Gradient Temporal-Difference Learning Algorithms

Rich Sutton (Alberta)

Hamid Maei (Alberta)

Doina Precup (McGill)
Shalabh Bhatnagar (IIS Bangalore)
Csaba Szepesvari (Alberta)
Eric Wiewiora (Alberta)
David Silver (UCL)

The problem

- Learning to predict the outcome of a way of behaving
  - from fragments of its execution
  - in a practical, scalable way

➡ Off-policy TD learning with linear function approximation

Outline

- The promise of TD learning
- Value-function approximation
- Gradient-descent methods
- Objective functions for TD
- Gradient-descent derivation of new algorithm
- Proof of convergence (sketch and remarks)
- 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 in a crowd

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*

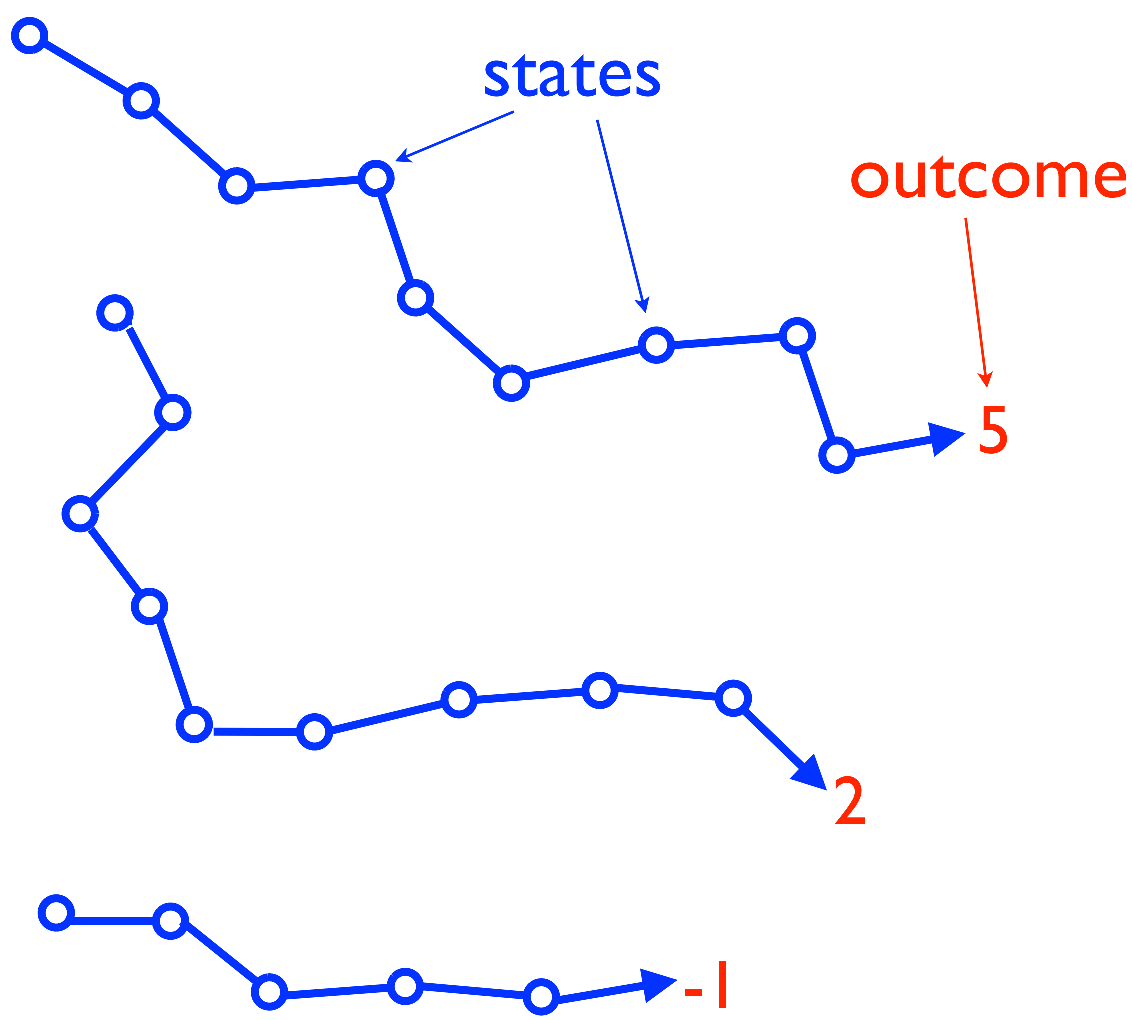
Off-policy learning

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

Outline

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

Value-function approximation from sample trajectories



- True values:

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

- Estimated values:

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

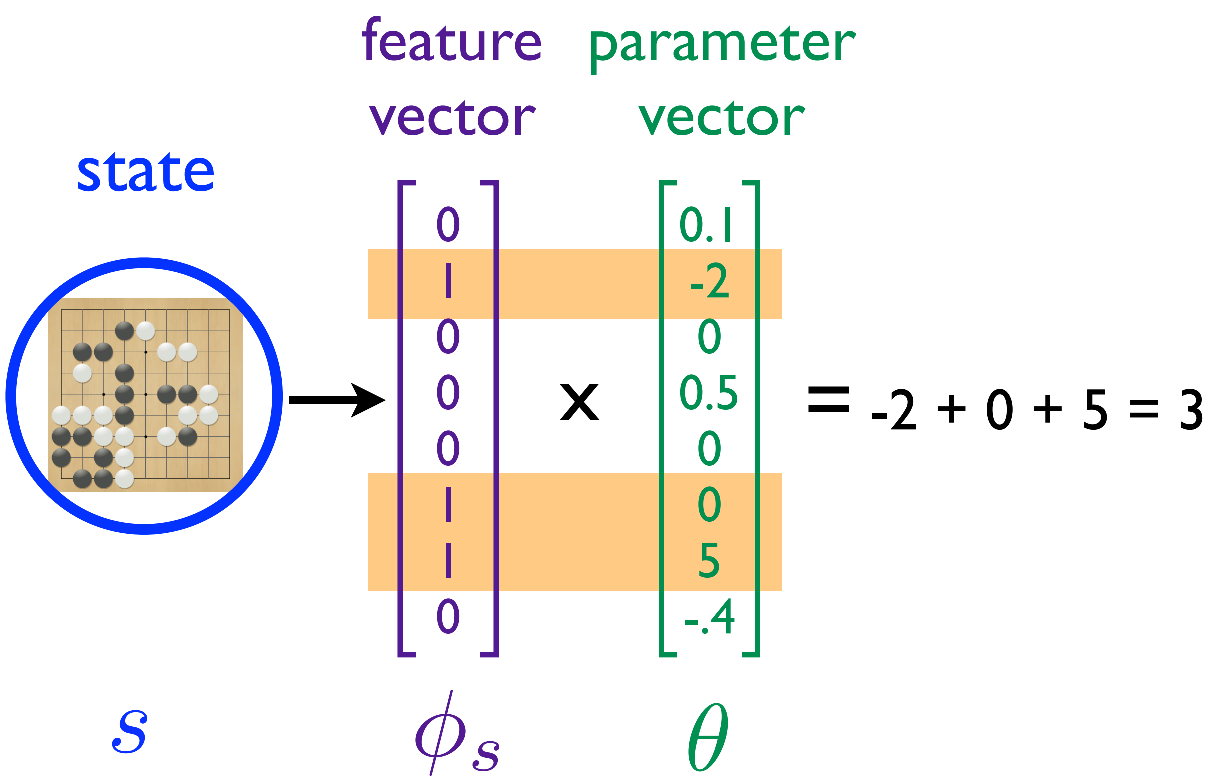
- Linear approximation:

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

modifiable parameter vector

feature vector for state s

Value-function approximation from sample trajectories



- True values:

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

- Estimated values:

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

- Linear approximation:

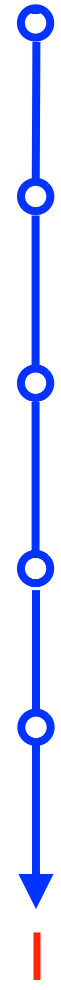
$$V_\theta(s) = \theta^\top \phi_s, \qquad \phi_s \in \Re^n$$

modifiable parameter vector

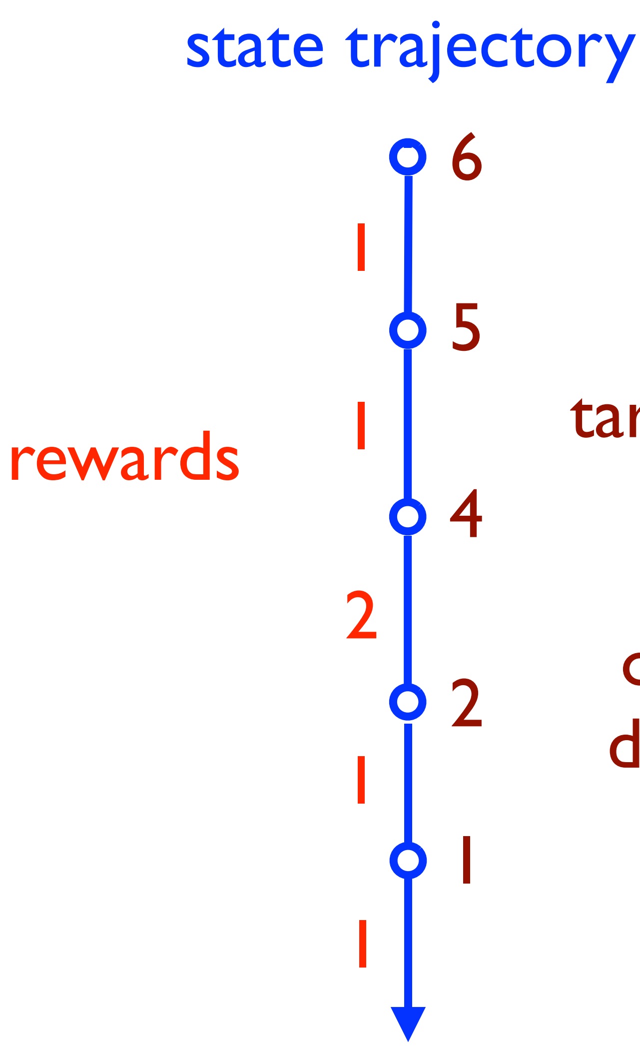
feature vector for state s

From terminal outcomes to per-step rewards

state trajectory



From terminal outcomes to per-step rewards



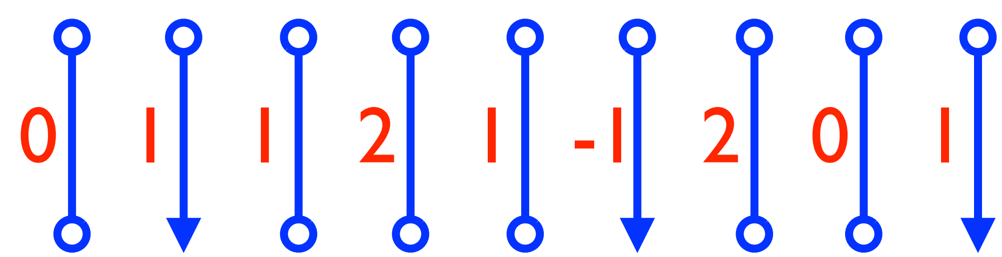
target values (returns)
= sum of future
rewards until end
of episode, or until
discounting horizon

- 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



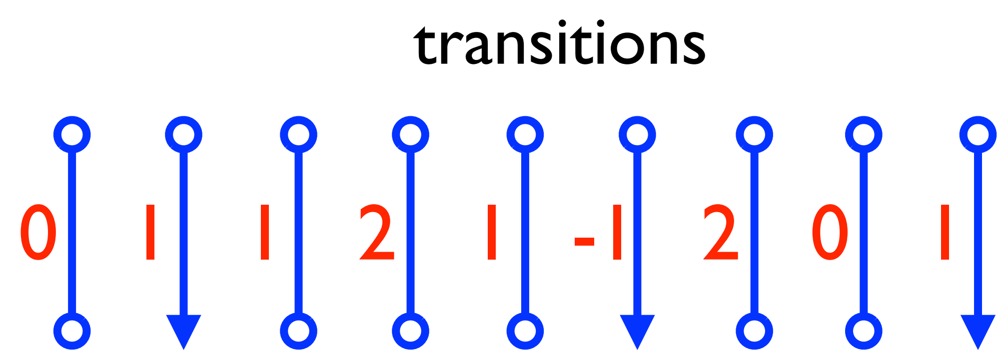
Training set is now a bag of transitions
Select from them i.i.d.
(independently, identically distributed)

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

TD(0) algorithm:
$$\delta = r + \gamma\theta^\top\phi' - \theta^\top\phi$$
$$\theta \leftarrow \theta + \alpha\delta\phi$$

TD methods operate on individual transitions

d_s - distribution of first state s
b_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')

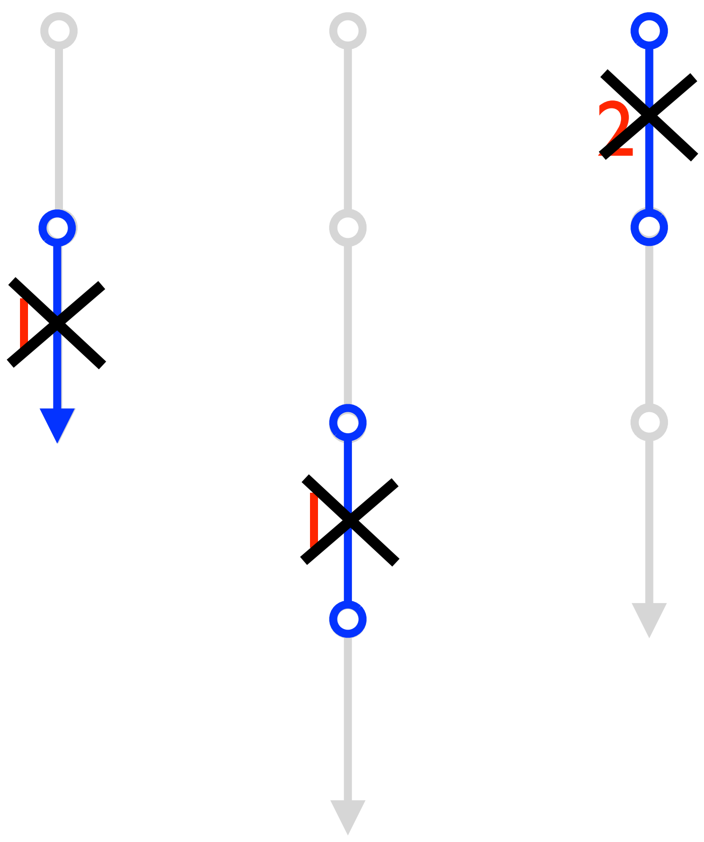
TD(0) algorithm:

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

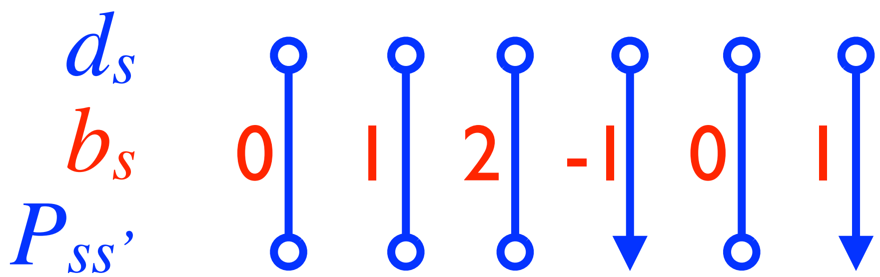
$$\theta \leftarrow \theta + \alpha\delta\phi$$

Off-policy training

trajectories



transitions

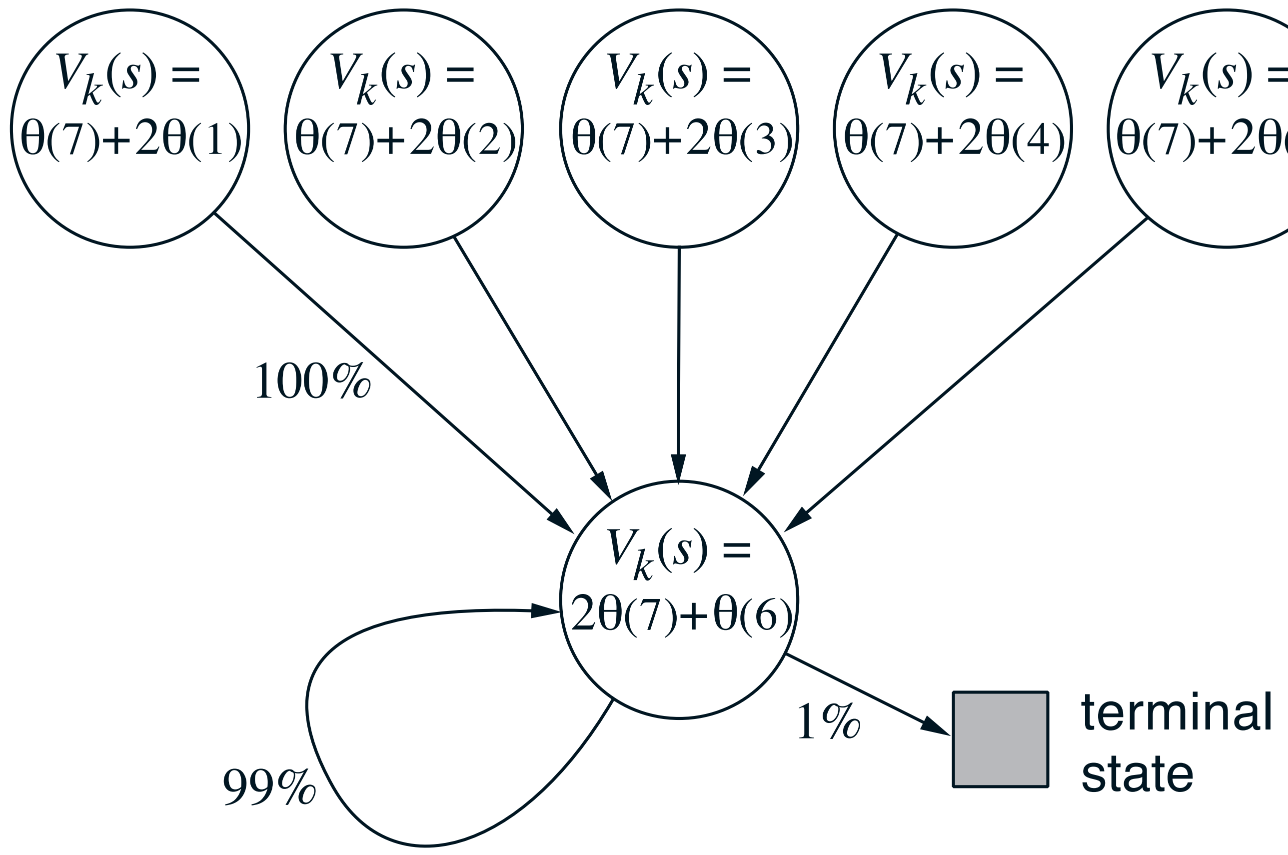


P and *d* are no longer linked

TD(0) may diverge!

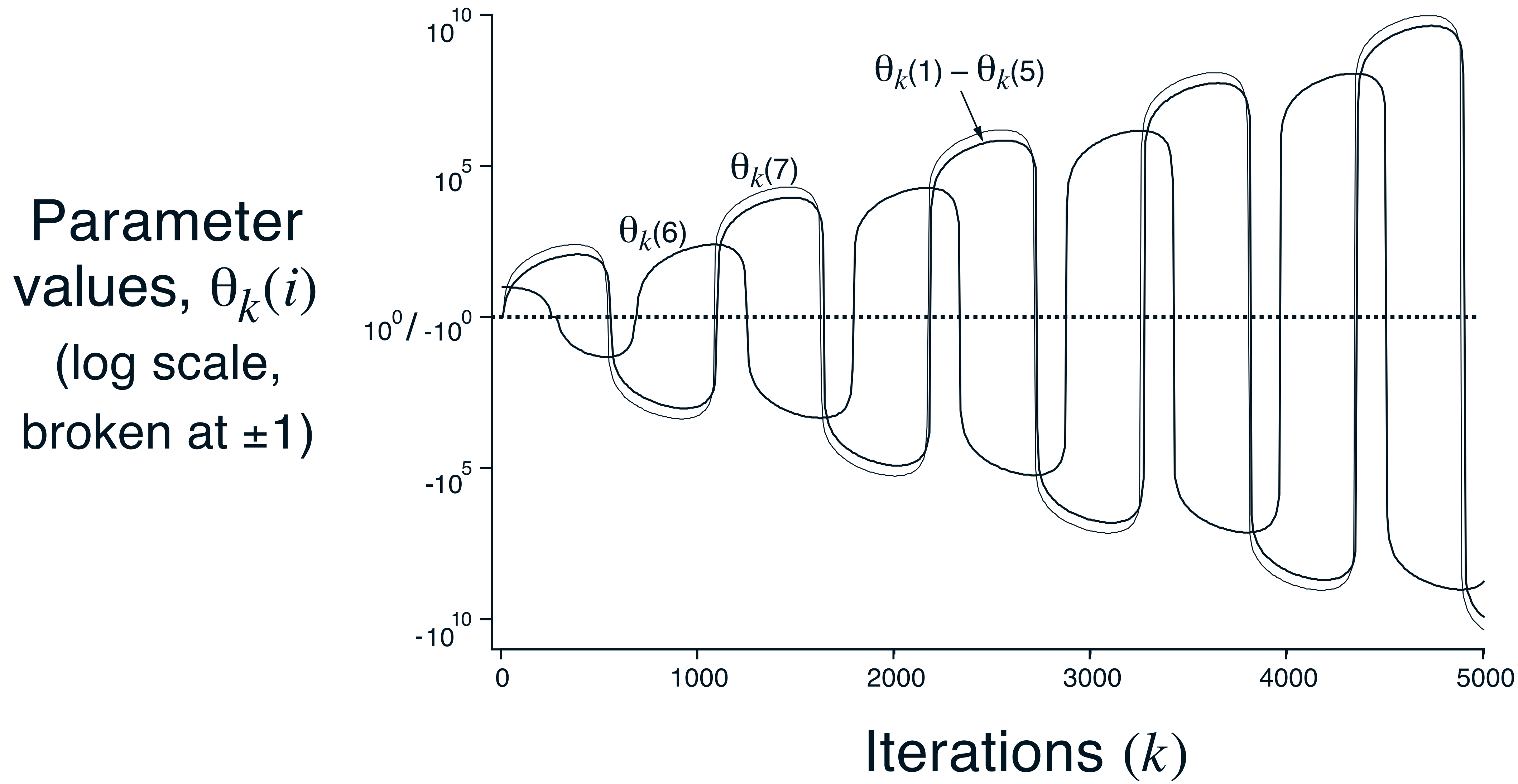
Baird's counter-example

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



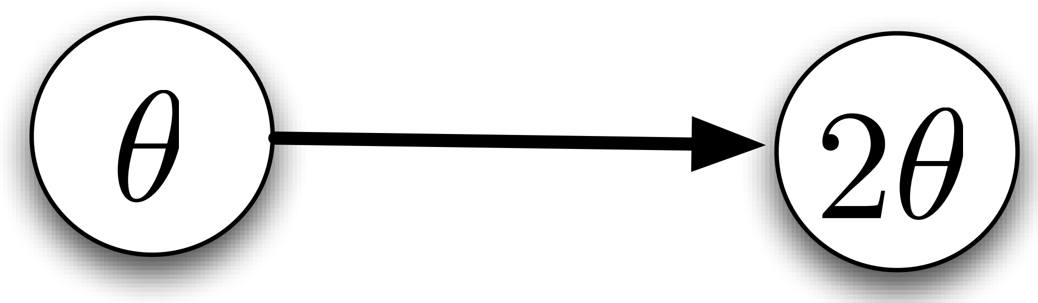
$r = 0$
on all transitions

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 of TD learning
- Value-function approximation
- **Gradient-descent methods**
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence (sketch and remarks)
- 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" $\nabla_\theta J_t(\theta)$ 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 of TD learning
- Value-function approximation
- Gradient-descent methods
- Objective functions for TD
- GD derivation of new algorithms
- Proofs of convergence (sketch and remarks)
- Empirical results
- Conclusions

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

- Close to the true values?

Mean-Square Error

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

True value function

- Or close to satisfying the Bellman equation?

Mean-Square *Bellman* Error

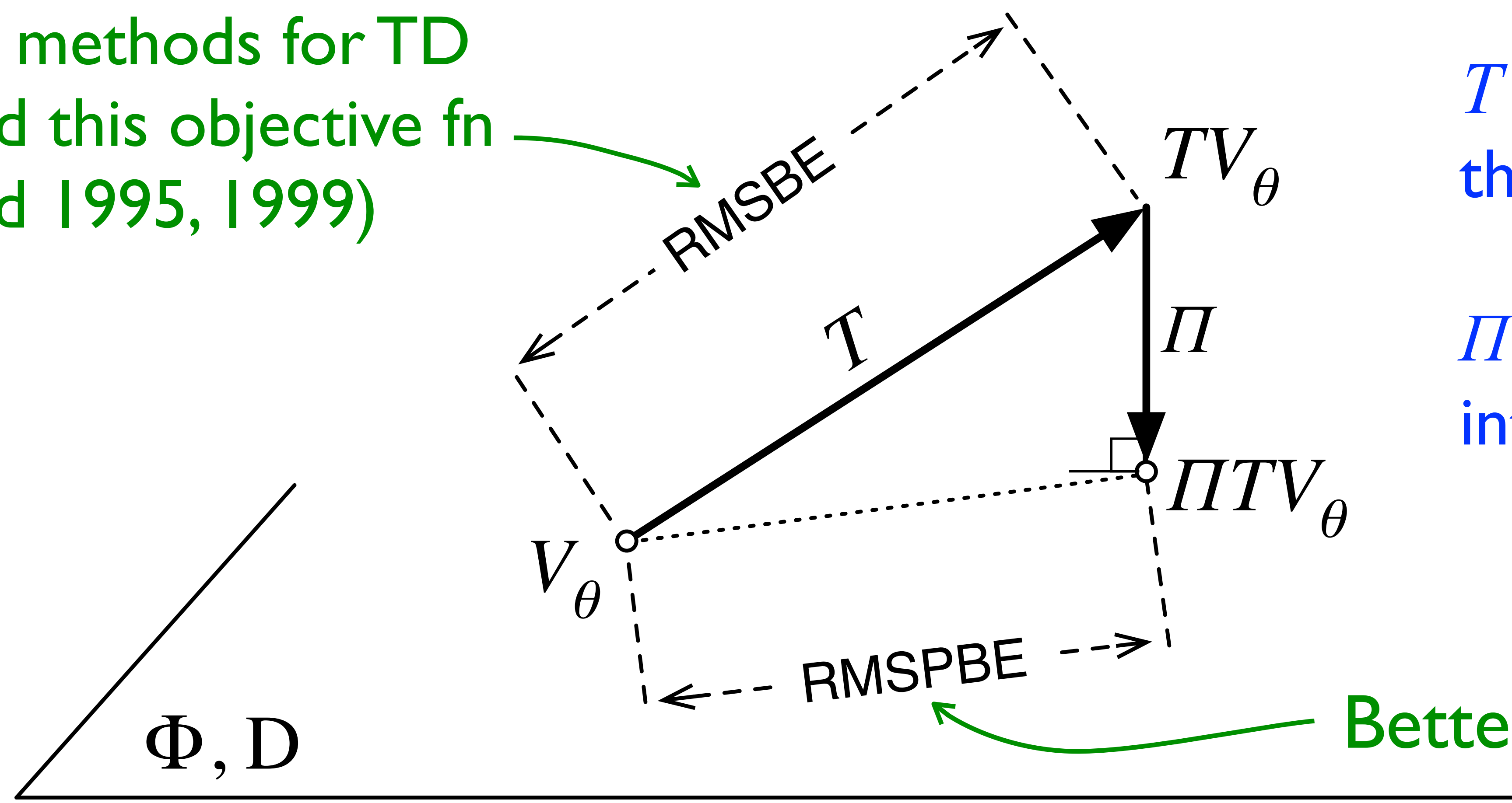
$$\mathrm{MSBE}(\theta) = \| V_\theta - TV_\theta \|_D^2$$

where T is the Bellman operator defined by

$$V = r + \gamma PV$$
$$= TV$$

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?

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

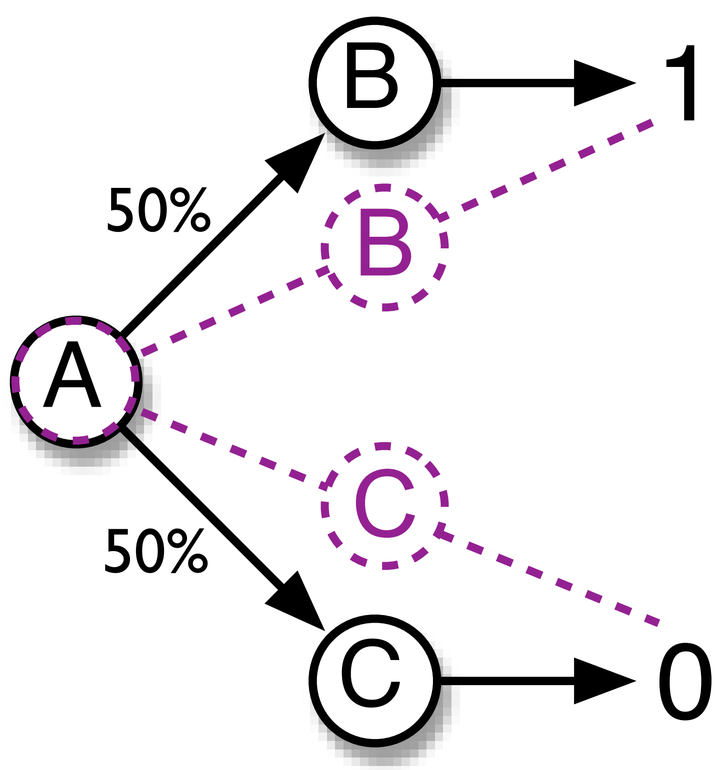
Is the TD fix-point

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

Mean Square *Projected* Bellman Error (MSPBE)

Backward-bootstrapping example (1)

(Dayan 1992)



Clearly, the true values are

$$V(B) = 1 \qquad V(C) = 0$$

$$V(A) = 0.5$$

But if you minimize the expected TD error:

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

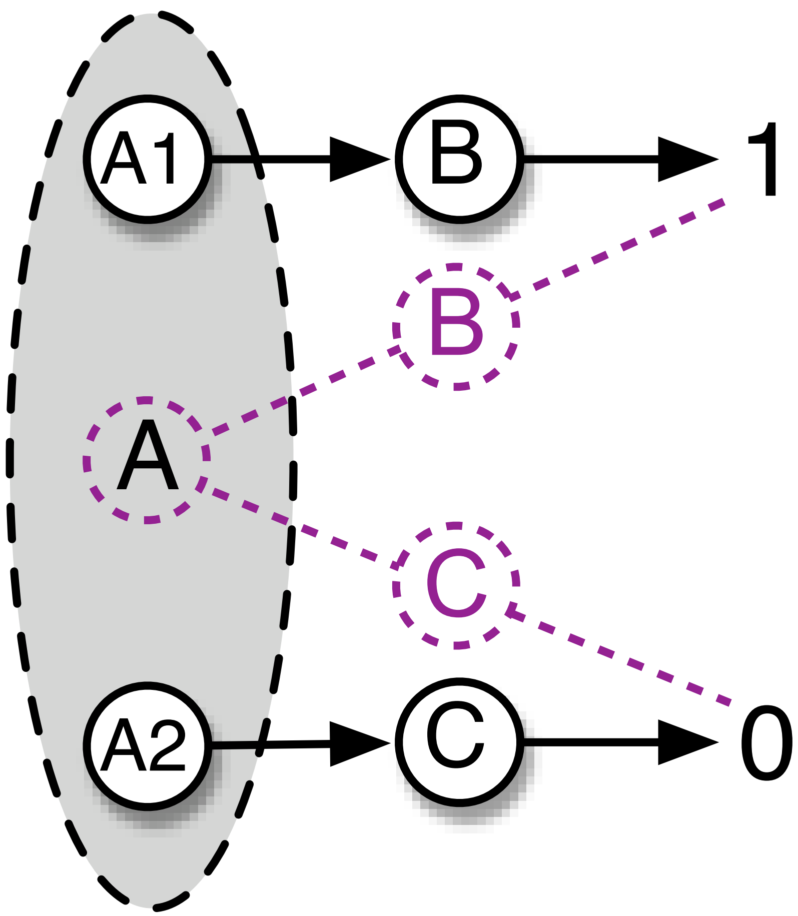
then you get the solution

$$V(B) = 0.75 \quad V(C) = 0.25$$

$$V(A) = 0.5$$

Even in the tabular case (no FA)

Backward-bootstrapping example (2)



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, but now the minimum mean-squared *Bellman error* solution is

$$V(B) = 0.75 \quad V(C) = 0.25$$

$$V(A) = 0.5$$

Outline

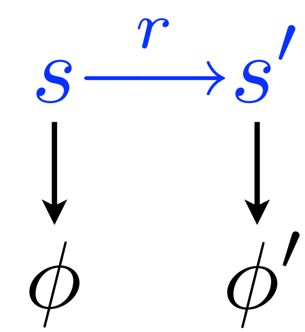
- The promise of TD learning
- Value-function approximation
- Gradient-descent methods
- Objective functions for TD
- **Gradient-descent derivation of new algorithms**
- Proofs of convergence (sketch and remarks)
- Empirical results
- Conclusions

Three new algorithms

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

- (also GQ(λ) and Greedy-GQ)

Derivation of the TDC algorithm



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

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

$$s \xrightarrow{r} s'$$
$$\downarrow \qquad \downarrow$$
$$\phi \qquad \phi'$$

- update two parameters

TD(0)

with gradient correction

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

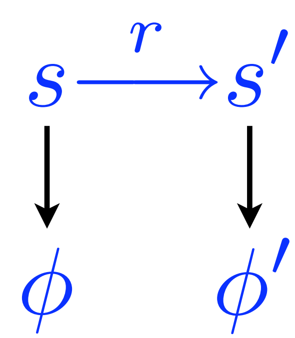
$$w \leftarrow w + \beta(\delta - \phi^\top w)\phi$$

- where, as usual

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

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, as usual

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

Outline

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

Stability and convergence

There exists a projected-Bellman-error objective function

$$J(\theta) = \| V_\theta - \Pi T V_\theta \|_D^2$$

vector of values, one per state

generalized Bellman operator

projection back into the space of representable functions

such that

$$E[\Delta\theta] = -\alpha \nabla_\theta J(\theta)$$

which guarantees convergence to $J(\theta) = 0$ (under step-size conditions)

Convergence theorems

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

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

- for GTD and GTD-2

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

- for TDC

$$\alpha = \frac{\beta}{\eta} \longrightarrow 0, \qquad \eta > \max(0, \lambda_{\max})$$

A little more theory

$$\begin{aligned}
\Delta\theta \propto \delta\phi &= \left(r + \gamma\theta^\top\phi' - \theta^\top\phi\right)\phi \\
&= \theta^\top\left(\gamma\phi' - \phi\right)\phi + r\phi \\
&= \phi\left(\gamma\phi' - \phi\right)^\top\theta + r\phi \\
\mathbb{E}\left[\Delta\theta\right] &\propto -\underbrace{\mathbb{E}\left[\phi\left(\phi - \gamma\phi'\right)^\top\right]}_{A}\theta + \underbrace{\mathbb{E}\left[r\phi\right]}_{b} \\
\mathbb{E}\left[\Delta\theta\right] &\propto -A\theta + b
\end{aligned}$$

convergent if A is pos. def.

therefore, at the TD fixpoint:

$$\begin{aligned}
A\theta^* &= b \\
\theta^* &= A^{-1}b
\end{aligned}$$

LSTD computes this directly

$$-\frac{1}{2}\nabla_\theta \mathrm{MSPBE} = -\underbrace{A^\top C^{-1} A}_{\text{always pos. def.}}\theta - b)$$

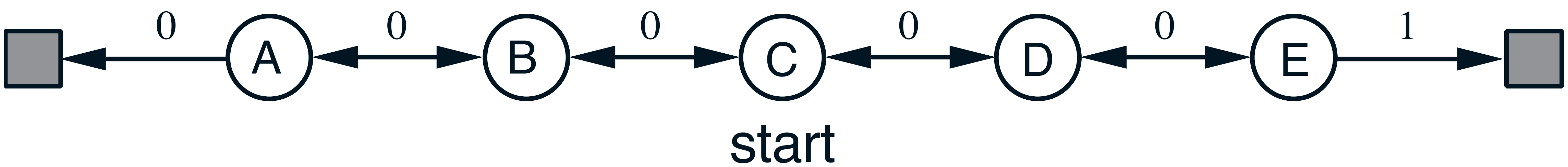
$$C = \mathbb{E}\left[\phi\phi^\top\right]$$

covariance matrix

Outline

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

Random walk problem (on-policy)

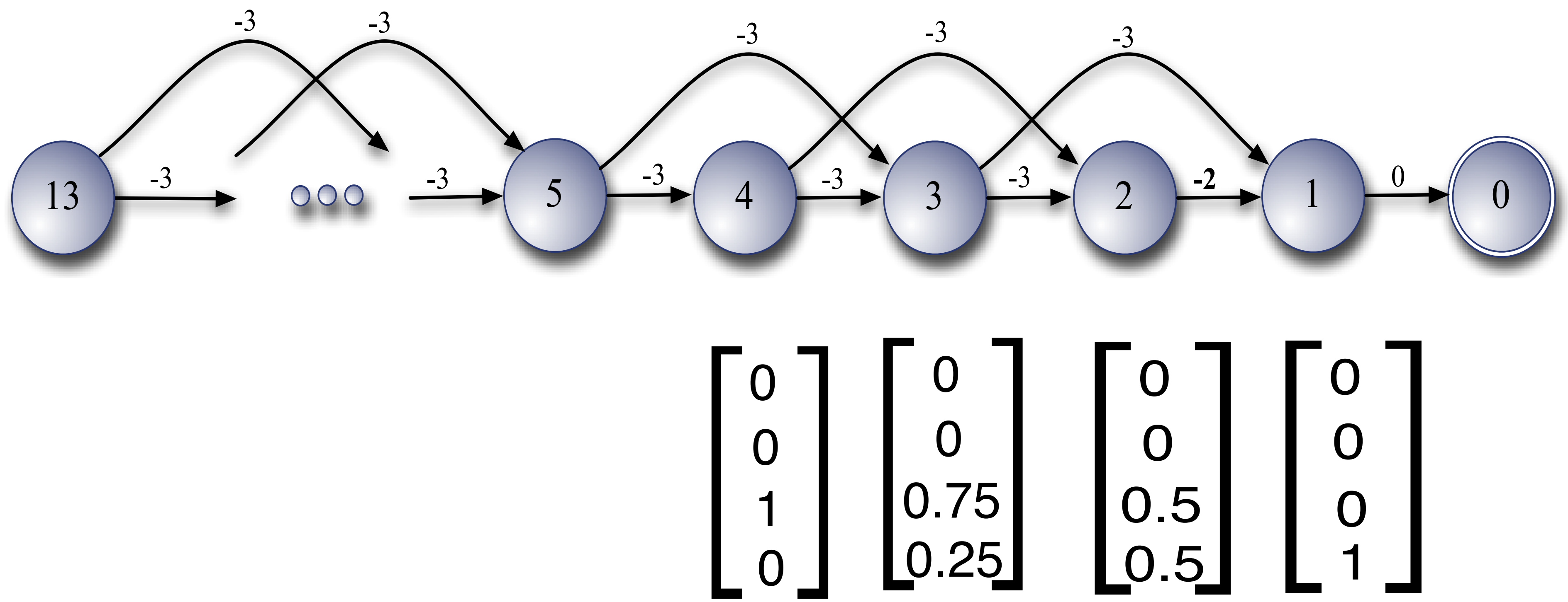


3 different feature representations.

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

Boyan chain problem (on-policy)

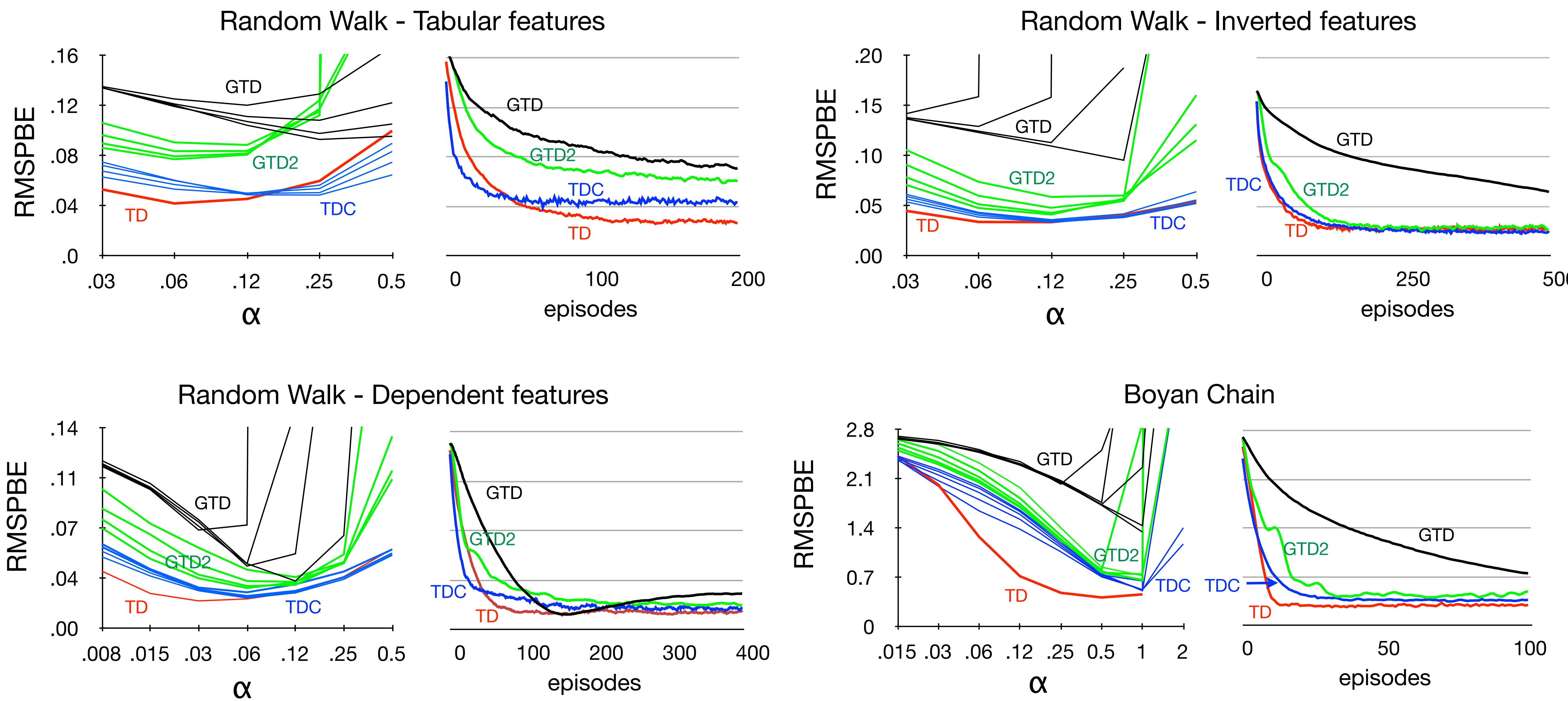
Boyan 1999



13 states, 4 features

Exact solution possible

Summary of empirical results on small problems



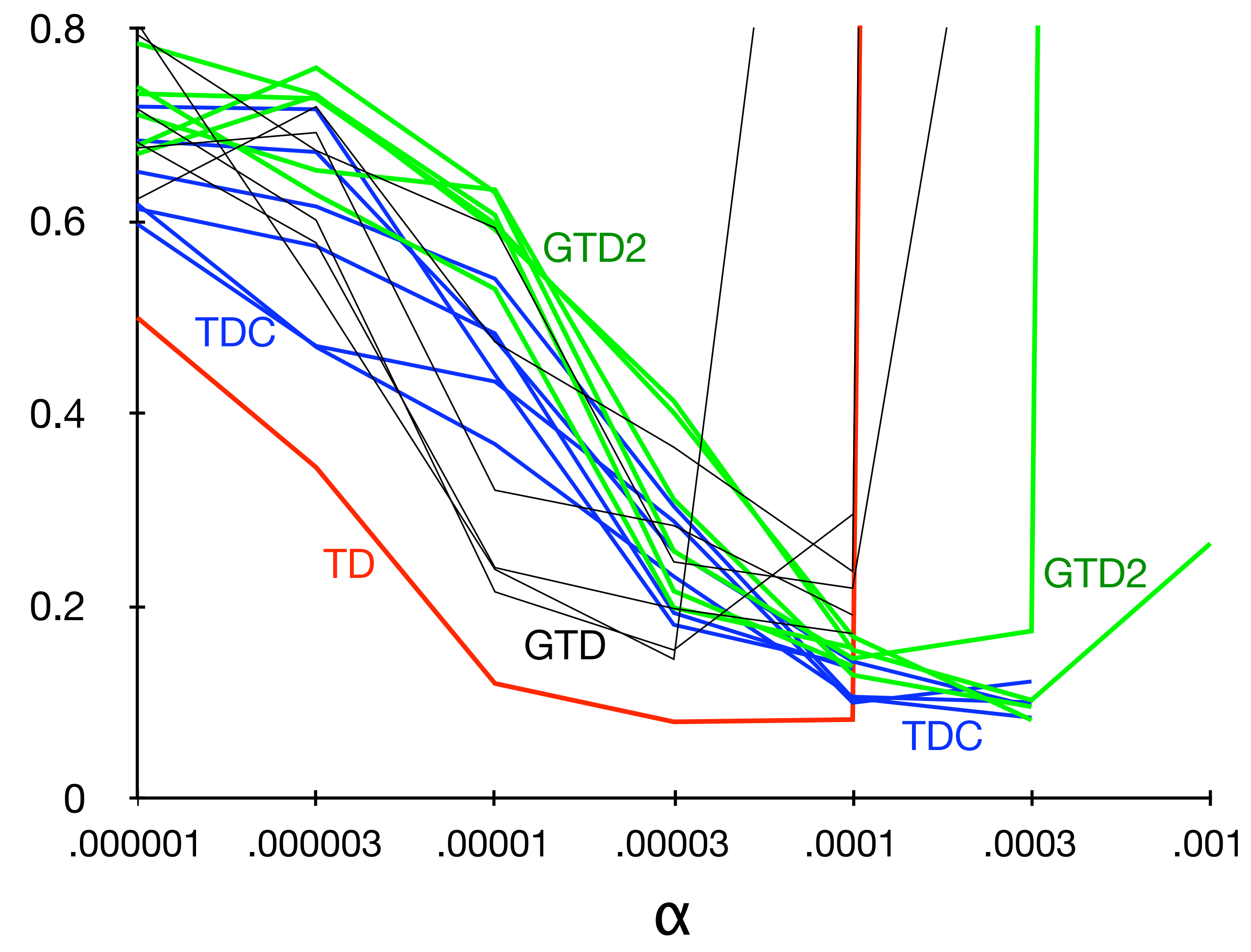
TD, TDC > GTD-2 > GTD

Sometimes TD > TDC

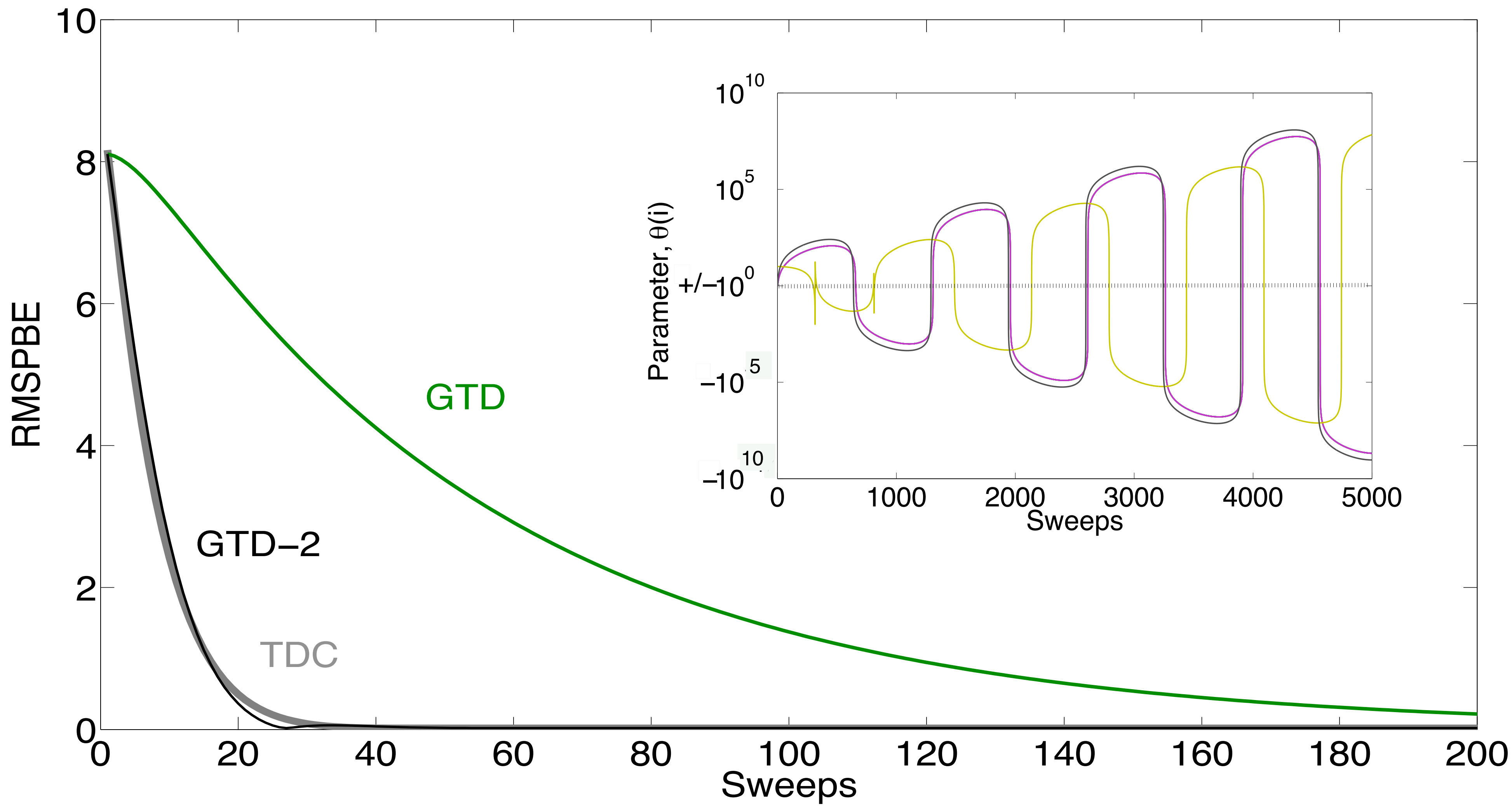
Computer Go experiment

- 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

$\| \mathbb{E}[\Delta\theta_{TD}] \|$



Off-policy result: Baird's counter-example



Gradient algorithms converge. TD diverges.

Further results with new *gradient-descent TD* methods

- Convergence with nonlinear function approximators (e.g., neural networks)
- Extensions to a very general form – GQ(λ)
  - action values (Q)
  - eligibility traces with state-dependent λ
  - state-dependent termination function γ
  - arbitrary behaviour policy
- First convergence result for the control case (changing target policy π) – Greedy-GQ

Specific conclusions

- TDC is roughly the same efficiency as conventional TD on on-policy problems
- and is guaranteed convergent under general off-policy training as well
- the key ideas appear to extend quite broadly

General conclusions

- The new gradient TD algorithms are a breakthrough in RL, solving two open probs:
  - convergent O(n) off-policy learning
  - nonlinear TD
- Function approximation in RL is now nearly as straightforward as supervised learning
  - the curse of dimensionality is broken
  - general learning from interaction is now practical
- Learning rate can probably still be improved; there are yet new algorithms coming