# PAC-MDP Reinforcement Learning with Bayesian Priors

**John Asmuth**[†] JASMUTH@CS.RUTGERS.EDU
**Lihong Li**[†] LIHONG@CS.RUTGERS.EDU
**Michael L. Littman**[†] MLITTMAN@CS.RUTGERS.EDU
**Ali Nouri**[†] NOURI@CS.RUTGERS.EDU
**David Wingate**[‡] WINGATED@MIT.EDU
[†]RL$^3$ Laboratory, Department of Computer Science, Rutgers University, Piscataway, NJ USA 08854
[‡]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139

In an effort to build on recent advances in reinforcement learning and Bayesian modeling, this work (Asmuth et al., 2009) combines ideas from two lines of research on exploration in reinforcement learning or RL (Sutton & Barto, 1998). *Bayesian RL* research (Dearden et al., 1999; Poupart et al., 2006) formulates the RL problem as decision making in the belief space of all possible environment models. As such, it becomes meaningful to talk about *optimal* RL—selecting actions that maximize the expected long-term reward given the uncertainty in the model. Although progress has been made in approximating optimal policies in model belief space, these techniques have not been shown to scale well and come with no finite-sample guarantees on the quality of the derived policies.

*PAC-MDP RL* approaches (Fiechter, 1994; Kearns & Singh, 2002; Brafman & Tennenholtz, 2002) relax the goal of optimal exploration to one of bounding, with high probability, the number of steps in which a learning algorithm does not have a near-optimal policy. Instead of choosing actions that maximize expected value, PAC-MDP RL algorithms are entitled to take a bounded number of sub-optimal actions. In contrast to Bayesian RL, the simpler goal of PAC-MDP RL allows provably efficient algorithms to be created for many model classes (Li et al., 2008; Li, 2009).

PAC-MDP RL algorithms are flexible in that different algorithms can be created to exploit different possible model classes, including no state–state generalization (Brafman & Tennenholtz, 2002), predefined conditional independence between state variables in the form of DBNs (Kearns & Koller, 1999), unknown DBNs (Strehl et al., 2007; Diuk et al., 2009), and outcome-based state-independent action models (Leffler et al., 2007). The main motivation for the present work is that in practice it can be difficult to choose among these algorithms. If a model class is chosen that is very narrow, there is a danger that the actual environment will fall outside this class and the learning algorithm will fail. If a model class is chosen that is overly broad, learning will be slower than necessary.

In this work, we provide an algorithm that makes it possible to provide a prior distribution over models that encourages our algorithm to consider the more efficiently learnable narrow classes first while still being able to provide guarantees on performance if the environment belongs to a broader model class. Like Bayesian RL, our approach represents the state of knowledge using a posterior distribution over model space. Like PAC-MDP RL, our approach achieves near-optimal reward with high probability with a polynomial sample complexity of exploration (Kakade, 2003), or sample complexity for short.

We call our algorithm "Best of Sampled Set" or BOSS. BOSS samples multiple ($K$) MDP models from the posterior whenever the number of transitions from a state–action pair reaches a pre-defined threshold ($B$). It then combines these models into an optimistic MDP, whose optimal policy is used to choose actions until model sampling happens again. This algorithm differs from previous model-sampling-based Bayesian RL algorithms: it explicitly specifies the number of models to sample, provides a concrete rule to decide when model re-sampling happens, and most importantly, defines how these sampled models are combined together for choosing actions.

These algorithmic choices are inspired by a PAC-MDP algorithm known as Rmax (Brafman & Tennenholtz, 2002) to encourage the agent to explore states where it does not have an accurate estimate of the true MDP model. In fact, we can show that, when the parameters $K$ and $B$ are chosen appropriately, BOSS achieves near-optimal reward with high probability with a sample complexity that is polynomial in relevant quantities including the speed at which the posterior distribution concentrates during learning.

BOSS performs favorably compared to state-of-the-art RL approaches in our experiments, including PAC-MDP and Bayesian ones. In a well-studied 5-state chain problem (*e.g.*, Poupart et al. (2006)), the agent has two actions: $A_1$ advances the agent along the chain, and $A_2$ resets the agent

| | Tied | Semi | Full |
|---|---|---|---|
| BEETLE | 3650 | 3648 | 1754 |
| exploit | 3642 | 3257 | 3078 |
| BOSS | 3657 | 3651 | 3003 |
| RAM-RMAX | 3404 | 3383 | 2810 |

*Table 1.* Cumulative reward in Chain

to the first node. $A_1$, when taken from the last node, leaves the agent where it is and gives a reward of 10—all other rewards are $0$. $A_2$ always has a reward of 2. With probability $0.2$ the outcomes are switched, however. The optimal behavior is to always choose $A_1$ to reach the end of the chain and high reward. As in Poupart et al. (2006), we used three types of priors: in the Full prior, the agent assumes each state–action pair corresponds to independent multinomial distributions over next states; in the Tied prior, the agent knows the underlying transition dynamics except for the value of a single slip probability that is shared between all state–action pairs; and the Semi prior is similar to Tied except that the two actions are considered independent. Posteriors for Full can be maintained using a Dirichlet (the conjugate for the multinomial) and Tied/Semi can be represented with a simple Beta distribution. Table 1 reports cumulative rewards of four algorithms in the first $1000$ steps, averaged over $500$ runs. All runs used $\gamma = 0.95$. BOSS used $B = 10$ and $K = 5$. Results for BEETLE and exploit are from Poupart et al. (2006), and RAM-RMAX is a non-Bayesian, PAC-MDP algorithm by Leffler et al. (2007). The optimal policy for this problem scores 3677. This experiment shows that BOSS can make effective use of informative Bayesian priors while remaining efficient when the prior contains less information.

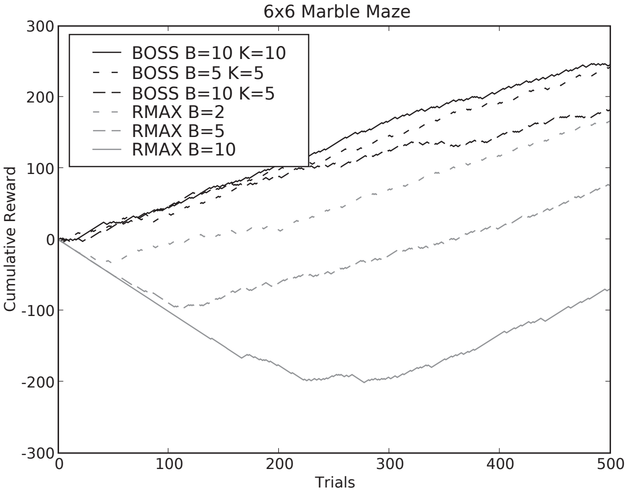

*Figure 1.* Comparison of algorithms on 6x6 Marble Maze.

BOSS is a modularized algorithm and so is flexible enough to be paired with a non-parametric Bayesian model (Chinese Restaurant Process or CRP) to discover structures, if any, in the MDP model based on weak prior knowledge. As a result, the algorithm is able to learn fast when many states' dynamics are similar, and on the other hand can still learn a near-optimal solution otherwise. We tried BOSS in two domains, including a $6 \times 6$ gridworld called Marble Maze (Asmuth et al., 2009). In this problem, the agent has four actions (N/E/S/W) whose transitions are stochastic. The agent succeeds in an episode if it reaches the goal or fails if it falls in a pit. The dynamics of this environment are such that each local pattern of walls (at most 16) can be modeled as a separate cluster. BOSS used $\gamma = 0.95$ and a CRP hyper-parameter of $\alpha = 10$. To sample MDP models, we used a Gibbs sampler that ran for a burn period of $100$ sweeps with $50$ sweeps between each sample. RMAX was run as a baseline in this domain. Figure 1 reports cumulative rewards of both algorithms with various parameters. BOSS learned the cluster structure and dominated RMAX that did not know or learn the cluster structure. The primary difference visible in the graph is the time needed to obtain the optimal policy. Remarkably, BOSS ($B = 10, K = 10$) latches onto near optimal behavior nearly instantaneously, whereas RMAX required $50$ to $250$ trials before behaving as well.

## References

Asmuth, J., Li, L., Littman, M. L., Nouri, A., & Wingate, D. (2009). A Bayesian sampling approach to exploration in reinforcement learning. *UAI*.

Brafman, R. I., & Tennenholtz, M. (2002). R-MAX—a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, *3*, 213–231.

Dearden, R., Friedman, N., & Andre, D. (1999). Model-based bayesian exploration. *UAI* (pp. 150–159).

Diuk, C., Li, L., & Leffler, B. R. (2009). The adaptive $k$-meteorologists problem and its application to structure discovery and feature selection in reinforcement learning. *ICML*.

Fiechter, C.-N. (1994). Efficient reinforcement learning. *COLT* (pp. 88–97).

Kakade, S. M. (2003). *On the sample complexity of reinforcement learning*. Doctoral dissertation, Gatsby Computational Neuroscience Unit, University College London.

Kearns, M. J., & Koller, D. (1999). Efficient reinforcement learning in factored MDPs. *IJCAI* (pp. 740–747).

Kearns, M. J., & Singh, S. P. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, *49*, 209–232.

Leffler, B. R., Littman, M. L., & Edmunds, T. (2007). Efficient reinforcement learning with relocatable action models. *AAAI*.

Li, L. (2009). *A unifying framework for computational reinforcement learning theory*. Doctoral dissertation, Rutgers University, New Brunswick, NJ.

Li, L., Littman, M. L., & Walsh, T. J. (2008). Knows what it knows: A framework for self-aware learning. *ICML* (pp. 568–575).

Poupart, P., Vlassis, N., Hoey, J., & Regan, K. (2006). An analytic solution to discrete Bayesian reinforcement learning. *ICML* (pp. 697–704).

Strehl, A. L., Diuk, C., & Littman, M. L. (2007). Efficient structure learning in factored-state MDPs. *AAAI* (pp. 645–650).

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. The MIT Press.