Imitating Driving Behavior in an Urban Environment

Malik Boudiaf (mboudiaf@stanford.edu), Ianis Bougdal-Lambert (ianisbl@stanford.edu)

Summary

Introduction

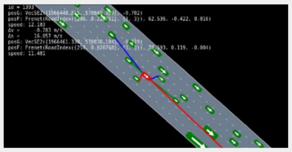
- Autonomous vehicles need to adapt to a wide range of situations, even unlikely
- Modern approaches imitate human drivers by fitting a control model using Behavioral Cloning (BC) or Reinforcement Learning
- They fail to generalize to unseen situations
- GAIL is a new framework that incorporates Imitation Learning into a Generative Adversarial model

Objectives

- Compare BC and GAIL on an urban dataset (only tested on highways so far)
- We show we can obtain good performances with simpler policy architectures

Background

- State s = set of a vehicle's features
- Action a = acceleration and turn rate
- Policy π = neural network with input s and output distribution over a
  $a \sim \pi(a|s; \theta)$
- Rolled-out in a simulation environment to get next state

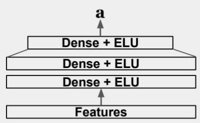


- **GANs training process**
  - Discriminator ψ = neural network trying to distinguish generated trajectories from expert trajectories
  - θ and ψ are optimized in a GAN fashion:

$$\max_{\psi} \min_{\theta} V(\theta, \psi) = \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{X}_E}[\log D_\psi(s,a)] + \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{X}_\theta}[\log(1 - D_\psi(s,a))].$$

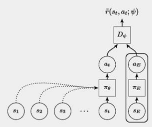
Model

Behavioral cloning

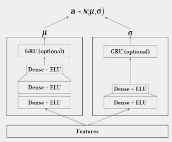


GAIL

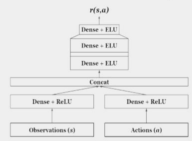
- π_Θ and D_ψ competing
- π_Θ can be **recurrent**



Policy Network π_θ



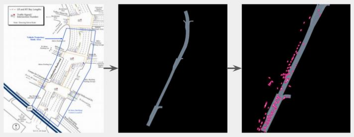
Discriminator D_ψ



- Modified to use **Wasserstein-GAN** instead of GAN
- **Discriminator → Critic**: outputs trajectories rewards instead of probability

Data

- Data downloaded from the NGSIM database
- Lankershim Blvd, LA: intersections + traffic lights
- Processed using AutoCAD → roadway model + trajs



Experiments & Results

- **Input**: features extracted from trajectories:
  - Core features: speed, veh. length/width, lane offset/rel. heading/curvature, dist. to left/right markings
  - Simulated lidar features + Indicator features (collision, off-road, reverse)
- **Output:** Trained policy π_Θ : $s \rightarrow \mu, \sigma$. Action sampled from: $a \sim N(\mu, \sigma)$
- **Different architectures implemented**

Model	π_θ		D_ψ
	μ_θ	Σ_θ	
Baseline BC-MLP small	*(128,128,64)*		
BC-MLP (5 layers)	(256,128,64,64,32)		
Static Gaussian	(32,32)	(32,32)	
GAIL-MLP	(128,128,64)	(128,64)	(128,128,64)
GAIL-GRU	(128,128,64) + (64)	(128,64) + (64)	(128,128,64)

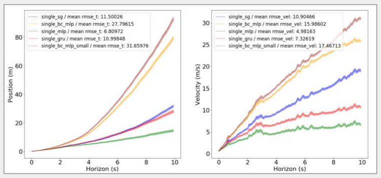
- **Training**
  - Different models trained for 1000 iterations (~4 days)
- **Evaluation**
  - Generate 10s trajectories in environment
  - Compute RMSE of position, lane offset and speed
  - Compute KL divergence KL ($p_\Theta(v)||p_{data}(v)$) for several variables v
- **RMSE**

v : variable from expert traj

$\hat{v}$: variable from generated traj

i , j : indices of traj

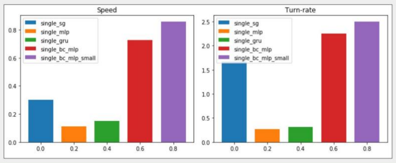
$$RWSE(t) = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(v_t^{(i)} - \hat{v}_t^{(i,j)})^2}$$



- **KL divergence**

Sample variable v and assume $p_{data}(v)$ and $p_{model}(v)$ are piecewise uniform (using B bins for both distributions)

$$D_{KL}(p_{data}||p_{model}) = \sum_{b=1}^{B} p_{b,data} log(\frac{p_{b,data}}{p_{b,model}})$$



Discussion

- BC fails to produce viable trajectories
- GAIL improves realism of generated trajectories
- Deeper policies π_θ don't necessarily result in better performance
  - Initial paper : GAIL's π_θ is (256,128,64,64,32)
  - Baseline way simpler than BC MLP but works better
- In reality, other drivers are influenced by our behavior -> multi-agent