Exploring methods of safety and robustness in autonomous driving

Final Report

Neha Desaraju
Department of Computer Science
University of Texas at Austin
Austin, TX 78705
ndesaraju@utexas.edu

Introduction

Deriving inspiration from methods such as those outlined in [4], we seek to propose methods to augment decision making in autonomous vehicles in order to better capture sensor noise in the environment and overall driver uncertainty. In practice, we formalize these definitions as *robustness* and *safety*. In order to control an agent vehicle in an uncertain environment (and through uncertainly learned representations of the environment), we must employ methods that allow for the best amount of exploration through the environment (despite "certainty" in a transition model) while keeping such exploration safe (robust against other driver randomness).

In this project, we experiment [1] with select model-based methods, in which a robust dynamics model is learned and leveraged in order to optimally and safely plan and control a vehicle. The environment we use to demonstrate our methods is provided by [3] as **highway-env**; in particular, we use the "roundabout-env" variation of the given environments, in which a car must navigate through a roundabout without crashing in set amount of time. We chose to use the continuous action space variation of this problem, which allows us to control the car by 2-dimensional controls for throttle and steering (as opposed to the 5-option discrete action space, which always keeps the car on the road). This introduces a more difficult layer of the problem, as the car (or rather, our control policy) must now learn to stay on the road while avoiding collisions.

Methodology

Because the observations served by the Gym environment are exact, we must manually perturb the outputs in order to simulate noise from sensor readings (e.g. LiDAR). Then, our dynamics model can learn the variation as uncertainty along with the mean itself. We used a simple normal distribution in order to shift all observations by a small amount [2].

Since we used model-based methods (as opposed to model-free, in which the reward and policy is directly learned), we formalize our dynamics problem as solving for a function D, which can be defined as the probability for the next state given a state, action pair. In other words, $D : (s_t, a_t) \rightarrow p(s_{t+1})$. Because the output is defined as a probability distribution, we can simply sample from the model's output distribution in order to "explore" while sampling trajectories during planning.

We chose to experiment with two distinct methods for deriving and learning the dynamics of the environment; a Gaussian Process (GP) method [5] as well as a model ensemble method [9].

The GP method was trained using deep approximate Gaussians [7], which involves training a GP model using stochastic priors. This allows us to sample interactions with the environment and

[1] Our code can be found at `https://github.com/estaudere/robust-cmdp`.

generalize towards a multi-task distribution. The model ensemble method is a set of five simple feed-forward neural networks, each independently trained from the dataset. The output is sampled from the five outputs, so that the probability distribution is a more discrete distribution from the outputs.

We also explored two methods of deriving a cost function for our planner; the first was manually defining a cost function from the observations. This involved calculating collision distance directly from the observation itself, as well as calculating on-road dynamics. We tried a variety of definitions, including $C : s \to c \in \{0, 1\}$ as well as $C : s \to c \in [0, 1]$, the latter of which is a continuous function of speed and distance to collision.

The second method of deriving a cost function was to train a gradient-boosted decision tree to classify states as either safe or unsafe [6]. This allows us to constrain the "safe" states as a geometry over the states (a stand-in for a [8], which is much harder to define in this context), while still allowing the model to learn its own representation of the environment.

Finally, we used the model-predictive control method that uses the learned (or defined) cost and dynamics functions in order to sample safe trajectories through an environment. We used the cross-entropy method (CEM) of sampling [1], which allows for some probability of exploration through the samples but progressively limits the sample range. Our planning method follows the Constrained Markov Decision Process (CMDP) framework [6], which is defined by a tuple (S, A, f, r, c), where f (the transition function) and c (the cost function) are learned from the data. When c is an indicator function, this becomes a strict CMDP. This uses the robust CEM as well.

Experiments and Analysis

We found that the GP method did not learn the dynamics of the environment well; we propose that future work in this area can look at more complex architectures, such as Spectral Normalization in lower neural network layers [5], in order to better generalize and learn the transition function. This model was also much slower to train and conduct inference from. The regression model ensemble very quickly generalized well to the training data.

Another thing we noticed was that thorugh manually defining the cost function is a suitable candidate as in this case the cost function si easy to define and it worked decently well at teaching the planner to avoid collisions, this method did not keep the car on the road for very long. We found that learning the cost function from environment feedback worked immensely better at keeping the car on the road; however, because speed feedback from the environment is nonexistent, we found that the car would learn to simply move back and forth or move very slowly on the road for some training iterations. This can be fixed by goal-conditioning the CMDP, which is a very possible improvement; however, this environment did not provide a suitable goal to follow rather than rewarding the agent for moving quickly. Thus, this may not be suitable for a strictly CMDP problem, in which the cost is an indicator function. However, the gradient-boosted decision tree performed incredibly well at state classification, reaching an accuracy of 100%.

In our best method — CMDP with CEM after learning a regression dynamics function and decision tree cost function — the car still moves slowly and fails on some iterations to make any progress. Thus, more work must be done to make our solution suitable and to generalize across many driving environments.

Conclusion

In this work, we proposed several solutions and began experiments of our methods on the given environment. In particular, we demonstrate the lack of feasibility of Gaussian dynamics functions and manually-defined cost functions with constrained MDPs, in which model-predictive planning occurs only in safe states (or only a few potentially unsafe state). We also apply these methods to continuous action spaces; the discrete action space version of this problem is much easier as the agent needn't learn staying on the road while avoiding collisions. This extra layer to the task necessitates models that can generalize to the tasks. Overall, more work should be done, though we have shown a promising step in the direction of robust and constrained planning in autonomous driving.

References

[1] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces, 2018.

[2] Raad Khraishi and Ramin Okhrati. Simple noisy environment augmentation for reinforcement learning, 2023.

[3] Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018.

[4] Edouard Leurent, Yann Blanco, Denis Efimov, and Odalric-Ambrym Maillard. Approximate robust control of uncertain dynamical systems, 2019.

[5] Jeremiah Zhe Liu, Shreyas Padhy, Jie Ren, Zi Lin, Yeming Wen, Ghassen Jerfel, Zack Nado, Jasper Snoek, Dustin Tran, and Balaji Lakshminarayanan. A simple approach to improve single-model deep uncertainty via distance-awareness, 2022.

[6] Zuxin Liu, Hongyi Zhou, Baiming Chen, Sicheng Zhong, Martial Hebert, and Ding Zhao. Constrained model-based reinforcement learning with robust cross-entropy method, 2021.

[7] Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes, 2017.

[8] Min Wen and Ufuk Topcu. Constrained cross-entropy method for safe reinforcement learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[9] Jingda Wu, Zhiyu Huang, and Chen Lv. Uncertainty-aware model-based reinforcement learning with application to autonomous driving, 2021.