Abstract

Predicting the behavior of surrounding vehicles is a critical problem for automated vehicles. We present a novel game theoretic behavior prediction model that improves the state of the art by explicitly reasoning about possible future interaction between agents. We evaluate our approach on NGSIM vehicle trajectory data set and demonstrate improvement over state-of-the-art methods.

1 Introduction

In order to plan and execute a safe and efficient trajectory through a traffic environment, an automated vehicle must have some understanding of how other agents are likely to move. Predicting the behavior of other drivers is therefore a critical problem in autonomous driving research and development. One of the key difficulties associated with behavior prediction is the interactive nature of traffic.

Existing behavior prediction models exhibit wide variation in the way they reason about interaction. Some models ignore it completely, predicting the future behavior of a target vehicle based solely on that vehicle’s previous motion [2], [13], [15], [10], [3], [18]. Among models that reason about interaction, some do so implicitly, conditioning motion prediction on the local traffic scene (including the current state and/or motion history of other nearby vehicles) [7], [8], [1], [16], [17], [14]. Still other models reason explicitly about interaction, addressing the prediction task from a game-theoretic perspective [9], [20], [19], [12], [21].

We present a novel game theoretic behavior prediction model, namely Multi-Fidelity Recursive Behavior Prediction (MFRBP for brevity), that performs better than the previous state of the art by explicitly reasoning about possible future interaction between agents. The proposed algorithm employs a recursive trajectory prediction scheme inspired by the Level $k$ [6] and Cognitive Hierarchy [11] recursive reasoning paradigms.

This paper gives a brief overview of the general Multi-Fidelity Recursive Behavior Prediction algorithm, and discusses several specific implementations of our model. Due to constraints on document size, we have omitted certain non-central aspects of the algorithm from discussion.

The models examined herein are heavily based on Convolutional Social Pooling for Vehicle Trajectory Prediction as proposed by Deo and Trivedi [7] in 2018. All experiments are conducted with the publicly available NGSIM data set.

2 Methods

Consider a traffic scene consisting of $n$ agents, where $v_i$ denotes the $i^{th}$ agent. The motion history of a traffic scene (up to and including the current time) can be compactly represented by the set of time histories $\mathcal{X}_{\text{hist}} = \{x_{1:t}^1, \ldots, x_{n:t}^t\}$, where $x_{i:t}^t$ is the state of agent $v_i$ at the current time $t$. The corresponding set of future trajectories is denoted by $\mathcal{X}_{\text{future}} = \{x_{1:t+1:t}^1, \ldots, x_{n:t+1:t}^t\}$.
The most general and complete solution to the behavior prediction task would be $\hat{P}(X_{\text{future}}) = \hat{P}(x_{1:t+1}^1, \ldots, x_{n:t+1}^n)$, an estimate of the joint probability distribution over the future trajectories of all vehicles. For real time computation, this expression needs to be approximated.

**Reasoning levels** The algorithm assigns a reasoning level $L$ to each agent in the scene, inspired by the Level $k$ [10] and Cognitive Hierarchy [11] models. Within this paradigm, an agent reasoning at level $k$ assumes that all other agents reason at level $k - 1$ or lower.

The recursive prediction phase begins by generating a level 0 trajectory prediction for each agent. Then, for each agent whose assigned reasoning level is higher than 0, the algorithm generates a level 1 trajectory prediction that is explicitly conditioned on the level 0 predictions of all neighbors. The recursive prediction process terminates once the highest level of reasoning has been reached.

**Multi-Fidelity Behavior Modeling** Consider a scene in which agent $v_i$ is driving very close to the robot, while agent $v_j$ is farther away, near the limits of the robot’s sensor range. The future motion of $v_i$ is likely to be more relevant than that of $v_j$ in the robot’s planning process. There may also be high uncertainty associated with the state of $v_j$ and its immediate vicinity. In such a case, it may be more important and more practical to have high-fidelity motion prediction for $v_i$ than for $v_j$.

MFRBP incorporates a simple but effective scheme for reasoning about future motion of surrounding vehicles at varying levels of fidelity: Each agent $v_i$ is assigned a set of policy models $\Pi_i = \{\pi_{i,0}, \ldots, \pi_{i,L_i}\}$. The level 0 trajectory prediction for agent $v_i$ is generated by $\pi_{i,0}$, the level 1 prediction comes from $\pi_{i,1}$, and so on up to level $L_i$. By using policy models of varying fidelity, MFRBP can (1) flexibly handle mismatch between models and available information and (2) allocate more computational resources to those agents whose future behavior is deemed most critical. We call this multi-fidelity modeling.

The general Multi-Fidelity Recursive Behavior Prediction algorithm is outlined in algorithm [1].

**Algorithm 1 Multi-Fidelity Recursive Behavior Prediction**

1: procedure \textsc{MultiFidelityRecursiveBehaviorPrediction($X_{\text{hist}}$)}
2: \hspace{1em} for $i \in \{1, n\}$ do
3: \hspace{2em} $L_i \leftarrow \text{ AssignReasoningLevel}(X_{\text{hist}}, v_i)$
4: \hspace{2em} \{ $\pi_{i,0}, \ldots, \pi_{i,L_i}$ \} \leftarrow \text{ AssignPolicyModels}(X_{\text{hist}}, v_i, L_i)$
5: \hspace{1em} for $k \leftarrow 0, \ldots, \max_{i \in \{1, \ldots, n\}} L_n$ do
6: \hspace{2em} if $k < L_i$ then
7: \hspace{3em} if $k = 0$ then
8: \hspace{4em} $(\hat{x}_{i,0}^{t+1:t}, \Sigma_{i,0}) = \pi_{i,0}(\{x_j^{t+1:t} \mid j \in \{1, \ldots, n\}, j \neq i\})$
9: \hspace{3em} else
10: \hspace{4em} $(\hat{x}_{i,k}^{t+1:t}, \Sigma_{i,k}) = \pi_{i,k}(\{x_j^{t+1:t}, \hat{x}_j^{t+1:t} \mid j \in \{1, \ldots, n\}, j \neq i\})$
11: \hspace{1em} return $\{ (\hat{x}_{1,L_1}^{t+1:t}, \Sigma_{1,L_1}), \ldots, (\hat{x}_{n,L_n}^{t+1:t}, \Sigma_{n,L_n}) \}$

**Policy models used in our experiments** We use combinations of three distinct policy models in our experiments. Two of these are level 0 policy models, meaning that they condition only on motion history. The other policy is a level 1 policy model that explicitly conditions on previously computed level 0 trajectory predictions. Vehicles are assigned policies based on their fidelity according to the traffic with respect to the ego vehicle. This is illustrated in Figure [1]. The left part shows an example of a traffic scene with ego vehicle in the center and the ‘high-fidelity’ vehicles whose trajectories are to be predicted. The rest on the periphery of the sensor range are ‘low-fidelity’ vehicles.

The Constant Velocity (CV) model is used as a “low-fidelity” level 0 model. CV predicts that a target vehicle will travel at constant velocity equal to the average velocity vector (both longitudinal and lateral components) over the last second.

The Convolutional Social Pooling (CSP) model is used as a “high-fidelity” level 0 model. CSP is a trajectory-based prediction model proposed by Deo and Trivedi [7]. CSP combines a Long Short-Term Memory network (LSTM) encoder-decoder architecture with a convolution neural network (CNN) “social pooling” architecture to generate multimodal trajectory predictions. This model is shown in the green-dashed box in Figure [1]. For each target, CSP accepts as input the state-histories of both the target vehicle and its neighbors (input rectangular region). The decoder LSTM layer
receives ‘social context’ and ‘vehicle dynamics’ from past. The output of this layer is a set of six Gaussian distributions, each with an associated likelihood, that represent six possible trajectories. In our experiments, we select the mode with the highest probability for prediction.

The \textit{Future-Conditional Convolutional Social Pooling (FC-CSP) model} is used as a “high-fidelity” level 1 model, shown in Figure 1. FC-CSP is a novel extension of CSP. We modify CSP to accept an additional input: the set of predicted future trajectories for all neighbors of the target vehicle. This is accomplished by adding a future social pooling block (lower block), which is architecturally identical to the history social pooling block. The decoder LSTM layer receives as input the concatenation of the both past and future “social contexts” and the “vehicle dynamics” vectors, and outputs a multimodal Gaussian trajectory distribution of the same form as the output of CSP.

3 \hspace{0.5em} \textbf{Experiments}

We perform three experiments to evaluate the performance of Multi-Fidelity Recursive Behavior Prediction. The first experiment is designed to compare our model against the existing state-of-the-art (the CSP baseline), while the other two are intended to evaluate our algorithm in a setting that is more representative of an autonomous driving scenario.

All three experiments are conducted on the publicly available NGSIM I-80 [5] and US101 [4] datasets. There are 3 subsets of both US-101 and I-80, consisting of vehicle trajectories recorded by overhead camera at a frequency of 10 Hz. The test set consists of one quarter of the vehicle trajectories from each of the subsets of the US-101 and I-80 datasets. We split the trajectories into segments of 8s (sampled at 10 Hz), where we use 3 s of track history and a 5 s prediction horizon. We use the common metric of \textit{Root Mean Square Error} (RMSE) for evaluating the performance of our models.

3.1 \hspace{0.5em} \textbf{Experiment 1 - Level 1 Recursive Behavior Prediction}

Our first experiment compares a specific implementation of Multi-Fidelity Recursive Behavior Prediction against the performance of the CSP baseline. This implementation is called Level 1 Recursive Behavior Prediction (L1-RBP). In L1-RBP, all agents are assigned a reasoning level of 1. The level 0 policy model for each agent is \textit{Convolutional Social Pooling} (CSP), and the level 1 policy for each agent is \textit{Future-Conditional Convolutional Social Pooling} (FC-CSP).

We train both the CSP (level 0 policy) model and FC-CSP (level 1 policy) model jointly from scratch. As in the original CSP implementation, we use the leaky-ReLU activation with $\alpha = 0.1$ for all layers and use Adam for optimization.

\textbf{Results for Experiment 1} \hspace{0.5em} We compare the final output of (L1-RBP) against the CSP baseline. The left parts of Table 1 shows RMSE values obtained at varying time horizons for CSP and L1-RBP. Note that CSP $\dagger$ corresponds to the original values reported in the paper by Deo and Trivedi [7] and CSP $\ast$ is our own implementation of the baseline CSP model.
Table 1: Comparison on RMSE.

<table>
<thead>
<tr>
<th>Horizon (s)</th>
<th>Baselines CSP</th>
<th>Experiment 1 L1-RBP</th>
<th>Experiment 2 L1-MFRBP</th>
<th>Experiment 3 L1-MFRBP (planning)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.62</td>
<td>0.53</td>
<td>0.54</td>
<td>0.54</td>
</tr>
<tr>
<td>2</td>
<td>1.29</td>
<td>1.19</td>
<td>1.20</td>
<td>1.19</td>
</tr>
<tr>
<td>3</td>
<td>2.13</td>
<td>1.95</td>
<td>1.99</td>
<td>1.95</td>
</tr>
<tr>
<td>4</td>
<td>3.20</td>
<td>2.87</td>
<td>2.97</td>
<td>2.88</td>
</tr>
<tr>
<td>5</td>
<td>4.52</td>
<td>3.97</td>
<td>4.16</td>
<td>4.01</td>
</tr>
</tbody>
</table>

3.2 Experiment 2 - Level 1 Multi-Fidelity Recursive Behavior Prediction

For experiment 2, we introduce Level 1 Multi-Fidelity Recursive Behavior Prediction (L1-MFRBP). L1-MFRBP targets the “ego-centric” prediction task, which is more representative of the autonomous driving use case: We randomly select a vehicle to treat as an “ego” agent, limiting the set of other agents in the scene to those within a plausible “sensor range” of this agent. This is repeated for many different “ego” agents during both training and testing. L1-MFRBP is identical to L1-RBP, except that it incorporates a multi-fidelity scheme: agents at the periphery of the designated ego agent’s sensor range are assigned to a lower reasoning level ($L = 0$) and a lower fidelity ($\pi = CV$) policy model. The L1-MFRBP CSP and FC-CSP policies are jointly trained, which includes the lower-fidelity constant velocity trajectory predictions.

Results for Experiment 2: It would take too long to exhaustively evaluate L1-MFRBP on the full test set (i.e. by treating every single vehicle in turn as the ego agent). Instead, we sample enough ego agents to ensure that a single level 1 prediction can be computed for each vehicle in the test set. We report the average results of the level 1 predictions from 10 full iterations through the test set in this manner. Our results are presented in Table 1 alongside the results from Experiment 1. L1-MFRBP exhibits slight improvement over CSP, indicating that a multi-fidelity recursive prediction scheme can enhance performance even if the “low-fidelity” models are very naive. Improvement is more pronounced over longer prediction horizons.

3.3 Experiment 3 - L1-MFRBP conditioned on Ego Future

An automated vehicle may figuratively ask: “If I do this, what will everyone else do?” Experiment 3 seeks to quantify the performance improvement that results from conditioning motion prediction on a candidate future ego trajectory.

To explore this question, we use the ground truth future trajectory as a surrogate for the ego agent’s “planned” trajectory. This takes the place of the ego agent’s level 0 trajectory in a “planning-aware” version of L1-MFRBP. In other words, we “cheat” by allowing the model to observe the ground truth future trajectory for each designated ego agent during training and testing. We wish to make it clear, therefore, that experiment 3 is not meant to compete with the other models.

Results for Experiment 3: RMSEs for planning-aware L1-MFRBP are shown in italics in the last column of Table 1. The numbers are better than for L1-MFRBP, which suggests that conditioning on a planned trajectory (as in a real automated driving scenario) can improve motion prediction.

4 Conclusion

We have demonstrated that motion prediction in traffic scenes can be improved by recursively reasoning about future interaction between agents. We have also shown that multi-fidelity modeling can be effectively incorporated in the recursive prediction process.

Immediate directions for future work include extending our method to reason about multiple possible future scenarios, incorporating a more flexible and diverse set of policy models, reasoning about input state uncertainty, and devising a more comprehensive set of experiments and performance metrics to evaluate our models. We aim to eventually implement a refined version of our algorithm on board a real automated vehicle.
References