# Bayesian Inverse Reinforcement Learning

Presented by Monica Dinculescu

McGill University
February 7, 2007

# Markov Decision Processes(MDP)

## Definition

An MDP is a tuple $(S, A, T, \gamma, R)$

- $S$ is a finite set of states
- $A$ is a finite set of actions
- $T : S \times A \times S \rightarrow [0, 1]$ is the transition probability function
- $\gamma \in [0, 1)$ is the discount factor
- $R : S \rightarrow \mathbb{R}$ is the reward function

# Bellman Equations

## Definition

A **policy** is a map $\pi : S \rightarrow A$

## Definition

The **value** of a policy $\pi$ under a reward function $R$

$V^{\pi}(s) = R(s) + \gamma \sum_{s'} T(s, \pi(s), s') V^{\pi}(s')$

## Definition

The **Q-function** of a policy $\pi$ under a reward function $R$

$Q^{\pi}(s, a) = R(s) + \gamma \sum_{s'} T(s, a, s') V^{\pi}(s')$

# Motivation

## Problem

Learn the reward function of an underlying Markov Decision Process given

1. behaviour of an expert
2. dynamics of the system

# Tasks

## Reward Learning

Estimate the reward function as accurately as possible

1. modelling opponents in competitive games
2. preference elicitation

## Apprenticeship Learning

Use observations of expert behaviour to decide own behaviour

1. policy learning
2. better generalization of tasks

# Bayesian Inference

- evidence is used to infer the probability that a hypothesis may be true
- given a hypothesis $H$ and evidence $E$, we define:

**Definition**

Prior probability $P(H)$

**Definition**

Posterior probability $P(H|E) = \frac{P(E|H)P(H)}{P(H)}$

# Main Idea

- the hypothesis: is the reward function that explains the agent's behaviour
- the evidence: observations about expert behaviour
- the evidence is used to infer the probability that a hypothesis may be true(i.e. the posterior distribution of the rewards, from a prior distribution)
- important how to choose the prior:
  - uniform distribution, if there is no information given
  - Laplacian or Gaussian, if most states have negligible rewards
  - Beta, if the MDP is a planning-type problem (most states have low rewards, but a few, goal states have high rewards)

# Reward Learning

- learn a reward function

**Definition**

Error loss function :$L(R, \overline{R}) = \| R - \overline{R} \|^2$, where:

- $R$ is the actual reward
- $\overline{R}$ is the estimated reward

- if $R$ is drawn from the posterior distribution, then $L(R, \overline{R})$ is minimized by setting $\overline{R}$ to the mean of the posterior
- use a maxiumum a posteriori estimator (MAP) as the estimator

# Apprenticeship Learning

- want to learn a policy (i.e. how to act), given expert behaviour

**Definition**

Class of policy loss functions :$L(R, \pi) = \| V^*(R) - V^{\pi}(R) \|^p$, where:

- $V^*$ is the vector of optimal values for each state, under the optimal policy $\pi$ for $R$

- want to find a $\pi$ that minimizes the policy loss over the posterior distribution for $R$
- direct minimization is hard; instead, find optimal policy $\pi$ for the mean reward function

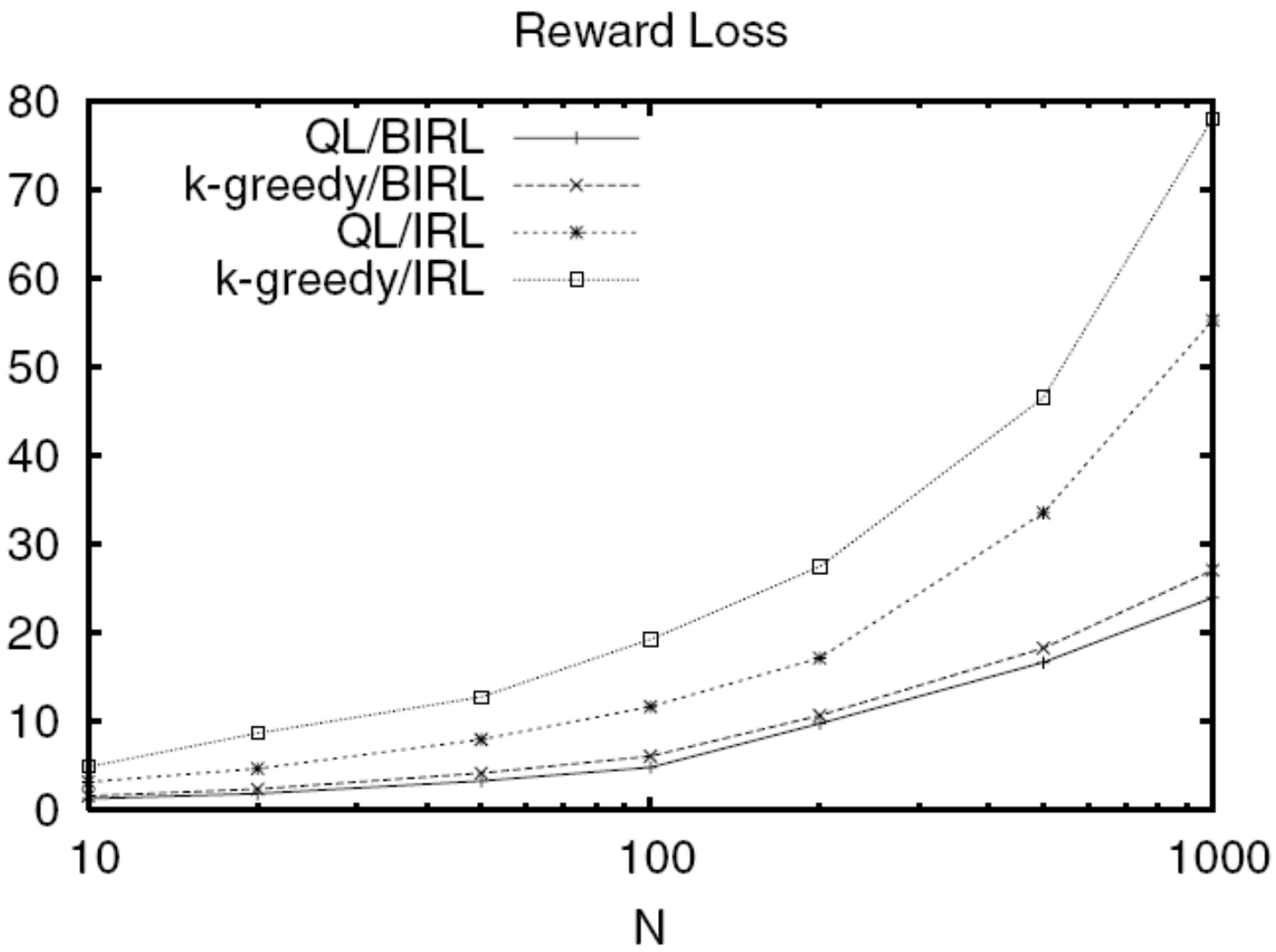
# PolicyWalk Sampling Algorithm

Algorithm `PolicyWalk`(Distribution $P$, MDP $M$, Step Size $\delta$ )

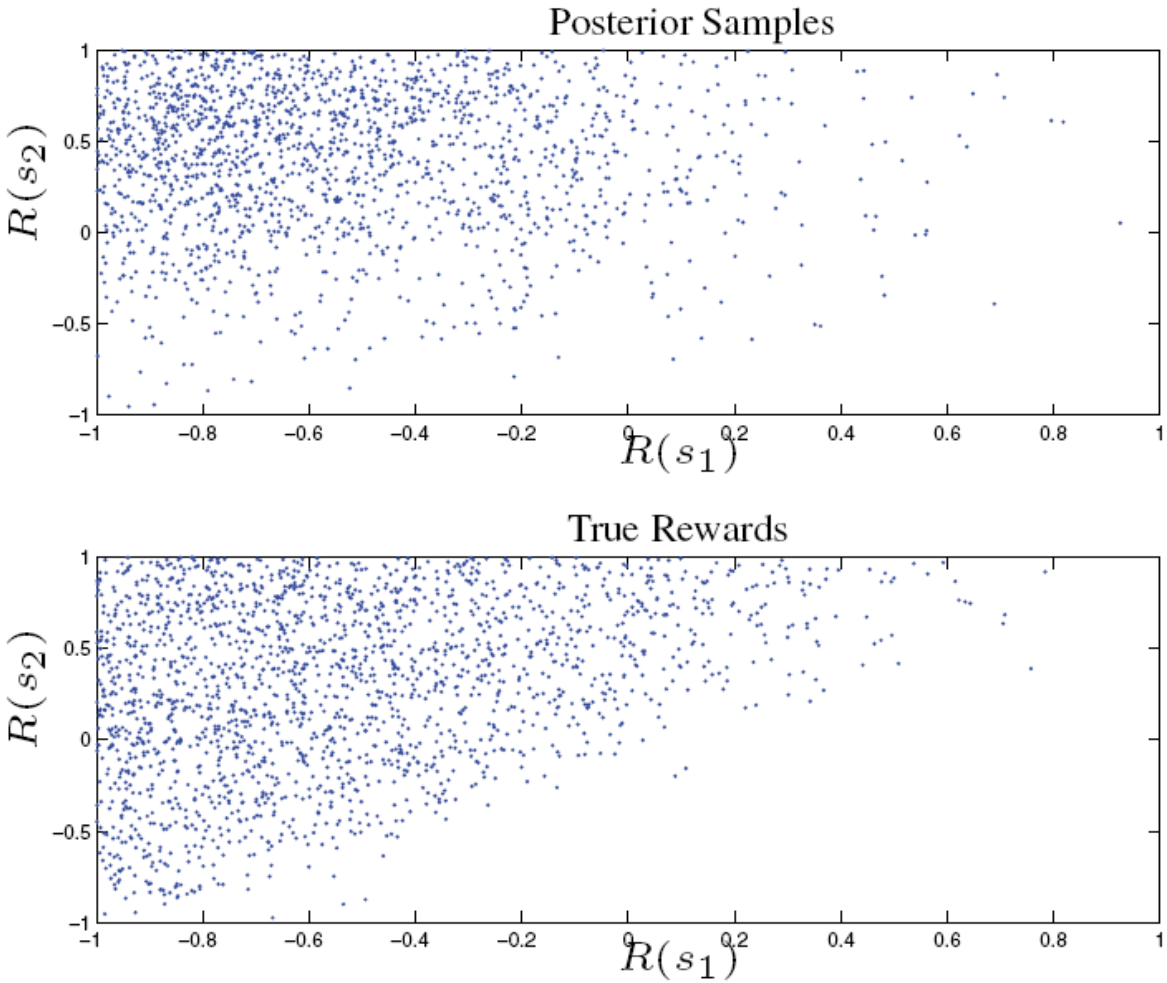
1. Pick a random reward vector $\boldsymbol{R} \in \mathbb{R}^{|S|}/\delta$.
2. $\pi := \texttt{PolicyIteration}(M, \boldsymbol{R})$
3. Repeat
   (a) Pick a reward vector $\tilde{\boldsymbol{R}}$ uniformly at random from the neighbours of $\boldsymbol{R}$ in $\mathbb{R}^{|S|}/\delta$.
   (b) Compute $Q^{\pi}(s, a, \tilde{\boldsymbol{R}})$ 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. $\tilde{\pi} := \texttt{PolicyIteration}(M, \tilde{\boldsymbol{R}}, \pi)$
      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 $\boldsymbol{R}$

- generate samples from the prior distribution (of rewards)
- the sample mean is the estimate of the true mean of the distribution

# Reward Loss


Reward Loss
QL/BIRL
k-greedy/BIRL
QL/IRL
k-greedy/IRL
80
70
60
50
40
30
20
10
0
10
100
1000
N

# Posterior Distributions



# Conclusions

- reward function can be learned by posing the problem as a Bayesian larning task
- algorithm yields a probability distribution over reward functions that is close to the true one
- policy learning is possible

# References


- D. Ramachandran and E. Amir, *Bayesian Inverse Reinforcement Learning,* In IJCAI, 2006
- P. Abbeel and A. Y. Ng. *Apprenticeship learning via inverse reinforcement learning*. In ICML, 2004