Forecasting and Decision Making in self-driving

Sanjiban Choudhury
“Trying to predict the future is a mug’s game...

... But increasingly it’s a game we all have to play because the world is changing so fast and we need to have some sort of idea of what the future’s actually going to be like because we are going to have to live there, probably next week.”

Douglas Adams
The Salmon on of Doubt
How the robot sees the world ...
Traditional Architecture

Raw sensor data

Perception → Forecasting → Motion Planning → Control actions
Is having cascaded blocks a good idea?
Lots of recent work on unifying perception and forecasting
Lots of recent work on unifying perception and forecasting

SpAGNN: Spatially-Aware Graph Neural Networks for Relational Behavior Forecasting from Sensor Data

Sergio Casas\textsuperscript{1,2}, Cole Gulino\textsuperscript{1}, Renjie Liao\textsuperscript{1,2}, Raquel Urtasun\textsuperscript{1,2}
Uber Advanced Technologies Group\textsuperscript{1}, University of Toronto\textsuperscript{2}
\{sergio.casas, cgulino, rjliao, urtasun\}@uber.com

MultiXNet: Multiclass Multistage Multimodal Motion Prediction

Uber Advanced Technologies Group
\{ndjuric, hcu2, sushacon, shxuanwu, satsang.wang, chou, luisas\}@uber.com
\{songf, ruizhu, yangxu, ade, sidney, becker, meyer, davidgt, wellington\}@uber.com
But what about forecasting and motion planning?
Shaky foundations of forecasting

Are we using the right model?

Are we collecting data correctly?

Are we using the right loss?
Example: Learning forecasts for merging actors

Goal

1. Predict 5s future trajectory
2. Plan with 5s future trajectory
Activity!
Example: Learning forecasts for merging actors

1. Predict 5s future trajectory

Data?
Model?
Loss?
Example: Learning forecasts for merging actors

2. Plan with 5s future trajectory

Cost function?

Planner?
Think (30 sec): Design choices for forecasting and motion planning

Pair: Find a partner

Share (45 sec): Partners exchange ideas

1. Predict 5s future trajectory
   - Data? Model? Loss?

2. Plan with 5s future trajectory
   - Cost Function? Planner?
Why is current state insufficient to predict future?

Simple latent variables:

Velocity, Acceleration may not be observable

Complex latent variables:

Intent (turning left, making a lane change) are not observable and must be inferred from past actions
Sequence Model
A very brief history of sequence prediction in robotics

Kalman Filter + Prediction
Hand design observation models, infer latent states, forward predict.

RNN, LSTMs
Learn the filter! Problem - forget long sequences since only one hidden state vector passed from one time step to next

Transformers
Retain all hidden state, pay $O(H^2)$ computation
Given sequence of English words, predict sequence of French
Transformer Architecture

INPUT: je suis étudiant

OUTPUT: I am a student
The animal didn't cross the street because it was too tired
Attention as a soft-memory look up
Back to forecasting
Transformers for motion prediction

$S_{t-2} \quad S_{t-1} \quad S_t \quad S_{t+1} \quad S_{t+2}$
What happens with a typical forecasting approach?

1. Collect lots of driving data of actors merging

2. Train a forecast model to predict actor future
Forecasts have huge variance!

Planner brakes aggressively!
Why is the forecast so whacky?
Why is the forecast so whacky?

Marginalizing over multiple modes!

Mode A:
Robot merges after

Mode B:
Robot merges before
Okay .. so why can’t we just predict multi-modal distributions?
Multi-modal forecasts do not solve the issue!

We are (incorrectly) telling the planner both modes can happen!

Mode A: Robot merges after

Mode B: Robot merges before
What robot does depends on other humans

What other humans do depends on the robot
Forecasting-or-planning: a chicken-or-egg problem
Why can’t we just forecast the robot motion?
Planning is NOT merely forecasting

Suppose you collected data from this vs data from this

Which data is useful for forecasting? For imitation learning?
Solving the chicken-or-egg problem

Train a *conditional* forecasting model

Normal forecasting

\[ P(s_{t:t+k} | s_{t:t-k}) \]

Conditional forecasting

\[ P(s_{t:t+k} | s_{t:t-k}, \xi_{\text{plan}}) \]
All actors in a scene influence each other

Robot is simply one actor among many in a scene

Need to jointly reason over all actors to produce forecasts
Problem: Space of joint trajectories is massive

Continuous space of trajectories + Exponentially with in actors

Conditional forecasting just makes this even harder
Reason in a space of discrete “modes”
3 fundamental modes of space-time paths

A Yields to B

B Yields to A

Not Yield
Mode $\equiv$ A single basin of forecast

R Yields to A
B Yields to R
Mode $\equiv$ A single basin of forecast

R Yields to A
R Yields to B
C Yields to R
Message Passing on a Graph

Given a set of modes chosen by the robot

Infer what modes others are likely to choose
Message Passing on a Graph

Given a set of modes chosen by the robot

Infer what modes others are likely to choose

Forecast actors given modes
Message Passing on a Graph

Given a set of modes chosen by the robot

Infer what modes others are likely to choose

Forecast actors given modes

Plan given forecast
Geometric XformerNet

**Input**
- **Node features** $f_i$ 
  state+history of each actor in different path frames
- **Edge features** $f_{ij}$ 
  source actor state+history in destination actor frame

**Encoder**
- $k = 1, \ldots, K$
- $K$ steps of message passing

**Output**
- **Edge output** $e_{ij}$
  Predict discrete modes
- **Node output** $n_i$
  Predict T-step trajectories

Builds on [Kumar et al. ’20]
R Yields to A
R Yields to B
C Yields to R
Shaky foundations of forecasting

Are we using the right model?
Conditional forecasting

Are we collecting data correctly?

Are we using the right loss?
What happens when we deploy model?

“The car will probably merge ahead, so I can slow down very smoothly…”

“What the heck does this truck want to do, go ahead or behind ?!?!?”

“?!@#!@“
We have seen this problem before!
Solution: DAGGER for SysID

Collect Trajectories with Current Policy & Exploration Policy

New Policy
Exploration Policy

New Model
Fit Model

Planner

New Transitions
State → Action → Next State

All previous transitions
Extends our previous work [1]. Similar to [2,3]
DAGGER for Forecasting!

Collect Data → Aggregate Data

Plan with forecasts → Train Forecaster
Shaky foundations of forecasting

Are we using the right model?
   Conditional forecasting

Are we collecting data correctly?
   Interactively collect data

Are we using the right loss?
What makes a forecast good?
What makes forecasts good?
Forecasting is just a model

Models are useful fictions
What makes a forecast model good?

\[ J_{M^*}(\hat{\pi}) - J_{M^*}(\pi^*) \]
The *Double* Simulation Lemma
Forecast Model Learning: It’s only a game!

\[ |J(\hat{\lambda}) - J(\lambda^k)| \]

\[
\begin{align*}
\hat{M} &= \min \max \sum_{t=1}^{T} E \left| \sum_{s_t \in S_{t+1}} \sum_{a_t \in A_{t+1}} M^*(s_t, a_t) V(s_{t+1}) - \hat{M}((s_t, a_t)) V(s_{t+1}) \right| + \\
&\quad \sum_{t=1}^{T} E \left| \sum_{s_t \in S_{t+1}} \sum_{a_t \in A_{t+1}} M^*(s_t, a_t) V(s_{t+1}) - \hat{M}((s_t, a_t)) V(s_{t+1}) \right|
\end{align*}
\]

Where \( \hat{\lambda} = \text{PLANNER}(\hat{M}) \)
Shaky foundations of forecasting

Are we using the right model?
Conditional forecasting

Are we collecting data correctly?
Interactively collect data

Are we using the right loss?
Performance Difference
tl;dr

But what about forecasting and motion planning?

Raw sensor data → Perception → Forecasting → Motion Planning → Control actions

Forecasting-or-planning: a chicken-or-egg problem

Shaky foundations of forecasting

Are we using the right model?  
   Conditional forecasting

Are we collecting data correctly?  
   Interactively collect data

Are we using the right loss?  
   Performance Difference