Probabilistic Inference for Solving Markov Decision Processes

Toussaint, Storkey, ICML '06

Noam Siegel

Computer Science M.Sc. Seminar
Ben Gurion University

May 15, 2022

Outline

1. Introduction
   - Motivation & Prior work
   - Main contribution
2. Research Problem
   - Solving Markov Decision Processes
3. Research Plan
   - Mixture of MDPs and likelihood
   - An EM-algorithm for computing the optimal policy
   - Relation to Policy Iteration
4. Experimental results
   - Discrete maze
   - Stochastic optimal control
5. Conclusion

1 Introduction
Motivation & Prior work
Main contribution

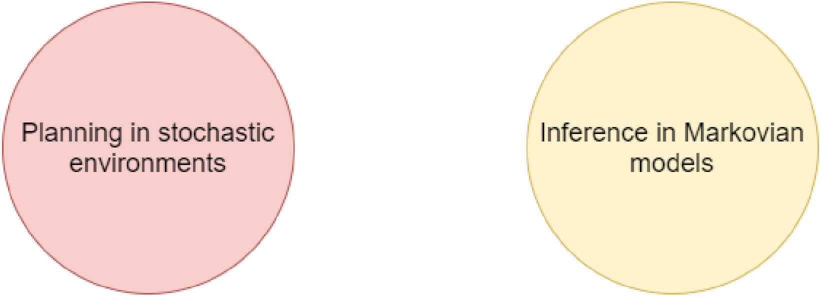
2 Research Problem

3 Research Plan

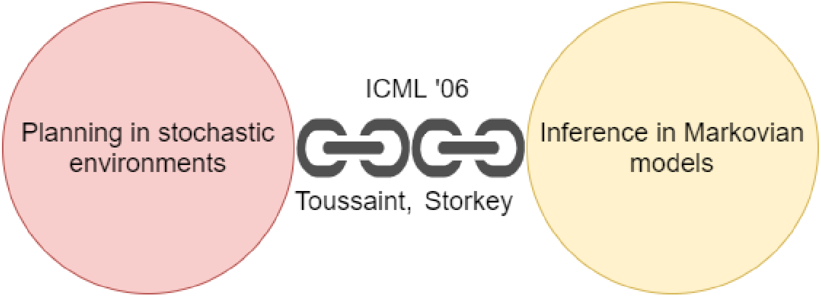
4 Experimental results

5 Conclusion

Motivation


Planning in stochastic environments
Inference in Markovian models

Motivation


Planning in stochastic environments
ICML '06
Toussaint, Storkey
Inference in Markovian models

Prior work

1. *Bui et al. (2002)* used inference on Abstract Hidden Markov Models for policy recognition, *but not for computing an optimal policy*.
2. *Attias (2003)* got close to translating the problem of planning to a problem of inference. However, *the total time T had to be fixed* and the MAP action sequence that is proposed as a solution *is not optimal*.
3. *Verma and Rao (2006)* used inference to compute plans, but again T has to be fixed and the plan is not optimal.

Main contribution

Contribution

1. Translate the problem of *maximizing the expected future return* exactly into a problem of *likelihood maximization in a latent variable model*, for arbitrary reward functions and episode lengths.

Main contribution

Contribution

1. Translate the problem of *maximizing the expected future return* exactly into a problem of *likelihood maximization in a latent variable model*, for arbitrary reward functions and episode lengths.
2. Demonstrate the approach on discrete & continuous stochastic optimal control problems,

1 Introduction

2 Research Problem
   Solving Markov Decision Processes

3 Research Plan

4 Experimental results

5 Conclusion

Examples 1: Discrete maze

(a)

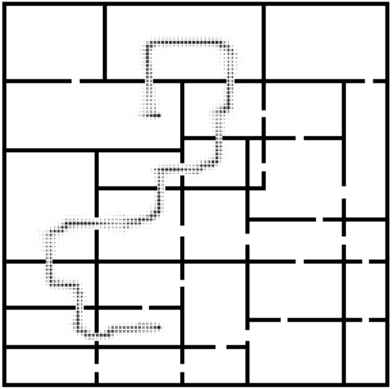


Figure 1: Posterior state-visiting-probabilities generated by our Probabilistic Infernce Planner (PIP)

Markov Decision Processes

Definition (MDP)

$$
\begin{aligned}
\text{state transition probability } & P(x_{t+1} \mid a_t, x_t) \\
\text{reward probability } & P(r_t \mid a_t; x_t), \qquad r_t \in \{0, 1\} \\
\text{action probability } & P(a_t \mid x_t; \pi), \qquad \pi \text{ a parameter}
\end{aligned}
$$

Markov Decision Processes

Definition (MDP)

state transition probability $P(x_{t+1} \mid a_t, x_t)$

reward probability $P(r_t \mid a_t; x_t), \qquad r_t \in \{0, 1\}$

action probability $P(a_t \mid x_t; \pi), \qquad \pi$ a parameter

Definition (Policy π)

The action probabilities are parameterized by a policy:

$$P(a_t \mid x_t = i; \pi) = \pi_{ai} \qquad s.t. \sum_a \pi_{ai} = 1$$

Markov Decision Processes

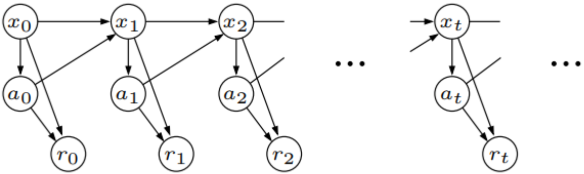


Figure 1. Dynamic Bayesian Network for a MDP. The x states denote the state variables, a the actions and r the rewards.

Definition (MDP)

$$\begin{aligned} \text{state transition probability } & P(x_{t+1} \mid a_t, x_t) & \\ \text{reward probability } & P(r_t \mid a_t; x_t), & r_t \in \{0, 1\} \\ \text{action probability } & P(a_t \mid x_t; \pi), & \pi \text{ a parameter} \end{aligned}$$

Research problem

Definition (solving an MDP)

Solving an MDP means to find a parameter π of the graphical model in Figure 1 that maximizes the expected future return $V^{\pi}(i) = E\{\sum_{t=0}^{\infty} \gamma^t r_t \mid x_0 = i; \pi\}$, where $\gamma \in [0, 1]$ is a discount factor.

Research problem

Definition (solving an MDP)

Solving an MDP means to find a parameter π of the graphical model in Figure 1 that maximizes the expected future return $V^{\pi}(i) = E\{\sum_{t=0}^{\infty} \gamma^{t} r_t \mid x_0 = i; \pi\}$, where $\gamma \in [0, 1]$ is a discount factor.

research problem

The problem is to *solve the MDP*, i.e. to find a policy that maximizes the expected future return.

1 Introduction

2 Research Problem

3 Research Plan
- Mixture of MDPs and likelihood
- An EM-algorithm for computing the optimal policy
- Relation to Policy Iteration

4 Experimental results

5 Conclusion

Mixture of MDPs

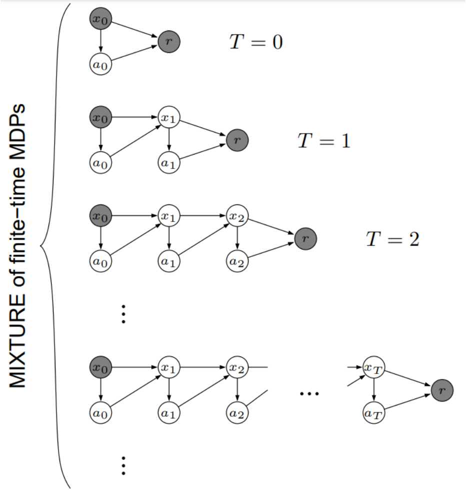


Figure 2. Mixture of finite-time MDPs.

Representing the joint distribution $P(\mathcal{X})$

full joint for finite time MDP

$$P(r, x_{0:T}, a_{0:T} \mid T; \pi) =$$

$$P(r \mid a_T, x_T) P(a_0 \mid x_0; \pi) P(x_0) \cdot \prod_{t=1}^{T} P(a_t \mid x_t; \pi) P(x_t \mid a_{t-1}, x_{t-1})$$

Representing the joint distribution $P(\mathcal{X})$

full joint for finite time MDP

$$P(r, x_{0:T}, a_{0:T} \mid T; \pi) =$$

$$P(r \mid a_T, x_T) P(a_0 \mid x_0; \pi) P(x_0) \cdot \prod_{t=1}^{T} P(a_t \mid x_t; \pi) P(x_t \mid a_{t-1}, x_{t-1})$$

full joint for mixture of finite-time MDPs

$$P(r, x_{0:T}, a_{0:T}, T; \pi) = P(r, x_{0:T}, a_{0:T} \mid T; \pi) P(T)$$

Representing the joint distribution $P(\mathcal{X})$

full joint for finite time MDP

$$P(r, x_{0:T}, a_{0:T} \mid T; \pi) =$$

$$P(r \mid a_T, x_T) P(a_0 \mid x_0; \pi) P(x_0) \cdot \prod_{t=1}^{T} P(a_t \mid x_t; \pi) P(x_t \mid a_{t-1}, x_{t-1})$$

full joint for mixture of finite-time MDPs

$$P(r, x_{0:T}, a_{0:T}, T; \pi) = P(r, x_{0:T}, a_{0:T} \mid T; \pi) P(T)$$

prior over the total time

$$P(T) = \gamma^T (1 - \gamma)$$

Defining the likelihood

Defining the likelihood

Definition (likelihood for a finite-time MDP)

$$L_T^\pi(i) = P(r = 1 \mid x_0 = i, T; \pi) = E\{r \mid x_0 = i, T; \pi\}$$

Defining the likelihood

Definition (likelihood for a finite-time MDP)

$$L_T^\pi(i) = P(r = 1 \mid x_0 = i, T; \pi) = E\{r \mid x_0 = i, T; \pi\}$$

Definition (likelihood for mixture of MDPs)

$$L^\pi(i) = P(r = 1 \mid x_0 = i; \pi) = \sum_T P(T) E\{r \mid x_0 = i, T; \pi\}$$

Defining the likelihood

Definition (likelihood for a finite-time MDP)

$$L_T^\pi(i) = P(r = 1 \mid x_0 = i, T; \pi) = E\{r \mid x_0 = i, T; \pi\}$$

Definition (likelihood for mixture of MDPs)

$$L^\pi(i) = P(r = 1 \mid x_0 = i; \pi) = \sum_T P(T) E\{r \mid x_0 = i, T; \pi\}$$

Corollary

$$L_T^\pi(i) = (1 - \gamma) V^\pi(i)$$

Theoretical Guarantee

Reminders

1. *Solving an MDP* means to find a parameter π of the graphical model in Figure 1 that maximizes the expected future return $V^{\pi}(i) = E\{\sum_{t=0}^{\infty} \gamma^{t} r_t \mid x_0 = i; \pi\}$.
2. The *likelihood for a mixture of MDPs* is given by $L^{\pi}(i) = P(r = 1 \mid x_0 = i; \pi) = \sum_{T} P(T) E\{r \mid x_0 = i, T; \pi\}$
3. This implies that $L_T^{\pi}(i) = (1 - \gamma) V^{\pi}(i)$

Theoretical Guarantee

Reminders

1. *Solving an MDP* means to find a parameter π of the graphical model in Figure 1 that maximizes the expected future return $V^{\pi}(i) = E\{\sum_{t=0}^{\infty} \gamma^t r_t \mid x_0 = i; \pi\}$.
2. The *likelihood for a mixture of MDPs* is given by $L^{\pi}(i) = P(r = 1 \mid x_0 = i; \pi) = \sum_T P(T) E\{r \mid x_0 = i, T; \pi\}$
3. This implies that $L_T^{\pi}(i) = (1 - \gamma) V^{\pi}(i)$

Theorem

(proved) Maximizing the likelihood in the mixture of finite-time MDPs is equivalent to solving the MDP.

An EM-algorithm for computing the optimal policy

What are EM algorithms?

A class of algorithms consisting of two modes:

1. E-step: estimates missing variables.
2. M-step: optimizes parameters of model to best explain the data.

An EM-algorithm for computing the optimal policy

What are EM algorithms?

A class of algorithms consisting of two modes:

1. E-step: estimates missing variables.
2. M-step: optimizes parameters of model to best explain the data.

An EM-algorithm for computing the optimal policy

What are EM algorithms?

A class of algorithms consisting of two modes:

1. E-step: estimates missing variables.
2. M-step: optimizes parameters of model to best explain the data.

An EM-algorithm for computing the optimal policy

What are EM algorithms?

A class of algorithms consisting of two modes:

1. E-step: estimates missing variables.
2. M-step: optimizes parameters of model to best explain the data.

E-step

For a given π, compute posteriors $P(T \mid x_0 = A, r = 1; \pi)$ and $P(x_{1:T}, a_{1:T} \mid x_0 = A, r = 1, T; \pi)$

An EM-algorithm for computing the optimal policy

What are EM algorithms?

A class of algorithms consisting of two modes:

1. E-step: estimates missing variables.
2. M-step: optimizes parameters of model to best explain the data.

E-step

For a given π, compute posteriors $P(T \mid x_0 = A, r = 1; \pi)$ and $P(x_{1:T}, a_{1:T} \mid x_0 = A, r = 1, T; \pi)$

M-step

Adapt parameters π to optimize $V^\pi(A)$

E-step: forward-backward in all MDPs synchronously

Simplifying notation

1. $p(j \mid a, i) = P(x_{t+1} = j \mid a_t = a, x_t = i)$
2. $p(j \mid i; \pi) = P(x_{t+1} = j \mid x_t = i; \pi) = \sum_a p(j \mid a, i)\pi_{ai}$

E-step: forward-backward in all MDPs synchronously

Forward Propagation

$$\alpha_0(i) = \delta_{i=A}$$

$$\begin{aligned} \alpha_t(i) &= P(x_t = i \mid x_0 = A; \pi) \\ &= \sum_j p(i \mid j; \pi)\alpha_{t-1}(j) \end{aligned}$$

E-step: forward-backward in all MDPs synchronously

Forward Propagation

$$\alpha_0(i) = \delta_{i=A}$$

$$\begin{aligned}\alpha_t(i) &= P(x_t = i \mid x_0 = A; \pi) \\ &= \sum_j p(i \mid j; \pi)\alpha_{t-1}(j)\end{aligned}$$

Backward Propagation (temp.)

$$\tilde{\beta}_T(i) = \hat{\beta}(i)$$

$$\begin{aligned}\tilde{\beta}_t(i) &= P(r = 1 \mid x_t = i; \pi) \\ &= \sum_j p(j \mid i; \pi)\tilde{\beta}_{t+1}(j)\end{aligned}$$

Where

$$\hat{\beta}(i) = P(r = 1 \mid x_T = i; \pi) = \sum_a P(r = 1 \mid a_T = a, x_T = i)\pi_{ai}$$

E-step: forward-backward in all MDPs synchronously

Backward Propagation (temp.)

$$\tilde{\beta}_T(i) = \hat{\beta}(i)$$

$$\begin{aligned}\tilde{\beta}_t(i) &= P(r = 1 \mid x_t = i; \pi) \\ &= \sum_j p(j \mid i; \pi)\tilde{\beta}_{t+1}(j)\end{aligned}$$

Where

$$\hat{\beta}(i) = P(r = 1 \mid x_T = i; \pi) = \sum_a P(r = 1 \mid a_T = a, x_T = i)\pi_{ai}$$

Backward Propagation (corrected)

$$\beta_0(i) = \hat{\beta}(i)$$

$$\begin{aligned}\beta_\tau(i) &= P(r = 1 \mid x_{T-\tau} = i; \pi) \\ &= \sum_j p(j \mid i; \pi)\beta_{\tau-1}(j)\end{aligned}$$

M-step: the policy update

Definition (expected complete log-likelihood)

$Q(\pi^*, \pi) = \sum_T \sum_{x_{0:T}, a_{0:T}} P(x_{0:T}, a_{0:T}, T \mid r = 1; \pi)$

M-step: the policy update

Definition (expected complete log-likelihood)

$Q(\pi^*, \pi) = \sum_T \sum_{x_{0:T}, a_{0:T}} P(x_{0:T}, a_{0:T}, T \mid r = 1; \boldsymbol{\pi})$

Fact

Maximizing $Q(\pi^*, \pi)$ w.r.t. π^* is achieved by setting $\pi^*_{ai} = P(a_t = a \mid x_t = i, r = 1; \boldsymbol{\pi})$

M-step: the policy update

Definition (expected complete log-likelihood)

$Q(\pi^*, \pi) = \sum_T \sum_{x_{0:T}, a_{0:T}} P(x_{0:T}, a_{0:T}, T \mid r = 1; \boldsymbol{\pi})$

Fact

Maximizing $Q(\pi^*, \pi)$ w.r.t. π^* is achieved by setting
$\pi^*_{ai} = P(a_t = a \mid x_t = i, r = 1; \boldsymbol{\pi})$

However

exploiting the structure of the MDP, we can also write the likelihood as

$$P(r = 1 \mid x_0 = i; \boldsymbol{\pi}) = \sum_{aj} P(r = 1; a_t = a, x_t = j; \boldsymbol{\pi}^*)\pi^*_{aj} \cdot P(x_t = j \mid x_0 = i; \boldsymbol{\pi}^*)$$

M-step: the policy update

However

exploiting the structure of the MDP, we can also write the likelihood as

$$P(r = 1 \mid x_0 = i; \pi) = \sum_{aj} P(r = 1; a_t = a, x_t = j; \pi^*)\pi^*_{aj} \cdot P(x_t = j \mid x_0 = i; \pi^*)$$

$$\pi^*_{ai} = \delta_{a=a^*(i)} \qquad \alpha^*(i) = \arg\max_a P(r = 1 \mid a_t = a, x_t = i; \pi)$$

Relation to Policy Iteration

E-step: Policy Evaluation

1. $\beta_\tau(i) \propto (V^\pi(i)$ *of the MDP of time* $T = \tau)$.

Relation to Policy Iteration

E-step: Policy Evaluation

1. $\beta_\tau(i) \propto (V^\pi(i)$ *of the MDP of time* $T = \tau)$.
2. $V^{pi}(i) = \frac{1}{1-\gamma} \sum_T P(T)\beta_T(i)$.

Relation to Policy Iteration

E-step: Policy Evaluation

1. $\beta_\tau(i) \propto (V^\pi(i)$ *of the MDP of time* $T = \tau)$.
2. $V^{pi}(i) = \frac{1}{1-\gamma} \sum_T P(T)\beta_T(i)$.
3. Hence, *the E-step performs a policy evaluation*.

Relation to Policy Iteration

E-step: Policy Evaluation

1. $\beta_\tau(i) \propto (V^\pi(i)$ *of the MDP of time* $T = \tau)$.
2. $V^{pi}(i) = \frac{1}{1-\gamma}\sum_T P(T)\beta_T(i)$.
3. Hence, *the E-step performs a policy evaluation*.
4. Additionally, *it yields the time, state and action posteriors*.

Relation to Policy Iteration

E-step: Policy Evaluation

1. $\beta_\tau(i) \propto (V^\pi(i)$ *of the MDP of time* $T = \tau)$.
2. $V^{pi}(i) = \frac{1}{1-\gamma} \sum_T P(T)\beta_T(i)$.
3. Hence, *the E-step performs a policy evaluation*.
4. Additionally, *it yields the time, state and action posteriors*.

Relation to Policy Iteration

E-step: Policy Evaluation

1. $\beta_\tau(i) \propto (V^\pi(i)$ *of the MDP of time* $T = \tau)$.
2. $V^{pi}(i) = \frac{1}{1-\gamma} \sum_T P(T)\beta_T(i)$.
3. Hence, *the E-step performs a policy evaluation*.
4. Additionally, *it yields the time, state and action posteriors*.

M-step: Policy Update

1. Maximizing the Q-function w.r.t. the action *a* and state *i*.

Relation to Policy Iteration

Thus,

the EM-algorithm using exact inference and belief representation is effectively equivalent to Policy Iteration but computes the necessary quantities in a different way.

Relation to Policy Iteration

Thus,

the EM-algorithm using exact inference and belief representation is effectively equivalent to Policy Iteration but computes the necessary quantities in a different way.

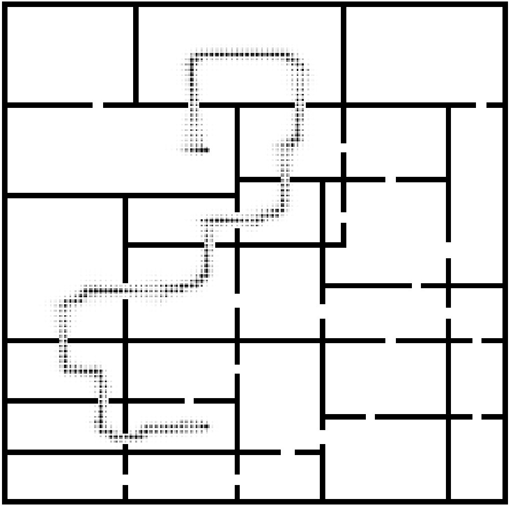
However,

when using approximate inference or belief representations

1. Introduction
2. Research Problem
3. Research Plan
4. Experimental results
   - Discrete maze
   - Stochastic optimal control
5. Conclusion

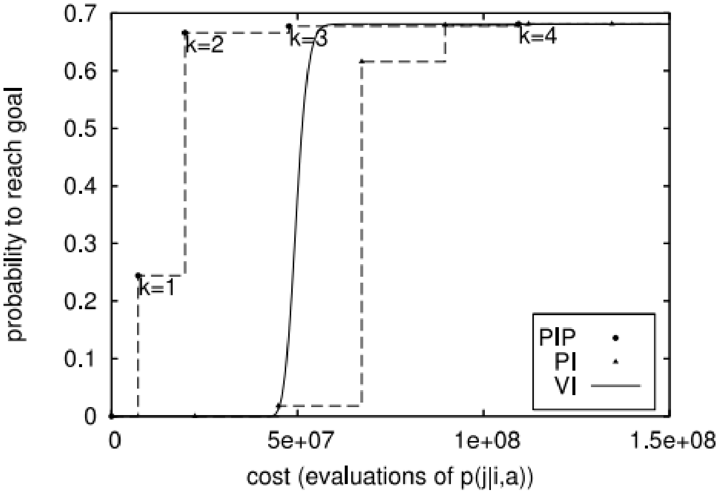
Discrete maze

(a)

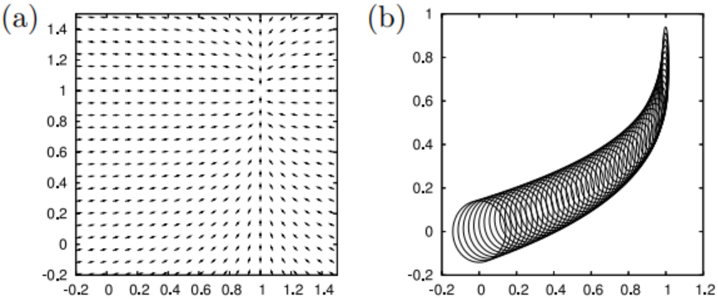


Discrete maze

(b)

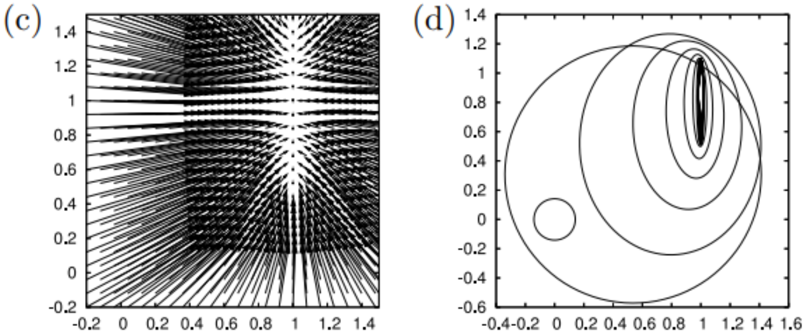


Stochastic optimal control - 'walker'

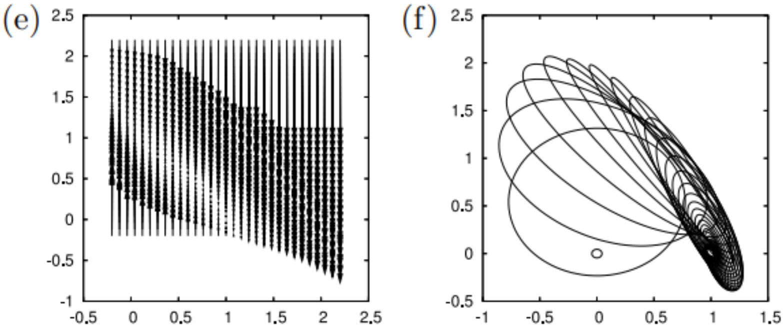


- $P(x' \mid u, x) = \mathcal{N}(x', \phi(u,x), Q + (|u|/\mu)^2 I)$ is *transitions*.
- $\alpha_0(x) = \mathcal{N}(x, (0,0), .01 I)$ is *start-state*.
- $P(r = 1 \mid x) = \mathcal{N}(x, (1,1), diag(.0001, .1))$ is *goal*.
- $\phi(u,x) = x + .1u$ is *control-law*.

Stochastic optimal control - 'golfer'



Stochastic optimal control - 'phase space'



1. Introduction
2. Research Problem
3. Research Plan
4. Experimental results
5. Conclusion

Conclusion

We present a model that translates the problem of planning into a problem of probabilistic inference.

Conclusion

We present a model that translates the problem of planning into a problem of probabilistic inference.

Main contributions

1. we do not have to fix a total time
2. likelihood maximization is equivalent to maximization of the expected future return

Conclusion

We present a model that translates the problem of planning into a problem of probabilistic inference.

Main contributions

1. we do not have to fix a total time
2. likelihood maximization is equivalent to maximization of the expected future return

We can compute posteriors over actions, states, and the total time.

Conclusion

We present a model that translates the problem of planning into a problem of probabilistic inference.

Main contributions

1. we do not have to fix a total time
2. likelihood maximization is equivalent to maximization of the expected future return

We can compute posteriors over actions, states, and the total time.

The full variety of existing inference techniques can be applied to solving MDPs.

End

Thank You