CS 294-115 Algorithmic Human-Robot Interaction **Fall 2015**

Lecture 8: Coordination: Infer/Express Intent

Scribes: Horia Mania, Dylan Hadfield-Menell

8.1 Mid-Semester Recap

8.1.1 Motion Planning

find feasible paths

*P*roblem: find a path in configuration space from a start configuration s to a goal configuration g that avoids obstacles.

A configuration $q \in C$ is a vector that completely characterizes the state/position of the robot in the world. For example, a configuration of a car might be given by the coordinates (x, y, θ).

The configuration space C can be partitioned into to set of free configurations C_{free} and the set C_{obs} of configurations in which the robot hits obstacles. Therefore $C = C_{free} \sqcup C_{obs}$. The goal of motion planning is to find paths from s to g in C_{free}.

Probabilistic Methods

PRM: Probabilistic road maps

- sample M milestones in C_{free}
- connect milestones to each other through a simple motion planning algorithm if possible. There are multiple ways to choosing candidate milestones: all pairs, in a radius of each other, k nearest neighbors
- connect s and g to the graph and do graph search
- if successful, return. Otherwise, sample new milestones and iterate.

Bi-RRT: Bidirectional Rapidly exploring Random Trees

This algorithm is a particular instance of PRM. Here $M = 1$ and the algorithm attempts to connect the sample only to the connected components of the trees containing the start and goal configurations.

Remark that these algorithms are probabilistically complete: if there is a solution, they will eventually find it.

8.1.2 Trajectory Optimization

find efficent paths

A trajectory is a function $\xi : [0, T] \to C$ and the cost is a functional $\mathcal{U} : \Xi \to \mathbb{R}^+$, where Ξ is the space of all trajectories.

Gradient descent can be used to optimize $\mathcal{U}(\xi)$: $\xi_{i+1} = \xi_i - \frac{1}{\alpha}\nabla_{\xi_i}\mathcal{U}$, where $\nabla_{\xi_i}\mathcal{U}$ is the gradient of $\mathcal{U}$ evaluated at ξ_i with respect to the Euclidean inner-product. Given another inner-product, defined by A, we have $\nabla^A_\xi \mathcal{U} = A^{-1}\nabla_\xi \mathcal{U}$.

8.1.3 Learning from Demonstrations

find paths that match user preferences

*P*roblem: find paths that match user's preferences as expressed by previously demonstrated paths.

A particular way of LfD is *Inverse Reinforcement Learning*. Next we recall two approaches to IRL.

MMP: Maximum Margin Planning

Assuming the demonstrated ξ_D is optimal, MMP is trying to find a cost function $\mathcal{U}$ such that ξ_D has a lower cost than other paths with a margin:

$$\mathcal{U}(\xi_D) \leq \min_{\xi \in \Xi \setminus \xi_D} \left(\mathcal{U}(\xi) - l(\xi, \xi_D)\right).$$

Then goal of MMP, for cost functions parametrized by $w^\top f_\xi$, is to solve the following optimization problem:

$$\min_w w^\top f_{\xi_D} - \min_{\xi \in \Xi \setminus \xi_D} \left(w^\top f_\xi - l(\xi, \xi_D)\right) + \frac{\lambda}{2}\|w\|^2.$$

Gradient descent for this problem translates to: $w_{i+1} = w_i - \frac{1}{\alpha}\left(f_{\xi_D} - f^*_{w_i} + \lambda w_i\right)$, where $f^*_{w_i}$ is the feature vector of a path minimizing $w_i^\top f_\xi - l(\xi, \xi_D)$.

Maximum Entropy IRL

In this model we assume we the agent is providing paths according to some reward function, but not necessarily optimal paths. We assume that

$$P(\xi|w) \propto e^{-w^\top f_\xi} \tag{8.1}$$

and we are trying to maximize the likelihood of the observed paths

$$w^* = \arg\max \log P(\tilde{\xi}|w).$$

This problem can be solved via gradient descent, i.e.

$$w_{i+1} = w_i + \frac{1}{\alpha}\left(f_{\xi_D} - \int P(\xi) f_\xi d\xi\right)$$

NOTE: 'Parametrization (8.1) was chosen because the maximum entropy distribution over paths is of that form.

To see this consider the problem

$$\max_P - \int P(\xi) \log P(\xi) d\xi$$
$$\text{s.t. } \mathbb{E}\left[\mathcal{U}(\xi)\right] = k,$$

for some expected cost k (closer k is to the optimal cost, the more optimal the agent is). To solve this problem we form the Lagrangian

$$\mathcal{L}(P, \lambda) = - \int P(\xi) \log P(\xi) + \lambda \left(\int P(\xi)\, \mathcal{U}(\xi) d\xi - k \right).$$

Taking the derivative with respect to P and equating it to zero, we obtain

$$1 + \log P(\xi) + \lambda U(\xi) = 0,$$

and therefore $P(\xi) \propto e^{-\lambda \mathcal{U}(\xi)}$.

8.2 Coordination

find paths that convey intent [also: infer human intent]

8.2.1 Infer Intent

Assume that an agent H starts in a configuration s and moves toward one of several possible goal states g_i. Given a path ξ_{s-q} from the start configuration to some configuration q, we would like to derive a method of estimating the probability $P(g_i|\xi_{s-q})$.

We will employ the method of Bayesian inference. The prior over goals can be uniform in the absence of any other information.

$$P(g_i|\xi_{s-q}) = \frac{P(\xi_{s-q}|g_i)P(g_i)}{\sum_g P(\xi_{s-q}|g)P(g)}.$$

Assume that the robot R has observed H and learned the cost function $\mathcal{U}$, then $P(\xi_{s-g}) \propto e^{-\mathcal{U}(\xi_{s-g})}$.

$$P(\xi_{s-q}|g) = \int P(\xi_{s-q}, \xi_{q-g}|g) d\xi_{q-g} = \frac{\int P(\xi_{s-q}, \xi_{q-g}) d\xi_{q-g}}{\int P(\xi_{s-g}) d\xi_{s-g}}$$

NOTE: For $\mathcal{U}$ quadratic this becomes

$$P(\xi_{s-q}|g) \propto \frac{e^{-\mathcal{U}(\xi^*_{q-g})}}{e^{-\mathcal{U}(\xi^*_{s-g})}}$$

Express Intent

The goal now is for R to find paths from which H can infer the end goal. The robot R models H as doing Bayesian inference as explained in the previous section [assumption backed up by papers next class].

Therefore, the robot needs to solve the following optimization problem

$$\max_{\xi} \int P(g|\xi_{s-\tilde{\xi}(t)})dt$$

with g being the actual robot goal.