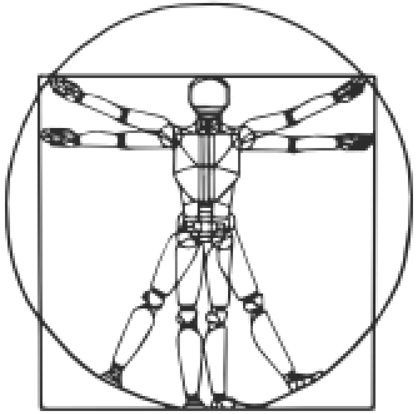


CSC2621
Imitation Learning for Robotics

Florian Shkurti

Week 10: Inverse Reinforcement Learning (Part II)

Today's agenda

- Guided Cost Learning by Finn, Levine, Abbeel
- Inverse KKT by Englert, Vien, Toussaint
- Bayesian Inverse RL by Ramachandran and Amir
- Max Margin Planning by Ratliff, Zinkevitch, and Bagnell

Today's agenda

- Guided Cost Learning by Finn, Levine, Abbeel
- Inverse KKT by Englert, Vien, Toussaint
- Bayesian Inverse RL by Ramachandran and Amir
- Max Margin Planning by Ratliff, Zinkevitch, and Bagnell

Recall: Maximum Entropy IRL [Ziebart et al. 2008]

$$p(\tau|\theta) = ?$$

$$\begin{cases} \underset{p(\tau|\theta)}{\operatorname{argmax}} \, \mathcal{H}(p) \\ \text{subject to} \quad \sum_{\tau} p(\tau|\theta) = 1 \\ \qquad\qquad\quad \mathbb{E}_{\tau \sim p(\tau|\theta)}\left[\mathbf{f}_\tau\right] = \frac{1}{|D|} \sum_{\tau \in D} \mathbf{f}_\tau \end{cases}$$

Assumption: Trajectories (states and action sequences) here are discrete

Recall: Maximum Entropy IRL [Ziebart et al. 2008]

$$p(\tau|\theta) = \frac{\exp(\theta^\top \mathbf{f}_\tau)}{Z(\theta)}$$

$$\begin{cases} \underset{p(\tau|\theta)}{\text{argmax}}\ \mathcal{H}(p) \\ \text{subject to} \quad \sum_{\tau} p(\tau|\theta) = 1 \\ \qquad \mathbb{E}_{\tau \sim p(\tau|\theta)}[\mathbf{f}_\tau] = \frac{1}{|D|} \sum_{\tau \in D} \mathbf{f}_\tau \end{cases}$$

Linear Reward Function

$$R_\theta(\tau) = \theta^\top \mathbf{f}_\tau$$

Recall: Maximum Entropy IRL [Ziebart et al. 2008]

$$p(\tau|\theta) = \frac{\exp(\theta^\top \mathbf{f}_\tau)}{Z(\theta)}$$

Linear Reward Function

$$R_\theta(\tau) = \theta^\top \mathbf{f}_\tau$$

$$p(\tau|\theta) = p(x_0) \prod_{t=0}^{T-1} p(x_{t+1}|x_t, u_t)\ \pi_\theta(u_t|x_t) = \frac{\exp(R_\theta(\tau))}{Z(\theta)}$$

Recall: Maximum Entropy IRL [Ziebart et al. 2008]

$$p(\tau|\theta) = \frac{\exp(\theta^\top \mathbf{f}_\tau)}{Z(\theta)}$$

$$p(\tau|\theta) = p(x_0) \prod_{t=0}^{T-1} \cancel{p(x_{t+1}|x_t, u_t)}\ \pi_\theta(u_t|x_t) = \frac{\exp(R_\theta(\tau))}{Z(\theta)}$$

Assumption: known and deterministic dynamics

Linear Reward Function

$$R_\theta(\tau) = \theta^\top \mathbf{f}_\tau$$

Recall: Maximum Entropy IRL [Ziebart et al. 2008]

$$p(\tau|\theta) = \frac{\exp(\theta^\top \mathbf{f}_\tau)}{Z(\theta)}$$

$$p(\tau|\theta) = p(x_0) \prod_{t=0}^{T-1} \cancel{p(x_{t+1}|x_t, u_t)}\ \pi_\theta(u_t|x_t) = \frac{\exp(R_\theta(\tau))}{Z(\theta)}$$

Assumption: known and deterministic dynamics

Linear Reward Function

$$R_\theta(\tau) = \theta^\top \mathbf{f}_\tau$$

Log-likelihood of observed dataset D of trajectories

$$L(\theta) = \frac{1}{|D|} \sum_{\tau \in D} \log p(\tau|\theta) = \frac{1}{|D|} \sum_{\tau \in D} \theta^\top \mathbf{f}_\tau - \log Z(\theta)$$

Recall: Maximum Entropy IRL [Ziebart et al. 2008]

$$p(\tau|\theta) = \frac{\exp(\theta^\top \mathbf{f}_\tau)}{Z(\theta)}$$

Linear Reward Function

$$R_\theta(\tau) = \theta^\top \mathbf{f}_\tau$$

$$p(\tau|\theta) = p(x_0) \prod_{t=0}^{T-1} \cancel{p(x_{t+1}|x_t, u_t)}\ \pi_\theta(u_t|x_t) = \frac{\exp(R_\theta(\tau))}{Z(\theta)}$$

Assumption: known and deterministic dynamics

Log-likelihood of observed dataset D of trajectories

$$L(\theta) = \frac{1}{|D|} \sum_{\tau \in D} \log p(\tau|\theta) = \frac{1}{|D|} \sum_{\tau \in D} \theta^\top \mathbf{f}_\tau - \log Z(\theta)$$

$$\nabla_\theta L(\theta) = \frac{1}{|D|} \sum_{\tau \in D} \mathbf{f}_\tau - \sum_\tau p(\tau|\theta)\ \mathbf{f}_\tau$$

Recall: Maximum Entropy IRL [Ziebart et al. 2008]

$$p(\tau|\theta) = \frac{\exp(\theta^\top \mathbf{f}_\tau)}{Z(\theta)}$$

$$p(\tau|\theta) = p(x_0) \prod_{t=0}^{T-1} \cancel{p(x_{t+1}|x_t, u_t)} \, \pi_\theta(u_t|x_t) = \frac{\exp(R_\theta(\tau))}{Z(\theta)}$$

Assumption: known and deterministic dynamics

Linear Reward Function

$$R_\theta(\tau) = \theta^\top \mathbf{f}_\tau$$

Hand-Engineered Features

Log-likelihood of observed dataset D of trajectories

$$L(\theta) = \frac{1}{|D|} \sum_{\tau \in D} \log p(\tau|\theta) = \frac{1}{|D|} \sum_{\tau \in D} \theta^\top \mathbf{f}_\tau - \log Z(\theta)$$

$$\nabla_\theta L(\theta) = \frac{1}{|D|} \sum_{\tau \in D} \mathbf{f}_\tau - \sum_\tau p(\tau|\theta) \, \mathbf{f}_\tau$$

Serious problem: Need to compute Z(theta) every time we compute the gradient

Guided Cost Learning [Finn, Levine, Abbeel et al. 2016]

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

$$p(\tau|\theta) = p(x_0) \prod_{t=0}^{T-1} \underbrace{p(x_{t+1}|x_t, u_t)}_{\text{True and stochastic dynamics (unknown)}} \pi_\theta(u_t|x_t) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Log-likelihood of observed dataset D of trajectories

$$L(\theta) = \frac{1}{|D|} \sum_{\tau \in D} \log p(\tau|\theta) = \frac{1}{|D|} \sum_{\tau \in D} -c_\theta(\tau) - \log Z(\theta)$$

Guided Cost Learning [Finn, Levine, Abbeel et al. 2016]

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

$$p(\tau|\theta) = p(x_0) \prod_{t=0}^{T-1} \underbrace{p(x_{t+1}|x_t, u_t)}_{\text{True and stochastic dynamics (unknown)}} \pi_\theta(u_t|x_t) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Log-likelihood of observed dataset D of trajectories

$$L(\theta) = \frac{1}{|D|} \sum_{\tau \in D} \log p(\tau|\theta) = \frac{1}{|D|} \sum_{\tau \in D} -c_\theta(\tau) - \log Z(\theta)$$

$$\nabla_\theta L(\theta) = -\frac{1}{|D|} \sum_{\tau \in D} \nabla_\theta c_\theta(\tau) + \sum_\tau p(\tau|\theta) \nabla_\theta c_\theta(\tau)$$

Serious problem remains

Approximating the gradient of the log-likelihood

$$\nabla_\theta\, L(\theta) = -\frac{1}{|D|}\sum_{\tau\in D}\nabla_\theta\, c_\theta(\tau) + \underbrace{\sum_{\tau}\, p(\tau|\theta)\,\nabla_\theta c_\theta(\tau)}_{\text{How do you approximate this expectation?}}$$

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

Approximating the gradient of the log-likelihood

$$\nabla_\theta L(\theta) = -\frac{1}{|D|}\sum_{\tau\in D}\nabla_\theta\, c_\theta(\tau) + \underbrace{\sum_\tau p(\tau|\theta)\,\nabla_\theta c_\theta(\tau)}$$

How do you approximate this expectation?

Idea #1: sample from $p(\tau|\theta)$
(can you do this?)

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

Approximating the gradient of the log-likelihood

$$\nabla_\theta L(\theta) = -\frac{1}{|D|}\sum_{\tau \in D} \nabla_\theta\, c_\theta(\tau) + \underbrace{\sum_{\tau} p(\tau|\theta)\, \nabla_\theta c_\theta(\tau)}$$

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

How do you approximate this expectation?

Idea #1: sample from $p(\tau|\theta)$
(don't know the dynamics ☹)

Approximating the gradient of the log-likelihood

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

$$\nabla_\theta L(\theta) = -\frac{1}{|D|}\sum_{\tau \in D} \nabla_\theta c_\theta(\tau) + \underbrace{\sum_\tau p(\tau|\theta) \nabla_\theta c_\theta(\tau)}$$

How do you approximate this expectation?

Idea #1: sample from $p(\tau|\theta)$
(don't know the dynamics ☹)

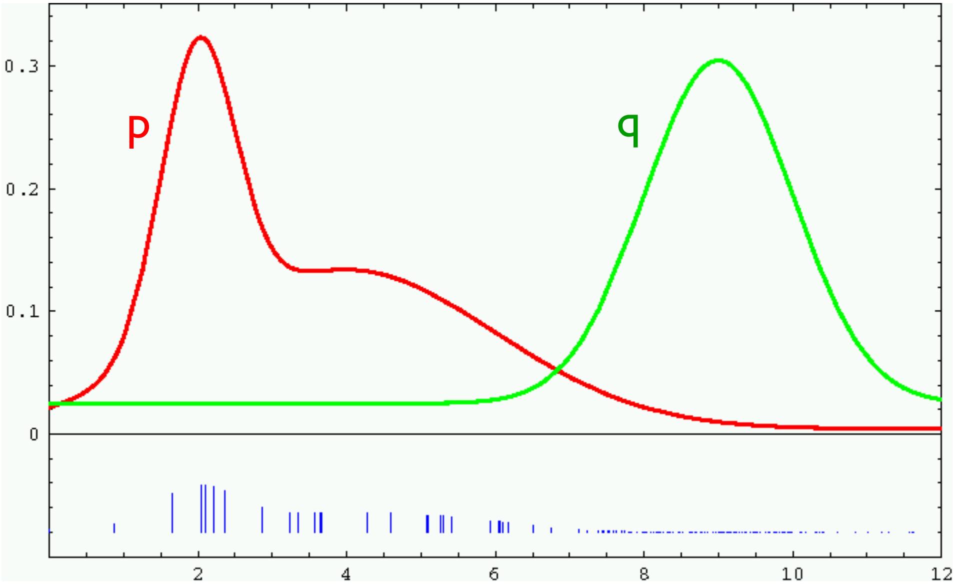
Idea #2: sample from an easier distribution $q(\tau|\theta)$
that approximates $p(\tau|\theta)$

Importance Sampling

see Relative Entropy Inverse RL
by Boularias, Kober, Peters

Importance Sampling

How to estimate properties/statistics of one distribution (p) given samples from another distribution (q)

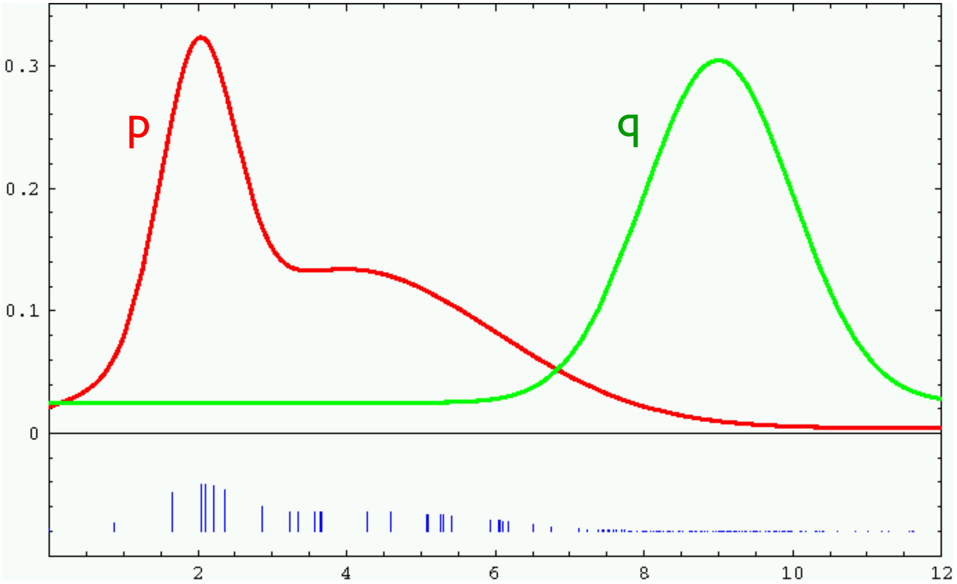


$$\begin{aligned}
\mathbb{E}_{x\sim p(x)}[f(x)] &= \int f(x)p(x)dx \\
&= \int \frac{q(x)}{q(x)} f(x)p(x)dx \\
&= \int \frac{f(x)p(x)}{q(x)} q(x)dx \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\frac{p(x)}{q(x)}\right] \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\; w(x)\right]
\end{aligned}$$

Weights = likelihood ratio,
i.e. how to reweigh samples to obtain statistics of p from samples of q

Importance Sampling: Pitfalls and Drawbacks

What can go wrong?



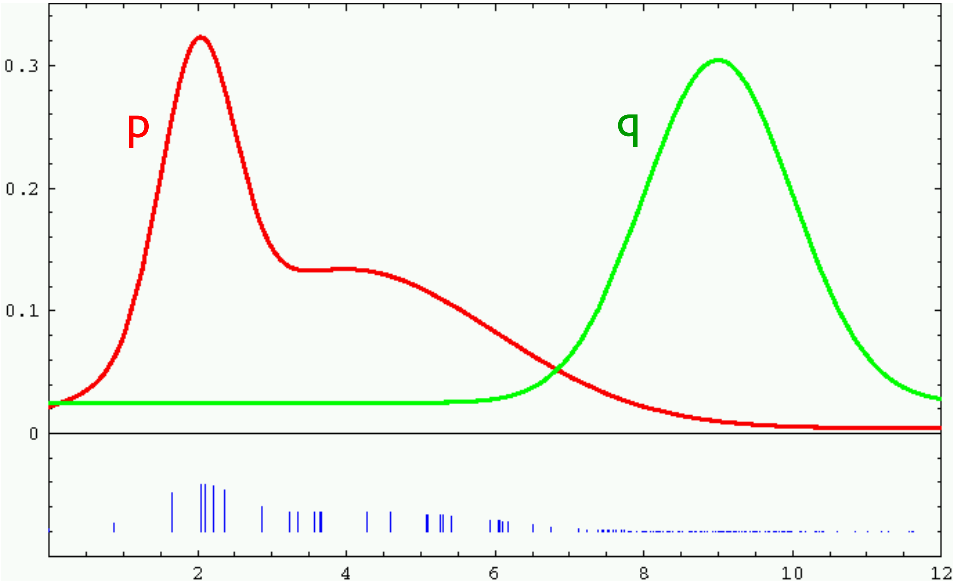
$$
\begin{aligned}
\mathbb{E}_{x\sim p(x)}[f(x)] &= \int f(x)p(x)dx \\
&= \int \frac{q(x)}{q(x)} f(x)p(x)dx \\
&= \int \frac{f(x)p(x)}{q(x)} q(x)dx \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\frac{p(x)}{q(x)}\right] \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\ w(x)\right]
\end{aligned}
$$

Problem #1:

If q(x) = 0 but f(x)p(x) > 0 for x in non-measure-zero set then there is estimation bias

Importance Sampling: Pitfalls and Drawbacks

What can go wrong?



$$\begin{aligned}
\mathbb{E}_{x\sim p(x)}[f(x)] &= \int f(x)p(x)dx \\
&= \int \frac{q(x)}{q(x)} f(x)p(x)dx \\
&= \int \frac{f(x)p(x)}{q(x)} q(x)dx \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\frac{p(x)}{q(x)}\right] \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\ w(x)\right]
\end{aligned}$$

Problem #1:

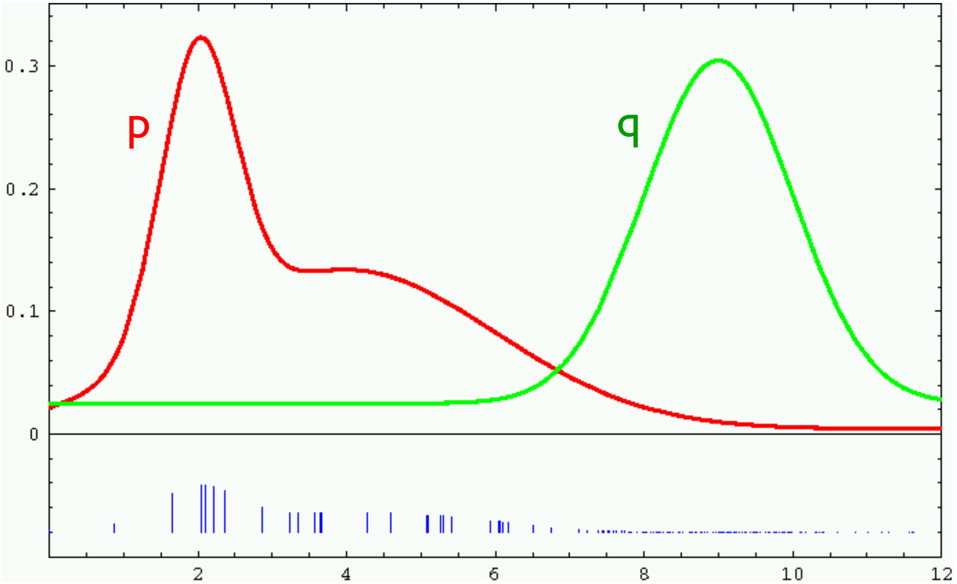
If q(x) = 0 but f(x)p(x) > 0 for x in non-measure-zero set then there is estimation bias

Problem #2:

Weights measure mismatch between q(x) and p(x). If mismatch is large then some weights will dominate. If x lives in high dimensions a single weight may dominate

Importance Sampling: Pitfalls and Drawbacks

What can go wrong?



$$
\begin{aligned}
\mathbb{E}_{x\sim p(x)}[f(x)] &= \int f(x)p(x)dx \\
&= \int \frac{q(x)}{q(x)} f(x)p(x)dx \\
&= \int \frac{f(x)p(x)}{q(x)} q(x)dx \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\frac{p(x)}{q(x)}\right] \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\ w(x)\right]
\end{aligned}
$$

Problem #1:

If q(x) = 0 but f(x)p(x) > 0 for x in non-measure-zero set then there is estimation bias

Problem #2:

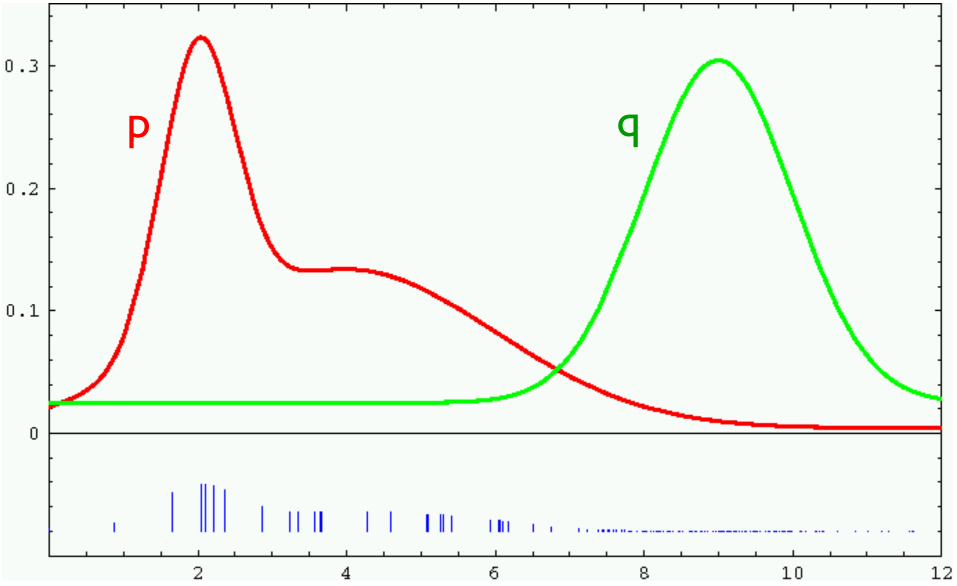
Weights measure mismatch between q(x) and p(x). If mismatch is large then some weights will dominate. If x lives in high dimensions a single weight may dominate

Problem #3:

Variance of estimator is high if (q – fp)(x) is high

Importance Sampling: Pitfalls and Drawbacks

What can go wrong?



$$
\begin{aligned}
\mathbb{E}_{x\sim p(x)}[f(x)] &= \int f(x)p(x)dx \\
&= \int \frac{q(x)}{q(x)} f(x)p(x)dx \\
&= \int \frac{f(x)p(x)}{q(x)} q(x)dx \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\frac{p(x)}{q(x)}\right] \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\; w(x)\right]
\end{aligned}
$$

Problem #1:

If q(x) = 0 but f(x)p(x) > 0 for x in non-measure-zero set then there is estimation bias

Problem #2:

Weights measure mismatch between q(x) and p(x). If mismatch is large then some weights will dominate. If x lives in high dimensions a single weight may dominate

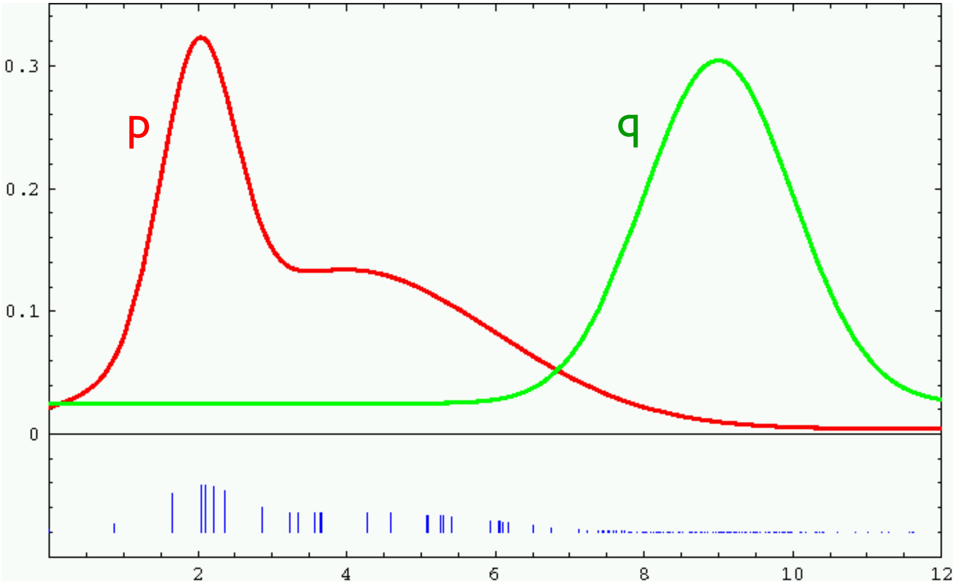
Problem #3:

Variance of estimator is high if (q – fp)(x) is high

For more info see:
#1, #3: Monte Carlo theory, methods, and examples, Art Owen, chapter 9
#2: Bayesian reasoning and machine learning, David Barber, chapter 27.6 on importance sampling

Importance Sampling

What is the best approximating distribution q?

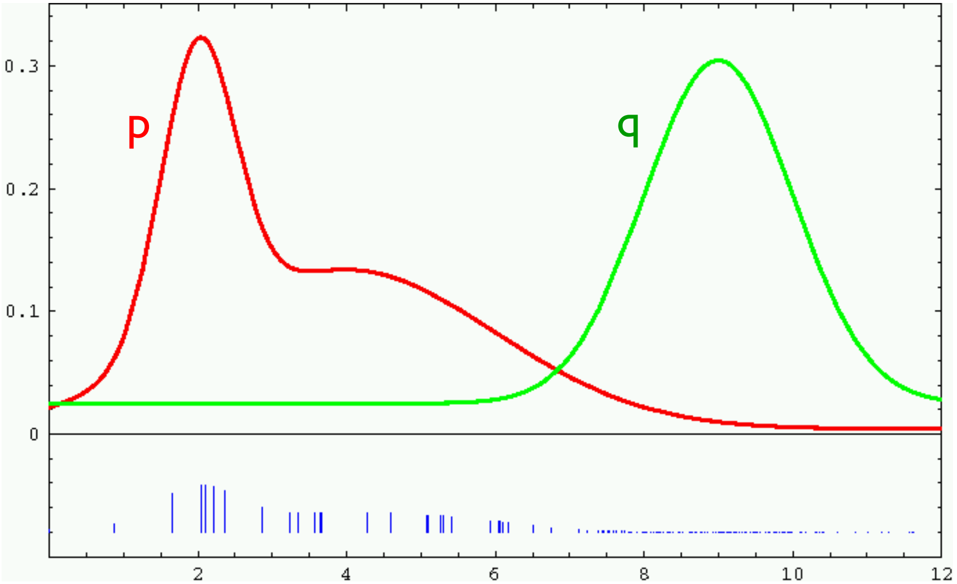


$$
\begin{aligned}
\mathbb{E}_{x\sim p(x)}[f(x)] &= \int f(x)p(x)dx \\
&= \int \frac{q(x)}{q(x)} f(x)p(x)dx \\
&= \int \frac{f(x)p(x)}{q(x)} q(x)dx \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\frac{p(x)}{q(x)}\right] \\
&= \mathbb{E}_{x\sim q(x)}\left[f(x)\ w(x)\right]
\end{aligned}
$$

Best approximation $q(x) \propto f(x)p(x)$

Importance Sampling

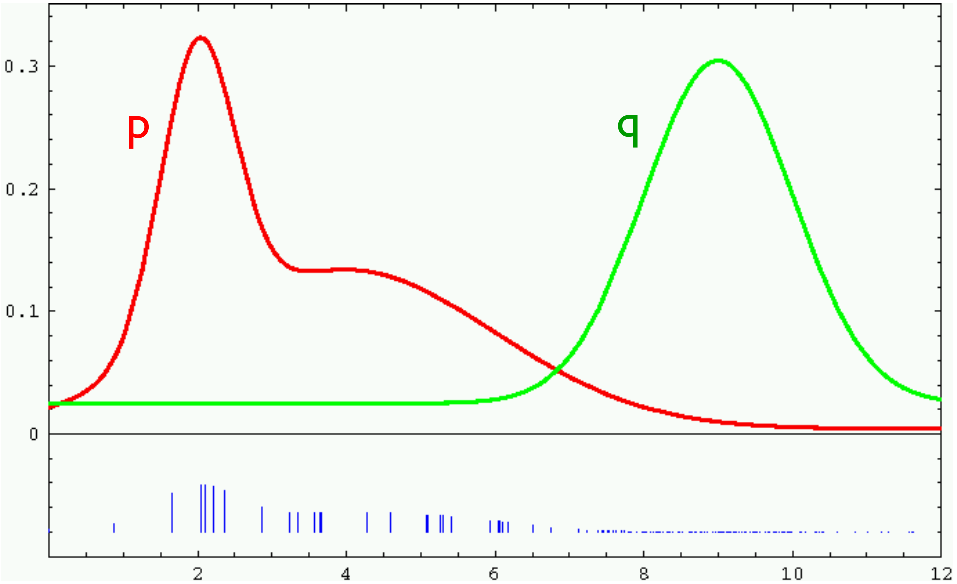
How does this connect back to partition function estimation?



$$
\begin{aligned}
Z(\theta) &= \sum_{\tau} \exp(-c_\theta(\tau)) \\
&= \sum_{\tau} \exp(-c_\theta(\tau)) \\
&= \sum_{\tau} \frac{q(\tau|\theta)}{q(\tau|\theta)} \exp(-c_\theta(\tau)) \\
&= \mathbb{E}_{\tau \sim q(\tau|\theta)} \left[\frac{\exp(-c_\theta(\tau))}{q(\tau|\theta)} \right]
\end{aligned}
$$

Importance Sampling

How does this connect back to partition function estimation?

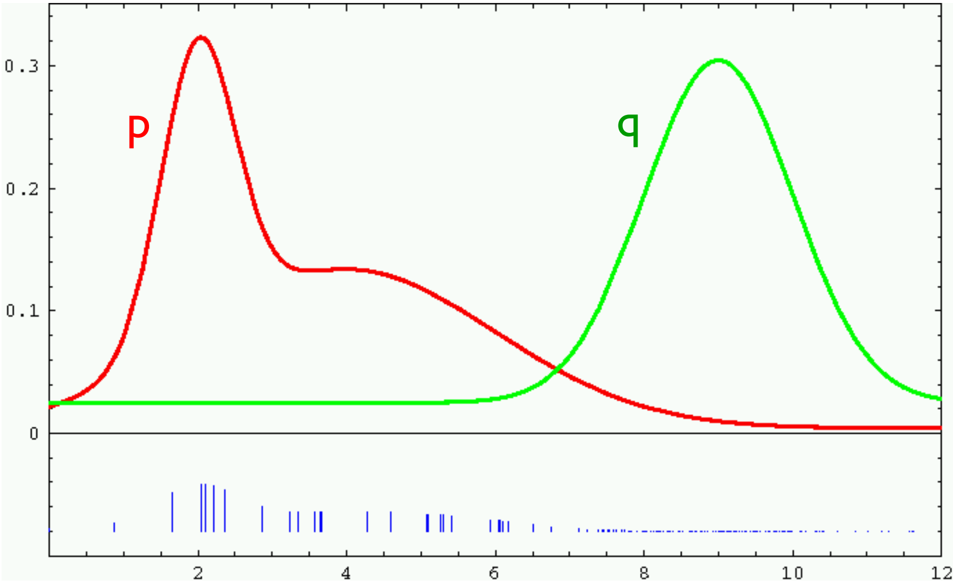


$$
\begin{aligned}
Z(\theta) &= \sum_{\tau} \exp(-c_\theta(\tau)) \\
&= \sum_{\tau} \exp(-c_\theta(\tau)) \\
&= \sum_{\tau} \frac{q(\tau|\theta)}{q(\tau|\theta)} \exp(-c_\theta(\tau)) \\
&= \mathbb{E}_{\tau \sim q(\tau|\theta)} \left[\frac{\exp(-c_\theta(\tau))}{q(\tau|\theta)} \right]
\end{aligned}
$$

Best approximating distribution $q(\tau|\theta) \propto \exp(-c_\theta(\tau))$

Importance Sampling

How does this connect back to partition function estimation?



$$
\begin{aligned}
Z(\theta) &= \sum_{\tau} \exp(-c_\theta(\tau)) \\
&= \sum_{\tau} \exp(-c_\theta(\tau)) \\
&= \sum_{\tau} \frac{q(\tau|\theta)}{q(\tau|\theta)} \exp(-c_\theta(\tau)) \\
&= \mathbb{E}_{\tau \sim q(\tau|\theta)} \left[\frac{\exp(-c_\theta(\tau))}{q(\tau|\theta)} \right]
\end{aligned}
$$

Best approximating distribution $q(\tau|\theta) \propto \exp(-c_\theta(\tau))$

Cost function estimate changes at each gradient step
Therefore the best approximating distribution should change as well

Approximating the gradient of the log-likelihood

$$\nabla_\theta L(\theta) = -\frac{1}{|D|}\sum_{\tau \in D} \nabla_\theta\, c_\theta(\tau) + \underbrace{\sum_{\tau} p(\tau|\theta)\, \nabla_\theta c_\theta(\tau)}$$

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

How do you approximate this expectation?

Idea #1: sample from $p(\tau|\theta)$
(don't know the dynamics ☹)

Idea #2: sample from an easier distribution $q(\tau|\theta)$
that approximates $p(\tau|\theta)$

Importance Sampling

see Relative Entropy Inverse RL
by Boularias, Kober, Peters

Approximating the gradient of the log-likelihood

$$\nabla_\theta L(\theta) = -\frac{1}{|D|}\sum_{\tau\in D}\nabla_\theta\, c_\theta(\tau) + \underbrace{\sum_{\tau} p(\tau|\theta)\, \nabla_\theta c_\theta(\tau)}$$

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

How do you approximate this expectation?

Idea #1: sample from $p(\tau|\theta)$
(don't know the dynamics ☹)

Idea #2: sample from an easier distribution $q(\tau|\theta)$
that approximates $p(\tau|\theta)$

Previous papers used a fixed $q(\tau|\theta)$

Importance Sampling
see Relative Entropy Inverse RL by Boularias, Kober, Peters

Approximating the gradient of the log-likelihood

$$\nabla_\theta L(\theta) = -\frac{1}{|D|}\sum_{\tau\in D}\nabla_\theta\, c_\theta(\tau) + \underbrace{\sum_{\tau} p(\tau|\theta)\, \nabla_\theta c_\theta(\tau)}$$

$$p(\tau|\theta) = \frac{\exp(-c_\theta(\tau))}{Z(\theta)}$$

Nonlinear Reward Function
Learned Features

How do you approximate this expectation?

Idea #1: sample from $p(\tau|\theta)$
(don't know the dynamics ☹)

Idea #2: sample from an easier distribution $q(\tau|\theta)$
that approximates $p(\tau|\theta)$

Previous papers used a fixed $q(\tau|\theta)$

Importance Sampling
see Relative Entropy Inverse RL
by Boularias, Kober, Peters

This paper uses adaptive $q(\tau|\theta)$

Adaptive Importance Sampling
see Guided Cost Learning
By Finn, Levine, Abbeel

Guided Cost Learning

How do you select q?

How do you adapt it as the cost c changes?

Guided Cost Learning: the punchline

How do you select q?

How do you adapt it as the cost c changes?

Given a fixed cost function c, the distribution of trajectories that Guided Policy Search computes is close to $\frac{\exp(-c(\tau))}{Z}$
i.e. it is good for importance sampling of the partition function Z

Recall: Finite-Horizon LQR

$$P_0 = Q$$

// n is the # of steps left

for n = 1…N

$$K_n = -(R + B^T P_{n-1} B)^{-1} B^T P_{n-1} A$$

$$P_n = Q + K_n^T R K_n + (A + BK_n)^T P_{n-1} (A + BK_n)$$

Optimal control for time t = N – n is $\mathbf{u}_t = K_t \mathbf{x}_t$ with cost-to-go $J_t(\mathbf{x}) = \mathbf{x}^T P_t \mathbf{x}$

where the states are predicted forward in time according to linear dynamics

Recall: LQG = LQR with stochastic dynamics

Assume $\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{w}_t$ and $c(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^T Q \mathbf{x}_t + \mathbf{u}_t^T R \mathbf{u}_t$

zero mean Gaussian

Then the form of the optimal policy is the same as in LQR $\mathbf{u}_t = K_t \hat{\mathbf{x}}_t$ estimate of the state

No need to change the algorithm, as long as you observe the state at each step (closed-loop policy)

Linear Quadratic Gaussian
LQG

Deterministic Nonlinear Cost & Deterministic Nonlinear Dynamics

$$
\begin{aligned}
u_0^*, \ldots, u_{N-1}^* = \quad & \arg\min_{u_0, \ldots, u_N} \quad \sum_{t=0}^{N} c(\mathbf{x}_t, \mathbf{u}_t) \\
\text{s.t.} \\
\mathbf{x}_1 &= f(\mathbf{x}_0, \mathbf{u}_0) \\
\mathbf{x}_2 &= f(\mathbf{x}_1, \mathbf{u}_1) \\
& \ldots \\
\mathbf{x}_N &= f(\mathbf{x}_{N-1}, \mathbf{u}_{N-1})
\end{aligned}
$$

Arbitrary differentiable functions c, f

iLQR: iteratively approximate solution by solving linearized versions of the problem via LQR

Deterministic Nonlinear Cost & Stochastic Nonlinear Dynamics

$$u_0^*, \ldots, u_{N-1}^* = \arg\min_{u_0, \ldots, u_N} \sum_{t=0}^{N} c(\mathbf{x}_t, \mathbf{u}_t)$$

$$\begin{aligned} \text{s.t.} \\ \mathbf{x}_1 &= f(\mathbf{x}_0, \mathbf{u}_0) + \mathbf{w}_0 \\ \mathbf{x}_2 &= f(\mathbf{x}_1, \mathbf{u}_1) + \mathbf{w}_1 \\ &\ldots \\ \mathbf{x}_N &= f(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) + \mathbf{w}_{N-1} \end{aligned}$$

Arbitrary differentiable functions c, f

$\mathbf{w}_t \sim \mathcal{N}(0, W_t)$

iLQG: iteratively approximate solution by solving linearized versions of the problem via LQG

Recall from Guided Policy Search

$$
\begin{aligned}
\underset{q(\tau)}{\operatorname{argmin}} \quad & \mathbb{E}_{\tau \sim q(\tau)}\left[c(\tau)\right] \\
\text{subject to} \quad & q(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t) \\
& \mathrm{KL}(q(\tau) \,||\, q_{\text{prev}}(\tau)) \leq \epsilon
\end{aligned}
$$

Learn linear Gaussian dynamics

Recall from Guided Policy Search

$$\begin{aligned} \underset{q(\tau)}{\text{argmin}} \quad & \mathbb{E}_{\tau \sim q(\tau)}\left[c(\tau)\right] \\ \text{subject to} \quad & q(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t) \\ & \mathrm{KL}(q(\tau) \mid\mid q_{\text{prev}}(\tau)) \leq \epsilon \end{aligned}$$

Learn linear Gaussian dynamics

$$\begin{aligned} q_{\text{gps}}(\tau) = \underset{q(\tau)}{\text{argmin}} \quad & \mathbb{E}_{\tau \sim q(\tau)}\left[c(\tau)\right] - \mathcal{H}(q(\tau)) \\ \text{subject to} \quad & q(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t) \end{aligned}$$

Recall from Guided Policy Search

$$
\begin{aligned}
\underset{q(\tau)}{\text{argmin}} \quad & \mathbb{E}_{\tau \sim q(\tau)} [c(\tau)] \\
\text{subject to} \quad & q(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t) \\
& \text{KL}(q(\tau) \,||\, q_{\text{prev}}(\tau)) \leq \epsilon
\end{aligned}
$$

Learn linear Gaussian dynamics

$$
\begin{aligned}
q_{\text{gps}}(\tau) = \underset{q(\tau)}{\text{argmin}} \quad & \mathbb{E}_{\tau \sim q(\tau)} [c(\tau)] - \mathcal{H}(q(\tau)) \\
\text{subject to} \quad & q(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t)
\end{aligned}
$$

$$
q_{\text{gps}}(\tau) = q(\mathbf{x}_0) \prod_{t=0}^{T-1} q(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) q(\mathbf{u}_t | \mathbf{x}_t)
$$

Linear Gaussian
dynamics and controller

Recall from Guided Policy Search

$$\begin{aligned} \underset{q(\tau)}{\text{argmin}} \quad & \mathbb{E}_{\tau \sim q(\tau)} [c(\tau)] \\ \text{subject to} \quad & q(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t) \\ & \text{KL}(q(\tau) \,||\, q_{\text{prev}}(\tau)) \leq \epsilon \end{aligned}$$

Learn linear Gaussian dynamics

$$\begin{aligned} q_{\text{gps}}(\tau) = \underset{q(\tau)}{\text{argmin}} \quad & \mathbb{E}_{\tau \sim q(\tau)} [c(\tau)] - \mathcal{H}(q(\tau)) \\ \text{subject to} \quad & q(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t) \end{aligned}$$

$$q_{\text{prev}} = q_{\text{gps}}$$

$$q_{\text{gps}}(\tau) = q(\mathbf{x}_0) \prod_{t=0}^{T-1} q(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) q(\mathbf{u}_t | \mathbf{x}_t)$$

Linear Gaussian
dynamics and controller

Run controller on the robot
Collect trajectories

Recall from Guided Policy Search

$$\begin{aligned} \underset{q(\tau)}{\operatorname{argmin}} \quad & \mathbb{E}_{\tau \sim q(\tau)}\left[c(\tau)\right] \\ \text{subject to} \quad & q(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t) \\ & \mathrm{KL}(q(\tau) \,||\, q_{\text{prev}}(\tau)) \leq \epsilon \end{aligned}$$

Learn linear Gaussian dynamics

$$\begin{aligned} q_{\text{gps}}(\tau) = \underset{q(\tau)}{\operatorname{argmin}} \quad & \boxed{\mathbb{E}_{\tau \sim q(\tau)}\left[c(\tau)\right] - \mathcal{H}(q(\tau))} \\ \text{subject to} \quad & q(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}; f_{xt}\mathbf{x}_t + f_{ut}\mathbf{u}_t, \mathbf{F}_t) \end{aligned}$$

$$\mathrm{KL}\left(q(\tau) \,||\, \frac{\exp(-c(\tau))}{Z}\right)$$

Given a fixed cost function c, the linear Gaussian controllers that GPS computes induce a distribution of trajectories close to

$$\frac{\exp(-c(\tau))}{Z}$$

i.e. good for importance sampling of the partition function Z

Guided Cost Learning [rough sketch]

Collect demonstration trajectories D
Initialize cost parameters θ_0

Do forward optimization using Guided Policy Search for cost function $c_{\theta_t}(\tau)$
and compute linear Gaussian distribution of trajectories $q_{\text{gps}}(\tau)$

$$\nabla_\theta\, L(\theta) = -\frac{1}{|D|} \sum_{\tau \in D} \nabla_\theta\, c_\theta(\tau) + \underbrace{\sum_\tau\, p(\tau|\theta)\, \nabla_\theta c_\theta(\tau)}_{\text{Importance sample trajectories from } q_{\text{gps}}(\tau)}$$

$$\theta_{t+1} = \theta_t + \gamma \nabla_\theta L(\theta_t)$$

Regularization of learned cost functions

$$g_{\mathrm{lcr}}(\tau) = \sum_{x_t \in \tau} [(c_\theta(x_{t+1}) - c_\theta(x_t)) - (c_\theta(x_t) - c_\theta(x_{t-1}))]^2$$

$$g_{\mathrm{mono}}(\tau) = \sum_{x_t \in \tau} [\max(0, c_\theta(x_t) - c_\theta(x_{t-1}) - 1)]^2$$

Guided Cost Learning:
Deep Inverse Optimal Control via Policy Optimization

Chelsea Finn, Sergey Levine, Pieter Abbeel
UC Berkeley

Today's agenda

- Guided Cost Learning by Finn, Levine, Abbeel
- Inverse KKT by Englert, Vien, Toussaint
- Bayesian Inverse RL by Ramachandran and Amir
- Max Margin Planning by Ratliff, Zinkevitch, and Bagnell

Setting up trajectory optimization problems

[e.g. for manipulation]

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$\Phi(\mathbf{x}_t)$ Non-learned features of the current state, e.g. distance to object

Features and their weights are time-dependent

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\operatorname{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Setting up trajectory optimization problems

[e.g. for manipulation]

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$\Phi(\mathbf{x}_t)$ Non-learned features of the current state, e.g. distance to object

Features and their weights are time-dependent

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\operatorname{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Constraints such as "stay away from an obstacle", or "acceleration should be bounded"

Setting up trajectory optimization problems

[e.g. for manipulation]

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$\Phi(\mathbf{x}_t)$ Non-learned features of the current state, e.g. distance to object

Features and their weights are time-dependent

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\operatorname{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Constraints such as "stay away from an obstacle", or "acceleration should be bounded"

Constraints such as "always touch the door handle"

Solving constrained optimization problems

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\operatorname{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Lagrangian function for this problem:

$$L_\theta(\mathbf{x}_{0:T}, \lambda) = c_\theta(\mathbf{x}_{0:T}) + \lambda^\top \begin{bmatrix} g(\mathbf{x}_{0:T}) \\ h(\mathbf{x}_{0:T}) \end{bmatrix}$$

KKT conditions for trajectory optimization

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\text{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Lagrangian function for this problem:

$$L_\theta(\mathbf{x}_{0:T}, \lambda) = c_\theta(\mathbf{x}_{0:T}) + \lambda^\top \begin{bmatrix} g(\mathbf{x}_{0:T}) \\ h(\mathbf{x}_{0:T}) \end{bmatrix}$$

One of the necessary conditions for optimal motion $\mathbf{x}^*_{0:T}$

$$\nabla_{\mathbf{x}_{0:T}} L_\theta(\mathbf{x}^*_{0:T}, \lambda) = 0$$

$$2J_\Phi(\mathbf{x}^*_{0:T})^\top \, \text{diag}(\theta) \, \Phi(\mathbf{x}^*_{0:T}) + \lambda^\top J_c(\mathbf{x}^*_{0:T}) = 0$$

KKT conditions for trajectory optimization

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\text{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Lagrangian function for this problem:

$$L_\theta(\mathbf{x}_{0:T}, \lambda) = c_\theta(\mathbf{x}_{0:T}) + \lambda^\top \begin{bmatrix} g(\mathbf{x}_{0:T}) \\ h(\mathbf{x}_{0:T}) \end{bmatrix}$$

One of the necessary conditions for optimal motion $\mathbf{x}^*_{0:T}$

$$\nabla_{\mathbf{x}_{0:T}} L_\theta(\mathbf{x}^*_{0:T}, \lambda) = 0$$

$$2J_\Phi(\mathbf{x}^*_{0:T})^\top \underbrace{\text{diag}(\theta)}_{} \Phi(\mathbf{x}^*_{0:T}) + \lambda^\top J_c(\mathbf{x}^*_{0:T}) = 0$$

What are the conditions on
the feature weights to ensure
optimality of demonstrated motion?

Inverse KKT conditions: optimality of cost

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\text{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Lagrangian function for this problem:

$$L_\theta(\mathbf{x}_{0:T}, \lambda) = c_\theta(\mathbf{x}_{0:T}) + \lambda^\top \begin{bmatrix} g(\mathbf{x}_{0:T}) \\ h(\mathbf{x}_{0:T}) \end{bmatrix}$$

One of the necessary conditions for optimal motion $\mathbf{x}^*_{0:T}$

$$\nabla_{\mathbf{x}_{0:T}} L_\theta(\mathbf{x}^*_{0:T}, \lambda) = 0$$

$$2J_\Phi(\mathbf{x}^*_{0:T})^\top \underbrace{\text{diag}(\theta)} \Phi(\mathbf{x}^*_{0:T}) + \lambda^\top J_c(\mathbf{x}^*_{0:T}) = 0$$

Minimize $\quad l(\theta) = ||\nabla_{\mathbf{x}_{0:T}} L_\theta(\mathbf{x}^*_{0:T}, \lambda(\theta))||^2$ $\Leftarrow$

What are the conditions on the feature weights to ensure optimality of demonstrated motion?

Inverse KKT conditions: optimality of cost

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\operatorname{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Lagrangian function for this problem:

$$L_\theta(\mathbf{x}_{0:T}, \lambda) = c_\theta(\mathbf{x}_{0:T}) + \lambda^\top \begin{bmatrix} g(\mathbf{x}_{0:T}) \\ h(\mathbf{x}_{0:T}) \end{bmatrix}$$

One of the necessary conditions for optimal motion $\mathbf{x}^*_{0:T}$

$$\nabla_{\mathbf{x}_{0:T}} L_\theta(\mathbf{x}^*_{0:T}, \lambda) = 0$$

$$2 J_\Phi(\mathbf{x}^*_{0:T})^\top \underbrace{\operatorname{diag}(\theta)}\ \Phi(\mathbf{x}^*_{0:T}) + \lambda^\top J_c(\mathbf{x}^*_{0:T}) = 0$$

What are the conditions on the feature weights to ensure optimality of demonstrated motion?

Minimize $\quad l(\theta) = \theta^\top \Lambda(\mathbf{x}^*_{0:T})\theta$

Inverse KKT conditions: optimality of cost

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\operatorname{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Lagrangian function for this problem:

$$L_\theta(\mathbf{x}_{0:T}, \lambda) = c_\theta(\mathbf{x}_{0:T}) + \lambda^\top \begin{bmatrix} g(\mathbf{x}_{0:T}) \\ h(\mathbf{x}_{0:T}) \end{bmatrix}$$

One of the necessary conditions for optimal motion $\mathbf{x}^*_{0:T}$

$$\nabla_{\mathbf{x}_{0:T}} L_\theta(\mathbf{x}^*_{0:T}, \lambda) = 0$$

$$2J_\Phi(\mathbf{x}^*_{0:T})^\top \underbrace{\operatorname{diag}(\theta)} \Phi(\mathbf{x}^*_{0:T}) + \lambda^\top J_c(\mathbf{x}^*_{0:T}) = 0$$

What are the conditions on the feature weights to ensure optimality of demonstrated motion?

$$\begin{aligned} \text{Minimize} \quad & l(\theta) = \theta^\top \Lambda(\mathbf{x}^*_{0:T})\theta \\ \text{subject to} \quad & \theta \geq 0 \end{aligned}$$

Inverse KKT conditions: optimality of cost

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$$\begin{aligned} \underset{\mathbf{x}_{0:T}}{\operatorname{argmin}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Lagrangian function for this problem:

$$L_\theta(\mathbf{x}_{0:T}, \lambda) = c_\theta(\mathbf{x}_{0:T}) + \lambda^\top \begin{bmatrix} g(\mathbf{x}_{0:T}) \\ h(\mathbf{x}_{0:T}) \end{bmatrix}$$

One of the necessary conditions for optimal motion $\mathbf{x}^*_{0:T}$

$$\nabla_{\mathbf{x}_{0:T}} L_\theta(\mathbf{x}^*_{0:T}, \lambda) = 0$$

$$2J_\Phi(\mathbf{x}^*_{0:T})^\top \underbrace{\operatorname{diag}(\theta)} \Phi(\mathbf{x}^*_{0:T}) + \lambda^\top J_c(\mathbf{x}^*_{0:T}) = 0$$

What are the conditions on the feature weights to ensure optimality of demonstrated motion?

$$\begin{aligned} \text{Minimize} \quad & l(\theta) = \theta^\top \Lambda(\mathbf{x}^*_{0:T})\theta \\ \text{subject to} \quad & \theta \geq 0, \quad \sum_i \theta_i = 1 \end{aligned}$$

Inverse KKT conditions: optimality of cost

$$c_\theta(\mathbf{x}_{0:T}) = \sum_{t=0}^{T} \theta_t^\top \Phi^2(\mathbf{x}_t)$$

$$\begin{aligned} \operatorname*{argmin}_{\mathbf{x}_{0:T}} \quad & c_\theta(\mathbf{x}_{0:T}) \\ \text{subject to} \quad & g(\mathbf{x}_{0:T}) \leq 0 \\ & h(\mathbf{x}_{0:T}) = 0 \end{aligned}$$

Lagrangian function for this problem:

$$L_\theta(\mathbf{x}_{0:T}, \lambda) = c_\theta(\mathbf{x}_{0:T}) + \lambda^\top \begin{bmatrix} g(\mathbf{x}_{0:T}) \\ h(\mathbf{x}_{0:T}) \end{bmatrix}$$

One of the necessary conditions for optimal motion $\mathbf{x}^*_{0:T}$

$$\nabla_{\mathbf{x}_{0:T}} L_\theta(\mathbf{x}^*_{0:T}, \lambda) = 0$$

$$2J_\Phi(\mathbf{x}^*_{0:T})^\top \underbrace{\operatorname{diag}(\theta)} \Phi(\mathbf{x}^*_{0:T}) + \lambda^\top J_c(\mathbf{x}^*_{0:T}) = 0$$

What are the conditions on the feature weights to ensure optimality of demonstrated motion?

Quadratic program
Efficient solvers exist (CPLEX, CVXGEN, Gurobi)

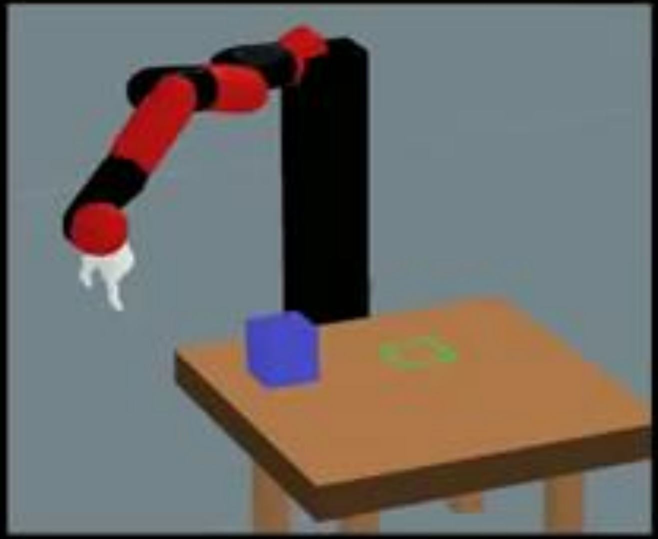
$$\boxed{\begin{aligned} \text{Minimize} \quad & l(\theta) = \theta^\top \Lambda(\mathbf{x}^*_{0:T})\theta \\ \text{subject to} \quad & \theta \geq 0, \quad \sum_i \theta_i = 1 \end{aligned}}$$

Inverse KKT: Learning Cost Functions of Manipulation Tasks from Demonstrations

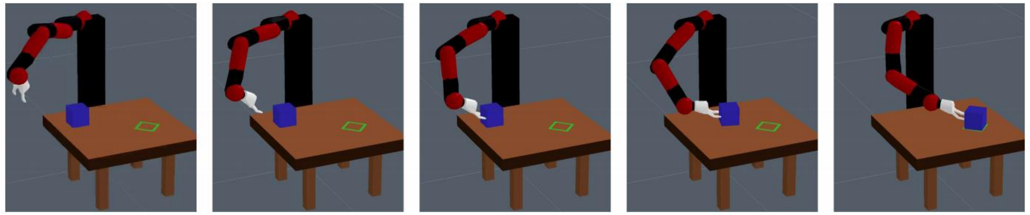
Peter Englert, Marc Toussaint
U Stuttgart

Part 1: Opening a door with PR2

Part 2: Sliding a box in simulation

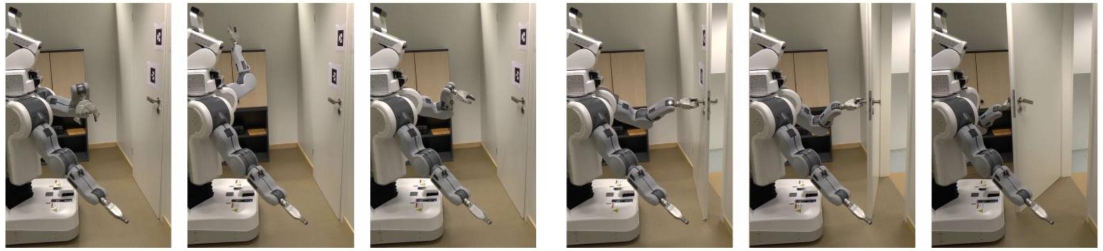


Features

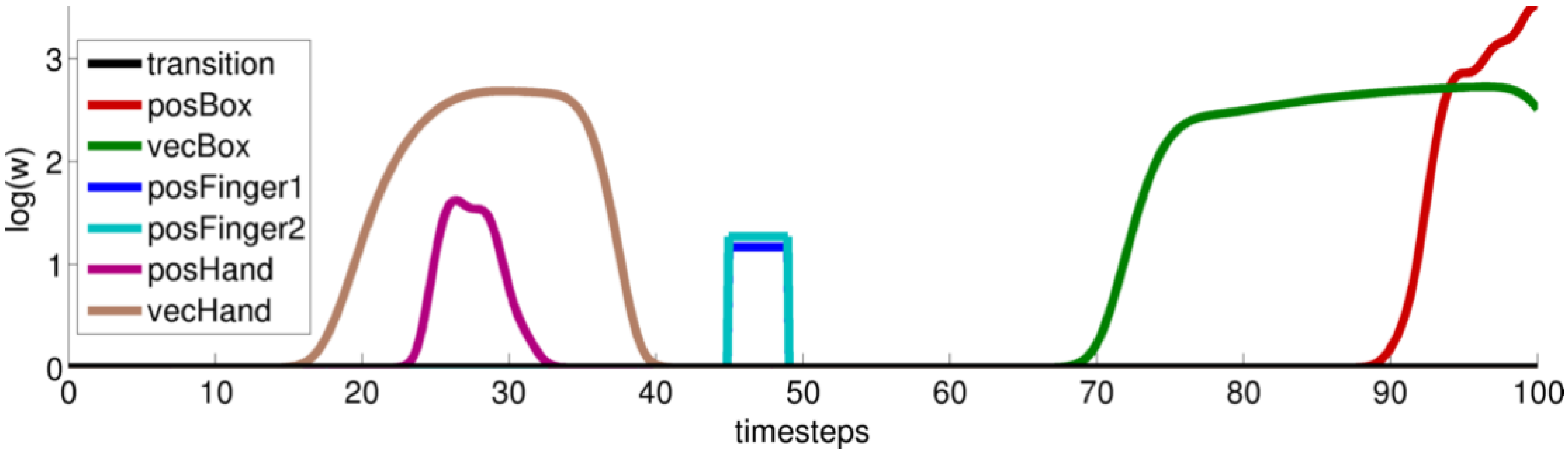


- **transition:** Squared acceleration at each time step in joint space
- **posBox:** Relative position between the box and the target.
- **vecBox:** Relative orientation between the box and the target.
- **posFinger1/2:** Relative position between the robots fingertips and the box.
- **posHand:** Relative position between robot hand and box.
- **vecHand:** Relative orientation between robot hand and box.

- Relative position & orientation between gripper and handle before and after unlocking the handle.
- end-effector orientation during the whole opening motion.
- Position of the final door state.



Feature weights over time



Today's agenda

- Guided Cost Learning by Finn, Levine, Abbeel
- Inverse KKT by Englert, Vien, Toussaint
- Bayesian Inverse RL by Ramachandran and Amir
- Max Margin Planning by Ratliff, Zinkevitch, and Bagnell

Bayesian updates of deterministic rewards

Demonstration trajectory
$\tau = \{(s_0, a_0, s_1, a_1, ..., s_T)\}$

Reward parameters θ

$$Q_\theta^\pi(s,a) = R_\theta(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)}\left[V_\theta^\pi(s)\right]$$

How does our belief in the reward
change after a demonstration?

$$p(\theta|\tau) = \frac{p(\tau|\theta)p(\theta)}{p(\tau)}$$

Bayesian updates of deterministic rewards

Demonstration trajectory
$\tau = \{(s_0, a_0, s_1, a_1, ..., s_T)\}$

Reward parameters θ

$$Q^{\pi}_{\theta}(s,a) = R_{\theta}(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} \left[V^{\pi}_{\theta}(s) \right]$$

How does our belief in the reward change after a demonstration?

$$p(\theta|\tau) = \frac{p(\tau|\theta)p(\theta)}{p(\tau)}$$

In this paper it is assumed that

$$p(\tau|\theta) \propto \exp\left(\eta \sum_{t=0}^{T} Q^{*}_{\theta}(s_t, a_t)\right)$$

MCMC sampling of the posterior

Algorithm $\texttt{PolicyWalk}$(Distribution P, MDP M, Step Size δ)

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 $\boldsymbol{R}$

Figure 3: PolicyWalk Sampling Algorithm

Next candidate reward vector picked randomly from current one

If the optimal policy has changed then do policy iteration starting from the old policy

The paper has results on mixing times for the MCMC walk

Interesting result

4.2 Apprenticeship Learning

For the apprenticeship learning task, the situation is more interesting. Since we are attempting to learn a policy π, we can formally define the following class of *policy loss functions*:

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

where $\boldsymbol{V}^*(\boldsymbol{R})$ is the vector of optimal values for each state acheived by the optimal policy for $\boldsymbol{R}$ and p is some norm.

Theorem 3. *Given a distribution $P(\boldsymbol{R})$ over reward functions $\mathbf{R}$ 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 $M = (S, A, T, \gamma, E_P[\boldsymbol{R}])$.*

Today's agenda

- Guided Cost Learning by Finn, Levine, Abbeel
- Inverse KKT by Englert, Vien, Toussaint
- Bayesian Inverse RL by Ramachandran and Amir
- Max Margin Planning by Ratliff, Zinkevitch, and Bagnell

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

Assumptions:

- Linear rewards with respect to handcrafted features
- Discrete states and actions

Main idea: (*reward weights should be such that) demonstrated trajectories collect more reward than any other trajectory, by a large margin*

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

Assumptions:

- Linear rewards with respect to handcrafted features
- Discrete states and actions

Main idea: (*reward weights should be such that) demonstrated trajectories collect more reward than any other trajectory, by a large margin*

How can we formulate this mathematically?

Detour: solving MDPs via linear programming

$$\begin{array}{ll} \underset{v}{\operatorname{argmin}} & \displaystyle\sum_{s \in S} d_s v_s \\ \text{subject to} & v_s \geq r_{s,a} + \displaystyle\sum_{s' \in S} T_{s,a}^{s'} v_{s'} \quad \forall s \in S, a \in A \\ & d \text{ is the initial state distribution} \end{array}$$

Detour: solving MDPs via linear programming

$$\begin{aligned}
\operatorname*{argmax}_{\mu} \quad & \sum_{s \in S, a \in A} \mu_{s,a} r_{s,a} \\
\text{subject to} \quad & \sum_{a \in A} \mu_{s',a} = d_{s'} + \gamma \sum_{s \in S, a \in A} T_{s,a}^{s'} \mu_{s,a} \quad \forall s' \in S \\
& \mu_{s,a} \geq 0
\end{aligned}$$

Discounted state action counts / occupancy measure

$$\mu(s,a) = \sum_{t=0}^{\infty} \gamma^t p(s_t = s, a_t = a)$$

Optimal policy

$$\pi^*(s) = \operatorname*{argmax}_{a \in A} \mu(s,a)$$

Detour: solving MDPs via linear programming

$$\begin{aligned}
\underset{v}{\operatorname{argmin}} \quad & \sum_{s \in S} d_s v_s \\
\text{subject to} \quad & v_s \geq r_{s,a} + \sum_{s' \in S} T_{s,a}^{s'} v_{s'} \quad \forall s \in S, a \in A \\
& d \text{ is the initial state distribution}
\end{aligned}$$

Primal LP

$$\begin{aligned}
\underset{\mu}{\operatorname{argmax}} \quad & \sum_{s \in S, a \in A} \mu_{s,a} r_{s,a} \\
\text{subject to} \quad & \sum_{a \in A} \mu_{s',a} = d_{s'} + \sum_{s \in S, a \in A} T_{s,a}^{s'} \mu_{s,a} \quad \forall s' \in S \\
& \mu_{s,a} \geq 0
\end{aligned}$$

Dual LP

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

$$\begin{aligned}
\underset{w,\mu}{\text{argmin}} \quad & ||w||^2 \\
\text{subject to} \quad & w^\top f_{\tau_i} \geq w^\top f_\tau \quad \forall \tau_i \in D,\ \forall \tau \\
& \mu_{s,a} \text{ is the visitation frequency of a state action pair} \\
& F \text{ a matrix indicating the presence of a feature at a state action pair} \\
& f_\tau \text{ is the accumulated feature frequency along trajectory } \tau \\
& w \in \mathbb{R}^d_+,\ \mu \in \mathbb{R}^{|S|\times|A|}_+
\end{aligned}$$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

$$\underset{w,\mu}{\text{argmin}} \quad ||w||^2$$

$$\text{subject to} \quad w^\top f_{\tau_i} \geq w^\top f_\tau \quad \forall \tau_i \in D,\ \forall \tau$$

$\boxed{\mu_{s,a} \text{ is the visitation frequency of a state action pair}}$

F a matrix indicating the presence of a feature at a state action pair

f_τ is the accumulated feature frequency along trajectory τ

$$w \in \mathbb{R}^d_+,\ \mu \in \mathbb{R}_+^{|S|\times|A|}$$

Is searching over visitation frequencies the same as searching over policies?

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

Feature frequencies
for i-th demonstrated trajectory

for any trajectory or state action pair

$$\underset{w,\mu}{\text{argmin}} \quad ||w||^2$$

$$\text{subject to} \quad w^\top \boxed{F\mu_{\tau_i}} \geq w^\top \boxed{F\mu} \quad \forall \tau_i \in D, \ \forall \mu$$

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

$$w \in \mathbb{R}^d_+, \ \mu \in \mathbb{R}^{|S|\times|A|}_+$$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

Impose large margin that is
dependent on state action pairs

$$\underset{w,\mu}{\operatorname{argmin}} \quad ||w||^2$$

$$\text{subject to} \quad w^\top F \mu_{\tau_i} \geq w^\top F \mu + \boxed{l_i^\top \mu} \quad \forall \tau_i \in D, \; \forall \mu$$

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

$w \in \mathbb{R}_+^d, \;\; \mu \in \mathbb{R}_+^{|S| \times |A|}$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

$$\begin{aligned}
&\underset{w}{\operatorname{argmin}} && ||w||^2 \\
&\text{subject to} && w^\top F \mu_{\tau_i} \geq \max_{\mu} \left[w^\top F \mu + l_i^\top \mu \right] \quad \forall \tau_i \in D
\end{aligned}$$

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

$w \in \mathbb{R}^d_+, \ \mu \in \mathbb{R}^{|S| \times |A|}_+$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

$$\underset{w,\zeta}{\operatorname{argmin}} \quad ||w||^2$$

$$\text{subject to} \quad w^\top F \mu_{\tau_i} + \boxed{\zeta_i} \geq \max_{\mu} \left[w^\top F \mu + l_i^\top \mu \right] \quad \forall \tau_i \in D$$

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

ζ_i a slack variable for optimality of reward

$w \in \mathbb{R}_+^d, \ \mu \in \mathbb{R}_+^{|S| \times |A|}, \ \zeta_i \in \mathbb{R}_+$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

Don't allow too much slack

$$\begin{aligned}
\underset{w,\zeta}{\operatorname{argmin}} \quad & ||w||^2 + \boxed{C\sum_{i=1}^{|D|}\zeta_i} \\
\text{subject to} \quad & w^\top F\mu_{\tau_i} + \zeta_i \geq \max_{\mu}\left[w^\top F\mu + l_i^\top \mu\right] \quad \forall \tau_i \in D
\end{aligned}$$

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

ζ_i a slack variable for optimality of reward

$w \in \mathbb{R}_+^d, \; \mu \in \mathbb{R}_+^{|S|\times|A|}, \; \zeta_i \in \mathbb{R}_+$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

Is this a proper formulation of a quadratic program? NO

$$\begin{aligned} \underset{w,\zeta}{\operatorname{argmin}} \quad & ||w||^2 + C\sum_{i=1}^{|D|}\zeta_i \\ \text{subject to} \quad & w^\top F\mu_{\tau_i} + \zeta_i \geq \boxed{\max_\mu \left[w^\top F\mu + l_i^\top \mu\right]} \quad \forall \tau_i \in D \end{aligned}$$

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

ζ_i a slack variable for optimality of reward

$$w \in \mathbb{R}_+^d,\ \mu \in \mathbb{R}_+^{|S|\times|A|},\ \zeta_i \in \mathbb{R}_+$$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

But it optimizes a linear objective with linear constraints,
so we can use duality in linear programming

$$\underset{w,\zeta}{\operatorname{argmin}} \quad ||w||^2 + C\sum_{i=1}^{|D|}\zeta_i$$

$$\text{subject to} \quad w^\top F\mu_{\tau_i} + \zeta_i \geq \boxed{\max_{\mu}\left[w^\top F\mu + l_i^\top \mu\right]} \quad \forall \tau_i \in D$$

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

ζ_i a slack variable for optimality of reward

$w \in \mathbb{R}_+^d,\ \mu \in \mathbb{R}_+^{|S|\times|A|},\ \zeta_i \in \mathbb{R}_+$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

$$\underset{w,\zeta}{\operatorname{argmin}} \qquad ||w||^2 + C \sum_{i=1}^{|D|} \zeta_i$$

$$\text{subject to} \quad w^\top F \mu_{\tau_i} + \zeta_i \geq \min_{v} \left[d_i^\top v \right] \quad \forall \tau_i \in D$$

v is the value function

d_i is the initial state distribution for demonstration i

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

ζ_i a slack variable for optimality of reward

$w \in \mathbb{R}_+^d,\ \mu \in \mathbb{R}_+^{|S| \times |A|},\ \zeta_i \in \mathbb{R}_+$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

$$\underset{w,\zeta,v_i}{\text{argmin}} \quad ||w||^2 + C\sum_{i=1}^{|D|}\zeta_i$$

$$\text{subject to} \quad w^\top F\mu_{\tau_i} + \zeta_i \geq d_i^\top v_i \quad \forall \tau_i \in D$$

$$v_i^s \geq \underbrace{(w^\top F + l_i)^{s,a}}_{\text{reward}} + \sum_{s' \in S} T_{s,a}^{s'} v_i^{s'} \quad \forall \tau_i \in D, s \in S, a \in A$$

d_i is the initial state distribution

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

ζ_i a slack variable for optimality of reward

$$w \in \mathbb{R}_+^d, \ \mu \in \mathbb{R}_+^{|S|\times|A|}, \ \zeta_i \in \mathbb{R}_+$$

Max Margin Planning [Ratliff, Zinkevitch, and Bagnell]

Is this sufficient to make v_i the optimal value function?

$$\begin{aligned}
\underset{w,\zeta,v_i}{\text{argmin}} \quad & \|w\|^2 + C\sum_{i=1}^{|D|}\zeta_i \\
\text{subject to} \quad & w^\top F\mu_{\tau_i} + \zeta_i \geq d_i^\top v_i \quad \forall \tau_i \in D \\
& v_i^s \geq (w^\top F + l_i)^{s,a} + \sum_{s' \in S} T_{s,a}^{s'} v_i^{s'} \quad \forall \tau_i \in D, s \in S, a \in A
\end{aligned}$$

d_i is the initial state distribution

$\mu_{s,a}$ is the visitation frequency of a state action pair

F a matrix indicating the presence of a feature at a state action pair

l_i a demonstration-specific weight vector for margins at each state action pair

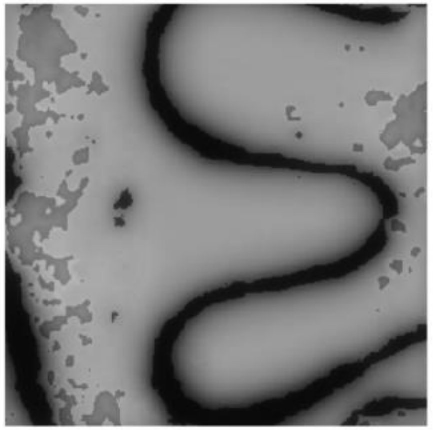
ζ_i a slack variable for optimality of reward

$w \in \mathbb{R}_+^d,\ \mu \in \mathbb{R}_+^{|S|\times|A|},\ \zeta_i \in \mathbb{R}_+$

Results

mode 1 - training

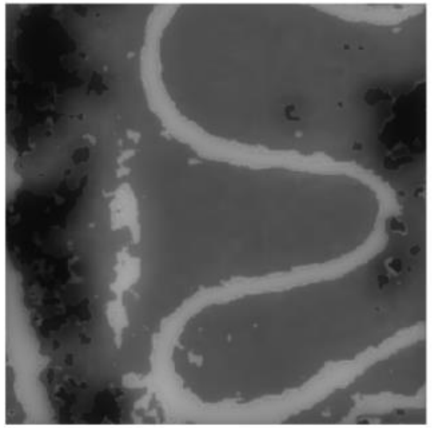
mode 1 - learned cost map over novel region



mode 1 - learned path over novel region

mode 2 - training

mode 2 - learned cost map over novel region



mode 2 - learned path over novel region