Predicting Collision-free Policy Sets for Safe Navigation among Human Agents

Luke Shimanuki
luke@shimanuki.cc

August 16, 2020

Abstract

Predicting human behavior is considered by many to be the largest barrier to autonomous navigation. The typical goal of predicting trajectories of other agents to avoid is the wrong problem to be solving, because it is not actually solvable, due to the unobservability and interactivity of the domain. Instead, I propose that we predict sets of policies that by construction do not allow collisions, leveraging the insight that the joint policy across all agents tends to avoid collisions. I design a simple policy set that provably prohibits collisions while still capturing nearly all actually observed behaviors in the inD dataset [2], allowing for a robot to follow such a policy with a collision rate over 10 times lower than demonstrated in prior work for high planning horizons. I claim that incorporating actual motion models and rules of the road will likely improve performance enough to achieve human-level safety, or at the very least, far better than what everyone is using right now.

1 Introduction

The goal of autonomous navigation – and let's be honest we only really care about self-driving cars – is generally to a) get to some desired location, and b) not hit things. Unfortunately, this gets complicated when the things can move, especially if such movements are based off unobservable processes (ie whatever is going on in other people's minds).

The standard approach to this is to create some kind of model of how the other agents move, predict where they're going to go, and then use a time-dependent motion planner to not hit where they are expected to be. And of course you can add fancy extensions like predicting probability distributions over trajectories, multimodal distributions, etc (this actually gets pretty complicated pretty quickly once you realize that planning over probability distributions is intractible in the general case, as is planning over multimodal obstacle models). The bigger problem with this is that it doesn't work.

Oh, you can get pretty decent results with clever engineering, lots of careful heuristics that work most of the time, or a near-infinite pool of data to throw a giant neural net at [3]. But what are we really measuring here? The usual evaluation metric is some variant of "how far was the actual trajectory from the predicted trajectory". And as it turns out, as you try to predict farther into the future (like a whole 3 seconds) it starts looking pretty dicey. You can't plan effectively when your uncertainty of the obstacles' locations is over 2 meters. And it gets worse (probably quadratically, given the main uncertainty is acceleration) with time. Sometimes they directly measure how safe planning based on these predictions is. And sometimes it looks pretty good, like half a percent collision rate over 3 seconds [3]. Until you realize that means a collision on average every 10 minutes. And that's assuming that risk increases only linearly with the time horizon (hint: it doesn't, at least not under these formalizations). It doesn't look like these methods are going to improve enough, because they are attempting to solve an inherently unsolvable problem. This is for 2 reasons: there is not enough information to infer the state of another human's mind, and the other people will be reacting to what you do (imagine trying to drive safely if nobody else knows you're there).

So how do humans avoid collisions so well? Certainly not by predicting trajectories of everyone else (I've always wanted to actually do a psychophysical experiment on human accuracy, but I never got around to it, and besides, we all know it's true anyways). Well, if you don't know what they will do, you do know what they won't. Like they won't teleport to the other side of the road. Or they won't drive into the opposing lane in front of an oncoming car. Maybe we don't know how humans avoid collisions so well, but what we do know is that *they won't collide* with each other, and it turns out, as we'll see later, that that's a tight enough constraint to plan around.

I propose a different paradigm by which we can approach this problem. Recall the 2 problems with predicting trajectories: unobservability and interactivity. To handle interactivity, we will predict *policies* rather than fixed trajectories. To handle unobservability, we will predict *sets* of policies rather than single policies (note that we could instead predict distributions over policies, but when planning we'd have to draw a risk cutoff at some point anyways, so might as well do that here). We know that the correct joint policy will avoid collisions, so one evaluation metric would be the number of situations where a collision would be possible given that every agent follows a policy from the predicted set. And since we need the predictions to actually match reality, the other evaluation metric would be the number of situations where an agent's actually observed behavior deviates from the policy set.

To illustrate with some trivial examples, consider the policy set where every agent is stationary. Then the collision rate would be zero, since stationary agents do not collide. But the deviation would be high, because nearly all the time, there will be a non-stationary agent. In the other extreme, consider the policy set where every agent is allowed any possible movement. Then the collision rate would be high, because a collision would always be achievable by some assignment of policies. But the deviation would be high, because anything that was actually observed fits within the policy set. An ideal policy set is then one that is narrow enough to prevent collisions but broad enough to contain every actual observation.

To motivate this approach, I construct a simple policy set that provably does not allow collisions, and has empirically low deviation over the inD dataset [2]. (I would have used SDD [1] like most papers do, but it is not suitable for actually measuring collisions because it only provides axis-aligned bounding boxes rather than rotated bounding boxes, leading to a large number of collisions in the recorded trajectories themselves)

2 Related Work

This isn't a real academic paper (yet), so sorry, no literature review unless I decide to actually submit this somewhere.

3 Problem Definition

We start by defining a highly structured deterministic partially observable markov decision process. Let O denote the observable state space for a single agent, including the pose, momentum, shape, and anything else that defines how it can move and interact with other agents, and let Π denote the unobservable state space including information such as where it wants to go or how it plans to get there. Suppose that an unobservable state $\pi \in \Pi$ is fully described by a policy $\pi : H \to A$, where H is the joint observable state space history $O^{n\tau}$ (where n is the number of agents and τ is the number of observed timesteps), and A is the action space for a single agent, typically actuator settings for the duration of a timestep. Let us further suppose that the transition model for a single agent $M : O \times A \to O$ is fully known and independent of the state and actions of other agents, and that there is a unique action that leads from any given state o_1 to o_2. For situations where it is not unambiguous, we specify $h^{(i)}$ as the history h but with agent i treated as the ego-agent, that is, the one that is being controlled.

We can now informally define a navigation problem.

Problem 1 (Navigation Problem)**.** *Given the current history $h \in H$ and a "typical" goal state $g \in O$ (describing just the robot), find a policy $\pi \in \Pi$ minimizing the number of fixed-time situations in which the goal cannot be reached or the robot collides with another agent, assuming each other agent is following a "typical" policy.*

Ideally, a solution would be evaluated based on some combination of how frequently it reaches the goal and how frequently it avoids collision. Unfortunately, neither of these can be measured for arbitrary policies offline, because we cannot record what policies other agents are following, and so don't have a notion of what a "typical" policy would be. Hence, a true evaluation would have to take place in the wild, which, unless the policy is actually safe, has high risk for human injury and property damage.

The data we do have access to is a set of observed trajectories of agents within a scene over a period of time. While this does not provide the full policies they are following (since if you changed what one of the agents did, the other agents' actions would also change), it does specify what action their policy returned given a very specific input. Therefore, in order to take advantage of what we do have access to, we instead propose an alternative problem.

Definition 1 (deviant)**.** *A set of policies $P \subset \Pi$ is deviant from a history $h \in H$ over a given time period (τ_1, τ_2) if for any timestep $\tau \in (\tau_1, \tau_2)$, any agent i took an action $a \in A$ during that time period for which there does not exist a policy $\pi \in P$ such that $\pi(h_\tau^{(i)}) = a$, where $h_\tau^{(i)}$ denotes the truncated history up to timestep τ.*

Definition 2 (collidable)**.** *A set of policies $P \subset \Pi$ is collidable over a history $h^{(i)} \in H$ and a given time period (τ_1, τ_2) if there exists an assignment of policies from P to each agent such that simulating forward starting at $h_{\tau_1}^{(i)}$ for $\tau_2 - \tau_1$ timesteps results in agent i colliding with some other agent.*

Problem 2 (Offline Navigation Problem)**.** *Given the full history of a scene $h \in H$, find a set of policies $P \subset \Pi$ minimizing the fraction of time periods for which P is deviant or collidable over h.*

This may seem like a strange objective, but it turns out that solving Problem 2 also solves Problem 1, assuming your dataset of recorded trajectories is representative, that is, contains all "typical" goals and policies. This is precisely because we know that people tend to reach their goals without colliding with each other. To spell it out, consider the 2 metrics in Problem 1. First, we observe that if your goal is something an agent in your dataset would have had as a goal, then that agent would likely be recorded as reaching the goal, and so either P contains a policy that leads to the goal or it is deviant. Second, we upper bound the fraction of situations in which a collision is possible by following a policy in P.

Theorem 1. *The total risk of collision R of following a policy in $P \subset \Pi$, which upper bounds the fraction of situations in which a collision is possible by following a policy in P, is given by $C + D$, where C is the fraction of situations in which P is collidable, and D is the fraction of situations in which P is deviant.*

Proof. Consider a situation for which following a policy in P leads to collision. If every other agent is also following a policy in P, then by definition this is a situation where P is collidable. If some agent is not following a policy in P, then by definition this is a situation where P is deviant. Then the number of such situations is upper bounded by the total number of situations in which P is collidable and those in which P is deviant, hence $R = C + D$. □

Short aside: we can think of the fraction of situations in which P is not deviant as the recall (since they are measuring how well the actual output is captured), and the fraction of situations in which P is not collidable as the precision (since they are measuring how well the model excludes bad outputs). Therefore we could consider measuring a variant of the F_1 score. However, the harmonic mean in this case doesn't mean much intuitively, whereas the sum upper bounds the risk of collision, so the sum turns out to be a nicer metric to optimize for.

Now representing sets of policies is fairly difficult, and moreso checking whether any contained policy allows collision, especially since the set is usually infinitely large. While under certain assumptions, this can sort of be measured by sampling a large number of policies and verifying that none of them allow collision, this is not foolproof, and it is trivial to construct policy sets for which policies that lead to collision always exist but are rarely sampled (eg suppose there are 10 agents and each is allowed to teleport up to 1000 meters – this policy set is always collidable, but the odds that a pose sampled for one agent would be in collision with that for another agent is small because of the large sample space).

Instead, we will construct a policy set that by definition is never collidable. Then, this policy set will be evaluated solely based on how frequently it is deviant.

4 Collision-free Policy Set

We now construct a set, first in very general terms that can be applied to any navigation domain with any set of priors, then with specific details for a 2D space with minimal priors.

4.1 A Parameterized Policy Set

We define a space P_{ϕ,π_0} of policy sets parameterized by 2 parameters. The first parameter is a claiming map $\phi : O \times T \times \mathbb{R}^d \to \mathbb{R}$ indicating the degree to which a given agent (with observable state O) claims a given point in task space for a particular timestep (T is the set of timesteps). If $\phi(h, i, \tau, x) > \phi(h, j, \tau, x)$ for each other agent j, then agent i is considered to claim x at time τ (and hence no other agent should occupy that space-time). From this scoring function we can derive a partitioning of $T \times \mathbb{R}^d$ into disjoint sets, denoted $\psi(h^{(i)})$ for each agent i. The second parameter is a default policy $\pi_0 \in \Pi$. Each agent will ensure that it is always able to safely begin following π_0 and that each other agent can safely begin following π_0 at any timestep.

The resulting policy set is then simple to construct. $\pi \in P_{\phi,\pi_0}$ iff the following conditions hold, starting from some history h_τ:

- Simulating forward the robot agent i following π starting at state $h_\tau^{(i)}$ for a single timestep does not collide with any point $x \notin \psi(h_{\tau-2}^{(i)})$.
- Simulating forward the robot agent i following π starting at state $h_\tau^{(i)}$ for a single timestep and then executing π_0 for $\eta - 1$ timesteps does not collide with any point $x \notin \psi(h_{\tau-1}^{(i)})$.
- Simulating forward another agent j following π starting at state $h_\tau^{(j)}$ for a single timestep does not collide with any point $x \in \psi(h_{\tau-2}^{(i)})$.
- Simulating forward another agent j following π starting at state $h_\tau^{(j)}$ for a single timestep and then executing π_0 for $\eta - 1$ timesteps does not collide with any point $x \in \psi(h_{\tau-1}^{(i)})$.

Here we use η to denote the time horizon (e.g. 3 seconds). Informally, this is basically saying that the robot is allowed to move anywhere in space-time that it claimed both 2 timesteps ago and 1 timestep ago (the intersection of these regions), as long as it ensures that it can execute the default policy afterwards without leaving the region it claimed 1 timestep ago, and that every other agent is allowed to do the same but with the complement of the robot's claimed region. The reason why the policy set is assymetric is because we are measuring only how often the robot in question is in collision. If our objective was to minimize all collisions regardless of which agents were involved, then we could restrict every agent to their specific claimed region. However, measuring only collisions involving the robot makes for a fairer comparison with other methods because that is what is usually measured in the literature. Also, I did not mention this previously, but we restrict the policies to act based on information at least one timestep old because neither humans nor computers can actually act on new information immediately. If we set a timestep to be 80ms long, then this will allow for a reasonable amount of time for a computer to react.

Theorem 2. *P_{ϕ,π_0} is not collidable for any history $h^{(i)}$ and time period $(\tau, \tau+\eta)$ if agent i following the default policy starting at time τ would not lead to collision.*

Proof. Suppose the robot agent i can execute a policy in P_{ϕ,π_0} for a single timestep starting at τ_1, where $\tau \leq \tau_1 \leq \tau+\eta$. Then each other agent is either also executing such a policy or the default policy; this is guaranteed to not be in collision because agent i avoids each other agent j's $\psi(h^{(j)}_{\tau_1-1})$, including any possible default trajectory. Suppose agent i cannot execute such a policy. Then it can execute the default policy, which is guaranteed to not collide with such a policy for any other agent because each other agent avoids $\psi(h^{(i)}_{\tau_1-1})$, which contains agent i's default trajectory. Agent i's default trajectory starting at τ_1 cannot collide with another agent j's default trajectory also starting at τ_1 because at time $\tau_1 - 1$ both agents selected trajectories such that an ensuing default trajectory is contained in $\psi(h^{(i)}_{\tau_1-2})$ and its complement, respectively, which are disjoint. If agent i initiates a default trajectory at τ_1 and some agent j initiates a default trajectory before τ_1, then at τ_1-1 agent i chose a trajectory that can be followed by a default trajectory at time τ while staying within $\psi(h^{(i)}_{\tau_1-2})$, which cannot intersect with any other agent's default trajectory starting at τ_1-1. If no such trajectory could be found, then agent i would have executed the default policy at $\tau_1 - 1$, and by induction we can see that the only situation where agent i's default trajectory can collide with any other agent's default trajectory is if those were the initial conditions (ie the preceding actions were not drawn from policies in P_{ϕ,π_0}). □

By extension we conclude that P_{ϕ,π_0} is not collidable for any history h and time period $(\tau, \tau+\eta)$ if it is not deviant from h over time period $(\tau, \tau+1)$ (since if the default policy starting from τ leads to collision, P_{ϕ,π_0} contains no valid policies for the colliding agents, and so whatever action that was observed would be classified as deviant). Therefore, the overall risk of P_{ϕ,π_0} is its rate of deviancy.

4.2 Parameter Details

We now construct parameters ϕ and π_0 (note that the results above hold regardless of what we do here, and so our goal is to choose these parameters to minimize deviancy, which will be empirically measured later).

Because I am lazy, and because I don't want to have to make separate cases for each kind of agent (e.g. car, bike, pedestrian, etc), I will make no assumptions about agents' motion models, aside from that their trajectories are continuous and that their acceleration is bounded.

For default policy π_0, the simplest is a stopping policy. An agent following π_0 will start moving along their current kinematic curve and slow down with a fixed deceleration until it stops. I don't have much to motivate this, aside from that if you're inching forwards at an intersection, you'll generally leave enough room to stop before going into dangerous territory until you know for sure that it's safe to go through. Notably, I don't expect that people are planning for complicated contingencies like how quickly they can swerve away from something, etc. If we had more specific motion models, we could do some more interesting things with like assuming it will still follow a lane and such, but this will work well enough as a starting point.

Now we construct ϕ. Suppose simulating an agent i forward following π_0 starting at state $h^{(i)}_\tau$ for j timesteps results in agent i being centered at point $y_0 \in \mathbb{R}^d$. Then we define $\phi(h, i, \tau + j, x)$ as the negative distance from x to the footprint of agent i at pose y_0. Informally, this scoring function says that an agent is allowed to move anywhere near its stopping trajectory, so long as another agent is not expected to be even closer.

5 Empirical Results

We evaluate our method on the inD dataset [2] for every vehicle (cars and trucks, ignoring bicycles and pedestrians) that moves at least 5 meters (mostly ignoring parked vehicles). We remove segments within 5 seconds of collisions in the ground truth, assuming that those were due to slight labeling error, as well as at the end of trajectories so we are not predicting beyond what has been recorded.

While we can't directly evaluate deviation and collidability on existing methods, we can still compare the risk bound to what has been reported in the literature. These comparisons come with the caveat that they were evaluated on different datasets, and so the comparisons might not be completely fair. In particular, Zeng et al. [3] evaluate their proposed model as well as a number of baselines on an internally collected dataset. The inD dataset [2] was collected at relatively complex unsignaled intersections, so I believe it to be at least as hard as other datasets since the rate of difficult interactions is fairly high (although it is missing unusual circumstances such as construction zones). Furthermore, the methodology used by others does not take into account how others will react to the robot. So take these comparisons with a grain of salt, but they're here to give a very rough picture.

- Ours: The policy set defined in Section 4 evaluated on the inD dataset [2].

- End-to-End Neural Network (NN): An end-to-end interpretable neural motion planner which produces intermediate representations of predicted future trajectories [3]. Evaluated on an internal Uber dataset.

- Ego-Motion (EM): A 4-layer MLP to predict future trajectory based solely on personal history [3]. Evaluated on an internal Uber dataset.

- Imitation Learning (IL): An imitation learning network to predict future trajectory of self based on personal history (encoded in same way as Ego-Motion) and the same backbone as End-to-End NN [3]. Evaluated on an internal Uber dataset.

- Adaptive Cruise Control (ACC): Following leading vehicle while staying in the center of the lane [3]. Evaluated on an internal Uber dataset.

- Manual Cost (MC): Same planning mechanism as NN but with a manually generated cost map based on simple heuristics [3]. Evaluated on an internal Uber dataset.

	Collision Rate (%)				
	1s	2s	3s	5s	10s
Ours	.00442	.0186	.0568	.465	1.34
NN	.01	.04	.78		
EM	.01	.54	1.81		
IL	.01	.55	1.72		
ACC	.12	.53	2.39		
MC	.02	.22	2.21		

6 Conclusion

As we can see, even a very rudimentary model with few priors around motion models and rules of the road can achieve very low risk of collision, beating existing approaches by a significant margin and scaling well with horizon. My main contribution is not this model, though. In fact I would suggest that no robotics company ever actually use this model in production. After all, the bound on the risk is still too high to be considered anywhere near safe. However, knowing that such a rudimentary model following this paradigm can work this well, imagine the performance a robust system incorporating all known priors and biases could achieve.

Really, existing approaches aren't too bad in practice, it's just that they cannot be effectively be measured safely, and they are optimizing for objectives that are tangential to the real goal. Taking an approach that explicitly optimizes for actual safety in the presence of unpredictable human agents and whose metrics can be tracked as improvements are made is likely to lead to much faster and more effective development.

The approach presented here has some nice properties. One nice side effect is that it reports exactly when it doesn't know what's going on, and so the robot can resort to some kind of fallback mechanism if necessary. It also optimizes for always allowing for the very reasonable fallback of immediately braking. Additionally, its safety is empirically testable offline without making assumptions about how agents' actions would change in response to the robot's actions. We see that its risk seems to scale better with the time horizon than other approaches (but I claim this is more due to limitations in ability to fairly evaluate these other approaches).

Honestly, I don't know whether this paradigm is the correct one. But it does seem better than what's out there. I know that we aren't getting anywhere with our current methods, and I firmly believe there is a simple concept that, when turned into a well-engineered robust implementation, can lead to actual human-level safety.

References

[1] A. Alahi S. Savarese A. Robicquet, A. Sadeghian. Learning social etiquette: Human trajectory prediction in crowded scenes. 2016.

[2] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. 2019.

[3] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. pages 8652–8661, 06 2019. doi: 10.1109/CVPR.2019.00886.