Temporal Difference Bayesian Model Averaging

A Bayesian Perspective on Adapting λ

Carlton Downey (downeycarl@ecs.vuw.ac.nz)

Scott Sanner (ssanner@nicta.com.au)

TD(λ) - Motivation

- TD(λ) is one of the most widely used reinforcement learning algorithms in the world. However…

- **Question:** data-dependent λ adaptation?

  - no manual λ tuning
  - adapt λ online

- **Answer:** Bayesian model averaging (BMA)

  - what model?
  - how to compute efficiently online?

Outline

- Motivation
- TD(λ) and Bayesian Model Averaging
- Derivation
- Comparative evaluation
- Conclusions & future work

Markov Decision Process (MDP)

- MDP is a tuple **⟨S, A, T, R⟩**
- **S:** finite set of states
- **A:** finite set of actions
- **$T(s',a,s) = P(s'|s, a)$:** transition function
  - stationary, Markovian
- **$R(s,a)$:** (stochastic) reward function

MDP Optimization

- **γ ($0 \leq \gamma < 1$):** discount factor
- **π(s,a):** (stochastic) exploration policy
  - $\pi(s, a) = P(a|s)$
- Objective to optimize:

$$Q_{\pi}(s, a) = E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} \cdot r_{t+1} \middle| s_0 = s, a_0 = a\right]$$

TD

- TD Update:

$$Q_\pi(s_t, a_t) = Q_\pi(s_t, a_t) + \alpha\left[\Delta \mathrm{Q}_{\mathrm{t}}^{\lambda}\right]$$

$$\Delta \mathrm{Q}_{\mathrm{t}}^{\lambda} = R_t - Q_\pi(s_t, a_t)$$

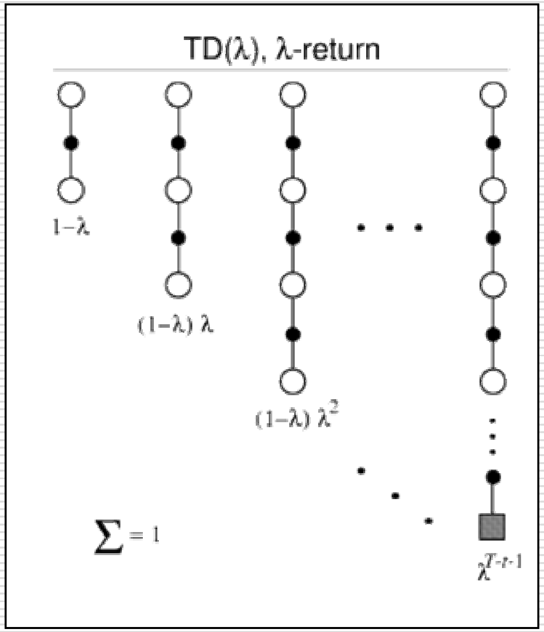
- An n-step return is a bootstrapped estimate.

$$\mathrm{R}_{\mathrm{t}}^{(\mathrm{n})} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \gamma^{i-1} r_{t+i} + \gamma^n \mathrm{Q}_\pi(\mathrm{s}_{\mathrm{t+n}}, \mathrm{a}_{\mathrm{t+n}})$$

TD(λ)

- Averaging many n-step returns gives us TD(λ).

$$R_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} R_t^{(n)}$$



- Expressed recursively

$$R_t^{\lambda} = r_{t+1} + \gamma\left[(1-\lambda)Q_{\pi}(s_{t+1}, a_{t+1}) + \lambda R_{t+1}^{\lambda}\right]$$

SARSA(λ)

- We thus obtain the TD(λ) update rule SARSA(λ):

$$Q_{\pi}(s_t, a_t) = Q_{\pi}(s_t, a_t) + \alpha[\Delta \mathrm{Q}_{\mathrm{t}}^{\lambda}]$$

$$\Delta \mathrm{Q}_{\mathrm{t}}^{\lambda} = \mathrm{R}_{\mathrm{t}}^{\lambda} - \mathrm{Q}_{\pi}(\mathrm{s}_{\mathrm{t}}, \mathrm{a}_{\mathrm{t}})$$

What’s wrong with TD(λ)?

- TD(λ) chooses a particular fixed function of λ.
- Is there a better way to weight each n-step return?
- Can we weight them with so as to reduce variance?

BMA view of TD(λ)

- BMA used for reducing estimator variance.
- BMA provides an intuitive data-dependent way to adjust the weights of multiple estimators.
- BMA weights models according to how much the data supports them.

BMA view of TD(λ)

- Can we look at TD(λ) from BMA perspective?
- We derive an expected return model R_t^{BMA}:

$$R_t^{BMA} = \mathrm{E}_{P(\vec{q}|D)}[q_{s,a}]$$

$$= \int_{q_{s,a} \in R} q_{s,a} P(q_{s,a} | D) dq_{s,a}$$

$$= \int_{q_{s,a} \in R} q_{s,a} \sum_{m \in MC,TD} P(q_{s,a} | m) p(m | D) dq_{s,a}$$

$$= \boxed{p(TD | D)\mathrm{E}_{P(q_{s,a}|TD)}[q_{s,a}] + p(MC | D)\mathrm{E}_{P(q_{s,a}|MC)}[q_{s,a}]}$$

BMA view of TD(λ)

- If we set P(MC|D) = λ and P(TD|D) = (1 − λ):

$$
\begin{aligned}
R_t^{BMA} &= \mathrm{E}_{P(q_{s,a}|TD)}[q_{s,a}]p(TD\,|\,D) + \mathrm{E}_{P(q_{s,a}|MC)}[q_{s,a}]p(MC\,|\,D) \\
&= \mathrm{E}_{P(q_{s,a}|TD)}[q_{s,a}](1-\lambda) + \mathrm{E}_{P(q_{s,a}|MC)}[q_{s,a}]\lambda \\
&= R_t^{\lambda}
\end{aligned}
$$

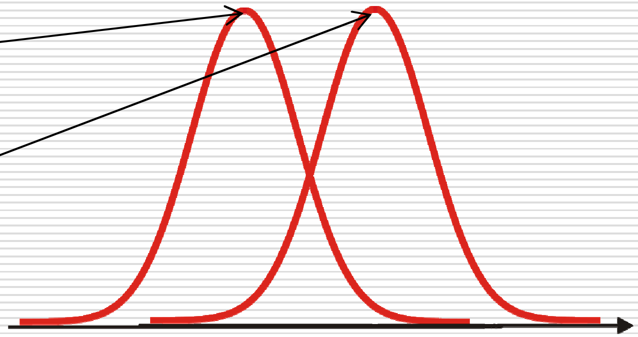
- We have exactly re-derived SARSA(λ).

- But from a BMA perspective should we fix λ?
  - We have data, we want to make lambda data dependent.

Gaussian Case

- Assume value of each pair (s,a) is independently gaussian.
- $P(q_{s,a} \mid m) = \mathrm{N}(q_{s,a}; \mu^m_{s,a}, (\sigma^m_{s,a})^2)$
- For each model Mean and S.D. are sufficient statistics.
- $\mu^{MC}_{s,a} = q^{MC}_{s,a} = r_{t+1} + \gamma R^{\lambda}_{t+1}$

  $\mu^{TD}_{s,a} = q^{TD}_{s,a} = r_{t+1} + \gamma Q_{\pi}(s_{t+1}, a_{t+1})$



- To avoid costly computation we use the S.D of D.

Gaussian Case

- Model Prediction is trivial:

$$\mathrm{E}_{\mathrm{P}(\mathrm{q}_{\mathrm{s,a}} | \mathrm{m})}[\mathrm{q}_{\mathrm{s,a}}] = \mu^m_{s,a} = q^m_{s,a}$$

- Model Weight is harder:

$$
\begin{aligned}
\mathrm{P}(\mathrm{m} | \mathrm{D}) &= \mathrm{P}(\mathrm{m} | \mathrm{D}_{\mathrm{s,a}}) \propto P(D_{s,a} | m) P(m) \\
&\propto P(D_{s,a} | m) \\
&= \prod\nolimits_{d \in D_{s,a}} P(d | m) \\
&= \prod\nolimits_{d \in D_{s,a}} N(d; q^m_{s,a}, \sigma^2_{s,a}) \\
&= \prod\nolimits_{d \in D_{s,a}} C e^{-\frac{(d - q^m_{s,a})^2}{2\sigma^2_{s,a}}} \\
&= C^{|D_{s,a}|} e^{\left(\frac{1}{2\sigma^2_{s,a}}\right) \boxed{\sum\nolimits_{d \in D_{s,a}} (-d^2 - (q^m_{s,a})^2 + 2 q^m_{s,a} d)}}
\end{aligned}
$$

Indep.

Unif. Prior

i.i.d

Bayes

Gaussian Case

- Simplying exponent of model weight further…

$$\boxed{\sum_{d\in D_{s,a}} \left(-d^2 - (q^m_{s,a})^2 + 2q^m_{s,a} d\right)}$$

$$= -\sum_{d\in D_{s,a}} d^2 - \sum_{d\in D_{s,a}} (q^m_{s,a})^2 + 2q^m_{s,a} \sum_{d\in D_{s,a}} d$$

$$= -|D_{s,a}|\left(\sigma^2_{s,a} + (\mu^D_{s,a})^2\right) - |D_{s,a}|(q^m_{s,a})^2 + 2q^m_{s,a}|D_{s,a}|\mu^D_{s,a}$$

Simplified using def. of variance

.

$$= -|D_{s,a}|\left(\sigma^2_{s,a} + (\mu^D_{s,a})^2 + (q^m_{s,a})^2 - 2q^m_{s,a}\mu^D_{s,a}\right)$$

$$= -|D_{s,a}|\sigma^2_{s,a} - |D_{s,a}|(q^m_{s,a} - \mu^D_{s,a})^2$$

Gaussian Case

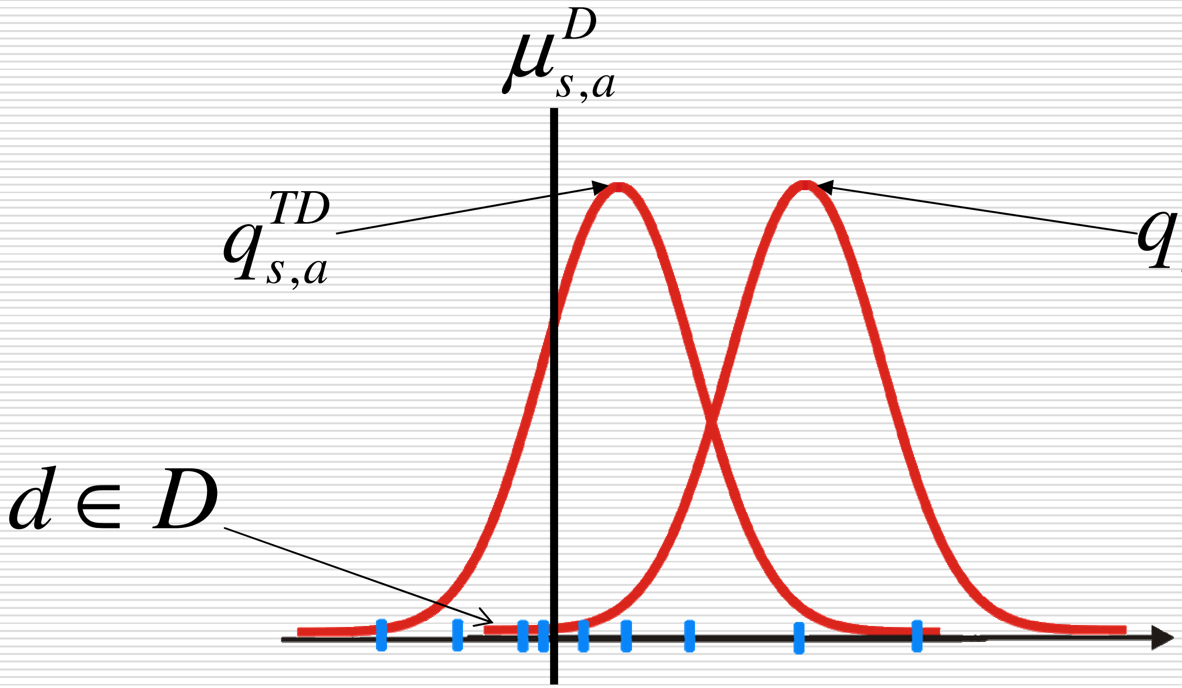
- Finally, substituting back for $\boxed{}$…

$$P(\mathrm{m} \mid \mathrm{D}) \propto C^{|D_{s,a}|} e^{\frac{|D_{s,a}|\sigma^2_{s,a}}{2\sigma^2_{s,a}} - \frac{|D_{s,a}|}{2\sigma^2_{s,a}}(q^m_{s,a} - \mu^D_{s,a})^2}$$

$$= e^{-\frac{|D_{s,a}|}{2}} \left[N(q^m_{s,a}; \mu^D_{s,a}, \sigma^2_{s,a}) \right]^{|D_{s,a}|}$$

- This result depends only $|D_{s,a}|$, $\mu^m_{s,a}$ and $\sigma^m_{s,a}$.
- These can be calculated online in constant time.

Intuition



- Clearly in this case $P(TD_{s,a} \mid D) > P(MC_{s,a} \mid D)$

Algorithm

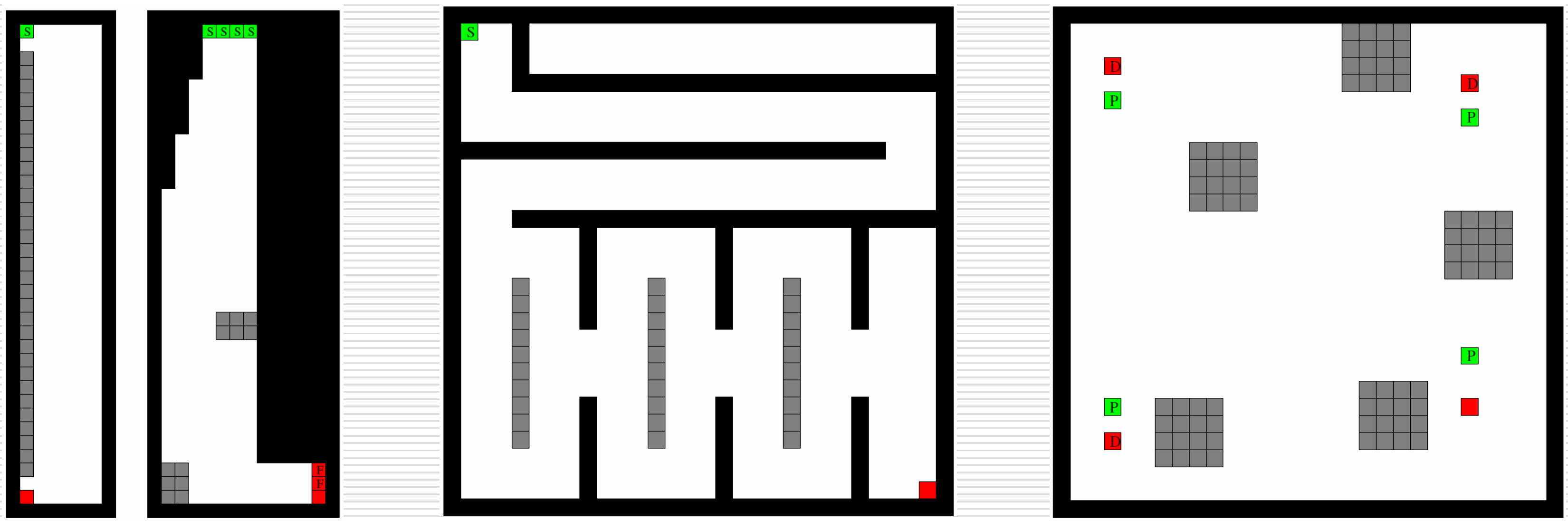
$$\mathrm{R}_{\mathrm{t}}^{\lambda} = \mathrm{r}_{\mathrm{t+1}} + \gamma\left[(1-\lambda)Q_{\pi}(s_{t+1}, a_{t+1}) + \lambda R_{t+1}^{\lambda}\right]$$

$$\lambda_{BMA} = \frac{P(MC_{s,a} \mid D)}{P(MC_{s,a} \mid D) + P(TD_{s,a} \mid D)}$$

$$= \frac{\left[N(q_{s,a}^{MC}; \mu_{s,a}^{D}, (\sigma_{s,a}^{D})^2)\right]^{|D_{s,a}|}}{\left[N(q_{s,a}^{MC}; \mu_{s,a}^{D}, (\sigma_{s,a}^{D})^2)\right]^{|D_{s,a}|}\left[N(q_{s,a}^{TD}; \mu_{s,a}^{D}, (\sigma_{s,a}^{D})^2)\right]^{|D_{s,a}|}}$$

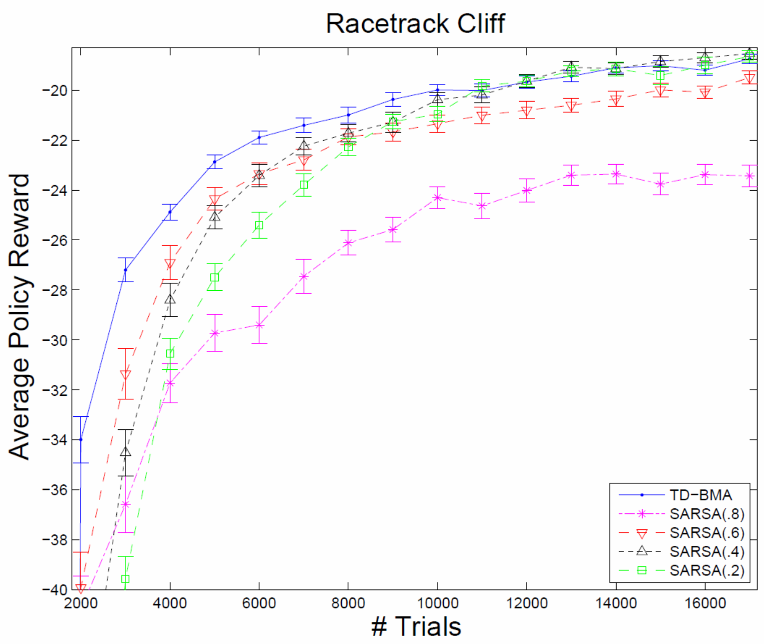
$$R_{t}^{BMA} = r_{t+1} + \gamma\left[(1-\lambda_{BMA})Q_{\pi}(s_{t+1}, a_{t+1}) + \lambda_{BMA} R_{t+1}^{BMA}\right]$$

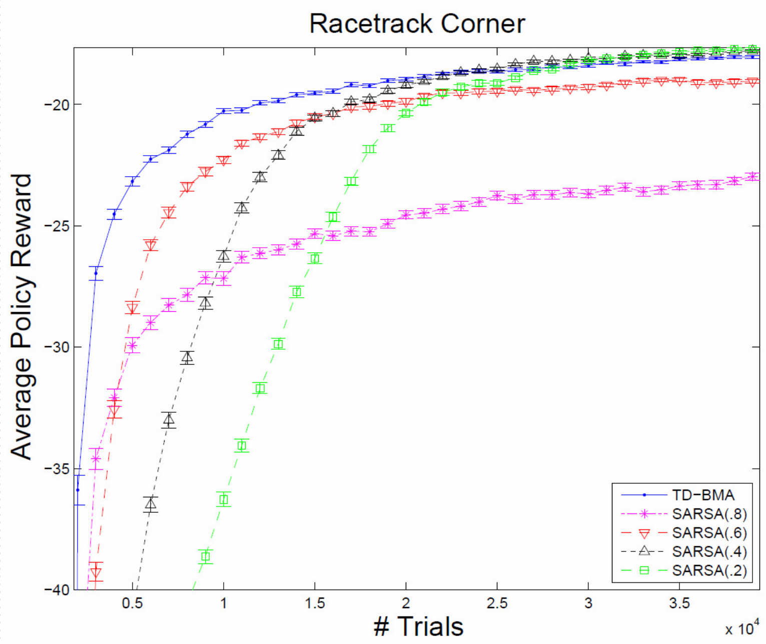
Experimental Setup



(a) Cliff (b) Corner (c) Curves (d) Mail

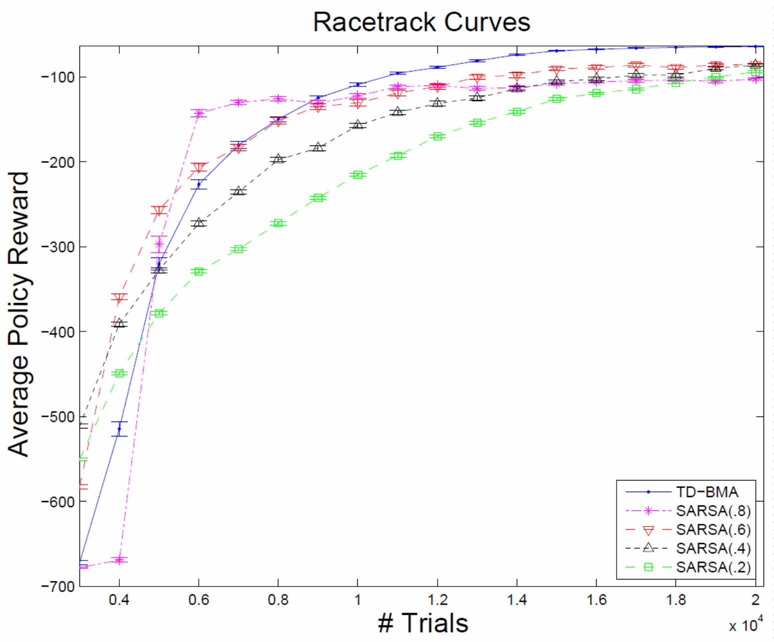
Results: Varying λ

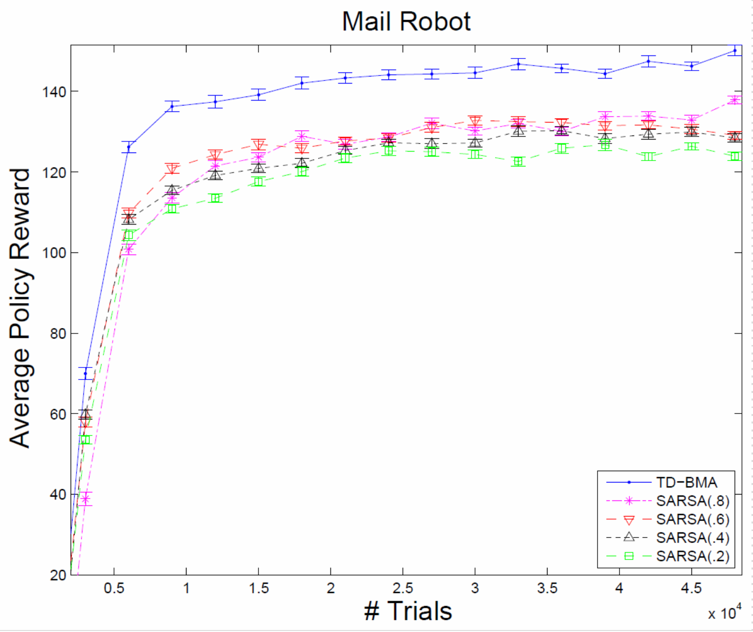




- TD-BMA outperforms SARSA for all values of λ.

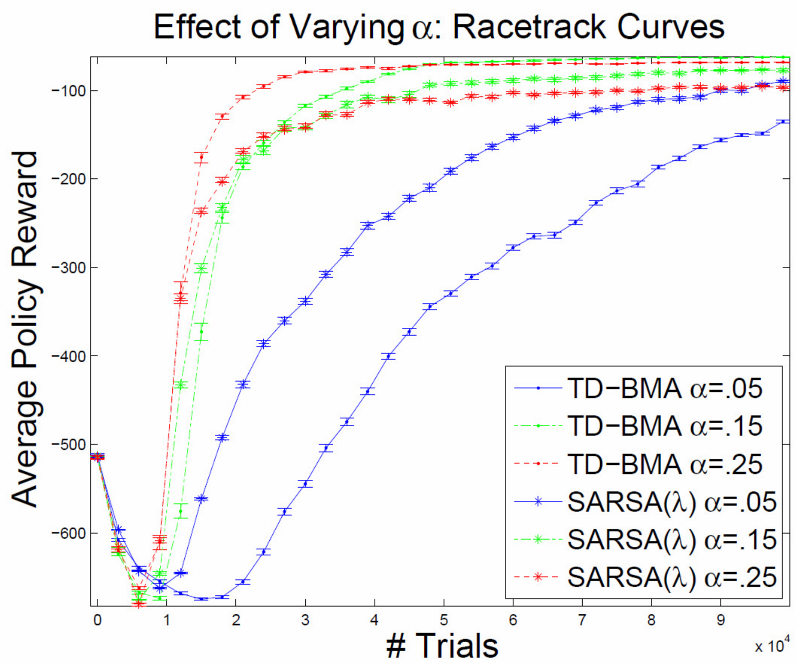
Results: Varying λ

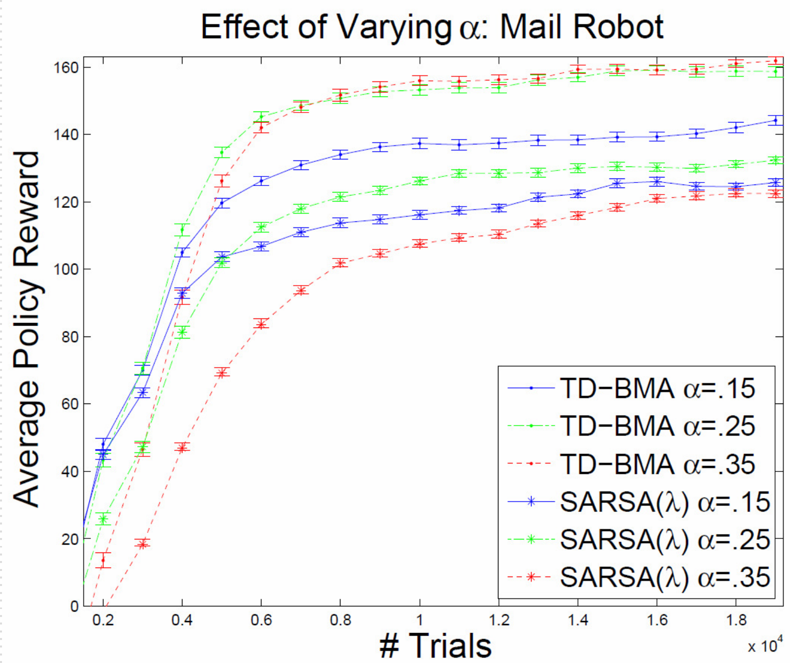




- TD-BMA outperforms SARSA for all values of λ.

Results: Varying α

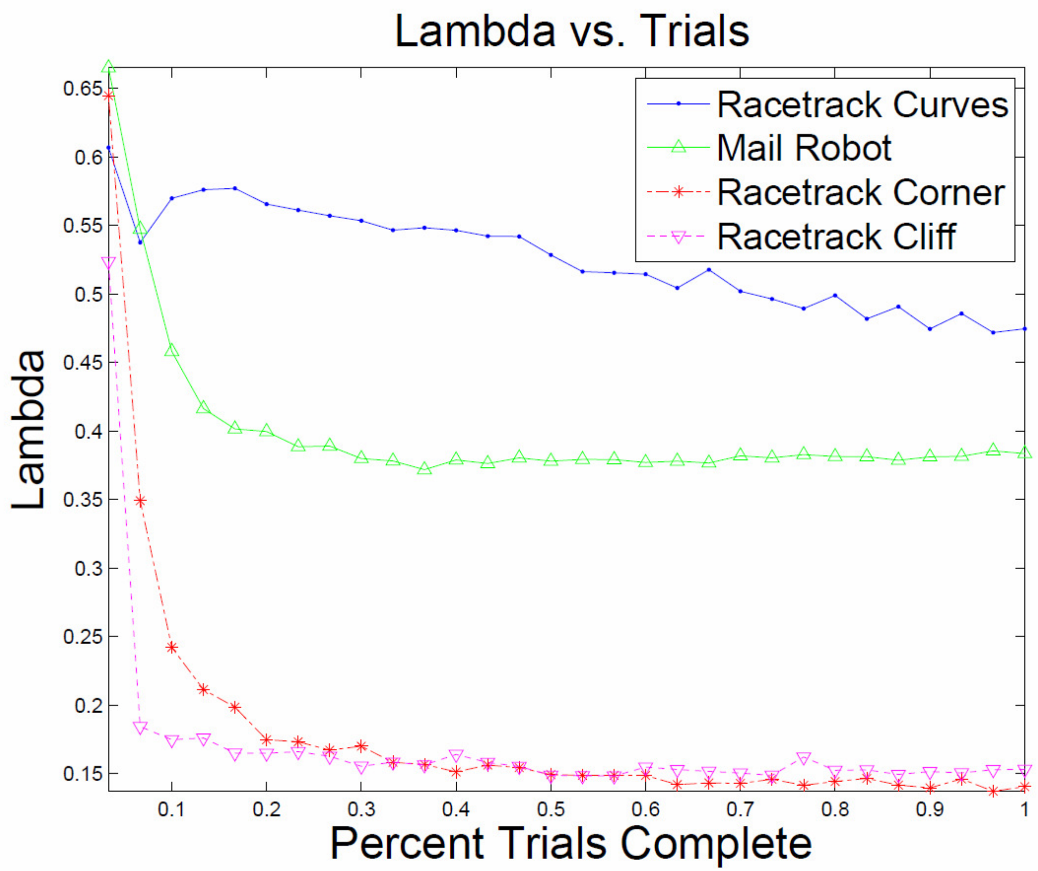




- TD-BMA outperforms SARSA for all values of α.

Results: λ over time

- Bootstrapped model trusted more over time.
- Mail is partially observable, bootstrapped model helps disambiguate state.



Related Work

- Focus of other Bayesian RL approaches is on balancing exploration vs. Exploitation.
  - E.g. GPTD
- Sutton and Singh propose three adaptive weighting schemes
  - First two restricted to acyclic.
  - Third requires maintaining an estimate of the model.

Conclusions and Future Work

- We derived a novel BMA approach to adapting λ in TD(λ).
- We contributed the efficient Gaussian-based TD-BMA algorithm.
- We showed that TD-BMA generally performs much better than SARSA for all fixed values of λ.
- Future work:
  - Derive an online version of TD-BMA.
  - Use this for TD-BMA with function approximation.