Bayesian Inverse Reinforcement Learning

Deepak Ramachandran and Deepak Ramachandran
University of Illinois at Urbana-Champaign

IJCAI-2007

Introduction

- Tasks of IRL-reward learning or apprenticeship learning.
- model the IRL problem from a Bayesian Perspective.
- Solve the problem with a modified MCMC algorithm.

Approach

- We observe expert's behavior $O = \{(s_1,\ a_1), (s_2,\ a_2) \dots (s_k,\ a_k)\}$.
- The expert's policy is stationary, make the following assumption:

$$\Pr(O|R) = \Pr((s_1, a_1)|R)\Pr((s_2, a_2)|R) \dots \Pr((s_k, a_k)|R)$$

$$\Pr((s_i, a_i)|R) = \frac{1}{Z_i} e^{\alpha Q^*(s_i, a_i, R)}$$

$$\Pr(O|R) = \frac{1}{Z} \mathrm{e}^{\alpha \sum Q^*(s_i, a_i, R)}$$

$$\Pr(R|O) = \frac{\Pr(O|R)\, P(R)}{Pr(O)}$$

$$= \frac{1}{Z'} \mathrm{e}^{\alpha \sum Q^*(s_i, a_i, R)} P(R)$$

Priors

- If we are completely agnostic about the prior :

Uniform distribution over $-R_{max} \leq R(s) \leq R_{max}$

- Most states have negligible rewards:

$$P_{Gaussian}(\boldsymbol{R}(s) = r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{r^2}{2\sigma^2}}, \forall s \in S$$

- most states to have low (or negative) rewards but a few states to have high rewards:

$$P_{Beta}(\boldsymbol{R}(s) = r) = \frac{1}{(\frac{r}{R_{max}})^{\frac{1}{2}}(1 - \frac{r}{R_{max}})^{\frac{1}{2}}}, \forall s \in S$$

Tasks-Reward Learning

$$
\begin{aligned}
L_{linear}(\boldsymbol{R}, \hat{\boldsymbol{R}}) &= \| \boldsymbol{R} - \hat{\boldsymbol{R}} \|_1 \\
L_{SE}(\boldsymbol{R}, \hat{\boldsymbol{R}}) &= \| \boldsymbol{R} - \hat{\boldsymbol{R}} \|_2
\end{aligned}
$$

The expected value of $LSE(R, \hat{R})$ is minimized by setting $\hat{R}$ to the mean of the posterior.

Similarly, the expected linear loss is minimized by setting $\hat{R}$ to the median of the distribution.

Tasks-Apprenticeship Learning

policy loss functions:

$$L^p_{policy}(\boldsymbol{R}, \pi) = \| \boldsymbol{V}^*(\boldsymbol{R}) - \boldsymbol{V}^\pi(\boldsymbol{R}) \|_p$$

So, instead of trying a difficult direct minimization of the expected policy loss, we can find the optimal policy for the mean reward function, which gives the same answer.

Theorem 3. *Given a distribution $P(\boldsymbol{R})$ over reward functions $\boldsymbol{R}$ for an MDP (S, A, T, γ), the loss function $L^p_{policy}(\boldsymbol{R}, \pi)$ is minimized for all p by π^*_M, the optimal policy for the Markov Decision Problem $M = (S, A, T, \gamma, E_P[\boldsymbol{R}])$.*

Algorithm

Algorithm `PolicyWalk`(Distribution P, MDP M, Step Size δ)

1. Pick a random reward vector $\boldsymbol{R} \in \mathbb{R}^{|S|}/\delta$.
2. $\pi :=$ `PolicyIteration`$(M, \boldsymbol{R})$
3. Repeat
   (a) Pick a reward vector $\tilde{\boldsymbol{R}}$ uniformly at random from the neighbours of $\boldsymbol{R}$ in $\mathbb{R}^{|S|}/\delta$.
   (b) Compute $Q^{\pi}(s, a, \tilde{\boldsymbol{R}})$ for all $(s, a) \in S, A$.
   (c) If $\exists(s, a) \in (S, A), Q^{\pi}(s, \pi(s), \tilde{\boldsymbol{R}}) < Q^{\pi}(s, a, \tilde{\boldsymbol{R}})$
      i. $\tilde{\pi} :=$ `PolicyIteration`$(M, \tilde{\boldsymbol{R}}, \pi)$
      ii. Set $\boldsymbol{R} := \tilde{\boldsymbol{R}}$ and $\pi := \tilde{\pi}$ with probability $\min\{1, \frac{P(\tilde{\boldsymbol{R}}, \tilde{\pi})}{P(\boldsymbol{R}, \pi)}\}$

      Else

      i. Set $\boldsymbol{R} := \tilde{\boldsymbol{R}}$ with probability $\min\{1, \frac{P(\tilde{\boldsymbol{R}}, \pi)}{P(\boldsymbol{R}, \pi)}\}$
4. Return $\boldsymbol{R}$

Figure 3: PolicyWalk Sampling Algorithm

- Both reward learning and apprenticeship learning require computing the mean of the posterior distribution.
- MCMC combined with policy iteration.

Experiments

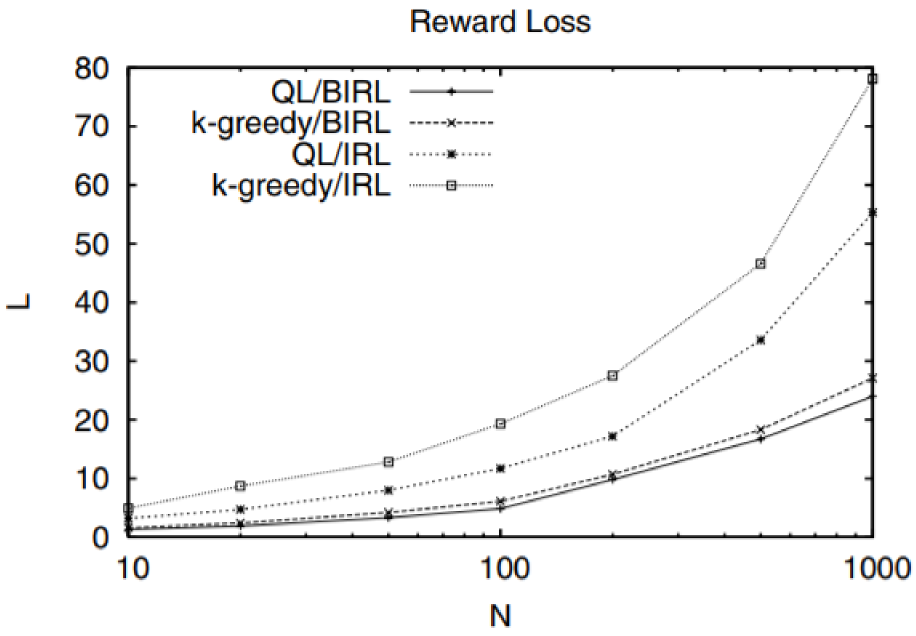

Figure 4: Reward Loss.

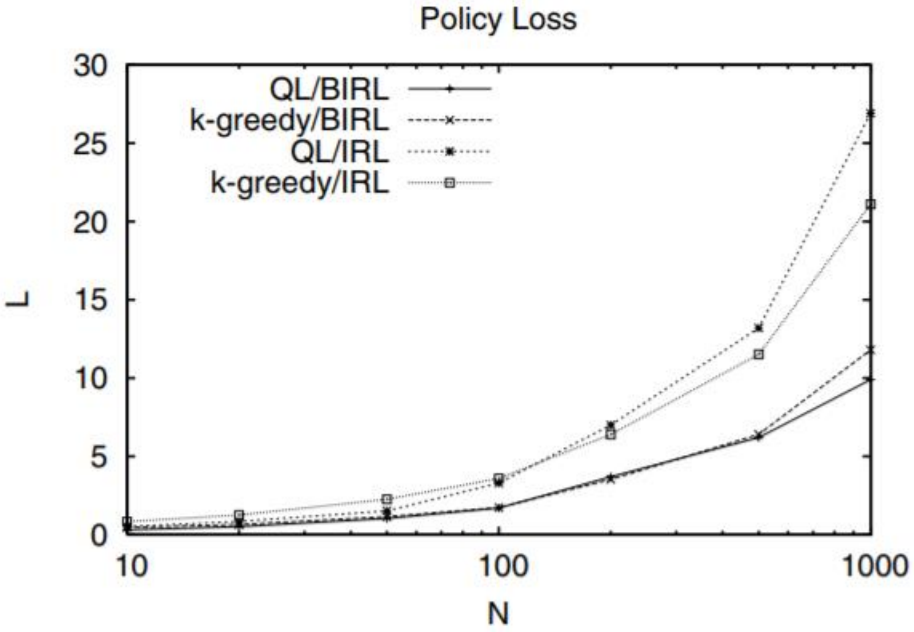

Figure 5: Policy Loss.

- BIRL vs IRL by(Ng and Russell, 2000)

Experiments

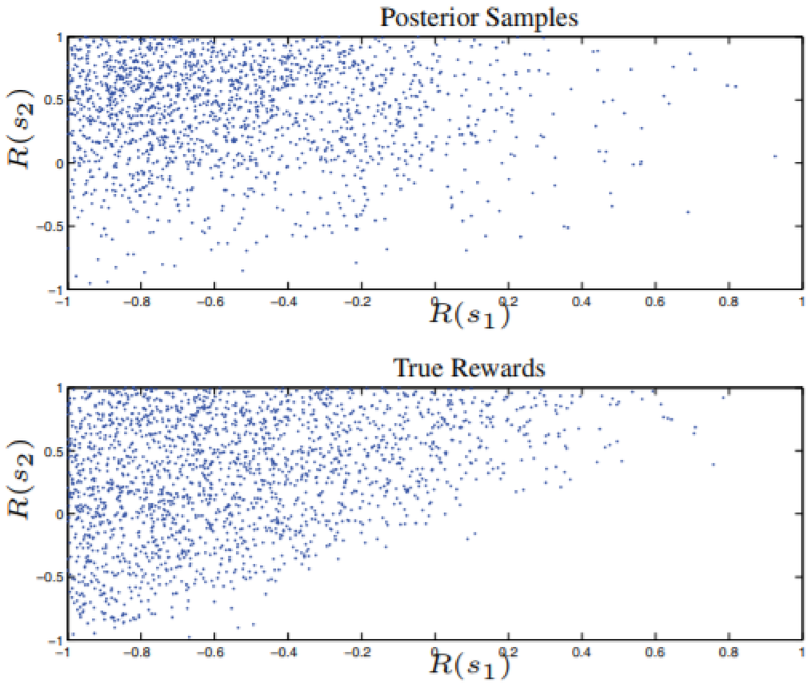

Figure 6: Scatter diagrams of sampled rewards of two arbitrary states for a given MDP and expert trajectory. Our computed posterior is shown to be close to the true distribution.

- Posterior samples vs true rewards

Experiments-domain knowledge

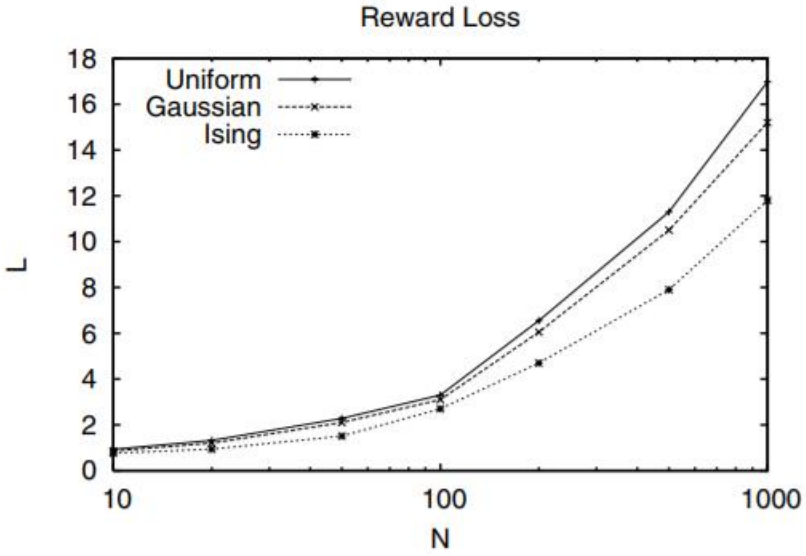

Figure 7: Ising versus Uninformed Priors for Adventure Games

A adventure game

Reward - Ising prior.

$$P_R(\boldsymbol{R}) = \frac{1}{Z}\exp(-J \sum_{(s',s)\in\mathcal{N}} R(s)R(s') - H \sum_{s} R(s))$$

Conclusion

- We model the IRL problem from a Bayesian Perspective.
- Solve the problem with a modified MCMC algorithm.
- We get improved solution.