A Worst-Case Comparison Between Temporal Difference and Residual Gradient with Linear Function Approximation

Lihong Li

Rutgers Laboratory for Real-Life Reinforcement Learning (RL^3)

Presented at ICML 2008
Helsinki, Finland
July 2008

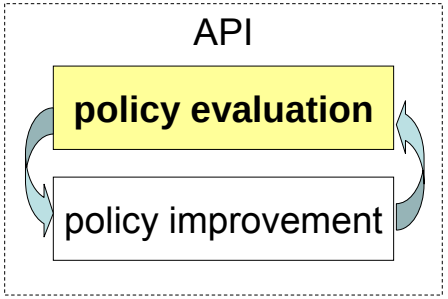
RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

Outline

- The sequential online learning model
- Related work
- Theoretical results
- Empirical evidence
- Conclusions

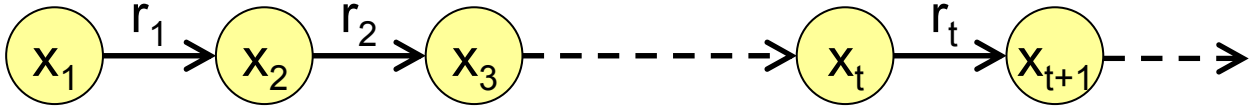
Policy Evaluation

- The problem: to predict long-term effect given short-term (immediate) feedback
- Critical in (approximate) policy iteration
  - Classic policy iteration (Howard, 60)
  - λ-policy iteration (Bertsekas & Ioffe, 96)
  - Least-squares policy iteration (Lagoudakis & Parr, 03)
- Curse of dimensionality (Bellman, 57)
  - Approximation is necessary
  - Linear function approximation
- Incremental algorithms
  - Temporal difference (Sutton, 88)
  - Residual gradient (Baird, 95)



online
fast
incremental

Sequential Online Learning Model: Notation



[Imagine $x_i = \phi(s_i)$ in RL]

Value (return) at t: $y_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$ "Bellman equation": $y_t = r_t + \gamma y_{t+1}$

For an algorithm w/ linear func. approx.

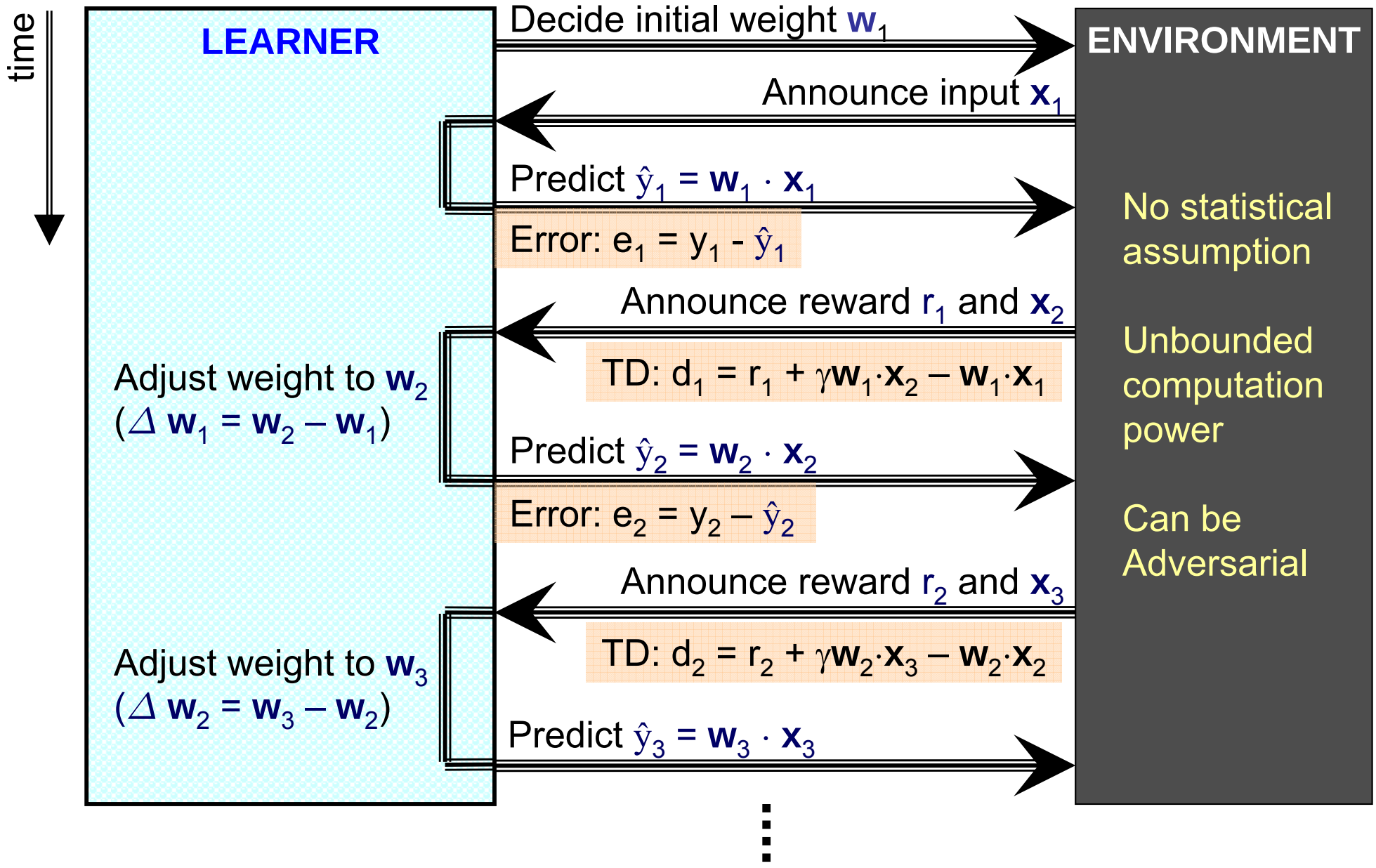
Prediction: $\hat{y}_t = \mathrm{w}_t \cdot \mathrm{x}_t$

Prediction error: $e_t = y_t - \hat{y}_t$

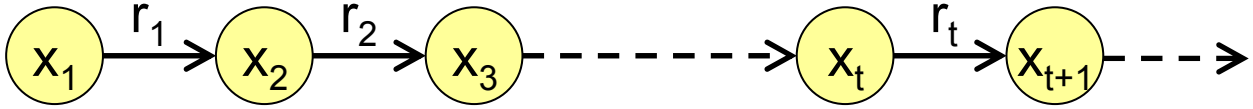
Temporal difference: $d_t = r_t + \gamma w_t \cdot x_{t+1} - w_t \cdot x_t$

Total losses: $\ell_P = \sum_t e_t^2$ and $\ell_{TD} = \sum_t d_t^2$

Learning Protocol



Sequential Online Learning Model: Notation



[Imagine $x_i = \phi(s_i)$ in RL]

Value (return) at t: $y_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$

"Bellman equation": $y_t = r_t + \gamma y_{t+1}$

For an algorithm w/ linear func. approx.

Prediction: $\hat{y}_t = \mathrm{w}_t \cdot \mathrm{x}_t$

Prediction error: $e_t = y_t - \hat{y}_t$

Temporal difference: $d_t = r_t + \gamma w_t \cdot x_{t+1} - w_t \cdot x_t$

Total losses: $\ell_P = \sum_t e_t^2$ and $\ell_{TD} = \sum_t d_t^2$

compared to the best (in hindsight) weight **u**

$$\ell_P^{\mathrm{u}} = \sum_t (y_t - \mathrm{u} \cdot \mathrm{x}_t)^2$$

$$\ell_{TD}^{\mathrm{u}} = \sum_t (r_t + \gamma \mathrm{u} \cdot \mathrm{x}_{t+1} - \mathrm{u} \cdot \mathrm{x}_t)^2$$

Goal: to upper bound $\ell_P / \ell_P^{\mathrm{u}}$ and $\ell_{TD} / \ell_{TD}^{\mathrm{u}}$ for *any* u

Why the Two Metrics?

- Prediction error (e_t) :
  - Measures prediction accuracy
  - Directly affects policy quality in approximate policy iteration (Bertsekas & Tsitsiklis, 96)

- Temporal difference (d_t):
  - Measures prediction consistency
  - Related to Bellman error
  - More robust in policy improvement (Munos, 03)

Three Online-Learning Algorithms

"Bellman equation": $y_t = r_t + \gamma y_{t+1}$

Temporal difference: $d_t = r_t + \gamma w_t \cdot x_{t+1} - w_t \cdot x_t$

- TD(0) (Sutton, 88)

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t + \eta d_t \mathrm{x}_t$$

- TD*(0) (Schapire & Warmuth, 96)

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t + \frac{\eta}{1 - \gamma\eta\, \mathrm{x}_t \cdot \mathrm{x}_{t+1}} d_t \mathrm{x}_t$$

- RG (Baird, 95)

$$\mathrm{w}_{t+1} \leftarrow \mathrm{w}_t + \eta d_t (\mathrm{x}_t - \gamma \mathrm{x}_{t+1})$$

Outline

- The sequential online learning model
- **Related work**
- Theoretical results
- Empirical evidence
- Conclusions

Related Work

- Sequential online learning model
  - First proposed by Schapire & Warmuth (96)
  - They proposed TD*(λ) and analyzed ℓ_P
- Online learning framework
  - Equivalently, $\gamma = 0$
  - E.g., (Cesa-Bianchi, Long & Warmuth, 96)
  - They analyzed ℓ_P for gradient descent

Related Work

- TD(0) converges faster than RG in a small problem (Baird, 95)

- TD(0) converges provably faster (Schoknecht & Merke, 03) when
  - Updates are synchronous
  - No function approximation is used
  - A certain matrix has real eigenvalues only

In contrast, our analysis has no such restrictions

Outline

- The sequential online learning model
- Related work
- **Theoretical results**
- Empirical evidence
- Conclusions

Proof Ideas

- Use $||\mathbf{w}_t - \mathbf{u}||^2$ as the potential function to measure progress of learning in each step
- Lower bound $||\mathbf{w}_t - \mathbf{u}||^2 - ||\mathbf{w}_{t+1} - \mathbf{u}||^2$ in terms of η, γ, d_t, e_t, and $\mathbf{u}$
- Upper bound d_t and e_t in terms of η, γ, and $\mathbf{u}$
- Upper bound $\sum_t d_t^2$ and $\sum_t e_t^2$ by algebraic manipulations
- Optimize η to minimize these upper bounds

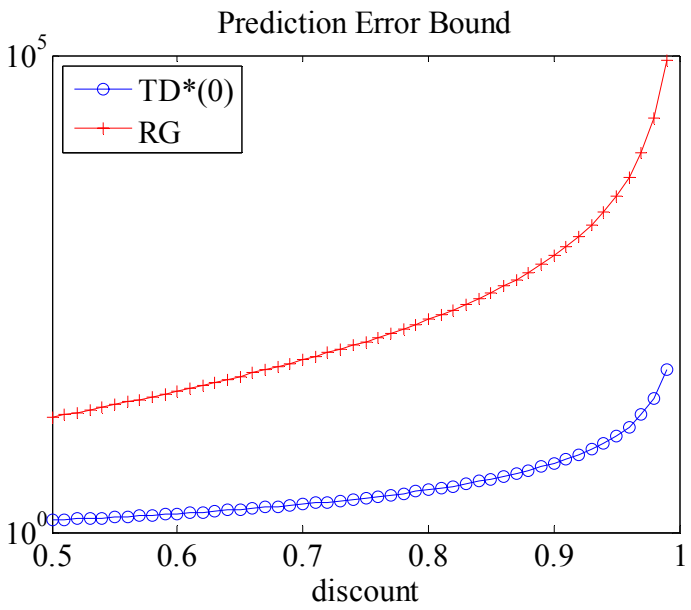
Regret Analysis Summary

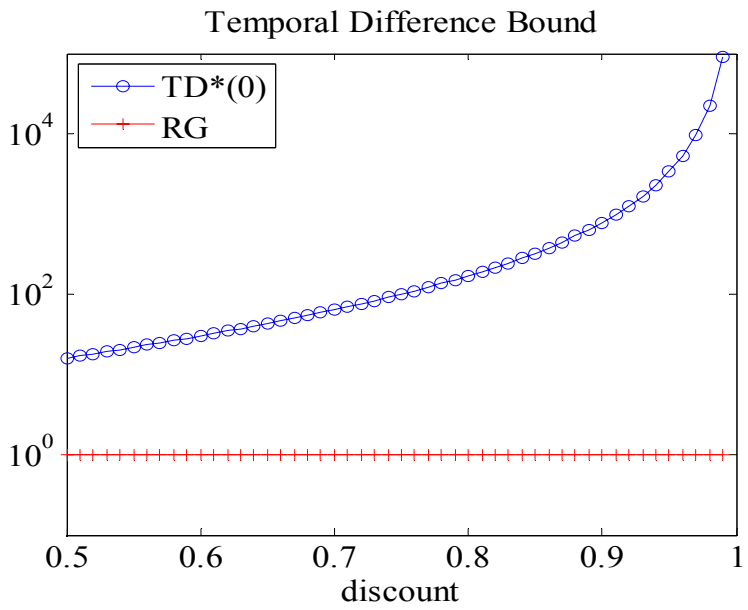
- TD*(0) and TD(0) yield smaller prediction errors
- RG yields smaller temporal differences
- The gaps become larger as $\gamma \rightarrow 1$

(Schapire & Warmuth, 96)

(Cesa-Bianchi, Long, & Warmuth, 96)

	$\ell_{\mathcal{P}}/\ell^{\mathrm{u}}_{\mathcal{P}}$	$\ell_{\mathcal{TD}}/\ell^{\mathrm{u}}_{\mathcal{TD}}$
TD*(0)	$\frac{1}{1-\gamma^2}+o(1)$	$\frac{(1+2\gamma)^2}{(1-\gamma)^2}+o(1)$
RG	$\frac{(1+2\gamma)^2}{(1-\gamma)^2}+o(1)$	$1+o(1)$





Regret Analysis Summary (cont'd)

- Results are valid for non-Markovian problems
  - Partially observable environments
    - Often the case when function approximation is used
  - Multi-agent systems
  - Time-varying situations

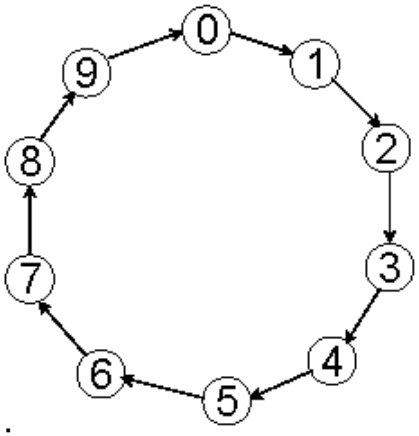
- But … these are upper bounds
  - Open question: how tight are they?

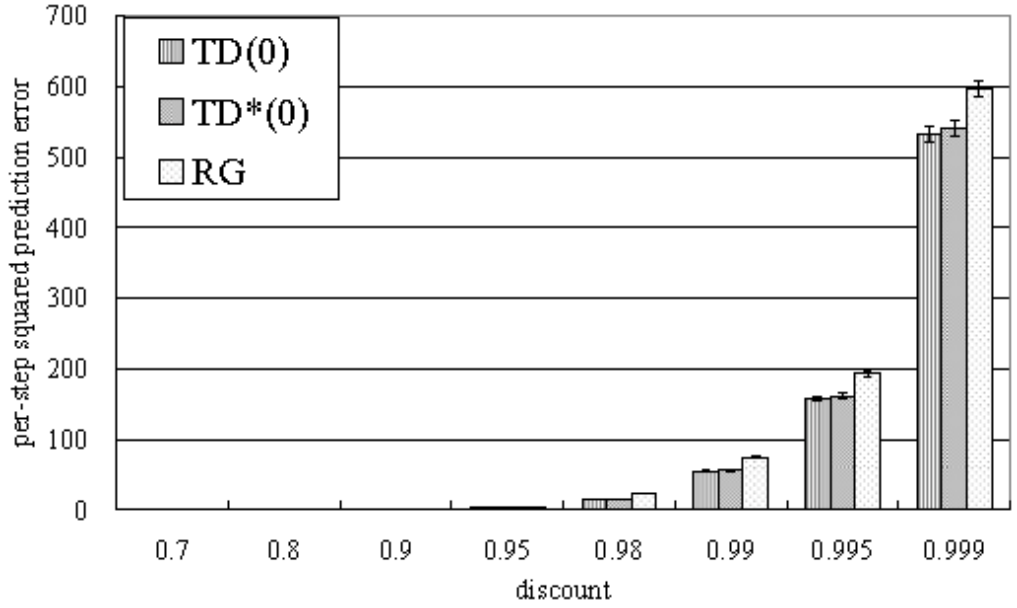
Outline

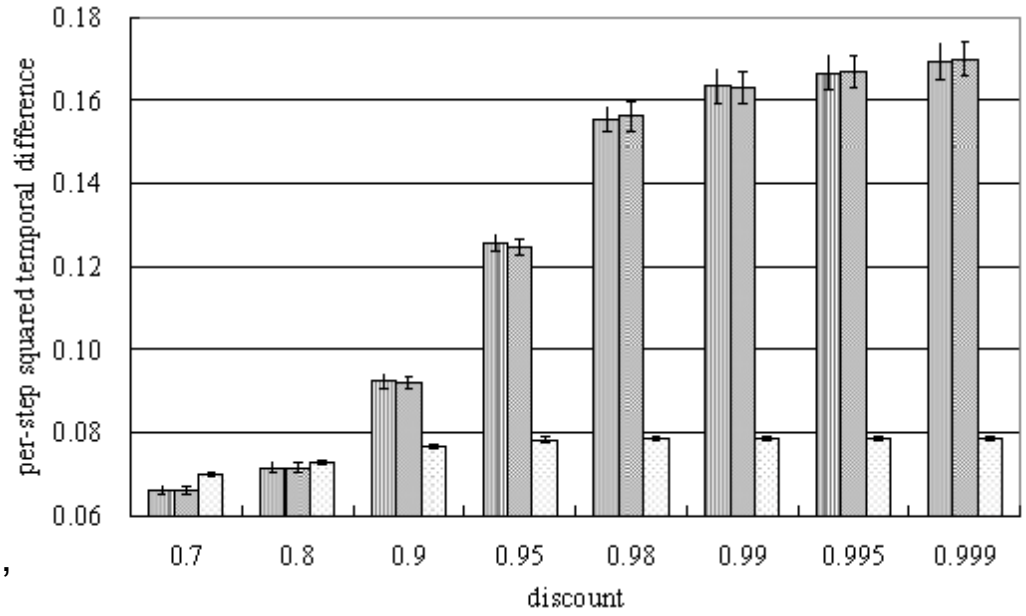
- The sequential online learning model
- Related work
- Theoretical results
- **Empirical evidence**
- Conclusions

Ring Domain

- Random features
- k=5
- $\mathbf{E}[y_t] = 0$
- η was optimized
- 500 runs
- 500 steps per run

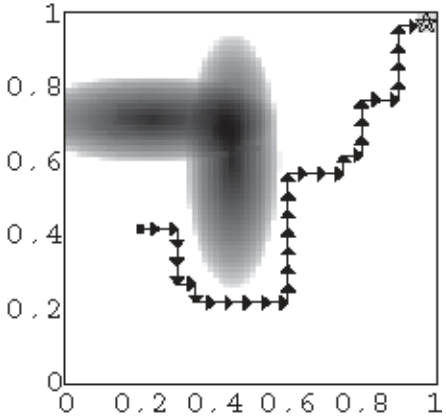




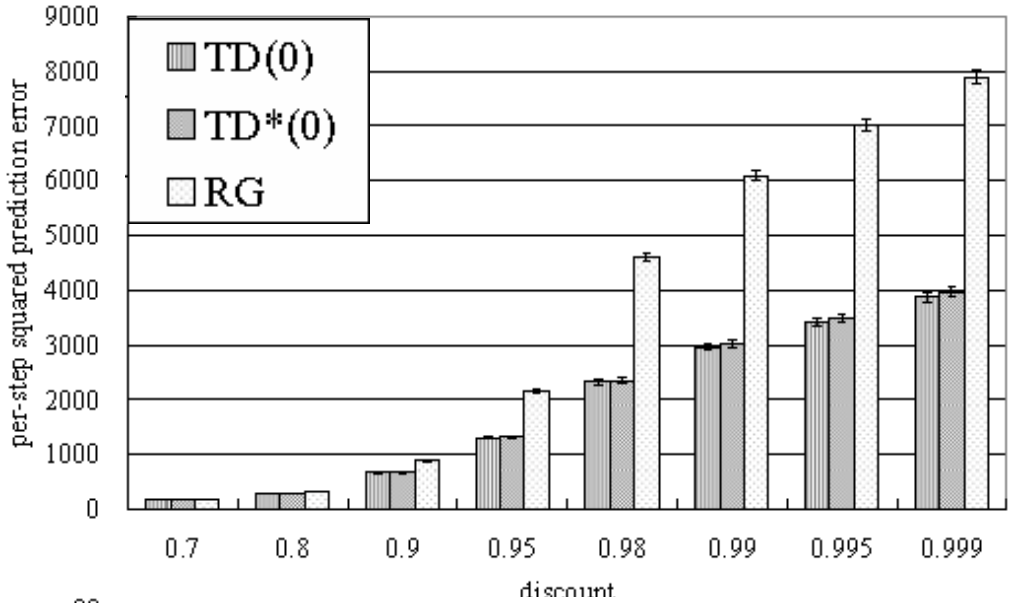


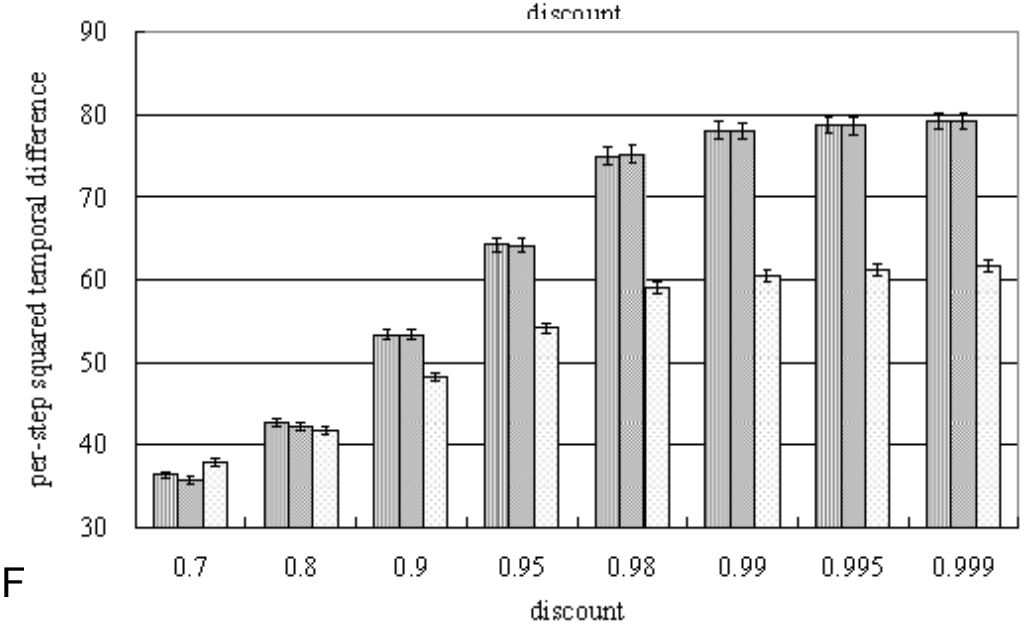
PuddleWorld Domain

- RBF features
- k = 16
- Stochastic policy (N/E)
- η was optimized
- 500 runs
- 50 episodes per run



From
(Boyan & Moore 95)





Outline

- The sequential online learning model
- Related work
- Theoretical results
- Empirical evidence
- **Conclusions**

Conclusions

- We compared TD(0), TD*(0), and RG
  - in sequential online learning model
  - when linear function approximation is used
- Theoretical implications
  - TD predictions are more accurate
  - RG predictions are more consistent
  - Supported by empirical evidence
- Online learning techniques can be useful in RL analysis

Future Work

- Extend analysis of temporal differences to TD(λ) and TD*(λ)
- Find non-trivial lower bounds
  - Ideally, matching lower bounds
- Analyze multiplicative forms of TD
  - E.g., (Kivinen & Warmuth, 97; Precup & Sutton, 97)
  - Could be useful when many features are irrelevant
    - For instance, CMAC features

Thank You!