representations.

- 5 tabular features
- 5 inverted-tabular features
- 3 features (genuine FA)

Boyan chain problem (on-policy)

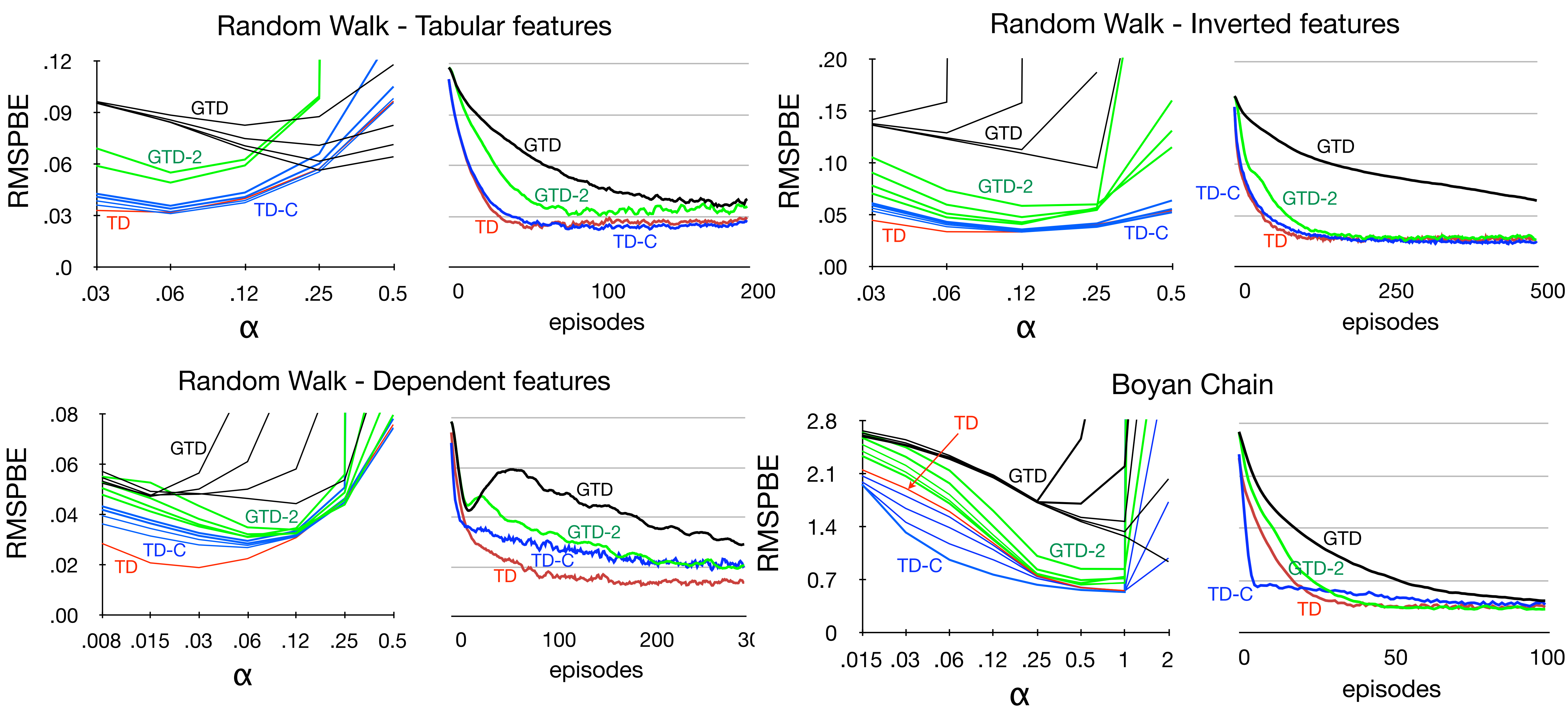
Boyan 1999



$$\begin{bmatrix}0\\0\\1\\0\end{bmatrix} \begin{bmatrix}0\\0\\0.75\\0.25\end{bmatrix} \begin{bmatrix}0\\0\\0.5\\0.5\end{bmatrix} \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

13 states, 4 features
Exact solution possible

Summary of empirical results on small problems



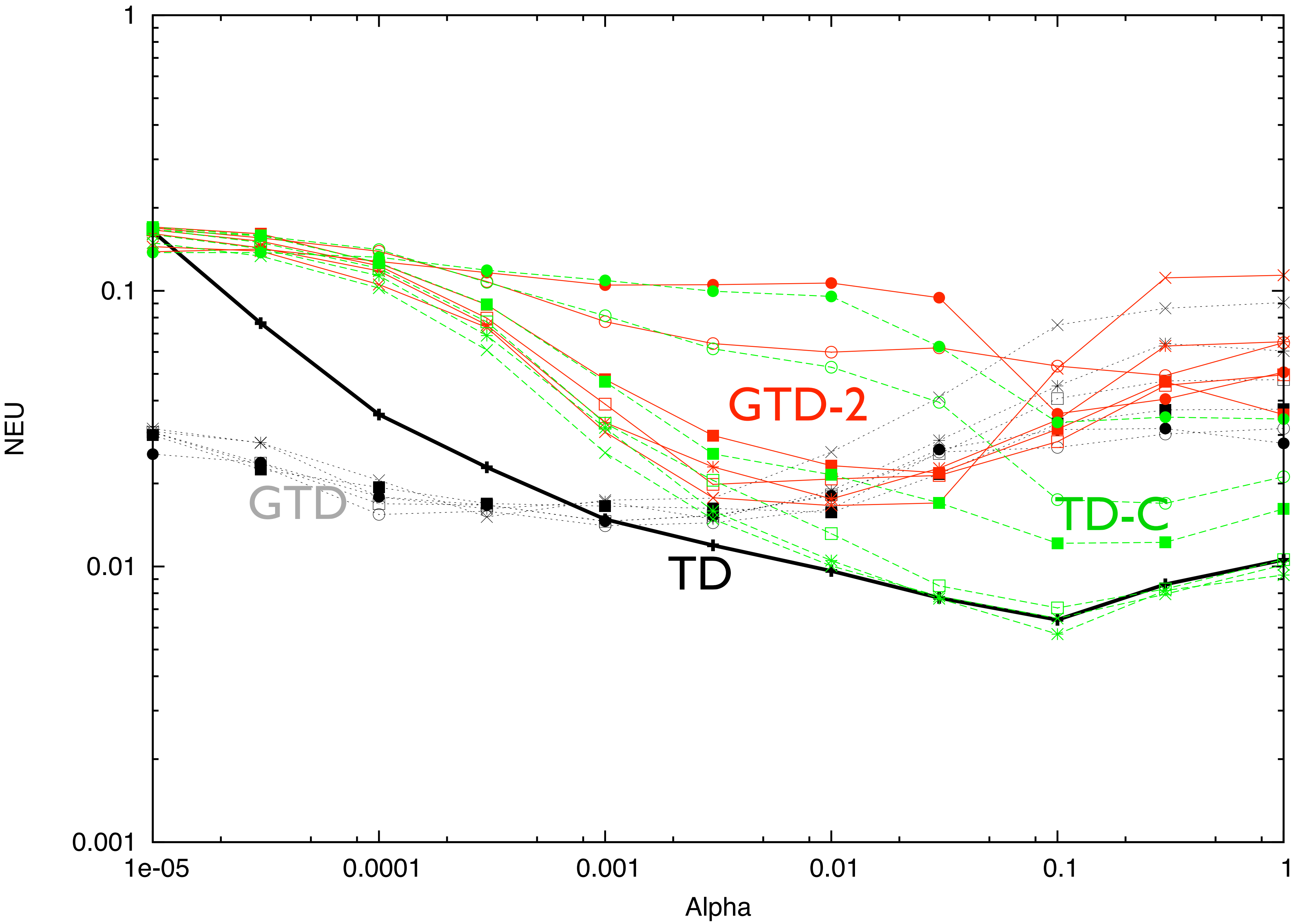
TD, TD-C > GTD-2 > GTD

Sometimes TD > TD-C

Computer Go experiment

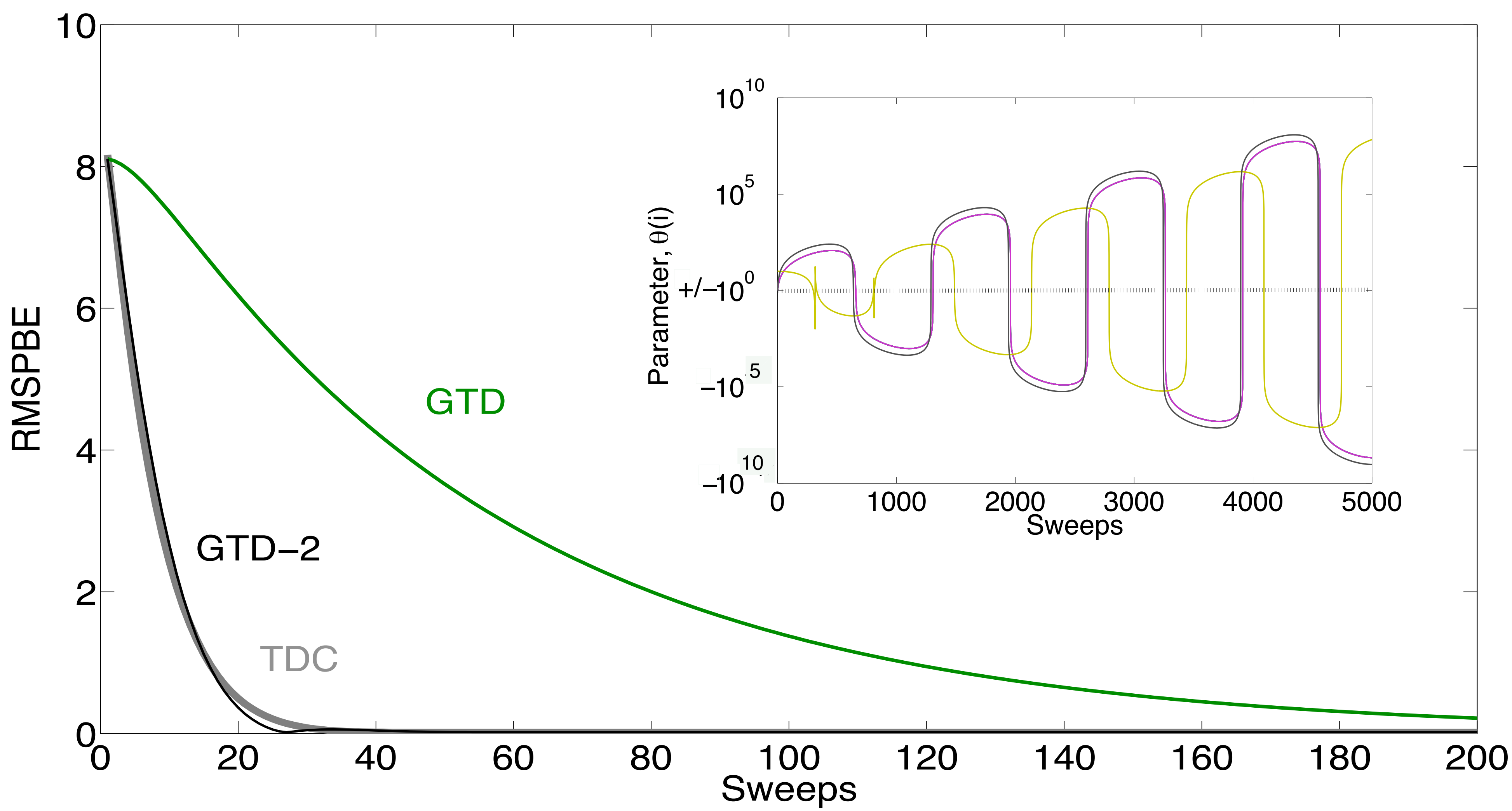
- Learn the value function (probability of winning) for 5x5 Go
- Lots of features, linearly combined, then passed through a logistic non-linearity
- An established experimental testbed
- Tried the various algorithms
- Results are still preliminary

Computer Go results



TD-C, TD > GTD, GTD-2

Off-policy result:
Baird's counter-example



Gradient algorithms converge. TD diverges.

Conclusions

- The first O(n) methods to work off-policy (and meet all the other desiderata)
- New methods (GTD-2 and TD-C) are much faster than original GTD
- Not clear yet whether or not TD-C is sufficiently close to TD speed on on-policy problems
- But it is at least a major step closer. And it works off-policy