Learning from Richer Human Guidance

Chandrayee Basu, Anca Dragan

November 27, 2017

Our prior study showed that users do not want their autonomous car to drive like them. We therefore cannot simply ask users to drive the car and learn driving from their demonstrations: our study suggested that they can drive in their style, but not in their preferred style [1]. Comparison-based learning [4, 5, 3] has emerged as a promising alternative approach to Inverse Reinforcement Learning in situations like this where demonstrated behavior does not reflect the correct reward or where demonstrations are difficult to provide. There, the robot iteratively shows users two possible trajectories (often in the same environment, for the same starting state), and asks which they prefer. It then uses the answer to update its understanding of the reward parameters.

Our insight is that we can extract richer guidance than just comparisons from people when learning reward functions.

Rich Queries combining Comparisons and Features. Building on the prior work on feature queries in the context of learning skills from demonstration [2] and comparison-based reward learning [5] we designed richer queries: comparison query augmented with feature query. The feature query is of the form: which feature in the reward function was responsible for the preference between the two options.

Learning from Rich Queries. We introduce a unifying formalism whereby the person’s answers are all treated as nosily-optimal responses conditioned on the true reward, and perform Bayesian inference to estimate the reward parameters.

Active Query Selection. To speed up learning, we derive a rich query selection method that optimizes for gathering as much information as possible from each query.

Analysis of Rich Queries. We conduct thorough experiments in simulation showing that rich queries learn faster than comparison-only queries, and follow-up with an in-lab study on learning driving style. We find that rich queries learn a reward that is significantly closer to the users’ internal preference.

Learning Algorithm

Imagine learning the reward for an autonomous car, so that it drives according to the user’s desired driving style. The environment is an initial state $x^0$, like where the road and obstacles are, along with the car’s initial position, orientation, and velocity. A trajectory $u$ consists of a sequence of controls $(u^0, ..., u^T)$ for the robot. A query is thus $q = (x^0, u_A, u_B)$.

Comparison Only. Prior work has used such queries to ask ”Which of the two trajectories do you prefer?” These queries represent our baseline.

Let the answer for these be $c \in A, B$: $A$ if $u_A$ is preferred, $B$ otherwise. Following prior work, we assume that people are nosily rational in identifying the correct trajectory:

$$P(c = A|\theta, q) = \frac{\exp(\beta^c R(x^0, u_A))}{\exp(\beta^c R(x^0, u_A)) + \exp(\beta^c R(x^0, u_B))} \quad (1)$$
with \( R(x^0, u_A) \) the cumulative reward in the environment for trajectory \( u_A \) and \( \beta^c \) a rationality coefficient: the higher \( \beta^c \) is, the less likely for the user to make a mistake in identifying the trajectory with higher desired reward. Now we propose to also ask "Which feature is most responsible for the difference in your preference between these two trajectories?". The answer is a feature ID \( f \) out of the set of \( F \) features that the robot is using in its reward. We introduce an observation model for such queries, based on the assumption that people will noisily identify which feature, when combined with how important it is (its weight in \( \theta \)), best accounts for the difference in reward between the two trajectories:

\[
P(f|\theta, q) = \frac{\exp(\beta^f \theta^f \cdot |\Phi_f(x^0, u_A) - \Phi_f(x^0, u_B)|)}{\sum_i \exp(\beta^i \theta^i \cdot |\Phi_i(x^0, u_A) - \Phi_i(x^0, u_B)|)}
\]

with \( \beta^f \) a rationality coefficient for answering these types of queries, \( \Phi_f \) the value of \( f \)th feature accumulated across the trajectory, and \( \theta^f \) its weight in \( \theta \).

**Rich Queries.** We propose to use rich queries, that combine comparisons and features. From the answer to the comparison, the robot will know which trajectory is better. From the answer to the feature query, the robot will know more about why.

The answer is a tuple \( a = (c, f) \). Each part serves as an observation about \( \theta \):

\[
p(a|\theta, q) = p(c, f|\theta, q) = P(c|\theta, q)P(f|\theta, q)
\]

We model he answers as conditionally independent given the reward – this is a design choice, and it means that people don’t have to get the comparison correct to tell us the main feature that matters in the comparison.

**Outcomes**

We validated our hypothesis in simulation and through user studies. We find that rich queries learn a reward that is significantly closer to the users’ internal preference.
References


